Systems Optimization for Mobility Management

Ashutosh Dutta

Submitted in partial fulfillment of the
Requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2010

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

Systems Optimization for Mobility Management

Ashutosh Dutta

Wireless network mobility frees the user from location dependence but requires additional mechanism to preserve network connectivity. Mobility events occur when user movement causes one network connection to be replaced by another. A network connection has associated with it properties, for example, network attachment points, network identifiers, and security associations. The mechanisms supporting mobility events rebind these properties, often requiring operations at multiple layers of the protocol stack. The rebinding is a sequential process and each process takes a finite amount of time. This overall process generates a period of time in which network service is degraded by transient data loss and increased end-to-end delay. Application specific and protocol specific ad hoc solutions are available for mitigating the service disruption. However, formal techniques to characterize this problem and to develop optimization methodologies for these processes have not been studied. This dissertation develops a systematic and formalized systems model that analyzes the basic operations associated with a mobility event, studies the behavioral properties of the system and characterizes several systems optimization techniques for these processes.

The proposed formal mobility systems model represents these basic operations in the form of a discrete event dynamic system. I analyze this general mobility systems framework and develop several methodologies that can model systems optimization techniques. In particular, I develop methodologies to model the optimization for many of the basic op-

erations, such as discovery, configuration, binding update, and media redirection. Then, I apply these methodologies to prototype mobility systems that can support subnet, domain and inter-technology roaming for real-time and streaming applications in the wireless Internet. Some of these methodologies include proactive network and resource discovery, pre-authentication, pre-configuration, reduction of binding delay, and minimizing the effect of media redirection delay by means of dynamic buffering and multicasting.

I validate the mobility systems model by using it to analytically assess related mobility protocols for both interactive and multicast streaming traffic. The model can result in an analytical assessment of the techniques that can be directly compared to my experimental results under certain resource constraint and various networking parameters, such as mobility rate, packet-to-mobility ratio, simultaneous mobility, distance between the communicating nodes, and network access characteristics. I perform a comparative analysis of our optimized mobility management schemes with similar network layer mobility protocols and other fast-handoff mechanisms.

Contents

List of Figures

List of Tables

Acknowledgements

First of all, I must express my sincere and foremost gratitude to my thesis advisor Prof. Henning Schulzrinne, for his insightful guidance, thoughtful coaching, constructive ideas, and tremendous efforts for my thesis related research. I will be forever indebted to him as I learned the art of research under his expert guidance. His constant guidance has helped me to improve my ability to pay attention to the details and quality of research. I would like to thank Dr. Bryan Lyles who supervised me at Telcordia and kept me on track. Prof. Prathima Agrawal and Dr. Toshikazu Kodama have inspired me during my Ph.D study.

I am privileged to have been a member of the Internet Real-Time (IRT) Laboratory at Columbia University headed by Prof. Henning Schulzrinne. I learned a lot from fellow IRT members through collaboration and group meetings. I would like to acknowledge the support from the staff of Computer Science department, namely, Rosemary Addarich, Susan Tritto, and Pat Hervey. I would like to acknowledge Prof. Yechiam Yemini who motivated me to work on my research in earlier years. I would like to thank my thesis committee members Prof. Nicholas Maxemchuk and Prof. Dan Rubenstein whose feedback on my research helped me to focus my thesis. I am indebted to my employer Telcordia Technologies for supporting my study. I thank my colleagues at Telcordia, Toshiba, Toyota and KDDI for the research collaboration. I am thankful to Prof. Jiacun Wang and Dr. Hesuan Hu for Petri net related discussions.

Finally, I wish to thank my wife Sarmistha, children Srijoy, and Arijit, my parents, friends and relatives for their constant encouragement and support during my Ph.D. study.

Chapter 1

Introduction

Wireless connectivity to communications and information has advanced the world towards ubiquitous computing. In the space of less than thirty years, cell phones have become ubiquitous and wireless data access has become common. However, this access has brought with it a variety of technical problems. Radio physics and power constraints, the need to reuse spectrum, economic constraints on facility placement, and service balkanization due to competitive and political factors force us to implement wireless systems as cells of limited range. Furthermore, cells may use very different wireless technologies or provide fundamentally different services, such as VoIP (Voice-over-IP), streaming or direct short range communications for telematics. We then need handoff mechanisms, often at multiple protocol layers, to allow a mobile terminal to move from cell to cell and maintain service continuity.

Mobility can be described as terminal movement resulting in the release of terminal's binding to the current cell (point of attachment to the network) and establishment of bindings to the new cell being entered while preserving the existing sessions associated with higher level services. The cellular telephony community has long implemented service and technology specific mobility protocols that hand off voice sessions as the user moves from cell to cell. Because voice service quality is highly sensitive to service interruptions, cell-

to-cell handoffs in a cellular environment have been highly optimized and are not noticed by the public. Tripathy et al. [TRV98] summarize some of the handoff technologies associated with cellular mobility. Pollini [Pol96] discusses some of the trends in handover design in the cellular networks that may affect the handover performance optimization.

For IP traffic, the IETF has defined mobility protocols for both IPv4 [Per02c] and IPv6 [JPA04]. However, IP traffic is dramatically more diverse than cellular voice in the range of link layer technologies used to support IP traffic, the number of economic units supplying IP service as well as the authentication protocols and services running above IP. This diversity meant that the IETF could not easily design the access specific handoff optimization techniques such as soft-handoff [WL97], [CM03] often seen in cellular voice into the mobility standards. As a result, unoptimized mobile IP handoffs can take a few seconds to perform a handoff and degrade the desired quality of service in the process.

IP's transformation from a service supporting email and file transfer to the base layer for network convergence means that the constraints on handover performance are becoming much more stringent. Handovers cannot interrupt real-time services. The mechanisms and design principles needed for building optimized handovers in the context of mobile Internet services are poorly understood and need better analysis. To the best of my knowledge, none of the existing work has attempted to model the systems aspects of a mobility event nor was intended to systematically analyze the elementary operations involved with a mobility event. This body of work also lacks in the ability to predict the performance of any mobility event. Some of the existing work have focused on optimizing only parts of a mobility event in an ad hoc manner, specific to a mobility protocol, without providing a comprehensive approach to solving the optimization at all layers or functional modules. I provide an overview of this related work for each of these mobility functions along with the detailed description of my proposed techniques in Chapter 5.

This thesis contributes to a general theory of optimized handover, especially with respect to mobility of Internet-based applications. The contributions fall into four categories:

1. Identification of fundamental properties that are rebound during a mobility event. Analysis of these properties provides a systematic framework for describing mobility management and the operations that are intrinsic to handover.

2. A model of the handover process that allows to predict performance for both an un-optimized handover and for specific optimization methodologies under resource constraints. This model also allows to study the behavioral properties of the handoff system such as data dependency and deadlocks.

3. A series of optimization methodologies, their experimental evaluation and optimization techniques that can be applied to link, network, and application layers and preserve the user experience by optimizing a handover.

4. Application of model to represent optimizations and comparison of the model-based results against experimental data.

1.1 Types of mobility

There are several types of mobility, such as terminal mobility, personal mobility, session mobility, and service mobility. Schulzrinne and Wedlund [SW00] introduce different types of mobility to support multimedia traffic for an IP-based network. I briefly review each type of mobility. However, the focus of my research is on terminal mobility.

1.1.1 Terminal mobility

Terminal mobility allows a device to move between networks while continuing to be reachable for incoming requests and maintaining the existing sessions during the movement. It allows an established call or session to continue when an MS (Mobile Station) moves from one cell to another without interruptions in the call or session.

Terminal mobility may also arise due to change in network condition, whereby a mobile switches between two neighboring networks even without any movement. I describe terminal mobility that arises due to different types of handoffs. Handoff, often also known as handover, is a process that results when a mobile disconnects from one point of attachment in the network and re-connects to another point of attachment in the same or different network.

The handoff process can be either hard or soft. With hard handoff, the link to the prior base station is terminated before or as the user is transferred to the new cell's base station. Thus, the mobile is linked to no more than one base station at a given time. Initiation of the handoff may begin when the signal strength at the mobile received from the target base station is greater than that of the current base station. As the mobile moves into a new cell, its signal is abruptly handed over from its current cell (or base station) to the new one rapidly. In the old analog systems, hard handover could be heard as a click or a very short beep. In the digital cellular systems, this is not noticed. However, in an IP-based handoff scenario, hard handover contributes to 4-15 seconds of delay [DKZ+05]. With soft hand-off [WL97] the MS (Mobile Station) continues to receive and accept radio signals from the base stations that are part of previous as well as its new cell for a limited period of time. The MS signal is also received at multiple base stations. In order to ensure the layer two independence requirement of mobility management scheme, a maximum acceptable hand-off time (MAHT) is required that will vary based on the access type.

In end-to-end wireless IP environment, three logical levels of hand-off procedures can be defined:

i. Layer 2 handoff - It allows an MS to move from one layer 2 point-of-attachment to another layer 2 point-of-attachment that belongs to the same subnetwork. Each layer 2 point-of-attachment may be equipped with same or different type of radio access technology. One subnetwork may consist of multiple layer 2 radio access networks. The IP address of the mobile host remains the same during this handoff.

ii. Subnet hand-off - It allows an MS to move from a radio access network within a subnet to an adjacent radio access network within another subnet that belongs to the same administrative domain. The IP address of the mobile may or may not change.

iii. Domain hand-off - It allows an MS to move from one subnet within an administrative domain to another in a different administrative domain. Typically it involves handoff between two wireless carriers.

As part of my thesis, I have analyzed, modeled, and experimented with all these three types of handoff in a heterogeneous access environment such as CDMA (Code Division Multiple Access) and IEEE 802.11. I focus on systems optimization aspects of terminal mobility in my thesis. Systems optimization techniques minimize the delay and packet loss contributed by several handoff components under certain resource constraints in the mobile and network, namely battery power, CPU cycles, and network capacity.

1.1.2 Personal mobility

The concept of personal mobility was initially introduced as part of Universal Personal Telecommunications (UPT) as explained by Zaid in [Zai94]. Personal mobility removes the fixed association between the terminal and user, thereby allowing an additional degree of mobility over and above terminal mobility in mobile networks.

Figure 1.1 shows some fundamental differences between personal mobility and terminal mobility by showing the relationship among path identification and terminal identification and user identification.[1]

For multimedia communication, personal mobility is a form of mobility by which a user can be reached at different terminals using the same logical address or resource identifier [S+96] regardless of its point of attachment and the identifier that it obtains when it attaches to a network. As the mobile changes its point of attachment, it acquires a new identifier in the new network and updates this identifier by means of registration to a cen-

[1]These terms are defined in the definition section as part of Appendix C

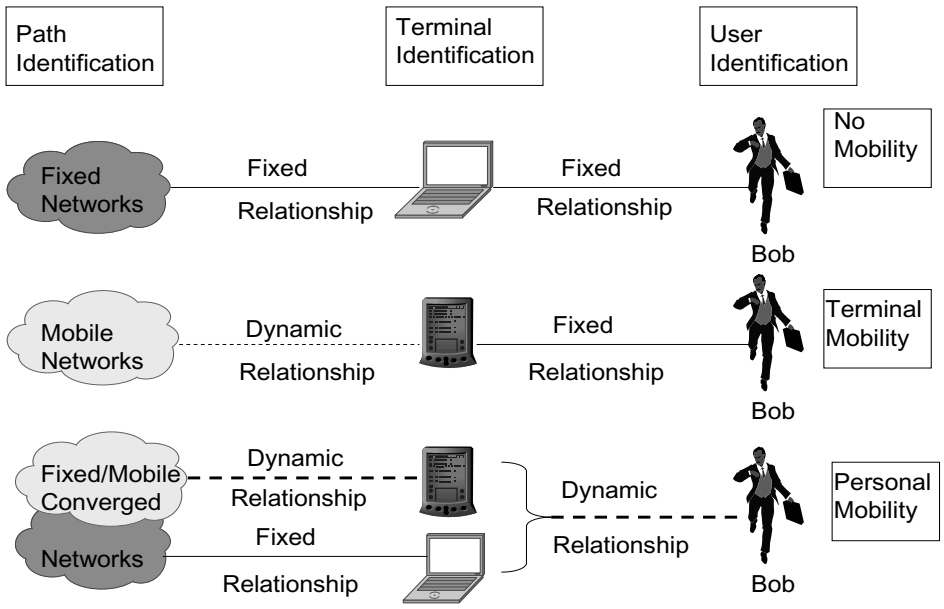


Figure 1.1: Personal and terminal mobility

tral authority either in the home network or in the visited network. The central authority is often a SIP (Session Initiation Protocol) registrar [RSC+02] that keeps a binding between the new terminal identifier and the unique resource identifier that is assigned to the mobile user and is unique for each mobile. An URI is typically a SIP URI and is of the form sip:alice@xyz.edu as defined by IETF RFC 2396 [BLFM98]. Personal mobility can involve both 1-to-*n* mapping, where one address can be associated with many potential terminals and *m*-to-1 mappings, where multiple addresses map to one device. Thus, by having a mapping between the terminal identifier and the resource identifier, it is possible to direct the data to one or more interfaces, where the interfaces can be part of the same device or multiple devices.

1.1.3 Session mobility

Session mobility allows a user to continue an existing multimedia session or part of the session as the user moves from one device to another device. For example, a user that is part of a multimedia session including voice and video in a cell phone can transfer the video to

another device such as TV, thus splitting the existing multimedia session into two devices. Similarly, an existing audio session can be transferred from the cell phone to a desktop phone. MEGACO [CGR+00], third-party call control [RPSC04] and REFER mechanisms [Spa03] are some of the approaches that can be used to implement session mobility. Most recently, 3GPP (Third Generation Partnership Project) has been following an approach called Voice Call Continuity (VCC) [A07] that allows a user to move between IP network and cellular network. Shacham et al. [SSTK08], [SSTK07] discuss the technical details about session mobility and describe how multimedia sessions can be transferred between different terminals.

1.1.4 Service mobility

Service mobility allows one user to maintain one's access to its services even when changing devices and network service providers. In a VoIP (Voice over IP) environment, the typical services that users may like to maintain include speed dial lists, address books, call logs, media preferences, buddy lists and call handling instructions. However, in order to be able to obtain the same service independent of the service provider, the mobile device may often require cross provider relationship.

1.2 Performance requirement

In order to provide the desirable quality of service for interactive VoIP and streaming traffic, one needs to limit the end-to-end delay, network jitter and packet loss to an acceptable level. The performance requirement will vary based on the type of application and its characteristics such as delay and loss tolerance. Different standards organizations have defined the limit for these metrics. For example, 3GPP TS23.107 [Gre01] defines four application classes: conversational, streaming, interactive and background (e.g., file transfer, email) each with different sets of end-to-end delay and QoS requirements. Based on the type of

applications (e.g., interactive, streaming, data), these values may vary. For example, for one-way delay, ITU-T G.114 [Tim00] recommends 150 ms as the upper limit for VoIP applications and considers 400 ms as generally unacceptable delay. Similarly, the streaming class has the tolerable packet (SDU) error ratio ranging from 0.1 to 0.00001 and a transmission delay limit of less than 300 ms.

In general, the handoff process contributes to the packet loss, network jitter, and adversely affects the overall throughput of data traffic because of interruption and retransmission of data caused due to a change in the network point of attachment. A mobility event contributes to two kinds of delays that affect the performance, namely *handoff delay* and *one-way delay* of the packet. Handoff delay is defined as the time between the last packet received in the old point of attachment and the first packet received in the new point of attachment. An end-to-end delay (one-way delay) consists of several components, namely, transmission delay, propagation delay, network delay, operating system delay, codec delay and application delay. Wenyu and Schulzrinne have done a complete analysis of these delays in [JS00]. Handoff contributes to the network delay component of the end-to-end delay.

During a mobile's handoff process, in-flight transient traffic cannot reach the mobile. In-flight packets are defined to be the packets that are in the transit during the mobile's movement from one network point of attachment to another. Network jitter contributes to the variation in inter-packet arrival time of the consecutive packets at the receiver. This is caused due to variation of one-way transmission delay of the consecutive packets. These in-flight packets could either be lost or buffered. If the in-flight packets are lost, then it contributes to inter packet arrival delay between the last packet before handoff and first packet after handoff. If these packets are buffered, packet loss is minimized, but there is additional jitter for the in-handoff packets when these are flushed after the handoff. Buffering during handoff avoids the packet loss, but at the cost of additional one-way-delay. A trade-off between one-way-delay and packet loss is desired based on the type of application. For

real-time communication, if a packet is received after a certain delay threshold, it is also considered lost.

I have verified experimentally that in the absence of any optimization technique, a mobile can be subjected to a handoff delay between 4 seconds and 17 seconds [DKZ+05] resulting in transient service interruption and packet loss. This value varies depending upon the type of mobility protocol used, types of handover, such as vertical i.e., handover between different network types or horizontal i.e., handover between same network types and types of access networks, such as 802.11 or CDMA. Thus, it is desirable to conduct a formal analysis of these discrete events that constitute the handoff process, build a system model that can predict the handoff performance, develop relevant optimization techniques for these operations, and analyze the dynamic behavior of the system during handoff including resource utilization.

While several mobility protocols have been defined at different layers, to the best of my knowledge, prior to my work there was no formal analysis or system model that can study the basic operations associated with mobility events and various systems optimization techniques.

1.3 Motivation

Following are the key issues that motivate my research:

1. Existing mobility protocols affect the performance of real-time communication because of the sequence of discrete events associated with the handoff event.

2. Existing mobility optimization mechanisms are tightly coupled with the respective mobility management protocols and do not provide a generalized approach to optimization. For example, it is impossible to apply mobility optimization mechanisms designed for Mobile IPv4 [Per02c] or Mobile IPv6 [JPA04] to MOBIKE [SE06]. Thus, it is desirable to develop a set of formal methodologies with specific design criteria to help formulate sys-

tems optimization techniques that can be applied to any mobility management protocol and access technologies.

3. Available mobility management techniques do not provide any systematic framework to formalize different states and transition processes involved with a mobility event. Thus, it becomes difficult to study the behavior of a handoff event and evaluate the performance of any mobility protocol or to devise any improvements.

(a) As far as I know, there has not been a systematic mobility system model that can analyze the behavioral characteristics of the handoff event such as deadlock and help formulate the system optimization techniques for cellular or IP-based mobility management protocols.

(b) Existing work does not provide a generalized mobility optimization framework that can support horizontal and vertical handovers across administrative domains.[2]

(c) There has not been any formal analysis of how a specific mobility optimization technique might affect other system resources in the network.

This thesis addresses the above issues. I analyze the basic operations associated with a mobility event in detail. I develop a formal mobility systems model by representing the basic operations associated with a handoff as a series of discrete events. I formalize the associated states and transitions in the form of a Discrete Event Dynamic Systems (DEDS) model. I then analyze the DEDS-oriented mobility model using Discrete Time Transition Petri nets (DTTP). Based on the analysis of these properties associated with a mobility event, I propose several systems optimization techniques for the basic operations associated with the handoff event. I demonstrate these techniques by models, experiments, and numerical analysis using a few network layer and application layer mobility protocols. I then apply these optimization techniques to a Timed Petri net model and compare with the experimental results. I perform a trade-off analysis between the utilization of systems resources and handoff performance metrics obtained through these optimization techniques.

[2] Administrative domain is defined as part in Appendix C.

Finally, I also use this model to study the behavioral properties of the handoff such as deadlock and data dependency under systems and network resource constraints.

1.4 Organization of the thesis

This thesis is organized as follows. Chapter 2 introduces mobility management in cellular and IP-based networks and discusses the related mobility protocols that are currently available. I provide a systematic analysis of the mobility event and associated handoff components in Chapter 3. In Chapter 4, I introduce a formal systems model that uses Petri nets to analyze the behavioral properties of a mobility event and the associated optimization techniques. Chapter 5 describes some key mobility optimization techniques for IP-based mobility protocols that I have developed for different components of the handoff event and demonstrate their validation by way of experiments. Chapter 6 discusses optimization techniques associated with multi-layer mobility protocols. Chapter 7 introduces simultaneous mobility, analyzes the probability of its occurrence and proposes respective optimization techniques for layer 3 and application layer mobility protocols. Chapter 8 describes optimization techniques for multicast stream delivery in a hierarchically scoped-based multicast architecture. Chapter 9 evaluates few handoff systems that I have prototyped using some of the optimization techniques by way of experiments and Petri net models, investigates the behavioral aspects of the handoff operations such as deadlocks and analyzes the trade-off between different schedules for handoff and the systems resources. Finally, Chapter 10 concludes my thesis with a discussion on best current practices of mobility optimization, summary of my contribution and some possible future research direction.

I include four appendices after the bibliography section. I define the RDF schema for application layer discovery in Appendix A. In Appendix B, as part of the glossary, I define an alphabetical list of the abbreviations that are used in the thesis. In Appendix C, I define many of the mobility related terms. Finally, I list my publication in Appendix D.

1.5 Summary of key contributions

I summarize the highlights of key contributions of Chapter 2 through Chapter 10 in Table 1.1. In each respective chapter, I describe the technical problems each of my proposed mechanisms solves, details of my proposed mechanisms and the key benefits with experimental results.

Table 1.1: Summary of key contributions

No.	Chapter title	Summary of key contributions
2	Analysis of mobility protocols for multimedia	Comprehensive analysis and comparison of several generations of mobility protocols (e.g., 1G, 2G, 3G and 4G) to extrapolate the common abstract functions for a mobility event.
3	System analysis of mobility events	Development of a new synthesis that derives a fundamental taxonomy of handover functions and their relationships. This taxonomy provides a basis for describing and characterizing optimization at each layer. Experimental analysis of the handoff delays for the application layer and network layer mobility protocols based on this handover taxonomy.
4	Modeling mobility	• Data dependency analysis and resource analysis of the handover components based on the mobility taxonomy • Design of the first mobility system model for the handoff processes using Deterministic Timed Transition Petri net (DTTPN) based on data dependency and resource dependency • Development of Petri net-based mechanisms to predict the systems performance and behavior of a handoff system • New mechanisms to investigate the opportunity for parallelism based on resource modeling
5	Mobility optimization techniques	Proposed proactive, reactive and cross layer mechanisms to optimize several handoff components as determined in Chapter 3.
	Discovery (Section 5.3)	New application layer discovery mechanism that discovers the network elements of the target networks in an access independent manner. By discovering these elements proactively and caching some of these at the mobile, network discovery latency is reduced to 4 ms.

Continued on next page

Table 1.1 – continued from previous page

No.	Chapter title	Summary of key contributions
	Authentication (Section 5.4)	First network layer assisted layer 2 pre-authentication mechanism that bootstraps layer 2 authentication process in the neighboring networks by deriving the pre-shared keys prior to mobile's handover. This mechanism reduces the authentication delay to 16 ms for both inter-subnet and inter-domain handover.
	Layer 3 configuration (Section 5.5)	• A new proactive IP address acquisition scheme that reduces the signaling exchange by obtaining the IP address from the target network over a secured tunnel before layer 2 handover. • A new reactive router assisted duplicate address detection mechanism, where the router multicasts ARP-cache in a periodic interval so that the mobile avoids the ARP checking for duplicate address detection.
	Layer 3 security association (Section 5.6)	• A new anchor agent assisted layer 3 mechanism that maintains the layer 3 security context by hiding the change of network layer identifier address of the mobile and reduces the delay by avoiding the re-keying process. • First pre-registration-based mechanism that establishes security context prior to mobile's handoff by generating the security keys. The handover delay due to layer 3 security association is completely eliminated.
	Binding update (Section 5.7)	• A new reactive hierarchical binding update mechanism that uses two level hierarchy of addresses and an anchor agent to limit the global signaling update during mobile's mobility within a domain. This mechanism achieves about 70 percent reduction in global signaling overhead for a 10 subnets/domain scenario. • A new proactive binding update mechanism over a secured tunnel that eliminates the binding update delay completely at the expense of maintaining the proactive tunnel between the mobile and the target network.
	Media re-routing (Sections 5.8 and 5.9)	• First reactive forwarding mechanism that redirects the in-flight data from the previous network using application layer mobility proxy in the previous network • First mobile controlled buffering mechanism that controls the buffering period dynamically based on handoff duration during proactive handoff • A new proactive multicasting mechanism that multicasts the in-flight data to the neighboring networks and reduces in-flight packet loss during handoff

Continued on next page

Table 1.1 – continued from previous page

No.	Chapter title	Summary of key contributions
	Route optimization (Section 5.10)	• A new packet interceptor-assisted mechanism that modifies the source and destination address of the packets at the end hosts to maintain the direct path between the communicating hosts. This mechanism reduces transport delay by 50% for the large packets. • First proxy-assisted packet interceptor that eliminates the trombone routing delay and reduces the signaling related delay by 60% in an IMS environment • First binding-cache-based mechanism that minimizes the end-to-end media transport delay by a factor of 5 by localizing the media traffic for localized mobility protocol (e.g., ProxyMIPv6)
	Media independent cross layer triggers (Section 5.11)	• First set of cross layer triggers based on abstract primitives that can pass information across layers and expedite handoff related operations independent of the access mechanisms (e.g., CDMA, 802.11). These proposed cross layer triggers got standardized in IEEE 802.21.
6	Optimization with multi-layer mobility protocols	First multilayer mobility management scheme that uses cross layer triggers from data link layers and application layers and optimizes several handoff operations, namely address configuration, layer 3 binding update and media traversal. My proposed mechanism increases the data throughput by 50% under high mobility scenario by reducing the binding update traversal.
7	Optimization for simultaneous mobility	• First proposal of an analytical framework for simultaneous mobility that can predict the probability of simultaneous mobility based on inter handoff time and binding update latency. • First proposal that outlines the solutions for simultaneous mobility problem for network layer and application layer mobility protocols based on timer-based retransmission, forwarding, redirecting mechanisms, and simultaneous bindings.
8	Handoff optimization for multicast streaming	• First hierarchical scope-based multicast streaming architecture and implementation that offer local and global program management, localized advertisement insertion using control information of real-time traffic (e.g., RTCP). • First to demonstrate reactive and proactive fast-handoff mechanisms using application layer triggering that reduces the *join* latency by a factor of 10 during subnet handover.

Continued on next page

Table 1.1 – continued from previous page

No.	Chapter title	Summary of key contributions
9	System evaluation	• Verification of the experimental results from a few mobility systems that I built using the optimization techniques of several handoff components that I developed and validation of these with the results from the corresponding Petri net models • Behavioral analysis to study the deadlocks and effect of concurrency on handoff operations
10	Conclusions and future work	Infers the best current practices for designing a mobility protocol based on the results from mobility taxonomy and systems optimization mechanisms.

Chapter 2

Analysis of mobility protocols for multimedia

Mobility management consists of two components: ***location management*** and ***handoff management***. *Location management* enables the network to discover the current point of attachment of the mobile user so that the new connection can be established when a new multimedia call arrives. *Handoff management*, often known as terminal mobility, allows the network to maintain the user's connection binding as the mobile node moves from one attachment point to another in the network. I focus on *handoff management* in my thesis.

2.1 Summary of key contribution and indicative results

Over the last three decades, few generations of mobility protocols have evolved without any systematic design approach and these protocols use ad hoc mechanisms to optimize the handoff performances. Without any systematic analysis of the handover components and optimization mechanisms, it is difficult to predict the systems performance of these mobility protocols or design any new mobility protocol for next generation networks.

I analyze the system architecture of each of the available mobility protocols (e.g., 1G, 2G, 3G and several IP-based mobility protocols), describe the respective handoff mecha-

nisms, and then compare the handoff mechanisms in terms of their common mobility functions. For example, I extrapolate how discovery, configuration, authentication and media routing functions are performed for each of the cellular and IP-based mobility protocols and then map the respective network parameters for these mobility protocols with each of the common mobility functions.

There is no prior work that extrapolates these common mobility functions from the existing cellular and IP-based mobility protocols. My comparative analysis and extrapolation of the abstract primitives can determine the required handoff functions that are needed to design a new mobility protocol with certain resource parameters and design optimization mechanisms for these functions.

2.2 Introduction

As a mobile goes through a handover process, it is subjected to connection disruption because of the rebinding of its association at several layers of the protocol stack. Delays incurred due to rebinding within each of these layers affect the ongoing multimedia application and data traffic within the client. Several basic operations are associated with the re-establishment of the binding process across these layers. These operations can be affected by several factors, such as access characteristics (e.g., bandwidth, channel characteristics), access mechanism (e.g., CDMA, CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance), TDMA (Time Division Multiple Access)), re-configuration of identifiers, re-authentication, re-authorization, and rebinding of security associations at all layers.

Mobility protocols have evolved over a period of last three decades. Based on access characteristics and bandwidth, these can be classified into five main categories: 1G cellular, 2G cellular, 3G cellular, 4G, and IP-based mobility. The definition section in Appendix C define 1G, 2G, 3G and 4G. These mobility protocols exhibit certain similar functionalities

during their handoff operation. I highlight these similarities when I describe these protocols later in the chapter. Figure 2.1 shows the evolution of access technologies for the generation of mobility standards. Table 2.1 shows the details of the access characteristics, frequency and data rate of these protocols. In this section, I briefly discuss how mobility operation is performed in 1G, 2G, 3G, and 4G access networks including IP-based mobility protocols and but analyze the associated abstract functions in Chapter 3.

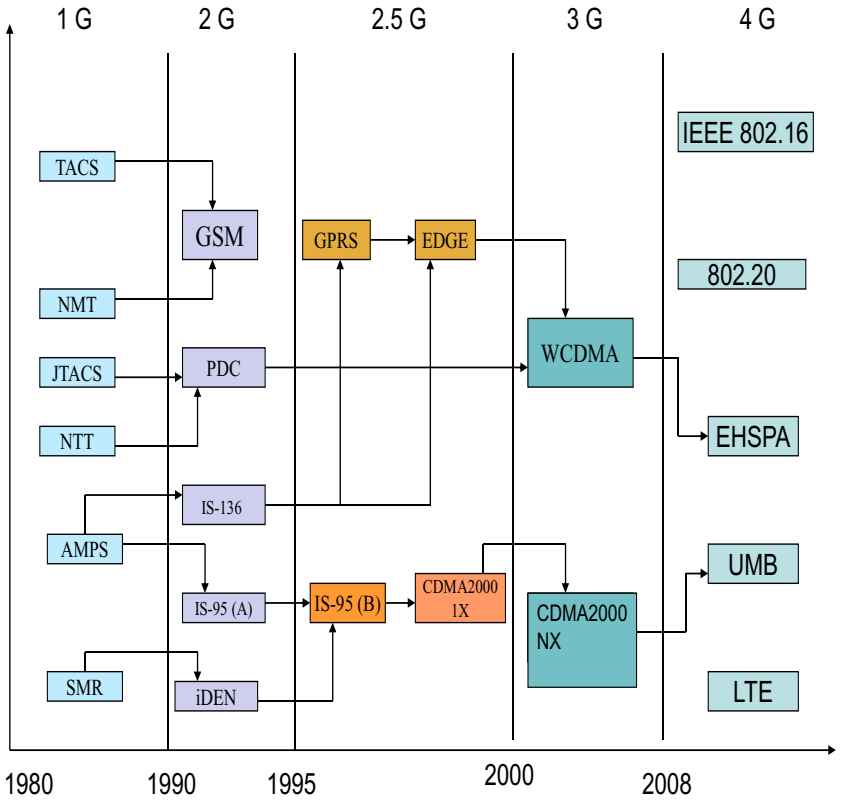


Figure 2.1: Evolution of wireless access technologies

2.3 Cellular 1G

1G refers to the first generation of wireless telephone technology. 1G cellular is based on analog technology and was introduced in the 1980s and continued until being replaced by 2G digital telecommunications. Several 1G cellular standards were developed in different

Table 2.1: Access characteristics of cellular protocols

Generation	System	Channel spacing	Access type	Uplink data rate
1G	AMPS	30 kHz	FDMA	N/A
	TACS	25 kHz	FDMA	N/A
	NMT	25 kHz	FDMA	N/A
	NTT	25 kHz	FDMA	N/A
2G	GSM	200 kHz	TDMA	9.6 kb/s
	PDC	30 kHz	TDMA	42 kb/s
	IS-136	30 kHz	F/TDMA	48 kb/s
	IS-95 (A)	1.25 MHz	F/CDMA	14.4 kb/s
	iDEN	25 kHz	F/TDMA	24 kb/s
2.5G	GPRS	200 kHz	TDMA	45 kb/s
	EDGE	200 kHZ	TDMA	236 kb/s
	IS-95 (B)	1.25 MHz	F/CDMA	115 kb/s
	CDMA2000 1X	1.25 MHz	CDMA	144 kb/s
3G	UMTS/WCDMA	5 MHz	CDMA/TDMA	2 Mb/s
	CDMA2000 1xEV-DO	1.25 MHz	CDMA	2 Mb/s
4G	LTE	20 MHz	OFDMA	50 Mb/s
	WiMAX	2.5 GHz	OFDM	40 Mb/s
	UMB	5 MHz	OFDMA	75 Mb/s

countries. One such standard is NMT (Nordic Mobile Telephone) used in Nordic countries, Eastern Europe and Russia. Others include AMPS (Advanced Mobile Phone System) used in the United States, TACS (Total Access Communications System) in the United Kingdom, JTACS in Japan, C-Netz in West Germany, Radiocom 2000 in France, and RTMI (Radio Telefono Mobile Integrato) in Italy. In 1979, the first analog cellular system, the Nippon Telephone and Telegraph (NTT) system was started. In 1981, Ericsson Radio Systems AB deployed the Nordic Mobile Telephone (NMT) 900 systems and in 1983 and ATT deployed the Advanced Mobile Phone Service (AMPS) as a trial in Chicago. However, the two most important 1G systems deployed in the world are AMPS and TACS. All the 1G systems use Frequency Division Multiple Access but each system works on different frequency range. For example, AMPS operates on 800 MHz frequency band, whereas TACS and NMT450 operate on 900-MHz and 450-MHz frequency bands, respectively. I briefly introduce the 1G architecture and describe how handoff is performed in 1G network.

2.3.1 System architecture

Figure 2.2 shows the simple system architecture of 1G system. The main components of the system are mobile host (MH), base station (BS), base station controller (BSC), and mobile switching center (MSC) or often known as mobile telephone switching office (MTSO) in AMPS. Base stations are considered to be the point of attachment (PoA) for the mobile host in a specific radio cell. MSC acts like a mobility anchor agent in the network. Definition of these terms are given in the glossary as part of Appendix B.

Mobile is assigned a mobile identifier number (MIN) that is equivalent to its home address. It includes an area code identifying the home address area, a three digit exchange number and a four digit subscriber identification number. Each mobile also has an electronic serial number (ESN) that is permanently assigned by the manufacturer. This is equivalent to a device identifier and helps to secure the mobile. While the voice traffic is transmitted using a traffic channel, signaling among the network components are done

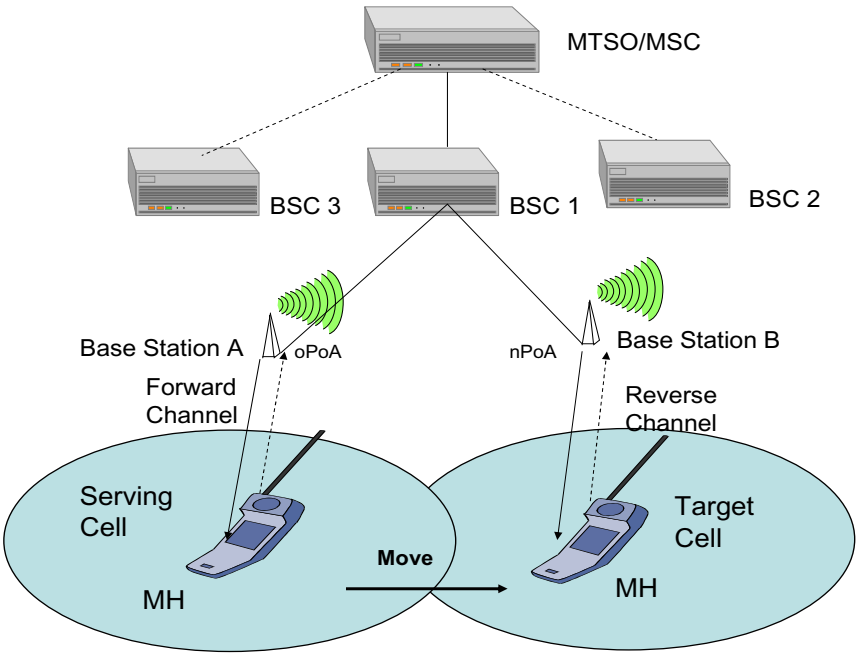


Figure 2.2: First generation cellular architecture

by control channels that are used to initiate the calls and for handoff related operations. There are two kinds of control channels: forward and reverse. For AMPS, the forward channel is known as the FOrward Control Channel (FOCC) and the other one is REverse Control Channel (RECC). FOCC is transmitted continuously by the base station so that it can be received by all the mobiles. The mobile uses FOCC to associate with the network and perform the handover. Mobile uses RECC to register with the network.

The network and base stations are provided with identification codes. For example, in USA, the System Identifier (SID) is assigned by FCC along with the carrier frequencies for each geographical area. Mobile host uses the system identifier to ensure that it is on the correct network. The MSCs are linked together to provide a fully integrated network. Base station controllers typically control a small number of base stations and are linked either by a wire land line or often a short-range microwave link.

2.3.2 Handoff procedure

First generation cellular networks follow network-initiated handover. During the call, the network entity base station (BS) measures the signal strength from a mobile device as received at the serving cell and passes this to MSC (Mobile Switching Center). As the signal-to-noise ratio begins to fall below a certain threshold, the nearby cells are requested to perform signal strength measurements. The scanning receiver within the BS measures the signal strength of the MSs in the neighboring cells and reports it to MSC. If the signal received is better in another base station, and if the MSC decides that a handover is necessary, it instructs the neighboring BS to allocate a channel for the mobile. A second voice path is then set up and bridged across to the existing one in preparation for the handoff. Once this is complete, the system or network generates a handoff instruction and sends it over the FOCC (Forward Control Channel). This handoff instruction carries information that includes the new channel to be used along with the SAT (Supervisory Audio Tone). The mobile accepts the information and sends an ST (Signaling Tone) for 50 ms. It then turns off the reverse channel transmission and re-tunes to the new channel. Once it reaches the new channel, the mobile turns its voice transmitter channel transmission on and retransmits the SAT of the new base station. This acts as the confirmation that the new link has been established. Accordingly, when the new base station receives this, it informs the mobile switching center that the handoff has been successfully accomplished. The new base station then informs the old base station to release the channel the mobile was previously using and make it available for further use. Thus, the original voice path via base station A (BS A) gets rerouted via base station B (BS B) as shown in Figure 2.2.

2.4 Cellular 2G mobility

Variety of second generation digital cellular systems have been developed at different parts of the world during early part of 1990. These include GSM/DCS1800/PCS1900 standard

in Europe, the PDC standard in Japan, and the IS 54/136 and IS-95 standards in the USA. In this section, I briefly discuss about two second generation mobile systems namely, GSM (Global System for Mobile communication) and IS-95 and highlight their handover procedures.

2.4.1 GSM

GSM (Global System for Mobile communication) system is considered to be a 2G cellular system that is a natural evolution from the TACS (Total Access Communication System) mostly used in European countries.

2.4.1.1 System architecture

Figure 2.3 shows a generalized architecture for GSM. The main elements of the system are the base transceiver station (BTS), the base station controller (BSC), the mobile switching center (MSC) and the registration and authentication components. BTS and BSC form the radio access network (RAN) part of the system. Registration functionalities are provided by home location register (HLR), the visitor location register (VLR). Equipment identity register (EIR) checks the validity of a mobile by checking its international mobile equipment identity (IMEI). Home location register (HLR) contacts the Authentication Center (AuC) in order to authenticate the mobile. These terms are defined in the glossary as part of Appendix B.

A mobile node can be associated with several types of identifiers each with its own function. SIM (Subscriber Identity Module) is a small memory card that provides information about the subscriber. IMEI is a fifteen digit number used to identify the equipment and is used by the EIR to decide if the mobile is properly authenticated. IMSI is like the home address of the mobile. It enables the operator to link the phone number and the subscriber. TMSI (Temporary Mobile Subscriber Identity) is like a Care-of-address for the mobile that is assigned when the mobile visits a network. TMSI changes whenever the mobile changes

its network. In order to provide proper authentication to the mobile, the authentication key is stored on the SIM card that is used to compute the cipher key (Kc). Cipher key is used in the encryption algorithm to prevent unauthorized listening to the mobile message.

There are a variety of control channels that are used to provide required functionality to enable the mobiles and BTS to communicate, set up and manage the calls. These could be split into groups, namely three broadcast channels for initial synchronization, three common control channels for initiating calls and three dedicated control channels to manage calls. BCCH (Broadcast Control Channel), Synchronization Channel (SCH), Paging Channel, Slow Associated Control Channel (SACCH), and Fast Associated COntrol Channel (FACCH) are mostly used for handoff management. SCH provides the BTS identification and allows the mobile to get associated with the BTS. BCCH is continually broadcast on the downlink channel by the BTS and contains the information including base station id and frequency allocations. Paging Channel is used to locate the mobile when there is incoming call. Downlink SACCH provides beacon frequencies of the neighboring cells, and uplink SACCH includes measurement report that gives the strength measurements from signals received from neighboring cell beacon transmissions.

However, I discuss the handoff procedure associated with GSM in the next section.

2.4.1.2 Handoff procedure

Unlike handoff in the 1G cellular systems, handover (handoff) in GSM could be either network initiated mobile assisted handoff or mobile initiated network controlled. Network initiated handoff is triggered by the network based on radio subsystem criteria such as RF (Radio Frequency) signal-to-noise ratio, distance from the base station as well as the network directed criteria such as current traffic load per cell. In order to support handover that is based on signal-to-noise ratio criteria, MS takes the radio measurements from the neighboring cells and report these to the serving cell on a regular basis. When the network determines that there is a need for handover, appropriate handoff procedures are followed.

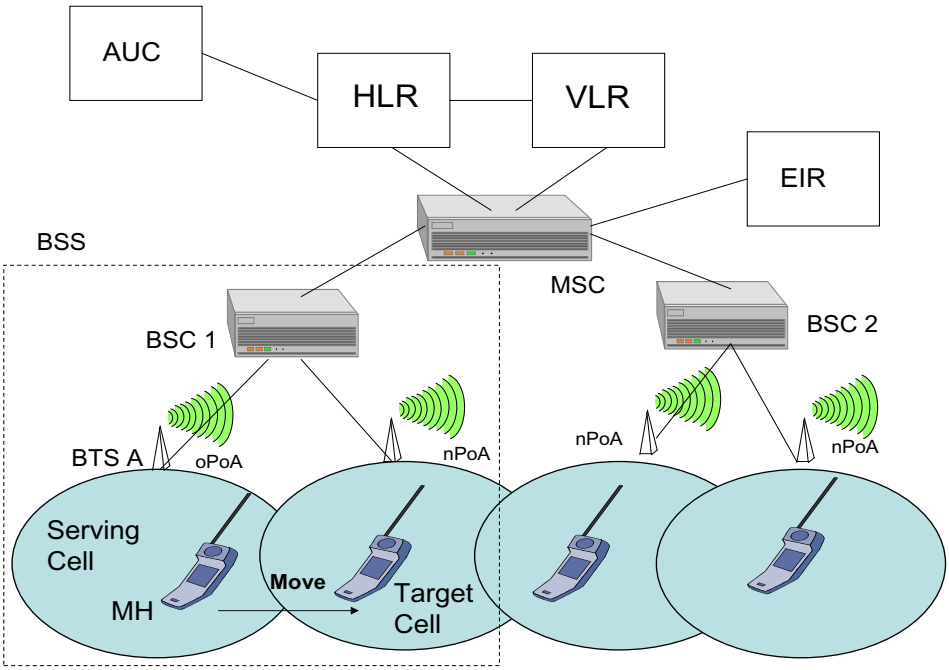


Figure 2.3: GSM-based mobility

Connections in GSM may be handed off between the radio channels in the same cell, between channels in different cells under the same BSS coverage or between the cells under the coverage of different BSSs and even different MSCs. Based on the types of movement, the handoffs can be categorized into two types, namely *internal connection handoff* and *external connection handoff*. As part of the *internal connection handoff* technique, BSS may autonomously handle the connection handoffs in the same cell or between the cells under its own coverage. The MSC is involved in managing the handoffs that need to take place between cells under the coverage of two different BSSs. This process is called *external connection handoff*. In general, when BSS indicates that an external handover is required, the decision of when and whether an external handover should occur is taken by the MSC. The MSC uses the signal quality measurement information as reported by the mobile stations that are pre-processed at the BSS for external handover determination. Details of connection handoff are discussed by Rahnema [Rah93].

Figure 2.4 shows an example of GSM-based mobility. It illustrates intra BSC, inter BSC

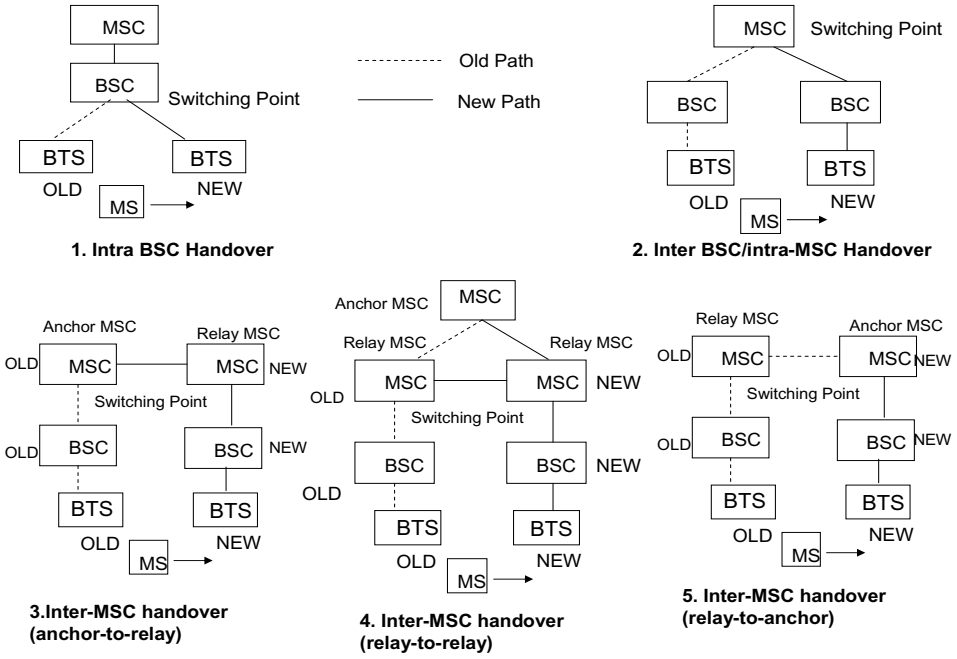


Figure 2.4: Types of GSM handover

and three different types of inter-MSC handover, namely anchor-to-relay, relay-to-relay and relay-to-anchor.

MSC acts like an anchor agent during the duration of the call. During the external handover, the original MSC handling a call keeps control of the call when the mobile is handed over to the target MSC or even to the subsequent MSC. When the BSS performs an internal connection handoff, it informs the MSC when the process is completed. Either the mobile or BSS indicates the impending need for connection handoff by way of FACH (Forward Access CHannel). The BSS usually monitors the quality of radio signal received and transmits such results to the MSC, that keeps a more global view on the radio channels belonging to its BSSs. The MSC may also initiate the need for a connection handoff in an attempt to balance out traffic load in the network.

Table 2.2 shows how different mobility related tasks such as radio channel measurement, handover request and confirm handover are taken care of by the network elements: MS, BS and MSC in a distributed manner.

Table 2.2: Distributed mobility functions for GSM networks

Task	MS	Old Base Station	New Base Station	MSC
Radio channel measurements	Make periodic measurements on current and neighboring channels Send results to BS	Monitor backwards channels Give measurements order to MS		
	Start measurements. Send results to BS			
Issue handover request		Send measurements results to MSC Request Handover		Evaluate handover request. Inform new BS
		Evaluate handover requests. Request handover		Inform new BS
Confirm/Disconfirm handover			Accept,block, delay Handover request	Permit,drop,delay handover

2.4.2 IS-95

Interim Standard Number 95 (IS-95) was developed by Qualcomm that got rebranded as CDMAOne in 1997. IS-95 is based on spread-spectrum technology platform that enables multiple users to occupy the same radio channel or frequency spectrum at the same time. IS-95 has two distinct versions, IS-95A and IS-95B. IS-95B provides additional ISDN-like data rates.

2.4.2.1 System architecture

Figure 2.5 shows the system architecture of IS-95 system. It is based on TR-45 [TIAb] reference model. The main components of this architecture are mobile station, base station, mobile switching center, home location register, visited location register, authentication

center, and equipment identity register. The base station consists of base transceiver system (BTS) and base station controller (BSC). Besides the switching functions, the MSCs behave as the anchor agents during handoff and help to direct voice traffic. In terms of their functionality, handoff related MSCs can be categorized as anchor MSC, border MSC, candidate MSC, originating MSC, target MSC, serving MSC and tandem MSC. For example, anchor MSC is the first MSC that gives radio contact in a call, a candidate MSC could possibly accept a call or a handoff, serving MSC currently provides service to a call, a tandem MSC provides trunk connection for a call in which a handoff has occurred and a target MSC is the MSC selected for a handoff.

The channels in CDMAOne can be split into forward link channel and reverse link channel. For forward link channel, there are four types of coded channels originating from the base station: pilot, sync, paging, and traffic. For the reverse link channel, there are two types of coded channels originating from the MS: access and traffic. The pilot, sync, paging, and access channels carry the necessary control data while the traffic channels carry digital voice. The pilot channel (PC) provides the MS with a beacon, timing and phase reference and signal strength for power control. The pilot channel is transmitted continuously by each sector of a base station. A mobile uses the pilot channel to identify a cell and can identify the strongest sector within a cell based on the measurement of signal-to-noise ratio of the pilot signal. This channel is useful to initiate the handoff operation. The sync channel (SC) provides the MS with critical time synchronization data. The message on the SC contains information necessary for the MS to align its timing with the pilot channel. The mobile uses the sync channel to discover the network and its parameters. The paging channel contains messages with parameters that the MS needs for access and paging. The messages convey system parameters, access parameters, and the neighbor list. This channel is used to communicate with the MS when there is no call in progress. Paging channel can be used to locate a mobile.

For the reverse link, there are two basic channels, namely the access channel and reverse

traffic channel. Mobile uses access channel to communicate with the base station when no traffic channel has been set up and reverse traffic channel to send date and control signals to the base station.

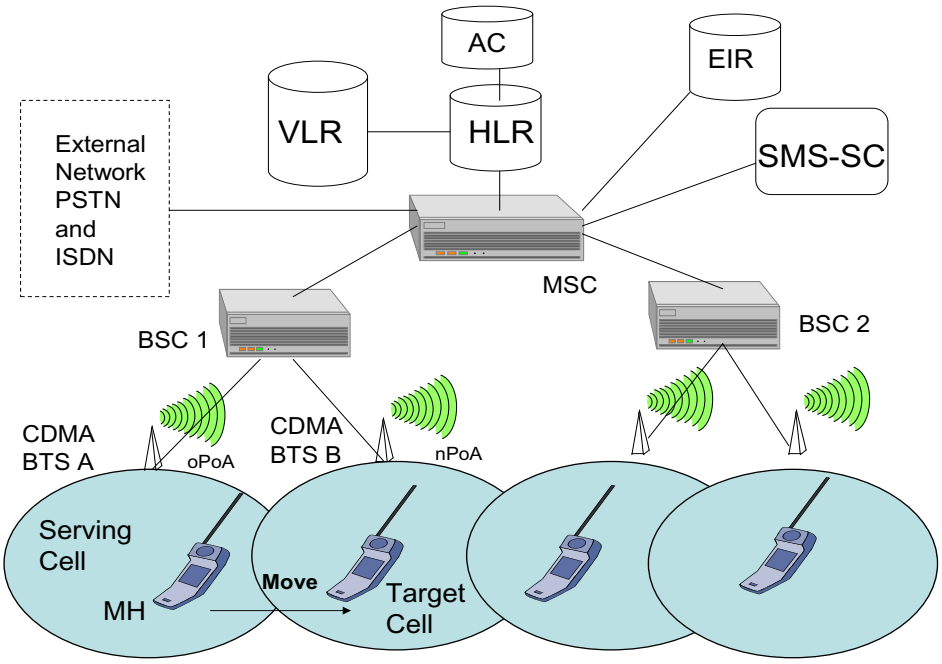


Figure 2.5: IS-95-based architecture

2.4.2.2 Handoff procedure

CDMAOne supports three main types of handover: soft handover, softer handover, and hard handover.

Soft handoff involves an inter-cell handoff and is a make-before-break connection. The connection between the mobile and the network is maintained by several base stations during the process. A soft handoff occurs only when the old base station and the new base stations are operating on the same CDMA frequency channel. The mobile communicates simultaneously with these BSs until it is clear that only one BS is required.

Softer handoff is an intra-cell handoff occurring between the sectors of a base station and is a make-before-break type. It takes place only at the serving base station.

Hard handoff process is functionally break-before-make type. The continuity of radio link is not maintained during the hard handoff. Hard handoff takes place when the mobile is switched between two BSs using different radio channels. Additionally, it is usually meant to enable a mobile to handoff from a CDMA call to an analog call. It is implemented in the areas where a mobile is subjected to handoff from a CDMA network to non-CDMA network because of non-availability of CDMA service in that area.

The handoff process begins when a mobile unit detects a pilot signal that is significantly stronger than any of the forward traffic channels assigned to it and starts to discover a new candidate point of attachment (PoA) where it can connect to. When the mobile detects the stronger pilot signal the following sequence usually takes place: The mobile sends a pilot strength measurement message to the base station and instructs it to initiate the handoff process. The network then sends a handoff direction message to the mobile unit directing it to perform the handoff. On the execution of the handoff direction message, the mobile unit sends a handoff completion message on the new reverse traffic channel.

There are basically two kinds of signaling protocols to take care of mobility related signaling in IS-95: ANSI-634 (American National Standard Institute-634) and ANSI-41. ANSI-634 takes care of signaling between BSC and MSC, and ANSI-41 takes care of the signaling between MSC/VLR and HLR and MSCs.

Figure 2.6 shows the flow for mobile controlled handoff based on IS-41. The handoff process begins with the handoff measurement procedure that determines which of the adjacent systems the mobile should switch to. IS-41-based inter-system handoff consists of two types of handoffs, namely *handoff-forward* and *handoff-back*. Handoff-forward defines a type of inter-system handoff where a mobile is handed off from one mobile switching center (MSCA) to another (MSCB), where $MSCA$ is the serving MSC. Handoff-back is the process of handing back to the serving MSC.

IS-95 also offers automatic roaming functionality by which it allows a mobile to originate a call in the visited system and receive the call destined for the roaming subscriber.

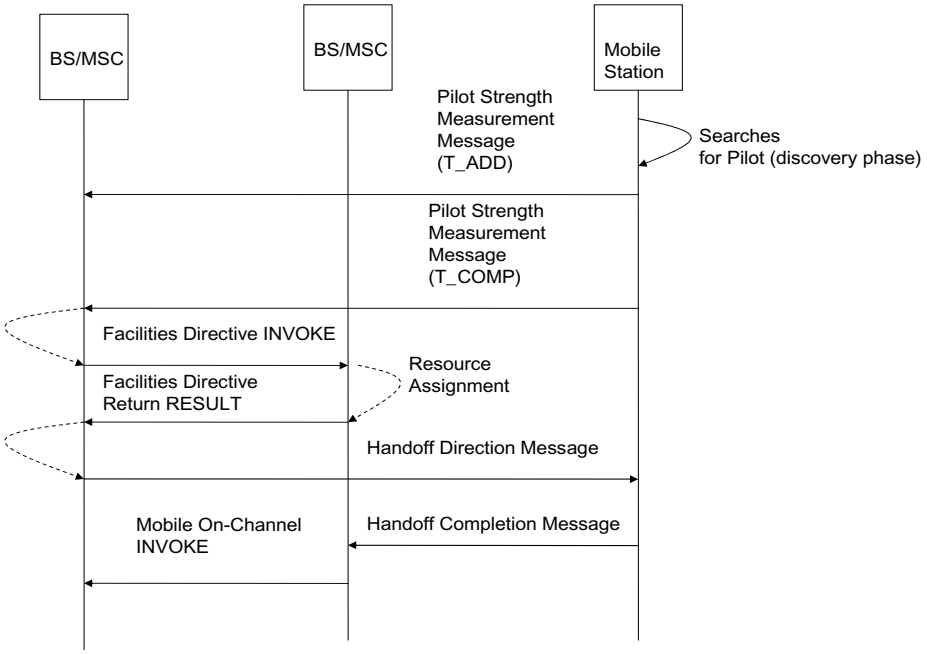


Figure 2.6: IS-41-based mobility

Three basic processes namely, registration, call origination and call delivery features constitute the roaming function.

2.5 Cellular 3G mobility

The main vision of third generation mobile networks (3G networks) is to provide ubiquitous wireless network access that can support voice, multimedia and high-speed data communication. IMT-2000 (International Mobile Telephone-2000) standards were developed by International Telecommunications Union Radio Communication (ITU-R) and International Telecommunication Union Telecommunications sector (ITU-T) to define the operations of 3G networks. One of the main attributes of IMT-2000 is the introduction of wireless wideband packet-switched data services for wireless access to Internet with speeds up to 2 Mb/s. IMT-2000 proposed ten different multiple access schemes. Two of these schemes were based on TDMA and the rest eight were based on CDMA. I describe two of these access schemes namely, WCDMA and CDMA2000.

2.5.1 WCDMA

WCDMA (Wideband Code Division Multiple Access) is the natural evolution of GSM-based systems. WCDMA networking specification was created by 3GPP (Third Generation Partnership Project). Compared to 2G version of cellular or CDMA2000, WCDMA uses a much wider spectrum of one 5 MHz channel for both data and voice providing data rate up to 2 Mb/s.

2.5.1.1 System architecture

WCDMA-based 3G networks consist of two major parts: the radio access network (RAN) and the core network (CN). The radio access network consists of base station and radio network controller (RNC). The base station (BS) in 3G (also known as node B) is an interface between the network and WCDMA air interface. The BS is responsible for taking care of channel coding, interleaving, rate adaptation and processing of the air interface. The RNC (Radio Network Controller) acts as an interface between the base station and the core network. It is responsible for controlling the radio resources. The core network in 3G network consists of two domains: a circuit switched (CS) domain and packet switched (PS) domain. Core network consists of components such as VLR, HLR, and MGW on the circuit switched side and SGSN (Serving GPRS Support Node) and GGSN (Gateway GPRS Support Node) on the packet switched side. Detailed description of these functional components can be found in [Tac02].

Figure 2.7 shows the system architecture of a WCDMA network. It shows the components of both packet switched and circuit switched parts of the network.

2.5.1.2 Handoff procedure

In this section, I describe the handoff procedure associated with WCDMA. I briefly discuss the mobility management of WCDMA system, both for circuit switched and packet switched domains. Mobility management for circuit switched domains in WCDMA con-

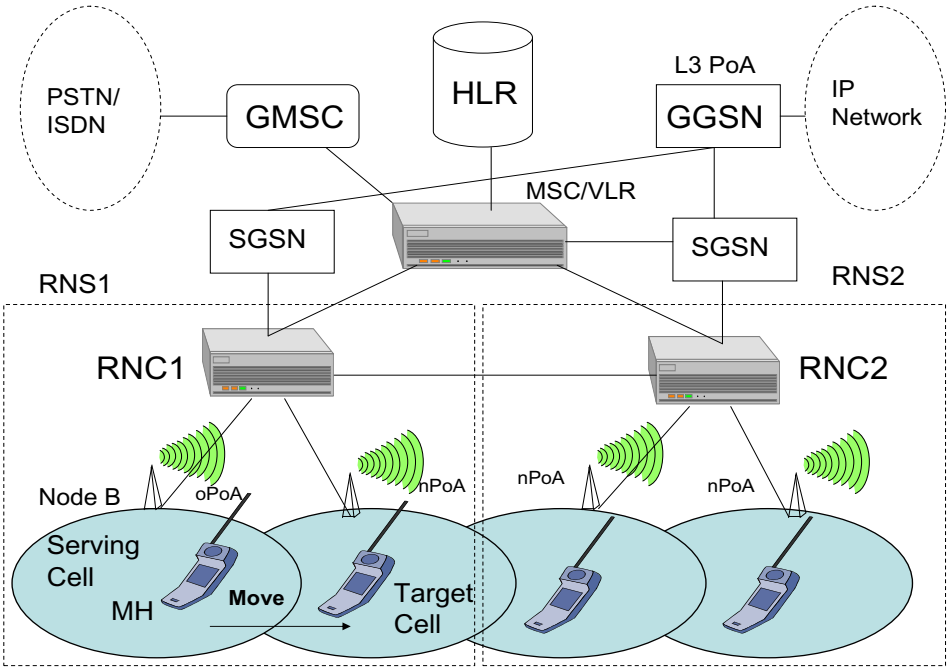


Figure 2.7: WCDMA architecture

sists of call origination, termination and handover control. In WCDMA, the mobile node is also called UE. As part of location updating process during *attach* procedure, a Temporary Mobile Subscriber Identity (TMSI) is assigned to the UE (User Equipment) that is identified by an international mobile subscriber identity (IMSI). IMSI is allocated permanently to each UE (User Equipment). However, every time the mobile changes its network, it obtains a new TMSI and it binds it to an IMSI. This process can be defined as the general binding update process of the handoff procedure.

The handover decisions are generally handled by the RNC (Radio Network Controller). RNC continually monitors information regarding the signals being received by both the mobile node and node B. When a particular link quality falls below a given level and another better radio channel is available, the mobile initiates a handover. As part of this monitoring process, the UE measures the received signal code power (RSCP) and received signal strength indicator (RSSI) and information is then returned to the node B and hence to the RNC on the uplink channel.

There are several cases of handovers in WCDMA based on the movement pattern of

the UE with respect to the cell and other nodes (e.g., Node Bs, RNCs) within the UTRAN (UMTS Terrestrial Radio Access Network). WCDMA-based handoff can be categorized into three types: soft handover, softer handover and hard handover.

Soft handover: Soft handover takes place when the operating frequency remains same between the neighboring cells. Several kinds of soft handover types are discussed here.

1. Intra Node B/intra RNS (Radio Network Subsystem): This handover type is done if the UE moves from one cell to another cell where both belong to the same Node B.

2. Inter Node B/inter RNS/intra SGSN: If the UE moves from a cell in one Node B to a cell in another Node B that belong to different RNS.

3. Inter Node B/inter RNS/inter SGSN: In this case, the UE moves from a cell in one Node B to a cell in another Node B that belongs to a different RNS. The Node Bs are connected to different RNCs and those two RNCs are also connected to different SGSNs. In this case, the UE may even move between two SGSNs.

4. Inter GGSN: GGSN is the layer 3 point of attachment that acts as the default router. Unless the mobile moves between GGSNs the layer 3 identifier of the mobile does not change.

Softer handoff: When the UE moves between two different sectors on the same base station it is called softer handoff. Since the processing elements are shared, it enables a softer handoff to be accomplished more easily than a soft handoff.

Hard handover: Hard handover (or inter-frequency handover) is only needed if the mobile needs to change its frequency after the handover or interface does not exist between two RNCs in case of a soft handover (inter Node B/inter RNS). A frequency change may take place as a result of change of W-CDMA cell level i.e., from a macro cell to a satellite or another change of the radio access technology (inter RAT handover), for example from UMTS to a WLAN or GSM network. A hard handover occurs quite rarely and differs a lot from the soft handover types described above.

2.5.2 CDMA2000

CDMA2000 builds on CDMAOne to provide an evolution path to 3G. Like WCDMA, it supports both kinds of access: FDD-based (Frequency Division Duplex) DS/CDMA and TDD-based (Time Division duplex) T/CDMA. There are several versions of CDMA systems available, namely CDMA2000 1X, CDMA 1xEV-DO (Evolution Data Only), and CDMA 1xEV-DV (Evolution Data and Voice).

2.5.2.1 Architecture

Compared to WCDMA, it uses a spectrum of 1.25 MHz per channel. CDMA2000 1X doubles the voice capacity of CDMAOne system and also supports high-speed data services. It can support peak data rate of up to 153 kb/s. CDMA 1xEV-DO is an evolution of CDMA 2000 that is designed for data-only use and it provides a peak rate capability of over 2.45 Mb/s on the downlink. On the other hand, CDMA2000 1xEV-DV is an evolution of CDMA2000 that can simultaneously transmit voice and data. The peak data rate is 3.1 Mb/s on the forward link and reverse link is limited to 384 kb/s.

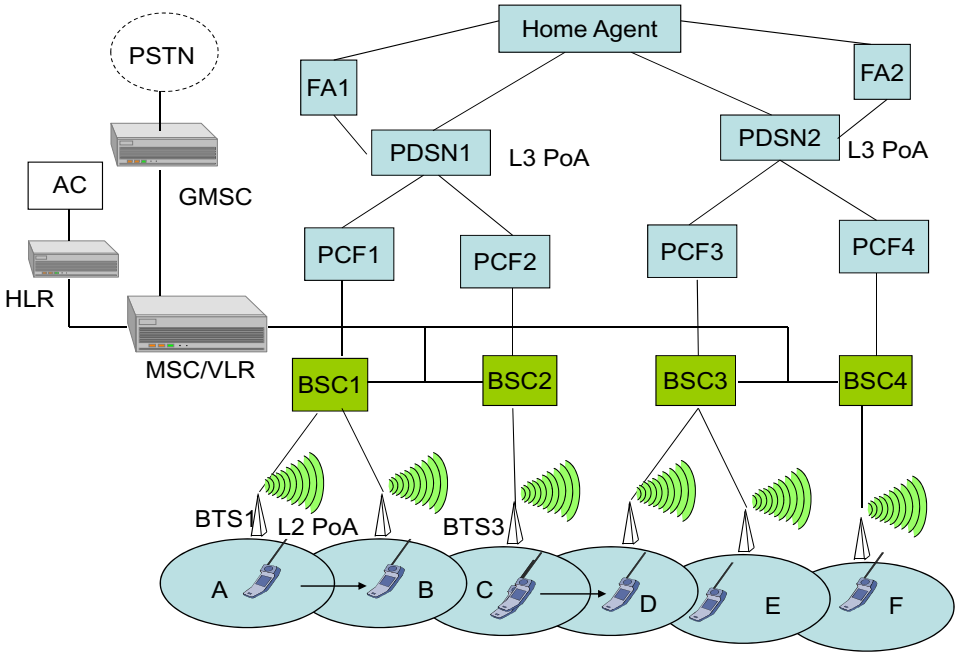


Figure 2.8: CDMA2000 architecture

In order to be able to support packet based services more efficiently, it upgrades the existing network elements of IS95, namely BTS and BSC and adds many new network elements, like PDSN (Packet Data Serving Node), AAA, and home agent (HA). BSC is equipped with additional IP routing functionality and new PDSN establishes, maintains, and terminates point-to-point sessions with the subscriber and initiates authentication, authorization and accounting (AAA) for the mobile station client to the AAA server. In addition, it also augments the functionality of HLR and VLR. For an architecture based on CDMA2000 in conjunction with legacy network components such as MSCs, HLRs, signaling between BSC and MSC are taken care of by TIA 2001 [TIAa] standards, and signaling between MSC, VLR and HLR and between MSCs are taken care of by ANSI-41 standards. TIA-2001 describes the overall system functions including services and features required for interfacing a Base Station with the Mobile Switching Center (MSC), with other Base Stations (BSs), with the Packet Control Function (PCF) and for interfacing PCF with the Packet Data Service Node (PDSN). MN usually initiates a data call via BTS which is the first point of network attachment. The BSC responsible for this BTS (Base Transceiver Station) forwards the call to the associated PCF (Packet Control Function). The PCF selects a PDSN based on certain unique characteristics of the mobile and establishes a GRE (Generic Routing Extension) tunnel with the PDSN. At that point, MN initiates a PPP session that gets terminated at the PDSN. Thus, there is a single hop IP connectivity between the mobile and the PDSN (Packet Data Serving Node).

2.5.2.2 Handoff

Compared to IS-95, CDMA2000 introduces additional levels of thresholds to limit the frequent handovers. By limiting the unnecessary handovers, it increases the system capacity for the data traffic. Like WCDMA, CDMA2000 supports three types of handoff operations: hard handoff, soft handoff, and softer handoff. Figure 2.8 shows a hierarchical network architecture with all the networking elements of a CDMA2000 network. A node's mobil-

ity is taken care of at each level of the hierarchy, namely at BTS level, PCF level, and PDSN level. There are several movement scenarios that can take place within this hierarchy. When the mobile moves between two BTSs, as long as the movement is confined to the PDSN the mobile does not need to set up a new PPP session. But if the mobile moves between two BTSs that are controlled by two different PCFs, each PCF may choose a new PDSN in the hierarchy. This will require to terminate the PPP session and start a new one. When the mobile chooses a new PDSN to re-establish a new PPP session, it obtains a new layer 3 identifier such as IP address. During packet-based communication, a mobile node is identified with a simple IP addressing scheme or mobile IP. In case of simple IP addressing scheme, the mobile obtains the IP address from a DHCP server that usually co-locates with the PDSN. This identifier changes as the mobile moves between the PDSNs. However, if the mobile uses mobile IP-based approach, this IP address does not change. In case of IP address change, layer 3 mobility can be taken care of by mobility protocols like Mobile IP at PDSN layer as discussed in Section 2.4.1.

2.6 4G Networks

4G refers to the fourth generation of cellular wireless and is an evolution of 3G networks. 4G networks provide additional features such as higher data rate up to 100 Mb/s, seamless mobility support across heterogeneous access networks, secured IP-based communication, QoS support for many multi-media services such as mobile TV, MMS (Multimedia Messaging Service), DVB (Digital Video Broadcasting). 4G networks are based on several access technologies, namely OFDMA (Orthogonal FDMA), Flash-OFDM, SC-FDMA (Single Carrier-FDMA), MC-CDMA (Multi-carrier CDMA). LTE (Long Term Evolution) does support OFDMA, WiMAX (IEEE 802.16) supports Flash-OFDM, and IEEE 802.20 supports MC-CDMA as the access technique. However, I describe one specific 4G candidate network, LTE, and its mobility features in this section.

2.6.1 Evolved Packet System

2.6.1.1 System Architecture

EPS (Evolved Packet System) represents the very latest evolution of UMTS standard. EPS has been defined as part of release 8 within the 3GPP standards bodies. It consists of two parts: LTE (Long Term Evolution) and SAE (System Architecture Evolution). LTE takes care of the evolution of radio interfaces and SAE focuses on core network architecture evolution. Figure 2.9 shows the evolved core architecture (EPC). I briefly describe some of the functional components of LTE and EPS.

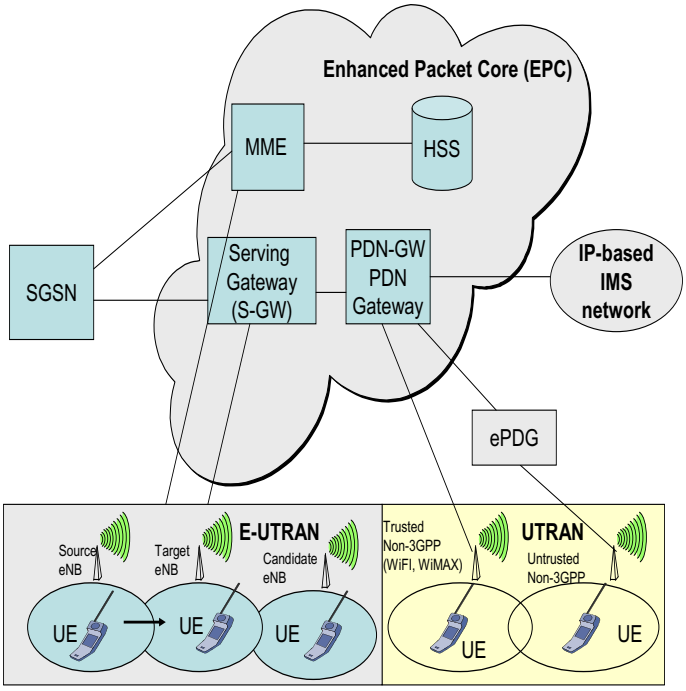


Figure 2.9: SAE/LTE architecture

LTE, which is also known as Enhanced UTRAN (E-UTRAN) defines a packet optimized access network that can efficiently support IP-based real-time and non-real time services providing performances comparable to circuit switched networks. Unlike UTRAN, E-UTRAN relies on a fully shared radio resource allocation scheme that allows maximizing resource usage by combining all radio bearers on a shared high bit rate radio channel.

The goals of E-UTRAN are to provide reduced latency, higher data rates, improved system capacity and coverage as well as reduced cost for the operator and packet optimized radio access technology. The most important component of E-UTRAN is eNode-B. Unlike UMTS, there is no separate RNC, but RNC functionality has been integrated into eNode-B.

SAE also defines a simplified core network composed of one packet domain that can support all packet switched services and inter-working capabilities with traditional PSTN. EPS represents a migration from the traditional hierarchical system architecture to a flattened architecture that minimizes the number of communication hops and distributes the processing load across the network. SAE is also known as EPC (Evolved Packet Core). EPC is composed of several functional entities, namely MME (Mobility Management Entity), serving gateway (S-GW), PDN (Packet Data Network) gateway, and PCRF (Policy and Charging Rules Function. MME is in charge of all the control plane functions and provides features such as security procedures, terminal-to-network session handing, idle terminal location management. The serving gateway (S-GW) is the termination point of the packet data interface towards E-UTRAN and serves as a local mobility anchor supporting intra E-UTRAN mobility and mobility with other 3GPP technologies such as 2G-based GSM and 3G-based UMTS. PDN GW (Packet Data Network Gateway) is the termination point of the packet data interface towards the packet data network. PDN GW supports policy enforcement features, packet filtering and enhanced charging support. PDN-GW also acts like a home agent.

EPS provides much better performance compared to the UMTS of the previous releases. A peak date rate of up to 100 Mb/s downlink and 10 Mb/s up-link can be obtained with EUTRAN. It offers a radio access network latency of 5 ms in user plane and 100 ms for control plane. Latencies for inter RAT (Radio Access Technology) are restricted to 300 ms for real-time traffic and 500 ms for non-real-time traffic.

2.6.1.2 Handoff procedure

LTE supports mobility both at radio layer and network layer. It supports UE-assisted network controlled hard handover when the UE is in active mode. There are two types of handoff defined for LTE: Intra-MME and Inter-MME. During Intra-MME handoff, the mobile moves between two eNBs that are connected to the same MME. Inter-MME defines the handover when the eNBs are controlled by two different MMEs. Two eNBs could be served by two different serving gateways (S-GWs).

LTE supports mobility with both 3GPP and non-3GPP access systems. Mobility across 3GPP access systems is called *local mobility* and mobility across non-3GPP access systems is called *global mobility*. Non-3GPP access systems include wireless systems, such as IEEE 802.11 and IEEE 802.16. Both GTP-based (GPRS Tunneling Protocol) mobility and mobile-IP-based mobility are used to support local mobility. However, only mobile IP-based mobility is used to support global mobility.

I describe below the specifics of a handoff process involving E-UTRAN. Handoff management within LTE network involves several stages, namely discovery, measurement, handover preparation and finally, the handover execution. The user equipment (UE) that is equivalent to a MH (Mobile Host) can be identified with a list of identifiers that are used to assist handoff. IMSI and IMEI provide subscriber ID and equipment ID, respectively. M-TMSI (M-Temporary Mobile Subscriber Identity) is used to identify the UE within the MME. GUTI (Globally unique temporary UE indentity) is allocated by MME. GUTI identifies a globally unique MME and the UE within the MME. The UE is assigned S-TMSI (S-Temporary Mobile Subscriber Identity) which is a shortened version of the GUTI and is used to locate the UE. UE is also assigned an ID that uniquely identifies the UE in a tracking area. An eNodeB sends the tracking area identifier (TAI) on Broadcast Control Channel (BCH). Upon power on the UE performs a cell search, discovers the EPS/LTE access system and performs access system and network selection. After the cell is discovered, it attaches to the radio network and is authenticated with the MME. After the successful

authentication, the MME registers with the HSS (Home Subscriber Server) as the serving UE in the HSS. Network layer configuration is done by the PDN-GW so that mobile can configure itself with an IP address and the default router.

Figure 2.10 describes intra-MME handover process where the UE is handed off from eNB1 to eNB2 that are part of the same MME. It shows the signaling between eNBs, MME and signaling gateway (S-GW).

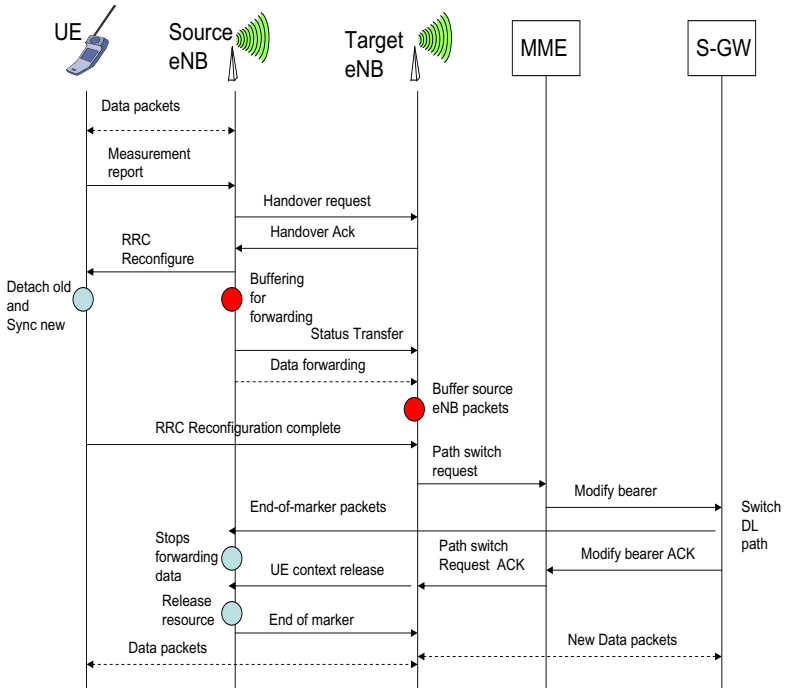


Figure 2.10: Intra-MME handoff call flow

Based on certain policy, the UE sends a measurement report to the source eNB or serving eNB. The source eNB makes a decision based on the measurement report and radio resource management (RRM) function to handover the UE to the target eNB. The source eNB sends a handover request to target eNB and passes the necessary QoS information it presently supports for the UE. This ensures that the target eNB configures the required resources for the UE when it arrives in the target network. A GTP-U tunnel is also set up between the source eNB and target eNB during this time. The mobile is then assigned the radio resources by the target eNB. During this time, the data destined to UE gets buffered

in the source eNB. Once all the details of the UE such as UL (Uplink) PDCP sequence number receiver status and DL (Down Link) sequence number transmitter status are known to the target eNB, it receives the buffered data from the source eNB and buffers it locally for further delivery to the UE at a later point. Once the UE connects to the target eNB successfully, the target eNB informs the MME to do the PATH SWITCH. On receiving the PATH switch request, the MME communicates with the S-GW to modify the bearer through modify bearer request. S-GW transmits the end of marker packets to the source eNB before starting transmission on the new path. This helps the source eNB to stop forwarding packets to the target eNB. Target eNB delivers the buffered packets before delivering packets on the new DL path. After the new path has been established, source eNB releases its resources. Thus, the new data is delivered to the UE via target eNB. Buffering at the source eNB and target eNB help to reduce the packet loss during handoff.

2.7 IP-based mobility

IP-based mobility management techniques can be implemented at several layers of the protocol stack, such as network layer, transport layer, and application layer. IP-based mobility protocols can be used to take care of mobility for 3G- and 4G-based systems. MIPv4 (Mobile IPv4) [Per02c] and its several variants, namely MIP-RO (MIP with Route Optimization), and MIP-RR (MIP with Regional Registration) [Per02b], MIP-LR (MIP with Location Register) [JRY+99], MIPv6 [JPA04], MOBIKE [SE06] are a few of the network layer mobility protocols that were defined within the IETF. Cellular IP [CGK+00], HAWAII (Handoff Aware Wireless Access Internet Infrastructure) [RLS+00], Proxy MIPv6 [GLD+08], IDMP (Intra Domain Mobility Protocol) [DDM+02] are the network layer micro-mobility protocols suitable for intra-domain mobility. Intra-domain mobility refers to a movement scenario when the mobile's movement is confined to the administrative domain. MSOCKS [MB98], TCP-Migrate [SB00], and SCTP (Stream Control Transport Pro-

tocol) [K+03] have been designed to take care of mobility at the transport layer. SIP-based mobility [SW00] takes care of mobility by means of application layer signaling, such as SIP (Session Initiation Protocol) [RSC+02]. HIP (Host Identity Protocol) [MN06] defines a shim layer between the network layer and transport layer to provide terminal mobility in a way that is transparent to both the network layer and transport layer.

I have provided a survey of the related mobility protocols and the issues in [DACS02], [DVC+01]. I have also experimented with several mobility protocols, namely, MIPv4, MIPv6, SIP-based mobility, MIP-LR, and ProxyMIPv6 and verified that these mobility protocols in their current form are not adequate to meet the delay, and packet loss performance requirements of real-time traffic [DKZ+05], [DMD+07] and hence, these protocols will benefit from overall systems optimization.

I briefly describe some of these IP-based mobility protocols and have categorized them into network layer macro mobility, network layer micro mobility, application layer mobility and transport layer mobility.

2.7.1 Network layer macro mobility

Network layer mobility can be categorized into two types: *macro* and *micro*. Macro mobility mechanism takes care of global mobility where the mobile moves between administrative domains. I describe two types of network layer macro mobility, namely Mobile IPv4 and Mobile IPv6.

2.7.1.1 Mobile IPv4

Mobile IP is a mechanism developed for the network layer to support mobility [Per02a]. Originally it was intended for travelers with laptops to provide portability, and later has been adopted by the wireless community. It supports transparency above the IP layer, including the maintenance of active TCP connections and UDP port bindings. A mobile host is identified by a node identifier such as fixed IP (home IP address). When the mobile host

connects to a visited network that is different than the one where its IP address belongs to, its home network forwards packets to the mobile. A router (or an arbitrary node) that is usually known as the home agent on the user's home network forwards the packets. There are two different methods to deliver the packets to a mobile host when it is on a foreign network. With the first method, the mobile host adopts a second (temporary) IP address known as the care-of address (CoA) and registers it with its home agent. When the home agent receives a packet for this user, it encapsulates the packet in another IP packet with the care-of address as the destination address and sends it to the foreign network [Per96a], [Per96b]. Encapsulating a packet within another packet until it reaches the care-of address is known as tunneling. Note that encapsulation adds between 8 and 20 bytes of overhead, which can be significant for voice packets of this size.

The care-of address in the first method is said to be co-located and it can be acquired via services like DHCP (Dynamic Host Configuration Protocol) [Dro97] or its optimized version such as DHCP with rapid commit [PKB05] in a local area network or via PPP [Sim94] in a point-to-point networking environment. With the second method, the mobile host first registers with a foreign agent (FA) in the network it is visiting. The foreign agent sends (registers) its address to the mobile host's home agent as the care-of address of the mobile host. Packets that are intended for the mobile host are sent to the foreign agent after the home agent encapsulates them with the IP address of the foreign agent. After decapsulating these packets, the foreign agent delivers them to the mobile host.

Figure 2.11 shows the functional details of Mobile IPv4. In this specific figure the mobile node moves from subnet 1 to subnet 2 and in the process changes its layer 2 point of attachment, layer 3 point of attachment and either reconfigures itself with a new care-of-address from a DHCP server or uses FA's address as its new CoA.

For the methods outlined above to be able to work, a mobile host needs to be able to learn that it has moved from its home network to a foreign network. For this purpose, home agents and foreign agents advertise their presence periodically in their own broadcast

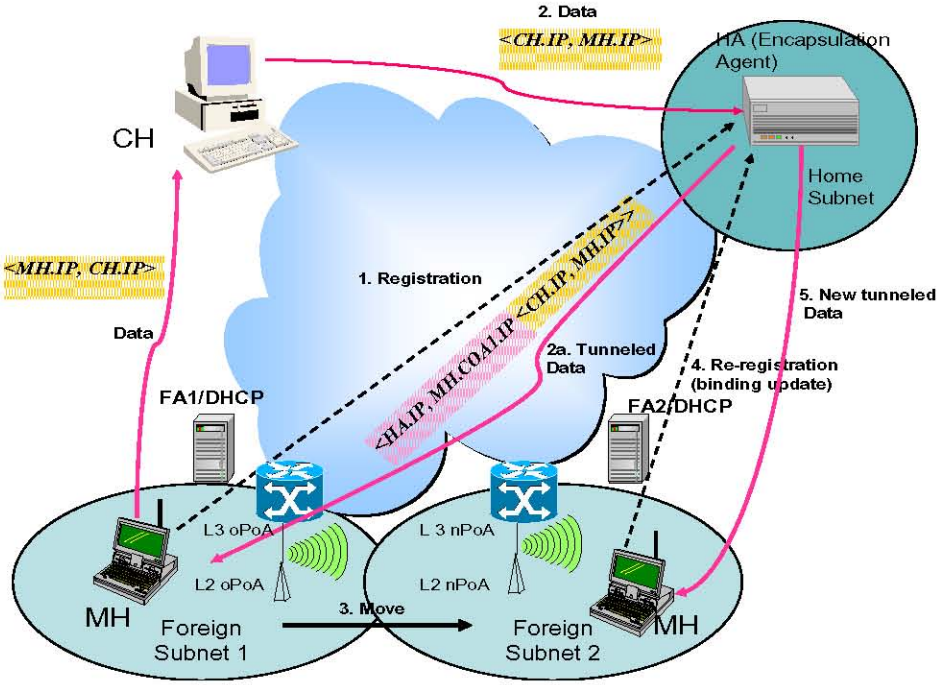


Figure 2.11: Mobile IPv4

domains. A mobile host can also solicit agent advertisements in case these advertisements are absent. Packets from the correspondent host must first travel to the home agent that are later forwarded to the mobile host either by way of the foreign agent or directly. Packets from the mobile host do not have to traverse the home agent; the mobile host sends them as usual with its home IP address as the source address, which is known as triangular routing.

Routing of all incoming packets via the home network may cause additional delays and waste of bandwidth. However, if the correspondent host knows where the mobile host is, it can send packets directly to the care-of address of the mobile host. This is achieved by route optimization [Per02a] process that enables the mobile to send mobility binding updates directly to the corresponding host. Binding updates are sent from a home agent upon request of the mobile, or can be sent upon receiving a warning message from a foreign agent if the mobile host changes location during a communication session. In the second case, the former foreign agent will keep forwarding packets to the new foreign agent until the correspondent host updates its mobility binding cache (known as smooth handoff).

Another optimization proposed is regional registration [CMP03], where the mobile locally registers within a visited domain. In base Mobile IP, a mobile host is required to register with its home agent each time it changes care-of address, thus causing signaling delay for the registration if the mobile host is far away from its home agent. Regional registration attempts to decrease the number of home registrations by maintaining a hierarchical structure of the foreign agents. As long as a mobile host's foreign agent is located hierarchically under a so-called gateway foreign agent (GFA), it is unnecessary to relay registration messages back to the home agent since the home agent has already registered the GFA's address as the care-of address. To make Mobile IP handoffs (i.e., the registration process) more suitable for real-time and delay-sensitive applications, Malki et al. [Mal04] proposed two additional methods. With the first of these methods, called the Network Assisted, Mobile and Network Controlled (NAMONC) handoff method, the mobile host is informed (assisted) by the network that a layer 2 handoff is anticipated. It proposes to use simultaneous bindings (multiple registrations at a time) in order to send multiple copies of the traffic to potential points of attachment before the actual movement. The other method, called the Network Initiated, Mobile Terminated (NIMOT) handoff method proposes extensions to the base mobile IP so that foreign agents can utilize information from layer 2. Specifically, foreign agents use layer 2 triggers to initiate a pre-registration prior to receiving a formal registration request from the mobile host. Both methods assume considerable involvement of information from layer 2.

I have proposed many of the Mobile IP related optimization techniques in Chapter 5.

2.7.1.2 MIP-LR

Mobile IP with Location Registers (MIP-LR) avoid encapsulation of packets [JRY+99] and provide survivable features in case of failure of location registers. In MIP-LR, each subnet may contain a host that functions as a visitor location register (VLR) and/or a host that functions as a home location register (HLR). Each mobile host can be served by multiple

HLRs. Each VLR advertises its presence on its local subnet using agent advertisement messages similar to Mobile IP. When a mobile host is located at its local subnet, it is not registered at either the HLR or the VLR. When the mobile moves to a foreign network it obtains a care-of address (CoA) from the pool of addresses that VLR has. The mobile host registers with the foreign VLR using the CoA it has obtained, which in turn relays the registration to the mobile host's HLR. The HLR returns a registration reply containing the allowed lifetime for this registration; the VLR records the mobile host's CoA and the lifetime and forwards the reply to the mobile host. A correspondent host, wishing to send a packet to the mobile host for the first time, issues a query to the HLR, which returns the mobile host's CoA as well as the remaining registration lifetime. The correspondent host then directly sends the packet to the mobile host's CoA. The correspondent host caches a binding for the mobile host's CoA and uses this binding for subsequent packets destined to the mobile host. The correspondent host must refresh its binding cache by querying the HLR again before the mobile host's remaining registration lifetime expires. In MIP-LR, unlike Mobile IP, HLR can be geographically distributed anywhere. As part of my thesis, I have implemented an extension of MIP-LR using application layer module that does not need any kernel changes. Having an application layer module, it allows the mobile to use a policy-based approach to trigger MIP-LR for certain types of application. Figure 2.12 shows the functionalities of Mobile IP with location registers when the mobile moves from one subnet to another and in the process changes its layer 2 point of attachment and layer 3 point of attachment. In this case, there is no foreign agent in the visited network and it is also not a requirement that the location register needs to be in the home network.

After a mobile host moves, if the mobile host was previously registered at some other foreign VLR, the new VLR de-registers the mobile host at the old VLR. The de-registration is required so that the mobile host's old CoA can eventually be released for use by some other mobile host. If a VLR runs out of CoAs temporarily, it can still issue its own IP address as a CoA, and, when a mobile host registers using this CoA, inform the HLR

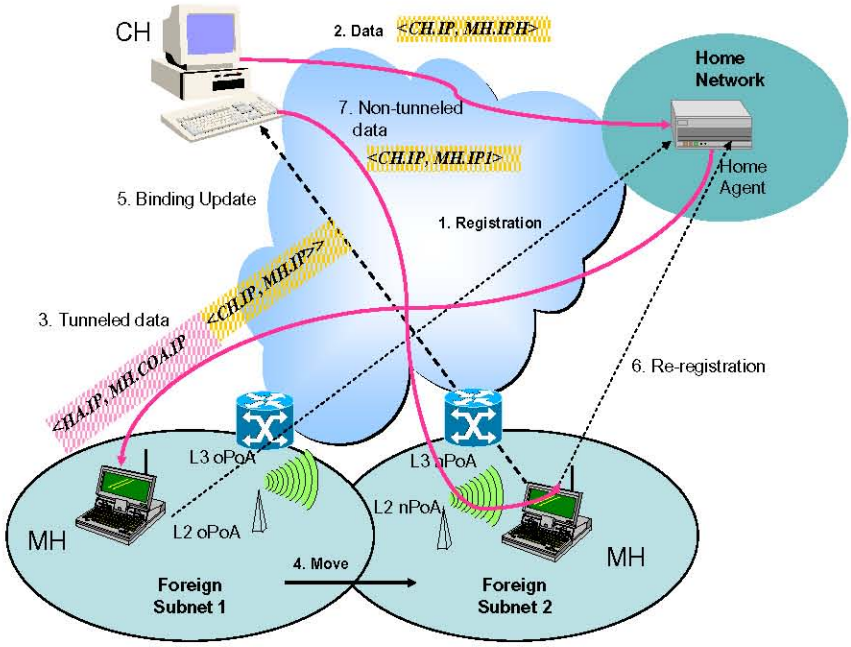


Figure 2.12: Mobile IP with location register

accordingly.

2.7.1.3 Mobile IPv6

IPv6's [DH98] increased address space and inherent support for security and auto-configuration have made it an attractive candidate to support mobility for the next generation Internet. Mobile IPv6 is the protocol to support mobility for IPv6 nodes. Since address auto-configuration is standard part of MIPv6, MH will always obtain a CoA routable to the foreign network. Thus, there is no need to have a foreign agent (FA) in MIPv6. When the mobile node moves to a new foreign network it acquires a temporary care-of-address using stateless auto-configuration [TN98] or via DHCPv6 [DEB+03].

Figure 2.13 shows the functional components of Mobile IPv6. Unlike Mobile IPv4, the visited networks do not have any foreign agent. MIPv6's route optimization feature also enables direct data delivery from the CH to the mobile node.

Although Mobile IPv6 is defined as a network layer approach and one needs to install

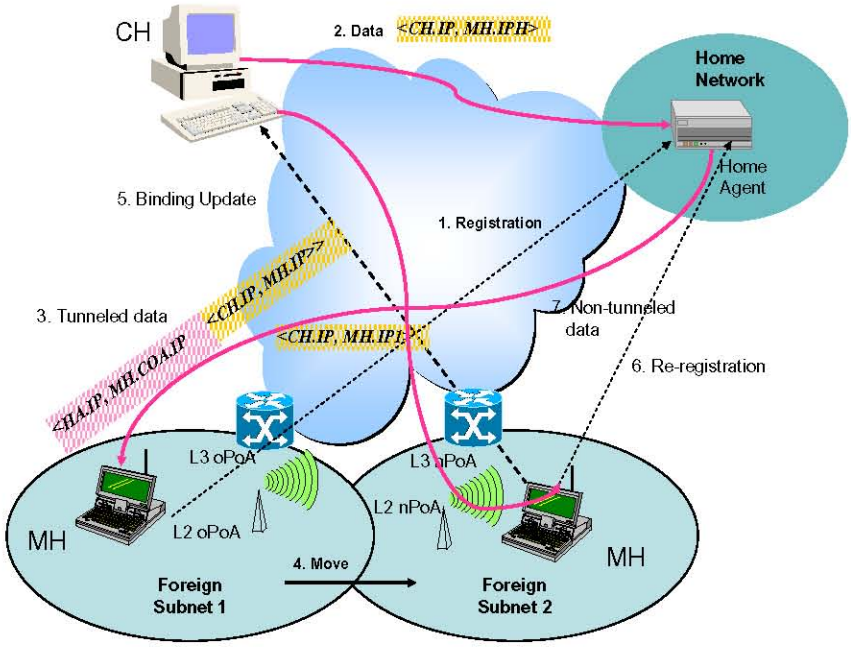


Figure 2.13: Mobile IPv6

MIPv6 stack so as to support mobility in IPv6 space, in future, any standard operating system will come with inherent mobile IPv6 support.

While mobile IPv6 provides a way of making sure the uniqueness of an address as the mobile moves to a new router space it also adds delay for the binding update and binding acknowledgement like mobile IPv4. However, compared to regular Mobile IP, there are inherent advantages in MIPv6. Route optimization is a standard feature of MIPv6, thus, there is no need for the CH to be equipped with additional software like MIP-RO (Mobile IP with route optimization). MH sends binding update directly to the correspondent host (CH) and makes use of the home address destination option as part of the binding update. This allows the correspondent host to keep a binding cache that maps the care-of-address of the mobile with the mobile's home address. For the on-going traffic it avoids triangular routing, thus packets from CH to MH need not be encapsulated but are sent directly to MH with its COA as the source route. However, when a new CH needs to communicate with the mobile for the first time, the packets from CH need to travel to HA and get tunneled to

the mobile host. As the mobile moves during the packet transfer process, the subsequent packets are tunneled directly to the mobile host without being routed via the home agent.

2.7.2 Network layer micro mobility

There are a few network layer micro mobility protocols that are meant to optimize mobility when mobile's movement is confined within a domain. These protocols avoid the overhead associated with tunneling over the air, and reduce the signaling overhead when the mobile's movement is confined to a domain. I describe a few of those network layer micro mobility protocols, namely, cellular IP, HAWAII, IDMP and ProxyMIPv6. I mainly focus on the general mechanisms that they use to optimize the mobility.

2.7.2.1 Cellular IP

Cellular IP [CGK+00] is a micro-mobility management protocol. It separates local and wide area mobility, adopts a domain-based approach, and uses Mobile IP for inter-domain (wide area) mobility. Cellular IP isolates the wireless access network from the core of the Internet via a gateway that acts like a foreign agent and deploy network elements (base stations) specialized for mobility management. Isolating the wireless access from the core is necessary since Cellular IP itself provides an IP forwarding engine replacing IP in the wireless access network. This approach reduces the signaling updates and localizes it within a domain.

Figure 2.14 shows how the packets destined to the mobile host are routed through the cellular IP nodes as the mobile moves between cells within the same domain and between the domains. Packets from a correspondent host are first sent to the mobile host's home agent and then tunneled to the gateway where they are decapsulated. Hosts are identified by their home addresses inside the Cellular IP cloud. Packets generated by mobile hosts are sent to the gateway and later to the correspondent host. Each cellular IP domain is equipped with a gateway router that periodically broadcasts beacon. This beacon is broadcast to all

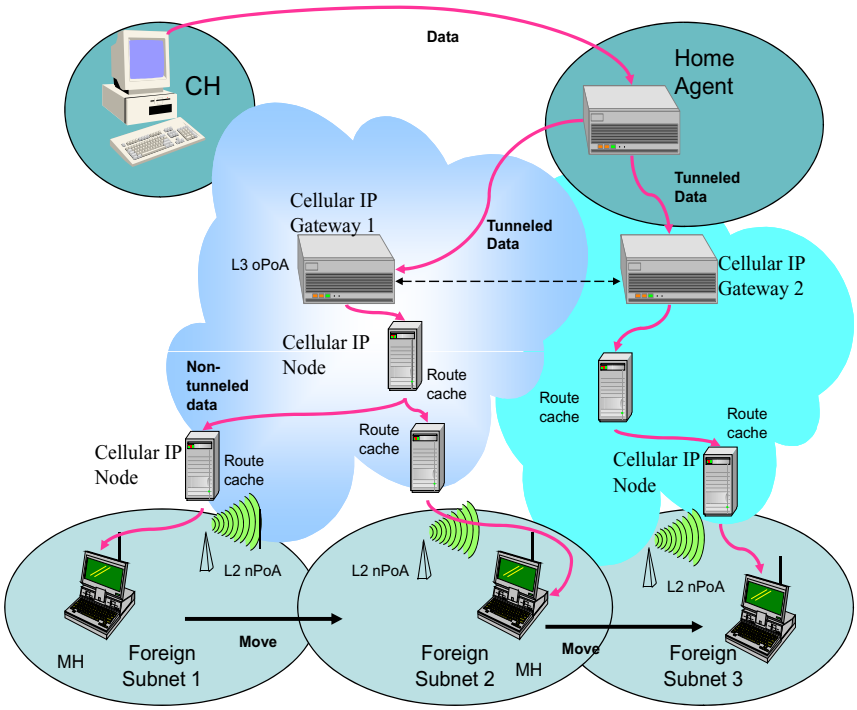


Figure 2.14: Mobility in cellular IP

the cellular IP nodes within the domain and thus each node between the mobile and gateway learns of its neighbor's address in the uplink and uses that information to route any data traffic destined to the gateway. Cellular IP base stations snoop actual data packets sent from mobile hosts to the gateway to cache the path taken by them (actually, base stations record only the host IP number and the neighboring base station from which the packet was received). To route packets from the gateway to the mobile host, base stations use the reverse of this path. Hosts that have not transmitted packets for a while are removed from the routing cache of the base stations. Location-tracking method of mobile hosts depends on whether the hosts are active or idle. An idle host is one that has not received or transmitted a packet for a specific time. It is adequate to maintain the position of idle hosts as a distributed paging cache. To achieve this, a technique known as passive connectivity in cellular telephone systems is mimicked in Cellular IP layer. Base stations are geographically grouped into paging areas, where a paging area may include more than one base station. Idle hosts send infrequent paging-update packets to the gateway. For active hosts, a dis-

tributed routing cache maintains the exact location of each host. Once a mobile host moves to another base station during a call, it sends a route-update packet back to the gateway. New base station(s) record this path accordingly.

2.7.2.2 HAWAII

Handoff-Aware Wireless Access Internet Infrastructure (HAWAII) is another effort to compensate for Mobile IP's inefficiency in supporting intra-domain mobility [RLS+00]. In that sense it is similar to Cellular IP's objective, but unlike cellular IP, HAWAII defines separate control messages to set up host routes on the intermediate routers to route the packets between the domain root router and the mobile host. HAWAII operates on the basic assumption that most user mobility occurs within a domain, therefore optimizing routing and forwarding for efficient support of intra-domain mobility will complement Mobile IP's inter-domain mobility support. Another assumption is that base stations are capable of IP routing. HAWAII segregates the network into a hierarchy of domains. On the top of hierarchy for each domain, there is a domain root router. Packets addressed to a mobile host in a specific domain, first reaches the domain root router and are then sent to the mobile host. As long as the mobile host moves within a domain, it retains its IP address. Once the mobile host moves into another domain it is assigned a co-located care-of address, and the home agent in the home domain tunnels packets to this address. Figure 2.15 shows a sample architecture for HAWAII.

The path (route) between the mobile host and the domain root router is specific to that host. It is established during power-up and updated during movement for that mobile host in the domain root router and pertinent intermediate routers. This information is refreshed periodically by the mobile host, which allows the routers to maintain path state. This idea is similar to Cellular IP and the regional registration in Mobile IP: a physically distant home agent involvement is not desirable each time the mobile host moves. Four path setup methods that can be used to re-establish path states when the mobile host moves within a

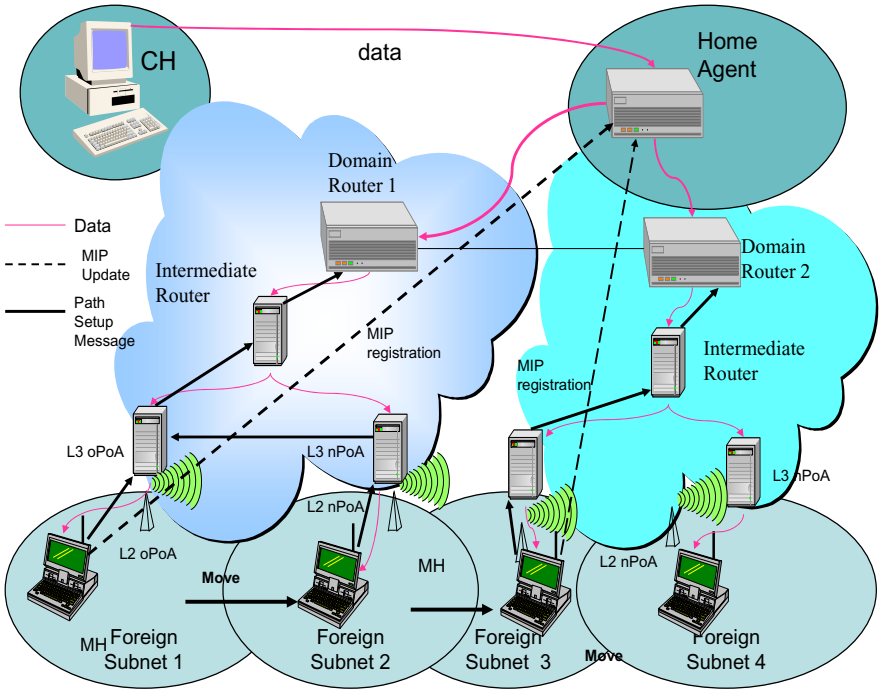


Figure 2.15: Mobility in HAWAII

domain are proposed in this scheme. Two of them forward packets from the old base station to the new base station for a short period (i.e., until the relevant routers update their entries for the specific host). The other two methods do not forward packets during a handoff. Rather, they either bi-cast the packets to two base stations or unicast them for hosts that can simultaneously listen to two base stations. Obviously, HAWAII requires all routers in a domain to be augmented with mobility support so that they are able to handle host-specific path setup messages.

2.7.2.3 TeleMIP

TeleMIP (Telecommunications-Enhanced Mobile IP) is an intra-domain mobility framework which uses two layers of scoping within a domain and is based on IDMP [DDM+02] (Intra Domain Mobility management Protocol). Figure 2.16 shows the basic architecture for TeleMIP. By specifying an intra-domain termination point called mobility agent (MA) it

helps to reduce the signaling updates during the movement within a domain. Mobility agent has the similar functionality like a foreign agent but is placed hierarchically at a higher level. This reduces the signaling traffic due to frequent hand-offs within a domain. Mobility agent acts like a domain-wide anchor point similar to gateway foreign agent (GFA) in MIP-RR. Unlike other proposed intra-domain mobility management schemes, IDMP uses two dynamically auto-configured care-of addresses (CoAs) for routing the packets destined to mobile nodes. The global care-of address (GCoA) identifies the mobile node's attachment to the current domain, while the local care-of address (LCoA) changes with subnet change and identifies the mobile's attachment in a specific subnet.

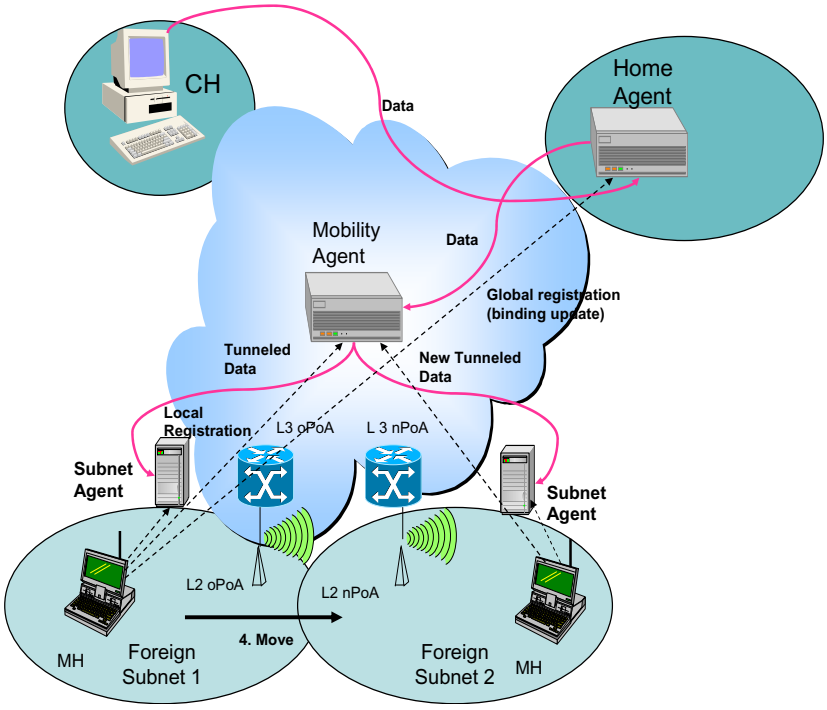


Figure 2.16: Mobility in TeleMIP

Intra-domain updates are sent only up to the MA, which provides a globally valid care-of address (CoA) to the mobile host. TeleMIP reduces the frequency of global update messages since the MA is located at a higher hierarchy than that of subnets, global updates (to home agent, correspondent hosts, etc.) only occur for inter-domain mobility. It also reduces the requirement of additional public addresses (IPv4) by adopting a two-level

addressing scheme.

In TeleMIP the network is divided into domains like Cellular IP and HAWAII. Each domain is identified by the mobility agent's IP address. Domain identifiers are broadcast as part of agent advertisement in each subnet. Each mobility agent is identified with a unique domain identifier. MA's address can also serve as a domain identifier. Network retains control of MA assignment. When a mobile host first moves into a domain, it obtains a global care-of address (mobility agent's address) as well as a local care-of address. MA's global CoA is sent in the registration message to HA. Mobile host also registers itself (using its local CoA) with its MA. There are foreign agents and DHCP servers organized hierarchically at the subnet level under an MA. They provide the mobile host with a locally-scoped address (LCoA), which identifies mobile location within the domain. The mobility agent provides mobile host with a global care-of address that stays constant as long as the mobile stays within the domain.

Each domain is equipped with at least one mobility agent. However, multiple MAs can be provisioned for load-balancing and redundancy within the domain. Mobile host's location is known only up to the MA-level granularity. A mobile host retains the same MA (global care-of address) within the same domain. All packets from the global Internet are tunneled to the MA, which acts as a single point of enforcement. The MA forwards packets to mobile host, using regular IP routing, by using the local CoA (co-located or FA) as the destination. On subsequent movement within the domain, mobile host only obtains a new local CoA. At that point there is no need to update the home agent or correspondent hosts. However, mobile host updates its MA with its new local CoA. With TeleMIP, if packets come from outside the domain, they go through the process of encapsulation twice once at the home agent and once at the mobility agent, thus adding delay to the packet delivery process due to additional processing at the mobility agent.

As part of my thesis, I have designed and implemented fast-handoff techniques associated with TeleMIP. I will describe these fast-handoff mechanisms in Chapter 5.

2.7.2.4 Proxy MIPv6

Advantage of local mobility management is to optimize many of the functions related to mobility and reduce the number of signaling messages over the air. A candidate mobility protocol called Proxy MIPv6 [GLD+08] has been standardized within the IETF to optimize local mobility management operations. This protocol is designed to take care of local mobility and is controlled by the network elements thereby reducing the load on the mobile nodes and number of signaling messages over the air. PMIPv6 does not use any mobility stack on the mobile node, but rather depends upon the proxies on the edge routers to perform the required mobility functions. These proxies are called proxy mobility agents (PMA) and can co-locate with the edge routers that are often called as media access gateway (MAG). As long as the mobile node moves within the same domain that has the same local mobility agent (LMA), the mobile node assumes that it is in a home link. The PMA is responsible for sending the proper mobile prefix as part of the router advertisement for stateless auto-configuration, or it can also act as a DHCP relay agent for stateful auto-configuration of the mobile nodes.

I briefly describe the operations of Proxy MIPv6. When the mobile node moves from one MAG (Media Access Gateway) to another MAG, and its movement is limited within one LMA, the following mobility related operations are performed: layer 2 movement, detection of new link by way of router solicitation, access authentication, profile verification, proxy binding update and address re-configuration.

Figure 2.17 illustrates the network elements associated with ProxyMIPv6 operation. After the mobile node connects to the new point-of-attachment as part of the initial bootstrapping process or after the movement to a new domain, access is authenticated with the designated AAA server. During this process, PMA sends the binding update to the LMA (Local Mobility Agent) with the address of the PMA that is specific to the home prefix of the mobile node. In the absence of a pre-existing tunnel, this process helps to set up a tunnel between the LMA and the respective PMA. The mobile node configures its ad-

dress using the prefix included in the router advertisement and interface-id, which can be assigned by PMA or created by itself. After the movement to the new access network, if the same prefix is advertised by the new PMA, then the IP address of the new mobile does not change. A tunnel is not desirable on the mobile node because it adds extra processing and bandwidth constraints to the wireless hop. ProxyMIPv6-based mobility protocol is preferred when mobility is confined within a domain and wireless service providers do not want to overload the mobile node's stack by setting up a tunnel between the mobile and the HA.

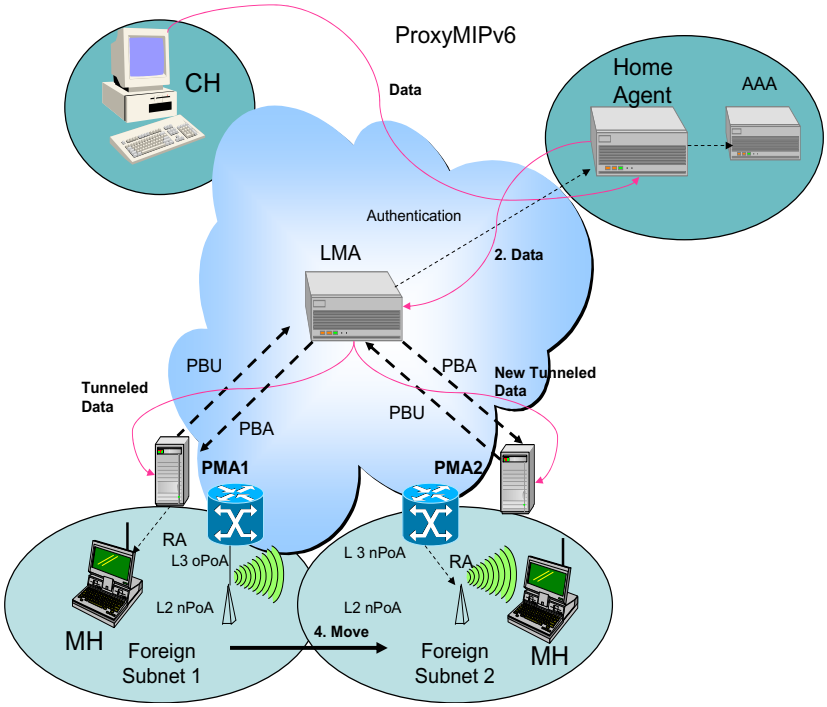


Figure 2.17: Proxy Mobile IPv6

2.7.3 Transport layer mobility

There are a few transport layer mobility solutions. TCP-Migrate [SB00] proposes a new set of migration options for TCP which provide a pure end-system alternative to network layer solutions. With this extension, established TCP connections can be suspended by a

TCP peer and reactivated from another IP address without a third party except for the involvement of dynamic DNS updates for locating the mobile host. However, this approach requires modifying the transport protocol at the end-terminals. MSOCKS [MB98] is another transport layer solution that introduces a proxy in the middle of a network and is built on top of the SOCKS protocol [LGL+96] often used for firewall traversal. Upon movement of the mobile and its address change, the intermediary proxy helps splice the TCP connection. The transport protocol SCTP (Stream Control Transport Protocol) [SXM+00] also adds support for mobility. It has a built-in ADDIP feature [K+03] that provides continuity support when the mobile's IP address changes. Figure 2.18 shows the splicing of TCP connection when there is a break in the communication due to mobile's movement.

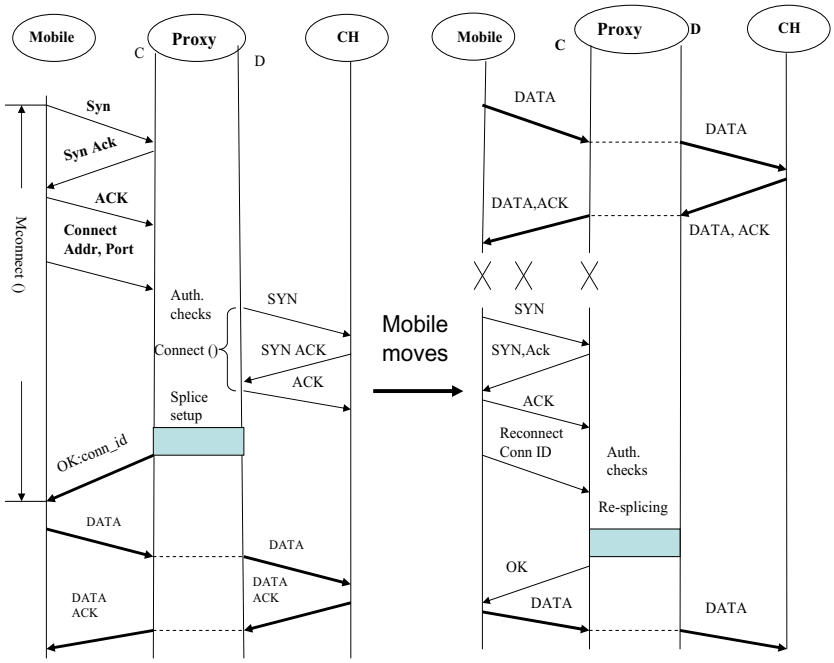


Figure 2.18: MSOCKS-based mobility

2.7.4 Application layer mobility

Application layer mobility uses the Session Initiation Protocol (SIP) [RSC+02] as the underlying signaling mechanism to provide host-based mobility solution. This mechanism

does not depend on the home agent or foreign agent in the network nor does it require to have additional mobility software in the end hosts. Figure 2.19 shows the functional components of SIP-based mobility. Just like other mobility protocols, SIP-based mobility illustrates a similar scenario where the mobile moves between two different subnets and in the process changes its layer 2 point-of-attachment and layer 3 point of attachment. Just like direct binding updates in case of MIPv6 and MIP-LR, SIP-based mobility management uses Re-Invite signal to the correspondent host as a way of direct binding update that helps to update the binding cache in the correspondent host. Thus, the data is delivered using an optimized direct path. Application layer mobility based on SIP can support variety of mobility mechanisms such as personal mobility, service mobility and terminal mobility. As part of my research, I have experimented with SIP-based application layer mobility protocol for both RTP- and TCP-based applications and have designed several optimization techniques for SIP-based application layer mobility protocol. These optimization techniques for SIP-based mobility management are described in Chapter 5.

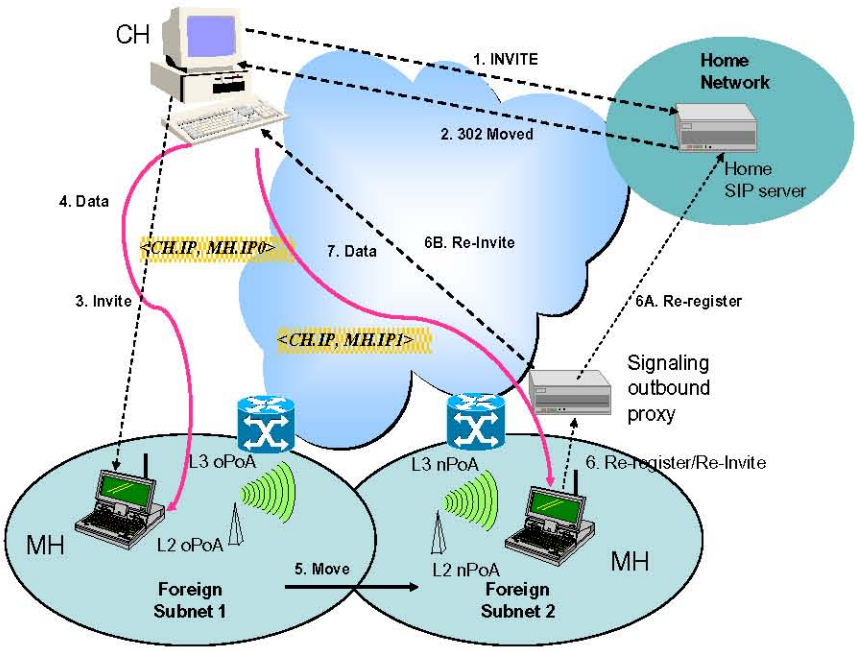


Figure 2.19: SIP-based mobility management

2.7.5 Host Identity Protocol

HIP (Host Identity Protocol) [MN06] defines a new protocol layer between inter-networking layer and transport layer to provide terminal mobility in a way that is transparent to both network layer and transport layer. The proposed host identity namespace takes care of some of the important gaps between the existing IP and DNS namespaces. Host Identity namespace consists of host identifiers (HIs). A host identifier is cryptographic in its nature and it is the public key of an asymmetric key pair. Each host identity uniquely identifies a single host. By doing so, IP addresses can be decoupled from the higher layer applications in a secure manner and the end hosts can be authenticated by a public key as they move around. Public key is one component of an asymmetric cryptographic key pair used as a publicly known identifier for cryptographic identity authentication. By decoupling network and transport layers, applications and transport connections can be made independent of underlying IP address changes thereby enabling alternative solutions to host mobility, host multi-homing, and network address translation. Potential benefit of HIP is that it can be directly integrated with IP security protocols IPSec. Figure 2.20 shows how the end-point address and the locator are separated whenever IP address changes.

Since IPSec security associations (SAs) are bound to host identifiers, not addresses, IP address change does not break the existing transport connections, nor does it trigger a re-establishment of IPSec SAs. HIP-enabled mobility provides optimization technique similar to route optimization technique of Mobile IPv6 by sending direct update to the correspondent host. However, unlike MIPv6, HIP does not have any home network. HIP uses a *Readdress* packet that is similar to Binding Update for MIPv6. However, unlike MIPv6, it inherently secures the readdressing process. In the HIP architecture, the end-point names and locators are separated from each other. IP addresses continue to act as locators. The Host Identifiers take the role of end-point identifiers. It is important to understand that the end-point names based on Host Identities are slightly different from interface names; a Host Identity can be simultaneously reachable through several interfaces.

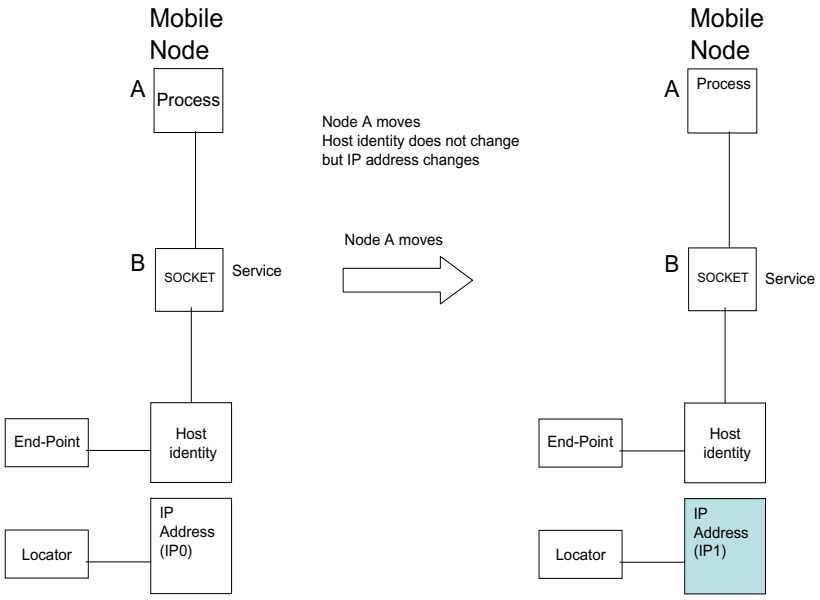


Figure 2.20: Host identity protocol

2.7.6 MOBIKE

MOBIKE (IKEv2 Mobility and Multihoming) [SE06] is an extension to IKEv2 [C.K05] that provides mechanisms so that mobile clients with VPN (Virtual Private Network) connectivity do not need to tear down the existing security association during their layer 3 handoff. In base IKEv2 protocol, the IKE SAs and tunnel mode IPSec security associations are created implicitly between the IP addresses that are used when IKE_{SA} is established. When the mobile moves, its IP address changes giving rise to the need for a new IKE process and new security association between the mobile's new IP address and the VPN gateway. This process results in suboptimal operation and media interruption. However, MOBIKE allows the IP addresses associated with IKEv2 and tunnel mode IPSec Security Associations to modify by initiating additional signaling message such as UPDATE_SA_ADDRESS. As a practical application, a mobile Virtual Private Network (VPN) client could use MOBIKE to keep the connection with the VPN gateway active while the mobile itself changes its address due to change in network point of attachment. Similarly, a multi-homed host could

use MOBIKE to move the traffic to a different interface if, for instance, the one currently being used stops working. MOBIKE is probably the most preferred mobility protocol to take care of mobility for the clients in mobile VPN environment, where the mobile does not need any additional home agent. In Chapter 5, I propose optimization techniques to improve the handoff delay performance for a mobile in a VPN environment.

Table 2.3 shows a qualitative comparison of some of the available IP-based mobility management protocols in the wireless Internet. Only network layer, transport layer and application layer mobility protocols are cited in this comparison. The mobility protocols with an "*" next to it are the candidate protocols that I have experimented with as part of my thesis. I briefly define the metrics against which these protocols are compared. Intra-domain encapsulation involves extra level of encapsulation due to additional tunneling in the network. Inter-domain encapsulation involves tunneling when the mobile moves between the domains. End system changes means whether additional software is needed to make that specific mobility protocol to work. Fast handoff support means whether the handoff delay is reduced.

Table 2.3: Qualitative comparison of IP-based mobility protocols

Mobility Protocol	Intra-domain encapsulation	Inter-domain encapsulation	End system changes	Triangle routing	Network change	Fast-handoff technique	Mobility layer	Mobility type
Mobile IPv4*	X	X	X	X	–	–	Network	Macro
MIPv6*	–	–	X	–	–	X	Network	Macro
MIP-RO	X	X	X	–	–	–	Network	Macro
MIP with FA assisted	X	X	–	X	X	X	Network	Macro
IDMP*	X	X	–	X	X	X	Network	Micro
MIP with LR*	–	–	X	–	–	X	Network	Macro
Cellular IP*	–	X	–	X	X	X	Network	Micro
HAWAII	–	X	X	X	X	X	Network	Micro
MSOCKS	–	–	X	–	X	–	Transport	Macro
TCP-Migrate	–	–	X	–	X	–	Transport	Macro
SIP*	–	–	–	–	–	X	Application	Macro

“X” - Yes “–”- No

2.7.7 Multicast mobility

IP multicasting allows IP packets to be delivered from a single source to a group of receivers that are part of the same multicast group using multicast routing protocol. Multicast packets are generally routed along a single shared tree or multiple source-based spanning trees for efficient content distribution. There are several proposed schemes to provide native IP multicast routing over a wide area network such as PIM [DEF+94], MOSPF [Moy93], DVMRP [WPD88], CBT [BFC93], BGMP [Tha04]. Traditional multicasting techniques do not handle large number of distinct multicast groups and do not provide means to handle multicast when some routers may not be multicast capable, while others are. Recently, there are however alternate techniques being developed, such as Source Specific Multicast (SSM) [HC06], UMTP (UDP Multicast Tunneling Protocol) [Fin03] and AMT [TV+02] that provide multicast support for the non-multicast enabled networks while providing novel ways to support specific application such as content distribution. Quinn and Almeroth [QA01] explain many of the issues associated with multicast deployment over wide area networking. Security, scalability and interoperability are some of deployment issues associated with multicast. However, I focus on mobile receiver issues as part of my thesis. Multicast group "join" and "leave" latencies are some of those deployment issues related to multicast mobility. Joining and leaving the multicast groups are usually handled through IGMP [Fen97], [CDK+02] although I have developed some alternative application layer techniques as part of my thesis and describe those in Chapter 8. In this section, I only highlight the mobility aspects of multicast traffic.

Unlike unicast traffic, multicast traffic distribution is receiver driven, where the mobile receiver makes a request to its access router to join a specific multicast group and router joins the upstream multicast tree. The mobile receiver periodically exchanges its group information with the router using Internet group management protocol (IGMP) [CDK+02] in IPv4 or multicast listener discovery (MLD) [ABO97] in IPv6 network. Mobile multicast introduces several challenges, namely general multicast routing issues, mobile receiver is-

sues, mobile source issues and deployment issues. Movement of a sender and receiver can introduce encapsulation and decapsulation of packets and routing state maintenance due to dynamic creation of multicast tree. Multicast receiver issues include join latency, packet loss, packet duplication, packet out-of-order and leave latency that affect the performance of real-time traffic. Reverse path forwarding (RPF), packet loss, multicast scoping, and source discovery are some of the issues associated with multicast delivery when the source is mobile. I describe some of the multicast mobility related work in this section.

Xylomenos and Polyzos [XP97], Varshey and Chatterji [VC99], and Acharya et al. [ABN95] describe many of the architectural issues associated with mobility support for multicast traffic. Romdhani [RKL+04] et al. categorize mobile multicasting primarily into four categories: home subscription-based, remote subscription-based solutions, hybrid solutions, and non-IP multicast-based solutions. I describe two of these approaches and some of mobile multicast protocols in each category.

2.7.7.1 Home subscription-based approach

Home subscription-based or bi-directional tunneling approach depends on home network and associated mobile IP entities such as home agent, foreign agent and uses a multicast router located in the home network. It puts the multicasting burden on the Home Agent (HA) by creating tunnels between the HA and the mobile or FA. However, tunneling multiple multicast packets to the foreign network is inefficient.

Figure 2.21 illustrates the home subscription-based architecture. Figure 2.22 describes the associated flow. In this architecture, the HA is multicast enabled and is responsible for periodically forwarding multicast group membership control messages to the mobile while it is away. To join a specific multicast group, the mobile node establishes a bi-directional tunnel with its HA and tunnels its membership message (e.g., IGMP) to the HA. It uses the same tunnel header used for routing unicast packets between the MN and HA. When the HA receives the join request it decapsulates and forwards it to the local multicast router on

the home link. The local multicast router (MR) intercepts this membership message and sends the join message to the upstream router on the home network. Once the multicast branch is established, the HA forwards the multicast traffic destined to the mobile over the tunnel. Thus, the HA acts like a multicast proxy for the mobile. When the mobile moves to the new foreign subnet, it does not need to join the multicast tree again, as the HA has already joined the multicast tree on behalf of the mobile. Although this method has the advantage that the mobile does not need to rejoin the multicast tree when the mobile moves from one network to another, this scheme suffers from triangle routing and tunnel overhead resulting in join latency. The HA needs to establish per-MN-tunnel to forward the multicast traffic. HA is also the central point of failure in this scheme.

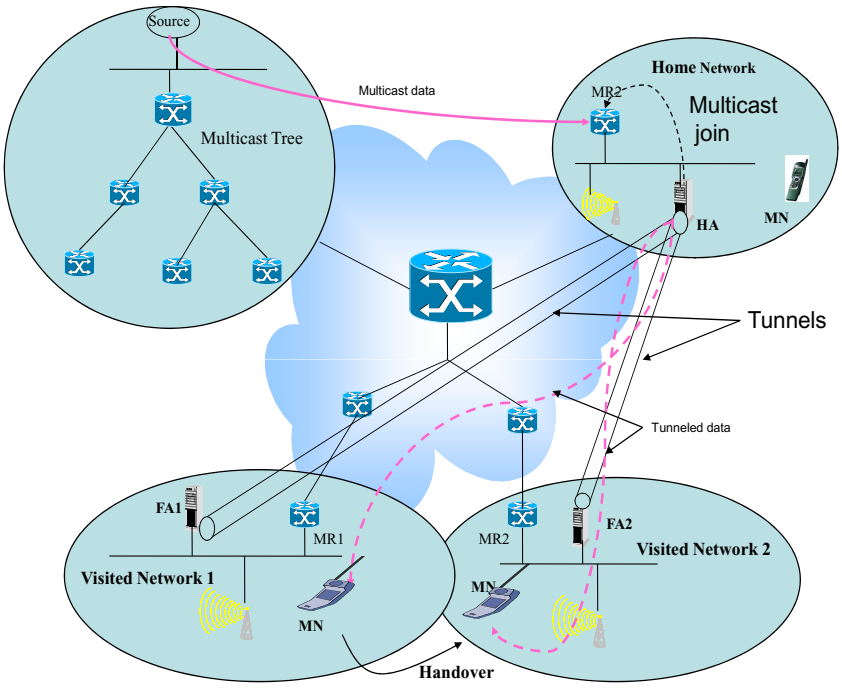


Figure 2.21: Multicast mobility: Home subscription-based

There are several mobile multicast protocols that belong to home subscription-based category and improve upon some of these drawbacks associated with basic home subscription-based approaches. I highlight some of these multicast protocols here. Mobile Multicast (MoM) [WHMB98] proposes to reduce the explosion problem in bi-directional tunneling

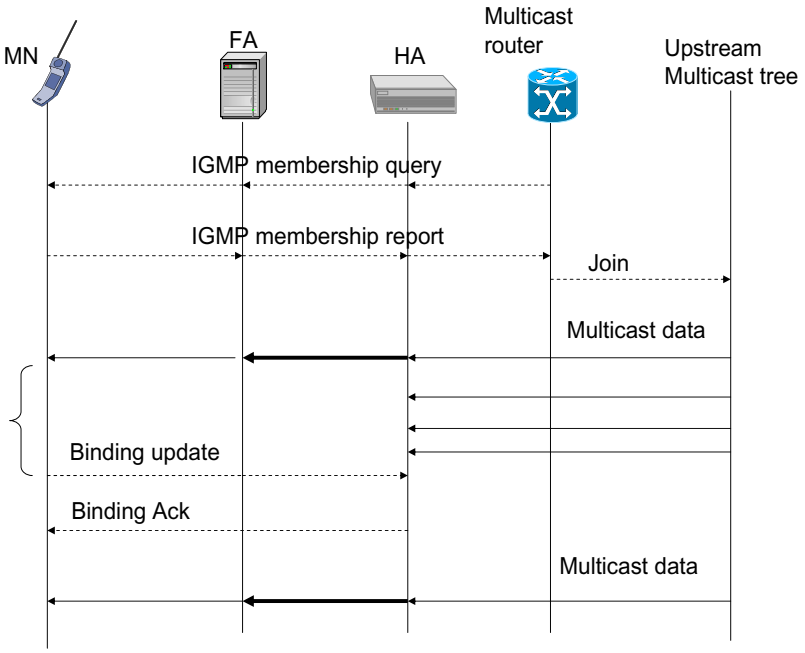


Figure 2.22: Home subscription-based flow

by electing one designated HA called designated multicast service provider (DMSP). MoM protocol uses single tunnel between the HA and the FA instead of using per-MN tunnel. FA in return uses link-level multicast to complete the delivery. Range-based MOM [LW00] takes MoM approach one step further and elects a multicast agent close to FA to tunnel multicast packets to the foreign network. MPDSR (Multicast Protocol With Dynamic Service Range) [YP01] enhances RBMOM protocol in the Mobile IPv6 environment. The main goal of this approach is to determine an optimal service range and avoid service disruption. It does so by introducing two other network elements such as core source node (CSN) and boundary foreign agent (BFA). In order to eliminate possible disruption triggered by the handover procedure, the mobile receiver pre-joins the multicast group prior to the handover.

2.7.7.2 Remote subscription-based approach

Remote subscription-based approach takes the burden off the home agent and does eliminate tunneling and avoids the duplication of multicast packets being tunneled to foreign networks. However, this requires that after each handoff, the user must rejoin a multicast group. In addition, the multicast trees used to route multicast packets will be updated after every handoff to track the multicast group members.

As shown in Figure 2.23, unlike home subscription-based approach, in remote subscription-based approach, the mobile receiver joins the multicast group via a local multicast router (MR1) on the foreign network. The mobile sends its membership report message to the local multicast router (MR1) in the visited network, where the mobile currently resides. The mobile node and the multicast router take care of group management using IGMP for IPv4 and MLD for IPv6. The local multicast router intercepts this message and joins the desired multicast group by joining the upstream routers. Figure 2.24 shows the flow associated with the remote subscription-based approach. Since this approach does not depend upon the home network or any of the home network elements, after the handover, the mobile obtains a new care-of-address (CoA) and sends a new membership report to the multicast router (MR2) in the new network. While MR2 is in the process of joining the upstream multicast tree, the router in the previous network MR1 leaves the upstream multicast tree if there is no other receiver tuned to the multicast group. This is called *leave latency* as defined in Appendix C. After the mobile node moves to the new network, it does not have to wait to complete mobile IP registration before it is able to join the multicast tree in the new network. Thus, the join latency in case of remote subscription-based approach is less than that of home subscription-based approach.

There are several remote subscription-based mobile multicast protocols available that try to reduce the join latency and tunnel convergence problem. Some of these protocols include: remote subscription-based Mobile IP, Mobicast [TP00b], hierarchical SSM [KHHK01], remote subscription with multicast agent, pe-registration with mobility sup-

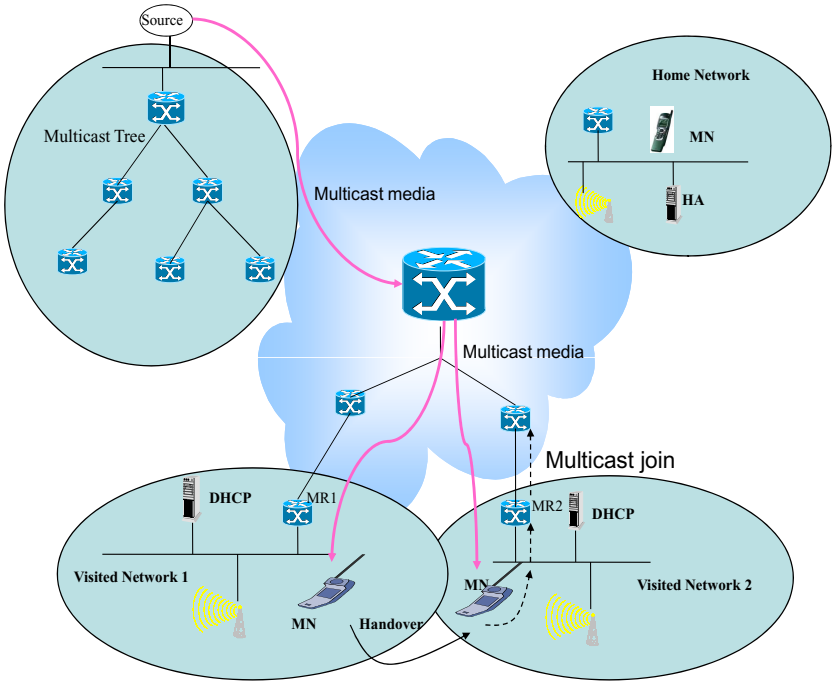


Figure 2.23: Multicast mobility: Remote subscription-based

port agent (MSA) [Wu99], multicast agent protocol (MMA) [SSK00], timer-based mobile multicast protocol (TBMOM) [Par02]. Mobility support agent (MSA) proposes a solution based on pre-registration to reduce the join latency and associated packet loss. Multicast agent protocol (MMA) uses a forwarding technique between the foreign networks instead of pre-registration method. This reduces the packet loss due to join latency. Timer-based mobile multicast protocol (TBMOM) selects a foreign multicast agent (FMA) to store the membership information of the mobile members in the foreign network. TBMOM protocol reduces multicast packet loss since unicast tunnels can be set up between FMAs of different foreign networks. Hierarchical SSM-based (Single Source Multicast) and Mobicast-based approaches provide hierarchical mobility management approaches by introducing hierarchical mobility agents such as border gateway router (BGR) and domain foreign agent (DFA), respectively. Both of these approaches split the multicast path into two level hierarchy and thus limits the multicast traffic distribution in the lower hierarchy whenever there is a movement within a hierarchy.

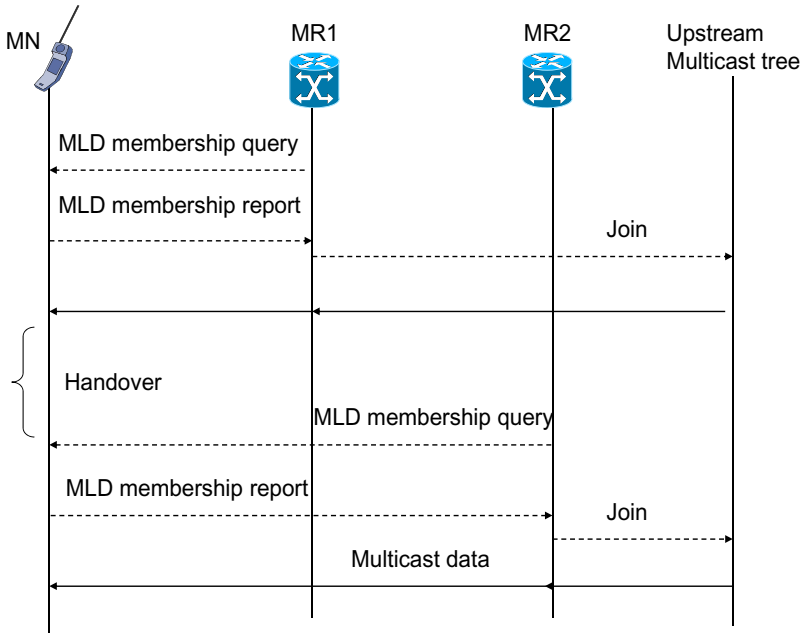


Figure 2.24: Remote subscription-based flow

I provide a summary of these two types of multicast protocols and their qualitative comparisons with respect to some of the performance parameters such as "JOIN" latency, point of failure, tunnel convergence, and hierarchical mobility properties in tables 2.4 and 2.5, respectively. Tunnel convergence problem arises when a number of mobiles belonging to different home networks are anchored at a specific foreign agent. If all those mobiles are subscribed to a specific multicast group, then each HA responsible for the corresponding mobile tunnels the multicast packet to the FA. This will result in duplicate packets for the same multicast group. Number of duplicate packets will increase as the number of mobile hosts subscribed to the same multicast group increases. Thus, it is desirable to reduce the problem due to tunnel convergence. I briefly explain the comparison of "JOIN latency" for both home subscription- and remote subscription-based protocols. For example, average join latency of home subscription-based approach is independent of multicast group size, as each mobile receives multicast datagram from its home agent independently. Average join latency for MOM is moderate and is similar to home subscription-based approach.

However, disruption in MOM protocol decreases with the increasing in group density due to the sharing of DMSP by the visiting group members of the same group in the same foreign network. Compared with home subscription and MOM, remote subscription (RS) approach and multicast agent approach provide less disruption since access to the multicast backbone is much closer. However, multicast agents have a lower disruption than remote subscription when multicast group size is small. As the group size increases, remote subscription method does better than the multicast agent approach. However, at a very high group density, both the approaches offer same performance as the probability of finding other group members in a new IP subnet approaches 1. Mobility support agent (MSA) approach reduces the "join" latency to a great extent due to its pre-registration ability before the handoff.

Table 2.4: Multicast mobility - Home subscription

Protocols	JOIN latency	Point of failure	Hierarchical mobility	Tunnel convergence	Key differentiator
Mobile IP Home Subscription	Remains same with group size	Home Agent	No	One tunnel per MN	-
MOM	Reduces with increase in group size	Designated Multicast Service Provider	No	One tunnel for all MNs	Introduces DMSP
RBMOM	Smaller than MOM	Designated Multicast Service Provider	No	One tunnel for all MNs	Introduces service range
MPDSR	Least handoff delay	BFA	No	One tunnel for all MNs	Reduces tunnel length

Multicast mobility protocols in their current form suffer from performance issues due to tunnel overhead, join, and leave latency. Thus, optimization techniques can be applied

Table 2.5: Multicast mobility - Remote subscription

Mobility Protocols	JOIN latency	Point of failure	Hierarchical mobility	Tunnel convergence	Differentiator
Remote subscription	Less than home subscription	FA or DHCP	No	N/A	
Multicast Agent	Same as RS	MA	No	Tunnel between FA and MA	Uses coordinator MA between FAs
MSA	Less than RS (pre-registarion)	MSA	No	No tunnel	Pre-registration
MMA	Less packet loss due to forwarding	MA and MF	No	One IP tunnel (MA and MF)	Forwarding technique reduces packet loss
TBMOM	Less than RS	DMSP	No	Does not use IP tunnel	Uses hybrid forwarding approach
Hierarchical SSM	Less delay than RS	BGR	Yes	IP tunnel between source and BGR	Reduces JOIN latency for micro mobility
Mobicast	Less packet loss buffering	DFA	yes	Does not use IP tunnel	Buffering solves packet loss

to the existing multicast mobility protocols to reduce the join latency and leave latency. In Chapter 8, I present some of the optimization techniques that I have developed for multicast stream delivery that reduces the join latency in a hierarchical multicast environment.

2.8 Concluding remarks

A careful survey and analysis of the handoff processes for the cellular mobility protocols and several IP-based mobility protocols extrapolate the common functions that are required to complete a handoff event. Since many of the functions such as discovery, authentication are access dependent, one needs to take into account the access characteristics of the

multi-layer handoff operations for the IP-based mobility while designing the optimization techniques. Lessons learnt from optimization techniques from cellular mobility can easily be applied to improve the optimization techniques for IP-based mobility. For example, mobility proxy-assisted forwarding technique for IP-based mobility benefits from GSM's MSC-anchored forwarding techniques to reduce packet loss by forwarding the media from the previous network. Soft-handoff technique for CDMA-based cellular networks can be used as guidelines while designing IP-based fast-handoff mechanisms by applying bicasting or multicasting mechanisms. Thus, while designing any new mobility protocol for next generation networks, or proposing a new optimization technique, it is very important to investigate the abstract handover primitives and study the optimization techniques for cellular mobility.

Chapter 3

Systems analysis of mobility events

In this chapter, I analyze the mobility event resulting out of a mobile node's handoff. In particular, I analyze the discrete operations of a mobility event in order to design a formal mobility systems model. However, I limit the analysis of the mobility systems model to infrastructure-based mobility only. Infrastructure-based mobility model assumes that the layer 2 and layer 3 network components of the core networks, namely access points, routers, and servers are not mobile and only the last hop of the access network changes. Figure 3.1 shows different functional components in an IP-based infrastructure-based mobility environment. This figure consists of layer 2 and layer 3 points of attachment, configuration agent, authentication agent, authorization agent, and signaling agent that perform the primitive handoff operations. During a mobility event, these network components are engaged in a distributed communication to take care of the handoff related operations. Figure 3.1 also shows how the mobile node changes its point of attachment as it moves between the layer 2 point of attachment, layer 3 point of attachment, different administrative domains and different access technologies. The markings A, B, C, and D in Figure 3.1 show the location of these mobile nodes as the mobile node connects to different point-of-attachment during its movement. The mobile starts from location marked as A and moves to locations marked as B, C, and D. During this process, the mobile is subjected to layer 2, layer 3 and

administrative domain handoff (e.g., handoff from domain A and domain B). Each of these handoffs (e.g., L2 handoff and L3 handoff) involves different types of handoff operations and different amounts of delays. For example, a layer 2 handoff contributes up to 900 ms delay in 802.11 environment, a layer 3 handoff contributes up to 4 second delay that is inclusive of layer 2 delay and an administrative domain handoff involving heterogeneous access results in a handoff delay up to 18 seconds [DKZ+05]. The networks N1 and N2 are shown as networks with different access technologies (e.g., 802.11 and CDMA). Figure 3.2 shows three different audio output from the mobile when it is subjected to layer 2, layer 3 and heterogeneous handoff. These output waveforms represent the media disruption due to delays contributed by the handoff related operations. Thus, it would be useful to investigate the delays related to each type of handoff operations. In this section, I analyze the delays due to handoff related operations at different layers.

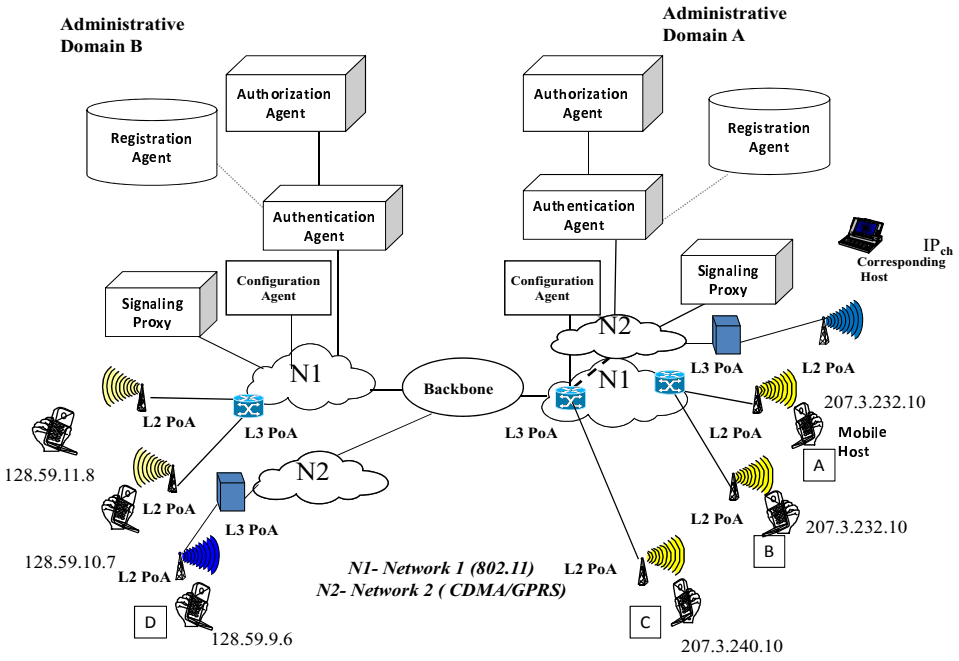


Figure 3.1: Functional components of infrastructure-based mobility

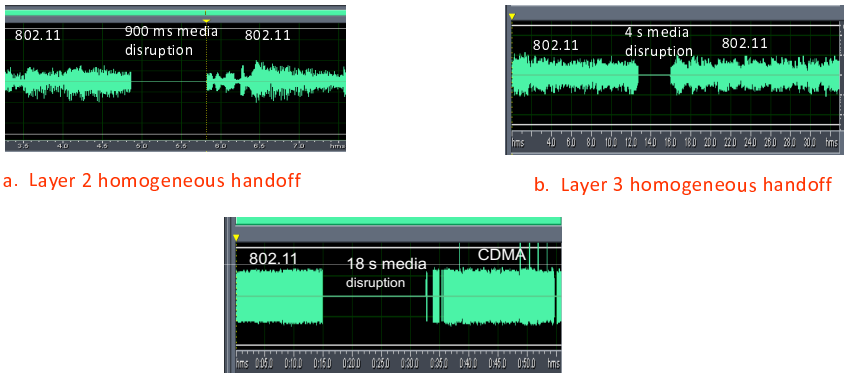


Figure 3.2: Media disruption due to handoff related operations

3.1 Summary of key contribution and indicative results

Currently, there is no work that provides a systematic analysis of the handoff processes and the associated sub-processes that are executed in all the layers during an IP-based handoff. Without this systematic analysis of the handoff processes, it is difficult to determine the data dependency among the handoff processes that could lead to the design of potential optimization techniques.

Based on the comparative analysis of mobility protocols in Chapter 2, I systematically analyze each of these handoff processes for an IP-based 4G network. This systematic analysis of the abstract functions highlights how these mobility functions are performed across several layers in an IP-based network. For example, a discovery function at layer 2 implies discovering a layer 2 point of attachment such as an access point while discovering a layer 3 point of attachment involves discovering a default router. Similarly, authentication, security association, and configuration are some other handoff components that take place across all layers. I analyze the basic primitive operations that are part of each of these handoff functions. Based on this handoff analysis, I built several un-optimized handoff systems such as MIP-based network layer mobility (MIPv4, MIPv6), and SIP-based application layer mobility and demonstrated experimental results for each of these un-optimized handoff components that show delays associated with each of these mobility components

(e.g., discovery, authentication, configuration).

From the systematic analysis (decomposition) of the handover operations and experimental results from some of the indicative un-optimized handoff systems that I built, it is possible to determine the delays associated with each of the primitive operations associated with the handoff event. Unlike other experimental analysis by Vogt et al. [Vog06] that were meant only for one type of mobility protocol (e.g., MIPv6), my experimental analyses are based on both network layer and application layer mobility protocols and work across access techniques (e.g., CDMA, 802.11). Comparative analysis of these experimental results from different mobility protocols verify the mapping between the handover delays with the common abstract functions independent of the types of mobility protocols and access mechanism.

In the rest of the chapter, I analyze the handover components and map these to different layers, demonstrate the experimental results from several un-optimized mobility protocols that I have experimented with and map these delays to respective handoff components.

3.2 Introduction

In order to determine the primitive operations that take place during a handoff event, I investigated and analyzed several cellular and IP-based mobility protocols at different layers. I have described the details of handoff operations for a set of cellular and IP-based mobility protocols in Chapter 2. In Table 3.1, I summarize how the various cellular and IP-based mobility protocols perform similar set of operations to support a handoff event. In particular, I illustrate how these mobility protocols carry out different handoff related functions, namely network discovery, resource discovery, network detection, configuration, authentication, encryption, binding update and media rerouting. The column headings in

Table 3.1: Mapping of basic operations of a mobility event

Mobility Type	Access Type	Network Discovery	Resource Discovery	Triggering Technique	Detection Technique	Configuration	Authentication	Encryption	Binding Update	Media Rerouting
GSM	TDMA	BCCH	FCCH	Channel Strentgh	SCH	TMSI	SRES A3	DES	MSC Control	Anchor
WCDMA	CDMA	PILOT	SYNC channel	Channel strength	Frequency	TMSI	SRES/A3	AES	Network Control	Anchor
IS-95	CDMA	PILOT	SYNC channel	Channel strength	RTC	TMSI	Diffie-Hellman CAVE	AES	MSC	PDSN MSC
802.11	CSMA/CA	Beacon 11R	11R 802.21	SNR at Mobile	Scanning Channel number SSID	SSID Channel number	802.1X EAPoL	WEP WPA 802.11i	Associate	IAPP
Cellular IP	Any	Gateway beacon	AP beacon	AP beacon ID	GW Beacon	MAC Address	IKE	AES	Route Update	Intermediate router
MIPv4	Any	ICMP RA, FA advertise ment	ICMP	FA advertise ment,L2 assn. isted	FA advertise ment	FA-CoA Co-CoA	IKE PANA EAP	AES	MIP Registra tion	FA RFA HA
MIPv6	Any	Stateless RA Proactive	CARD 802.21 802.11R	Router Adv.	Router Prefix	CoA	IKE PANA EAP	AES	MIP Update MIP RO	CH MAP HA
SIP-based mobility	Any	Stateless RA ICMP Router	802.21 802.11R	L3 Router Adv.	Router Prefix ICMP	CoA AOR Register	IKE EAP Invite Exchange	AES SRTP	Re-INVITE	B2BUA CH RTPTrans

Table 3.1 represent the primitive operations that are part of a handoff event for different types of mobility protocols that are represented in the row headings.

Operational aspects of these functions play an important role in the design of any specific mobility protocol. For example, the operational aspect of a discovery function could include many factors like, number of messages between the mobile and the server, discovery mechanism (e.g., network layer or application layer). Performance of any mobility protocol depends upon how efficiently these functions operate. In order to get a better understanding, I have performed a comparative analysis of application layer mobility protocol (e.g., SIP-based mobility) and network layer mobility protocol (e.g., MIPv6) and have illustrated how each of these primitive operations is carried out for both of these mobility protocols [DSC+06]. I briefly describe how some of the handoff operations are taken care of by each of these two mobility protocols.

3.2.1 Comparative analysis of mobility protocols

In Chapter 2, I have described mobility management techniques for both SIP-based mobility and mobile IPv6 (MIPv6). However, in this section, I compare how some of the basic operations such as discovery, registration, binding update, configuration, location management, tunneling, and security operations are done for each of these two mobility protocols.

In SIP-based mobility management, as part of the registration process, the mobile updates its IP address with the visited SIP proxy or home proxy. In case of MIPv6, the mobile sends the binding update to the home agent and corresponding host. Thus binding update to the home agent can be regarded as the registration process for MIPv6.

Both MIPv6 and SIP-based mobility do not require any foreign agent in the network and thus use co-located care-of-address as the new identifier. SIP-based mobility for IPv4 networks mostly depends upon DHCP for IP address configuration. Both SIP-based mobility for IPv6 networks and MIPv6 however use stateless auto-configuration to configure a layer 3 identifier.

In the absence of route optimization, mobile IPv6 tunnels payload packets between the mobile node and the home agent in both directions. This specific tunneling mechanism uses IPv6 encapsulation as specified by RFC 2473 [CD98]. In addition to the extra headers assigned to the original packet, additional time is spent due to processing of encapsulation and de-capsulation. SIP-based mobility on the other hand does not make use of tunneling as the media travels directly between the CH and MH. Thus, processing delay due to encapsulation and tunneling overhead is avoided when SIP-based mobility is used.

Binding update has been defined in Chapter 2. In IP-based environment, it is the process of notifying the correspondent node or other networking node such as home agent about the new layer 3 identifier of the mobile so that the data can get forwarded to the new address of the mobile after the handoff. In case of Mobile IPv6, as the mobile obtains a new care-of-address either via stateful DHCP server or stateless auto-configuration it notifies the correspondent host and the home agent. Since route optimization is an inbuilt mechanism for MIPv6, the new data during mid-session mobility does not need to get rerouted via home agent. On the other hand, SIP-based mobility sends the binding update as part of its Re-INVITE signal to the correspondent host only since there is no home agent for SIP-based mobility.

Mobile IPv6 by itself does not provide fast-handoff mechanism. However, there are extensions to Mobile IPv6, namely, FMIPv6 [Koo05] and HMIPv6 [SCeMB06] that provide fast-handoff during mid-session mobility. FMIPv6 provides fast-handoff mechanism by introducing several reactive and proactive mechanisms, whereas HMIPv6 introduces a mobility anchor point (MAP) to take care of intra-domain mobility. Similarly, SIP-based mobility has been extended to provide fast-handoff by introducing a network entity called B2BUA (Back-to-Back-User Agent) in the hierarchy to limit the binding update or by forwarding the transient traffic from the previous network.

Both of these mobility protocols provide security for the signaling and data. MIPv6 takes advantage of network layer security such as IPSec to protect the signaling between

the mobile and home agent. It can also use IPSec tunnel instead of IP-IP tunnel between the mobile and home agent to carry the tunneled data. SIP-based mobility on the other hand can provide a multilayer security. It can either choose IPSec to provide security at layer 3 for both signaling and media (RTP) or it can use S/MIME to secure the SIP signaling and use secure RTP (SRTP) to secure the media stream. However, SRTP can also be used to protect the data in case of MIPv6, but it will need a separate key distribution architecture unlike SIP-based mobility where the key is distributed by Invite exchange method.

Mobile IPv6 does provide return routability support by using CTI (Care-of-Test Init) and HTI (Home Test Init) messages. This actually verifies the new care-of-address of the mobile before the binding update is sent out. While it helps to avoid session hijacking etc., it does add delay to the binding update procedure. SIP, on the other hand, does not support inherent return routability testing, but new care-of-address of the mobile can be verified by using cryptographic technique such as SIP identity [PJ06].

3.3 Analysis of handoff components

Based on the analysis of the basic operations associated with both IP-based mobility protocols and cellular protocols, I categorize the handoff process into six main phases, namely, ***network discovery and selection***, ***network attachment***, ***configuration***, ***security association***, ***binding update*** and ***media reroute***. Figure 3.3 decomposes the mobility event into several processes and sub-processes.

Each of these operations involves several network elements, namely mobile node, access point, router, server and correspondent node. Figure 3.3 shows how each of these sub-operations involves several components in the network at several layers. For example, a layer 2 discovery involves interaction between the mobile node and the access point, whereas layer 3 discovery involves mobile node, access point and the router in the network. On the other hand, a binding update involves the core components in the network, such

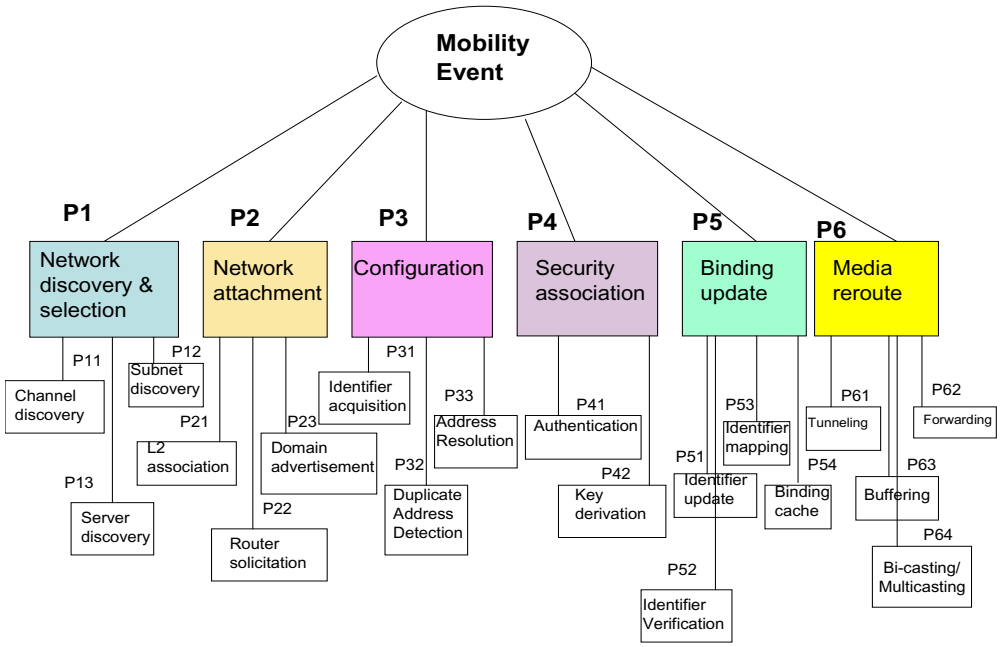


Figure 3.3: Systems decomposition of handoff

as home agent, server and core routers in the network. I describe below these handoff operations in details.

Handover initiation or handover preparation precedes a handover process. Handover initiation process does not contribute to the handover delay. Depending upon who initiates the handover and who has the primary control over the handover, a handoff process can be either mobile-initiated handover, network-initiated handover, mobile- controlled handover or network-controlled handover [EE04]. During this phase, the mobile or the network node determines the need for the handoff based on the measurement on the mobile, and initiates the preparation for the handoff operation. For example, during a mobile-initiated handoff, based on Signal-to-Noise Ratio (SNR) or the quality of service of the media traffic, the mobile makes the decision about the impending handoff and starts the network discovery process to determine the best available network to connect to. A handover process may result in re-initiation of layer 2 or layer 3 operations based on whether MN loses the layer 2 or layer 3 connectivity, respectively. For example, in case the mobile does not receive the layer 3 router advertisement, it initiates a layer 3 handoff process without necessarily

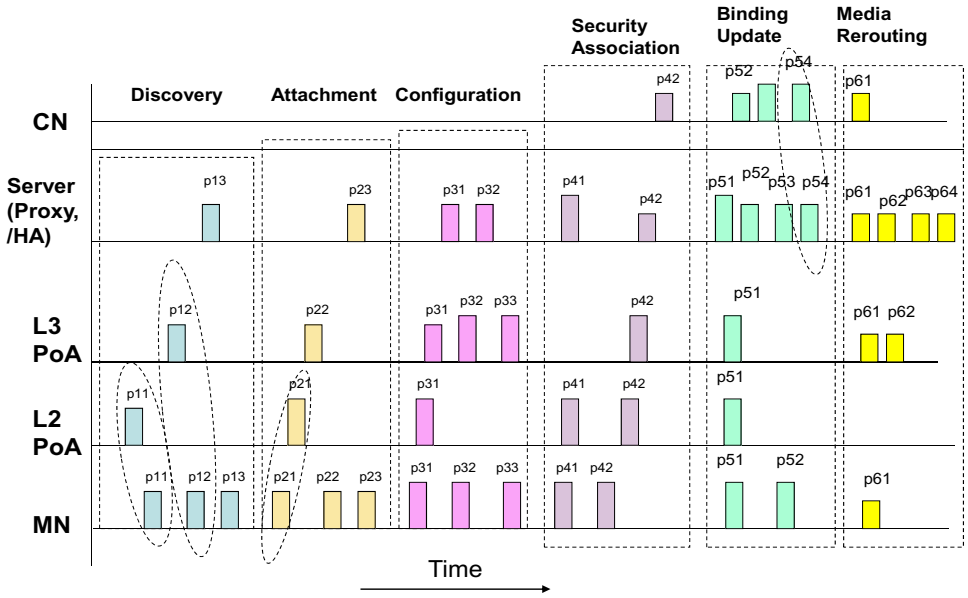


Figure 3.4: Handoff functions distributed across network elements

disturbing the layer 2 association. On the other hand, a layer 2 unreachability event resulting out of low signal-to-noise ratio or non-receipt of beacon will lead to handover initiation process in layer 2 that may or may not involve handoff at layer 3.

The IEEE 802.21 [802b] working group has defined several event service primitives as part of its Media Independent Handover Function (MIHF). “Link Up”, “ Link Going Down”, and “Link Down” are some examples of such primitives that can be used to provide the status of lower layer events and expedite the handoff process. For example, since the layer 2 association takes place before any other upper layer operations, it helps to send a layer 2 event notification, such as“Link up” to execute the upper layer mobility function like layer 3 configuration. Signal-to-Noise Ratio (SNR) threshold can possibly generate such an event notification. Thus, an event notification from lower layer helps to establish a successful link in the new point of attachment in an expedited manner.

3.3.1 Network discovery and selection

The very first phase of the handoff process is network discovery and selection phase. During this phase, the mobile or network discovers possible new points of attachment. This phase involves discovering both the neighboring networks and the required resources for handoff within the network. Once the target network is discovered, several resource parameters within the target network are retrieved including channel number, bandwidth, encryption algorithm, authentication server, registration server, and configuration server. The resource discovery process helps the mobile to associate with proper channel number and proper authentication parameters in the new network so that it can communicate successfully. Jari et al. [AAKB08] provide an overview of network discovery and selection, and its applicability to handoff. Based on the type of access technology, the discovery process can be passive, where the node listens for network announcements, or active where the node solicits network announcements. Different types of discovery process take place for each type of network, such as discovery of a new Location Area (LA) for GSM or discovery of new Routing Area (RA) in case of GPRS. For IP-based networks, discovery process can span across all the layers and include layer 2 PoA (Point of Attachment), layer 3 PoA, subnet or domain discovery. Here a domain is defined as an administrative domain.

Each layer 2 access technology provides different means of out-of-band mechanism for discovering the networks and resources. For example, GSM uses BCCH (Broadcast Control Channel), CDMA uses pilot channel and 802.11 uses active scanning and passive scanning to discover the new point-of-attachment. Based on the type of access characteristics of the target networks, discovery of the appropriate network takes different amounts of time. For example, in IEEE 802.11 network, without any optimization, active scanning operation takes about 500 ms [MSA03] including the channel probe delay, authentication and association delays. Network discovery timing in GSM network is determined by the frequency of broadcast channel. In case of IP-based mobility, some of the upper layer detection mechanisms, such as foreign agent advertisement or router advertisement can help

discover the servers in the network.

Network selection is a process by which a mobile node or a network element analyzes the information discovered about its neighboring networks, and then selects a network to connect to. The selection may be based on criteria such as required QoS, cost or user preferences. An appropriate selection mechanism helps the overall resource optimization process and increases the probability of successful handoff to the target network.

3.3.2 Network attachment

After the mobile has selected the target network, it attempts to connect to the new network point of attachment. Network attachment can take place in several layers. A mobile attaches to an 802.11 access point by means of layer 2 association and connects to a router by means of layer 3 association. Event notification from the lower layers, such as the availability of the new point of attachment or sudden lapse of an existing connection is usually passed to the upper layers in order to initiate the subsequent handoff related functions in an expedited manner. The IETF is currently working on standardizing a protocol for detection of network attachment called DNA [Cho05] that involves mechanisms both at layer 2 and layer 3 and can notify the upper layer about the network detection.

3.3.3 Configuration

Configuration phase helps to prepare the re-connection path of the mobile. During this phase, the mobile obtains a temporary identifier in the network the mobile is visiting, discovers servers such as outbound SIP server and default gateway, and maps the address with the appropriate network entity in the network. Configuration phase can be split into the following sub-phases, namely *identifier acquisition*, *duplicate address detection* and *address resolution*. During identifier acquisition process, the mobile acquires a new temporary L2 or L3 connection identifier at the new point of attachment of the network. For example, it could include Care-of-Address (CoA) in case of IP-based mobility or TMSI

(Temporary Mobile Subscriber Identity) in case of GSM. In IPv4-based networks, the mobile uses DHCP server to obtain the care-of-address, whereas in IPv6-based network the mobile obtains the address either from a DHCP server or using stateless auto-configuration method.

Duplicate address detection process tests the uniqueness of the mobile's address in the network. Address uniqueness testing is performed differently by IPv4 and IPv6 networks. For example, when an MN uses stateless address auto-configuration to assign an IPv6 address to its interface after getting a new RA (Router Advertisement), it sends a neighbor solicitation on the local link in order to verify whether any other node on the link has the same address. When the pre-determined time elapses and the MN does not receive any reply, the MN assumes that no other node on the link has this address and finally assigns that address to its interface. IPv4 node uses ARP broadcast to determine if any other node in the network has the same address.

Once the address uniqueness has been determined, the mobile and the default router keep the mapping between the IP address and MAC address of each other using address resolution mechanism. This address resolution mechanism helps the mobile and the default router to communicate with each other at layer 2 in the same subnetwork.

In general, completion of these processes requires a series of signaling messages between the mobile and servers in the network and thus contributes to the handoff delay.

3.3.4 Security association

A security association is defined as a secure channel between two endpoints that applies a security policy and keys to protect the information. Before a new communication path is established between the end-points, the mobile node needs to authenticate itself and then establish security association with other network nodes, such as routers, access points or the correspondent host. Establishing a security association involves *authentication and authorization* and *key derivation* procedures. The authentication and authorization processes

allow the mobile to get access to the network resources. An authentication process includes exchange of messages between the mobile, authenticator and authenticator server e.g., AAA server in the network. Successful authentication process generates the key that can be used to encrypt the data. Encryption process itself is considered to be separate from authentication and does not affect the handoff delay, unless the type of encryption algorithm changes after the handoff. However, the process of deriving the encryption key contributes to the handoff delay.

Security association may take place at several layers of the protocol stack. For example, during a layer 3 movement in an 802.11 access network, EAP messages need to be carried between the mobile node, authenticator and authentication server to generate the shared master key followed by a 4-way handshake between the mobile node and the access point to generate the PTK (Pairwise Transient Key) that can be used for encryption at layer 2. In an IP-based network, ISAKMP (Internet Security Association and Key Management Protocol) [MSST98] defines the mechanism for establishing the security association at layer 3 using IPSec. An IPSec security association is defined by the encryption algorithm, the authentication algorithm, and the shared session key.

Each mobility protocol also uses different mechanism for authentication. As shown in Table 3.1, the mobile uses SRES and A3 algorithm for the authentication in GSM. For 802.11 access networks, the mobile can use open system authentication or shared-key authentication WEP or IEEE 802.11i-based authentication in layer 2. Fathi et al. [FKC+05] demonstrate how different authentication mechanisms affect the association and transport delay at layer 2. Similarly, layer 2 access independent network layer authentication protocol such as PANA (Protocol for carrying Authentication to Network Access) [JLO08] add delay during layer 3 authentication. Georgiades [Geo04] shows that it takes up to 4 seconds to complete the authentication and authorization process. These operations could use a combination of EAP (Extensible Authentication Protocol) [ABV+04] over layer 2 for IEEE 802.1x-based authentication and EAP-TLS (Transport layer Security) [AS99] for

layer 3 authentication. At the time of reconnection after handoff, re-authentication process adds to the handoff delay. An authentication process is followed by authorization process in some cases. For example, an 802.1x-based or PANA-based authenticator usually communicates with an AAA server to complete the authorization process. In case of inter-domain mobility, the AAA server changes. Thus, the authentication process is further delayed as the mobile needs to go through the authorization process using the new AAA server.

3.3.5 Binding update

Binding update is the process by which a mobile can update its newly obtained network identifier so that the data after handoff can be rerouted to the new destination. A binding update process consists of three main phases, namely *identifier update*, *identifier mapping* and *binding cache update*. This identifier update process associates the new network identifier with the permanent identifier of the mobile. As the mobile connects to a new point of attachment and obtains a new temporary network identifier (e.g., TMSI in GSM, COA in MIPv6, FA-COA in MIPv4) in the new network, it needs to update the correspondent host and home agent so that the packets can be routed to the new destination.

Until the re-association of the new identifier is complete and binding cache in the correspondent node or home agent is updated, the transient in-flight data continue to go to the old network and is lost in the absence of any optimization mechanism, such as buffering or packet forwarding. For application layer mobility, binding update process also includes registration process. By means of registration process, the mobile establishes the mapping between the permanent locator, e.g., URI (Universal Resource Identifier) the temporary identifier, e.g., care-of-address for proper location management function. An optimized or hierarchical registration process expedites locating a user and provides faster delivery of the new data.

Binding update also needs to be authenticated. For example, MIPv6 introduces return routability procedure and adds two additional messages, namely CTI (Care-of-Test Init)

and HTI (Home-Test Init) to obtain the binding key that can authenticate the binding update message. However, this process contributes to additional delay for the binding update procedure after every handoff. Binding update can also be secured by way of IPSec. The security association for binding update in an IP-based network can be uniquely identified by a tuple consisting of a Security Parameter Index (SPI), an IP destination address, and a security protocol (AH or ESP) identifier.

3.3.6 Media re-routing

A media rerouting phase is the last phase during the handover process and follows the binding update process. This phase involves re-routing of the media so that the delivery of data changes from the old path to the new path according to the pre-defined service guarantee.

Once the binding update is complete, the home agent or correspondent host updates its binding cache and the data from the correspondent node gets routed to the mobile's new location. Media delivery can take place in several ways. In one way, the media delivery can take place using the direct path between the CN and the MN. In second scenario, the media is delivered using an indirect path and uses a network entity called home agent. However, until the time the new data is sent directly to the mobile's new point of attachment (nPoA), there are packets in-flight to the previous point-of-attachment. Media re-routing does also take care of re-routing in-flight packets. In both the cases, the in-flight data can be captured and redirected to the new point of attachment. Media rerouting process may include several elementary operations, such as encapsulation, decapsulation, tunneling, buffering, and store-and-forward techniques. During the media re-routing process, transient data may get lost or may get delayed because of these operations. There is data overhead associated with encapsulation, decapsulation and tunneling operations. Thus, there is a need to optimize these operations to ensure that the media delivery delay is reduced after

the handoff. Optimization techniques for the media delivery are often defined as route optimization methodologies. I describe some of these route optimization techniques in Chapter 5. As an example, operations in the network, such as buffering or forwarding mechanisms help to reduce packet loss but add delay to the packet traversal. Thus, the buffering period needs to be adjusted to compromise between packet loss and one-way packet delay.

3.4 Effect of handoff across layers

In this section, I illustrate how these basic handoff operations affect multiple layers in an IP-based network. Based on the type of mobility, a subset of operations are executed at each layer and the overall handoff delay is contributed by delays at all layers during a mobility event. For example, during an intra-domain movement, the mobile is not subjected to the delay due to authorization process unlike inter-domain mobility. Similarly, a layer 2 handoff does not involve delays due to other layer 3 operations, namely layer 3 identifier acquisition or duplicate address detection. Following is a description of how optimization techniques can be useful to each of these basic operations at every layer.

3.4.1 Layer 2 delay

In 802.11 environment, channel scanning, probing, authentication, and association are the basic functions that contribute to the delay before a mobile completes the network attachment at layer 2. Scanning is considered to be a discovery process in layer 2. Encryption and user authentication using WPA (Wi-Fi Protected Access) in conjunction with 802.1X [802a] and EAP (Extensible Authentication Protocol) contribute to the additional delays because of the associated 4-way handshake between the mobile and the access point. Mishra et al. [MSA03] provide a comprehensive delay analysis of the basic operations associated with a layer 2 handoff. Shin et al. [SSFR04] also discuss the probe delays, authentication

and association delays during layer 2 handoff in 802.11 networks. Both of these studies demonstrate that the probe delay constitute ninety percent of the total layer 2 handoff delay.

I have also taken measurements in the experimental testbed that I have created to study the layer 2 handoff related operations for two different operating systems, namely Linux and Windows, and used different layer 2 drivers, namely Aironet, Orinoco, DLink, and Centrino. From a thorough analysis of the layer 2 handoff event I found that scanning and probing operations contribute to most of the delay. For example, in our experimental setup, using active scanning with Orinoco driver in a Linux environment, it takes almost 100 ms for probing action to complete. This is followed by layer 2 open authentication and association that take about 2 ms and 20 ms, respectively. I have also experimented with layer 2 authentication, such as IEEE 802.11i and have found average EAP delay and 4-way handshake delay to be 79 ms and 616 ms for non-roaming and roaming cases, respectively. Roaming scenario involves interaction with AAA server during mobile's handoff, where as authentication is limited to the local authentication agent for non-roaming scenario.

3.4.2 Layer 3 delay

In an IP-based environment, network association at layer 3 involves several basic operations, such as discovery of layer 3 point of attachment (e.g., default gateway discovery), IP address acquisition, duplicate address detection, neighbor reachability, local access authentication, and authorization. Each of these operations involves a number of message exchanges between the mobile and other network entity such as router, DHCP server, authentication server. From the periodic router advertisement, the mobile can discover the new layer 3 point of attachment. There are several protocols, such as DHCP [Dro97], DHCPv6 [DEB+03], PPP [Sim94] and stateless auto-configuration [TN98] that help the mobile to acquire IP address. I have measured the time taken by each of these IP address configuration methods and have explained them in [DMCS06]. Table 3.2 shows IP address acquisition

delays due to different types of configuration protocols including zeroconf [CAG05], static (manual configuration) and proactive IP address acquisition technique. Time taken to configure an IP address using each of these protocols differs due to various factors, such as variation in the number of signaling messages to acquire the IP address and duplicate address detection technique. In a typical inter-domain mobility scenario, there are additional operations, such as re-authentication, re-authorization and profile verification by the AAA servers during the handoff process. In some cases, the AAA-related messages traverse all the way to the home AAA server before the network service is granted to the mobile in the new network. Thus, it is desirable to have an optimized method of interaction between the mobile and AAA server during the inter-domain handoff. After a layer 3 identifier is reconfigured, other layer 3 functions, such as the binding update from the mobile and media redirection at the correspondent host contribute to the additional delay.

Table 3.2: IP address acquisition delay

Layer 3 configuration methods	DHCP w/ ARP	DHCP w/o ARP)	IPv6 stateless	DHCPv6	PPP	FA CoA	Zero Conf	Manual	Proactive
Address acquisition delay	4 s	400 ms	160 ms	500 ms	8 s	2 s	5 s	100 ms	4 ms

3.4.3 Application layer delay

A mobile node suffers from application layer delay during a handoff operation, when some of the handoff related operations are performed at the application layer. Application layer delay is mostly attributed due to the operations such as binding update delay at the application layer, processing delay at the end hosts, registration, upper layer encryption, such as TLS (Transport Layer Security), and SRTP (Secured RTP). SIP-based mobility is a typical example where binding update is done at the application layer. SIP-based binding update (e.g., Re-INVITE) contributes to the delay because of three-way-exchange between the mobile and the correspondent host during identifier rebinding process.

3.4.4 Handoff operations across layers

Figure 3.5 illustrates a sample protocol flow where the mobile is subjected to subnet handoff and uses MIPv6 as the mobility protocol.

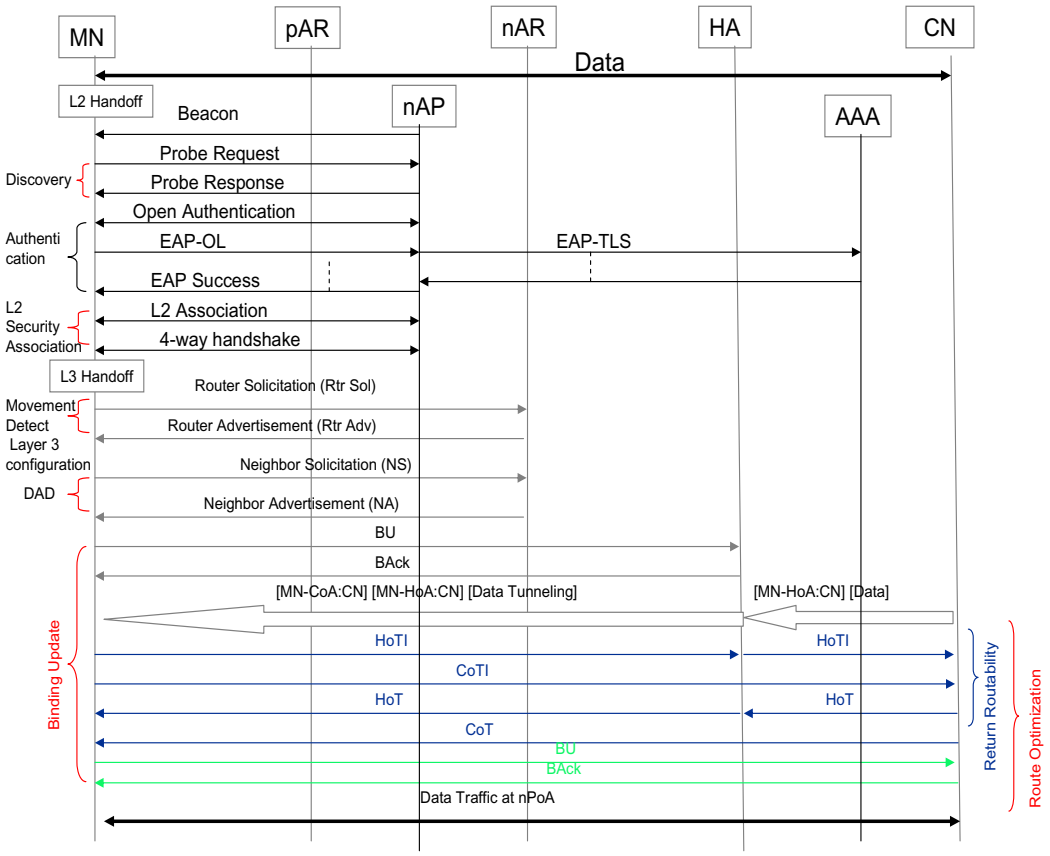


Figure 3.5: Protocol flow for MIPv6-based operations

In this specific example, the mobile moves from old layer 2 point of attachment (oPoA) to a new layer 2 point of attachment (nPoA) and in the process it disconnects from previous Access Router (pAR) and connects to the new Access Router (nAR) resulting in change in layer 3 point of attachment. This figure shows the message exchange between different components in the network that are used to take care of handoff operations, namely layer 2 discovery, layer 3 discovery, authentication, layer 2 security association, address acquisition, duplicate address detection and binding update. Each of these operations needs different number of signaling messages and carry varying amounts of data payload. Thus, consumption of systems and network resources (e.g., cpu cycles, battery power, bandwidth)

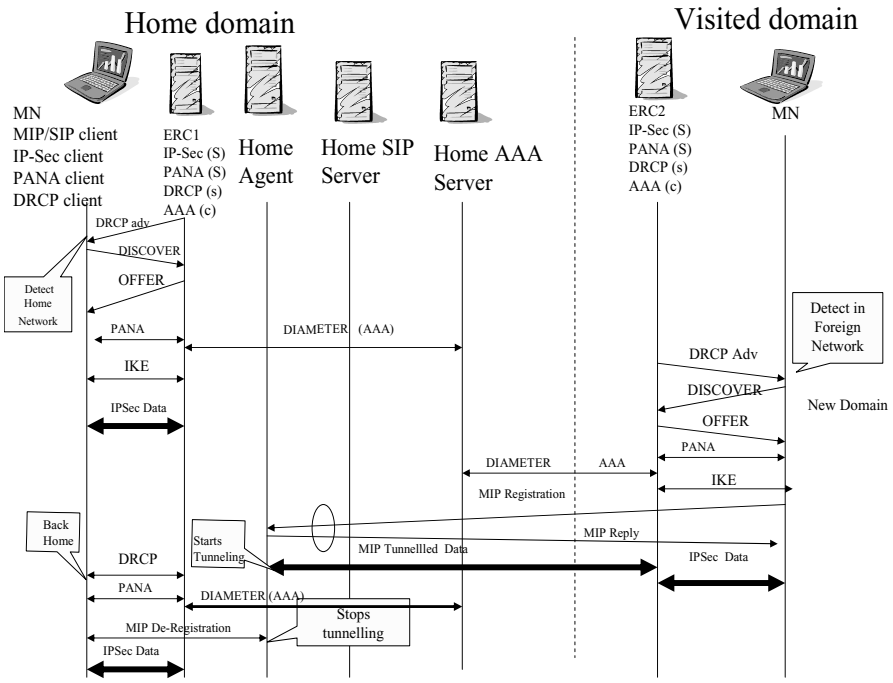


Figure 3.6: Call flow for mobile IP-based inter-domain handoff

and amount of time taken to complete each of these operations will vary.

In order to study the effect of different components on handoff delay, I experimented with secured inter-domain mobility using both network layer (e.g., Mobile IP) and application layer (e.g., SIP-based) mobility techniques [DAC+03] and [DDL+04]. Inter-domain mobility introduces additional operations, such as local authentication and interaction with an authorization server resulting in additional delay. These mobility prototypes include a combination of network detection, registration, configuration, authentication, security, location management functions, and roaming support for the wireless Internet [DVC+01]. I have used DRCP, a variant of DHCP for configuration, PANA [JLO08] for secure access control, Diameter [CLG+03] for authorization, and ESP (Encapsulating Security Payload) within IPSec [KA98a] and SRTP (Secured RTP) [BMN+04] for encryption.

Figures 3.6 and 3.7 show the call flows for mobile IP-based and SIP-based inter-domain handoff, respectively. ERC1 and ERC2 are the two edge router controllers that also act as DRCP server, PANA server and IPSec end point. These flows demonstrate the protocol in-

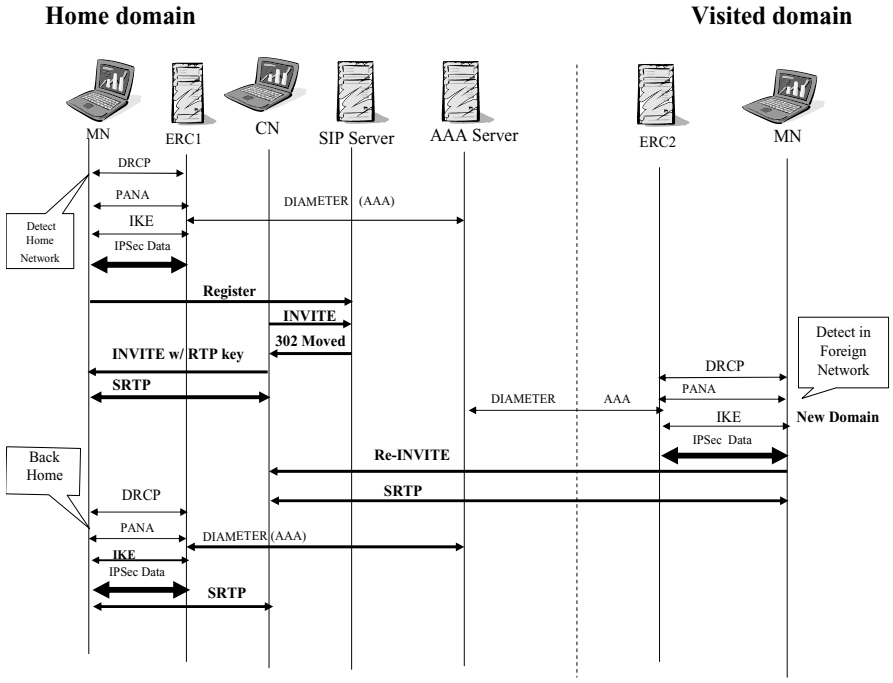


Figure 3.7: Call flow for SIP-based inter-domain handoff

teraction among several functional components, such as SIP user agent or Mobile IP client, DRCP server, PANA server, home agent, IPSec server and AAA server when the mobile moves from home domain to visited domain and return to the home domain. During this movement, the mobile changes its layer 2 point-of-attachment, layer 3 point-of-attachment, and then re-establishes layer 3 authentication, authorization and security association using IPSec. During the inter-domain handoff the mobile needs to communicate with the visited AAA server for re-authorization.

Table 3.3 shows the timing delay for some of the functional components during inter-domain handoff when SIP-based mobility and MIP are used as the mobility protocols. End systems processing time are not shown in this table. These experimental results demonstrate delays due to several handoff components, namely layer 2 beacon interval, subnet, and domain discovery, IP address acquisition, local authentication, authorization, and delay due to binding update, such as SIP re-INVITE and MIP registration. Pre-handoff media represents the time when the media is received prior to handoff in the old network and

post-handoff media represents the time media is received in the new point of attachment after the handoff. For both the cases, IKE (Internet Key Exchange) took almost 5 seconds for successful establishment of the IPSec-based security association.

Table 3.3: Experimental timing for handoff components

Mobility Type	Pre-handoff media	Layer 2 beacon adv.	Router advertisement	Layer 3 configuration	Layer 3 authentication delay	IKE process	Binding update	Post-handoff media	In-handoff delay
SIP	51.7 s	120 ms	500 ms	80 ms	10 ms	5130 ms	240 ms	53 s	1.3 s
MIP	23.8 s	120 ms	500 ms	80 ms	10 ms	4580 ms	20 ms	31.1 s	7.3 s

Table 3.4 summarizes how the basic operations of a mobility event as described in Section 3.1 are performed across different layers in an IP-based environment with 802.11 as the access media. Depending upon the layer at which the handoff takes place, (e.g., layer 2 or layer 3) appropriate operations at those layers are performed. For example, if a mobile's handoff operation does not involve change in subnet then many of the layer 3 related operations are not performed during the mobility event.

Table 3.4: Mapping of basic handoff operations across layers

Layers	Basic handoff operations						
	Discovery	Authentication	Security association	Identifier configuration	Address uniqueness	Binding update	Media routing
Layer 2	Scanning	Open auth, EAPoL	4-way handshake	ESSID, Beacon	MAC address	Update ARP cache	IAPP
Layer 3	Router advertisement	L3EAP IKE PANA	IPSEC	DHCP Stateless	ARP DAD	Update HA and CN	Encapsulation, Tunneling, Forwarding
Application Layer	AAA discovery	S/MIME	TLS SRTP	URI-IP mapping	Registration	SIP Re-INVITE	Direct routing

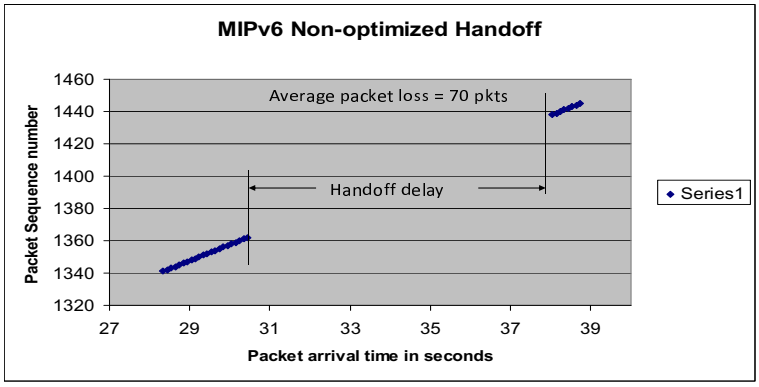


Figure 3.8: Results of FMIPv6 non-optimized handoff

While optimization of each of these specific operations minimizes overall handoff delay, scheduling among different tasks across layers can also lead to variety of optimization techniques. For example, it is not necessary for all the layer 2 operations to complete before layer 3 operations can start. Scheduling a layer 3 discovery process during layer 2 discovery process or configuring a layer 3 identifier before a layer 2 identifier is configured will help minimize the handoff delay. Figures 3.8, 3.9 and 3.10 illustrate the effect of non-optimized, reactive and proactive handoff on the mobile, respectively. I used Mobile IPv6 for non-optimized handoff and used Fast Mobile IPv6 to obtain the results for both reactive and proactive handoffs. As shown in these figures, in this specific laboratory testbed environment, a mobile suffers from almost 7 seconds delay and lose about 70 packets due to handoff when MIPv6 was used without any optimization. Whereas the reactive optimization associated with Fast MIPv6 [Koo05] reduces the delay to almost 1.5 seconds. Finally, when FMIPv6's proactive optimization techniques are applied, the handoff delay is little less than handoff delay during reactive optimization but the packet loss is reduced to zero due to buffering techniques applied at the next access router. It is important to note that these experiments did not include any optimization at layer 2.

I describe my proposed optimization techniques for each of the handoff components in Chapter 5.

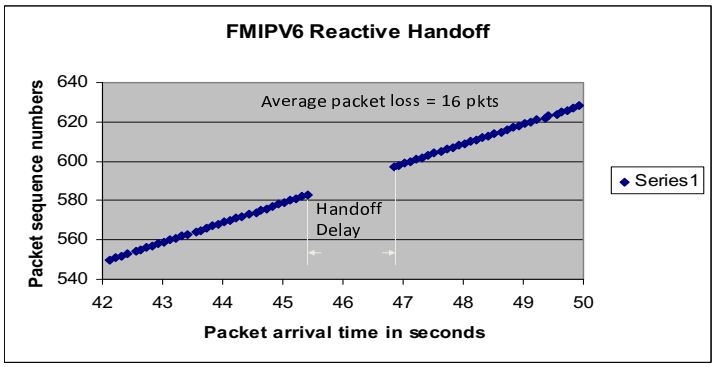


Figure 3.9: Results of FMIPv6 reactive handoff

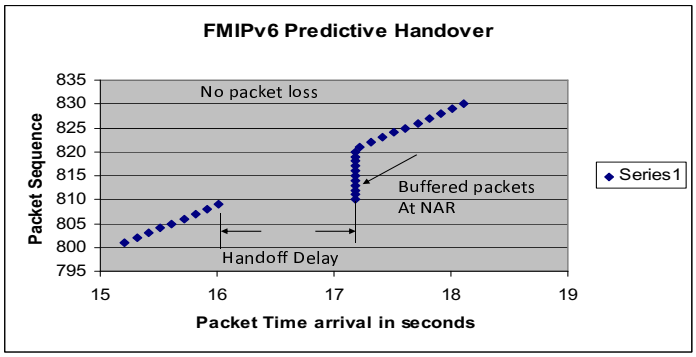


Figure 3.10: Results of FMIPv6 proactive handoff

3.5 Concluding remarks

From the systematic analysis of the un-optimized mobility protocols and their experimental results from different prototype handoff systems, amount of delays for different components of the handoff operations are determined for each of the layers (e.g., layer 2, layer 3 and application layer). Having the ability to determine which handover component is contributing to how much percentage of the total handoff delay one can get some hints for a new mobility system design. These predictions of delays would be useful while designing a handoff optimization at different layers. From these experimental results it is also

possible to find out what components of delays are due to signaling exchange between the mobile node, server and the correspondent node and what components of the delays are due to processing at the network nodes and end systems. Experimental results from the corresponding optimization protocols (e.g., MIPv6 vs. FMIPv6) give some insight into some of the existing mechanisms that optimize certain handover components (e.g., binding update, media forwarding). Any new optimization mechanism will benefit from this analysis and the above experimental results from both optimized and un-optimized systems but will need to take into account the amount of resources such as processing power or battery power to balance a tradeoff between handoff delay and systems resources.

Chapter 4

Modeling mobility

In this chapter, I develop a system model for the mobility event by incorporating the state transitions associated with the basic operations that take place during handoff. The system model decomposes a mobility event into various tasks and subtasks and analyzes the primitive operations.

4.1 Summary of key contribution and indicative results

In the absence of any formal mechanism to analyze the dynamics of handoff systems, it is difficult to predict or verify the systems performance of un-optimized handover or any specific handoff optimization technique. Without an existing mobility system model, it is difficult to design a new mobility protocol or design new optimization techniques of an existing mobility protocol in a systematic way.

I model the handoff-related processes as Discrete Event Dynamic Systems (DEDS) [CH90] and use Deterministic Timed Transition Petri Net (DTTPN) to build various un-optimized mobility models and their associated optimization techniques. I perform data dependency analysis and resource analysis of the handoff related operations to determine the possible sequence of operations and investigate behavioral properties such as deadlocks and liveness associated with the handoff operations using Petri net.

My proposed framework for the mobility model has the following key features:

- ◇ My proposed mechanism analyzes data dependency among the handover components and illustrates how handoff operations are distributed over the network components at different layers.
- ◇ Investigates resource dependency analysis of various handoff operations from an experimental testbed.
- ◇ Design a mobility system model using Timed Transition Petri net based on data dependency analysis and resource dependency.

Key benefits of the mobility model are as follows:

- ◇ The model can predict systems performance for optimized handoff operations.
- ◇ The model can design optimal path for sequence of execution of events based on expected performance and resource constraints.
- ◇ The model can verify systems behavior (e.g., deadlocks) during handover.
- ◇ Design of various Petri net-based approaches (e.g., Floyd algorithm, RTP-based, Matrix-based solution) to evaluate the mobility models for different handoff components.
- ◇ This system model can investigate parallelism and opportunity for optimization during a handoff operation. Using these models one can predict or verify the systems performance of an un-optimized handover or any specific handoff optimization technique.
- ◇ This model can predict the performance of any mobility protocol under any specific deployment scenario, such as intra-domain, inter-domain and heterogeneous handoff.

- ◇ The model can also analyze the trade-off between performance metrics and resources when a mobility event exhibits parallel, optimistic or speculative operations.

In the rest of the chapter, I discuss the related work, dynamics of the proposed Petri net based mobility model, detail mechanisms to evaluate the systems performance and investigate the opportunity for optimization using the model-based approach.

4.2 Introduction

Behavioral properties of any system are those properties that characterize the interaction among different components of the system. These properties define how the systems operate under varying working procedures. In order to understand the dynamic behavioral properties of the handoff system, study the interaction among the primitive operations of handoff, and investigate the trade-off between the performance metrics such as handoff delays and resource utilization during the handoff operations, it is important to design a formal mobility system model. For example, some of the questions related to systems dynamics could be things like - Will the system ever reach a conflict state under a given sequence of handoff operations where one operation cannot proceed because other operation has not started yet?; Under what conditions will there be a conflict due to resource sharing?; How does the system behave in the absence of conflicts; Can we obtain required performance measures for the system under some resource constraints? The model should also be useful to define the control aspects of the above system that can provide certain guidance to rectify a certain anomaly, unwanted behavior such as deadlocks. Non-availability of data and resources due to sharing could lead to possible deadlock situations that should be avoided by scheduling the handoff operations properly. This model can analyze the important behavioral system properties of the mobility event, such as possible deadlocks during handoff operation to verify the correctness of the system operation and predict possible sequence of transitions during handoff to meet certain performance criteria.

4.3 Related work

A few related efforts have attempted to model certain aspects of mobility management. Marshan et al. [MMGS01] have used a Petri net-based model to analyze the performance characteristics of wireless Internet access for GSM and GPRS systems. Amadio et al. [AP98] have modeled IP mobility using a process calculus approach and have applied this to a specific protocol, Mobile IPv6 [JPA04]. The process-calculi-based method has used a software agent approach for modeling the mobility event. Tuts and Sokol [TS01] provide a Petri net-based performance evaluation of bandwidth partitioning meant for multimedia session in wireless access. Molina-Ramirez et al. [MRMRLM94] and Jaimes-Romero et al. [JRMRMT97] have used a Petri net-based approach to model and analyze channel allocation schemes in cellular systems. Mostafa and Cicak [MC06] develop a Petri net-based model to calculate the end-to-end delay between the communicating hosts in a Mobile IP environment. However, none of these papers have attempted to model the interaction of different functional components of a handoff system or study the behavioral characteristics such as liveness and deadlocks in the system. None of these models were intended to systematically analyze the elementary operations involved with a mobility event or serve as a basis for optimizing the operations for a mobility event.

Based on the systems analysis of the mobility event, I develop a formal model to study the system design of an IP-based handoff. This model can be applied to analyze any type of mobility protocol, study the functional components and be able to predict the systems performance during handoff.

4.4 Modeling mobility as a discrete event dynamic systems

A mobility event can be viewed as the perturbation to the steady state of a communicating node that may affect different layers in the protocol stack. As a communicating node is subjected to handoff, it goes through a series of sequential discrete states before it attains

a steady state by returning to the communicating state again. Each of the basic operations described in Chapter 3 can be modeled as a transition event where the mobile moves from one state to another. As a result of a series of state transitions during the mobility event, a mobile's communication is interrupted because of the delay associated with each transition. A discrete event dynamic system (DEDS) [CH90] is a type of system where the state space is discrete and state changes are driven by external or internal events. A mobility event can thus be modeled as a discrete event dynamic system, since many of its components exhibit either concurrent, sequential, or competitive activities (i.e., two operations competing for the same resources) among many of the several handoff operations that are part of the mobility event.

I considered several tools as possible candidates to model and analyze mobility event using DEDS-based approach. These include software language, such as Esterel [BdSEC91], Markov chains, queuing theory, minimax algebra [CG91, PK91], GANTT charts [HD99], and Petri nets [Mur89]. Esterel is a synchronous programming language tailored for the development of reactive applications that show both concurrency and determinism. However, it cannot analyze the behavioral aspects of a system such as precedence relations among events, mutual exclusion, deadlocks and liveness. The existing algebra-based techniques such as minimax algebra cannot be used to analyze the dynamics of the handoff system as these techniques can only be applied to the subclass of DEDS that are described by a max-linear time-invariant system. On the other hand, handoff system is time-variant. Markov chain-based solutions are more suitable for systems that exhibit non-deterministic or stochastic behavior unlike the handoff system that is deterministic in nature. Although queueing theory can be used for performance evaluation, it involves stochastic framework and is meant for average long-term evaluation unlike handoff system that is deterministic and demonstrates transient behavior. GANTT charts can demonstrate the dependencies among the events but cannot verify the correctness of the model.

Finally, I chose Petri net as the tool to model the mobility event for the following rea-

sons. Petri nets can be used to model properties such as process synchronization, asynchronous events, sequential operations, concurrent operations, and conflicts or resource sharing. Timed Petri nets [Zub91] provide a uniform environment for modeling, design and performance analysis of discrete event systems. Timed Petri nets propose to assign time to the transitions and/or places of the Petri net. Timed Petri was originally developed by Ramchandani [Ram74] in 1974, where the firing times are associated with the transitions of a net, and tokens are removed from transitions' input places at the beginning of firings. The graphical representation makes Petri nets intuitively very appealing to represent any temporal events (e.g., events related to time). As a mathematical tool, a Petri net model can be described by a set of linear algebraic equations or other matrix-based mathematical models reflecting the behavior of the system. Several software tools are available (e.g., TimeNet [ZGFH99], MATLAB-based Petri net Tool [MMP03], SPNP [HTT00], UltraSAN [San95], TOMSPIN [KL92]) to automate the analysis of the Petri net model and obtain performance evaluation. Timed Petri net model can also be used with other existing techniques, such as simulated annealing [ZRS99] to perform a tradeoff analysis between different performance metrics, such as handoff delay, handoff probability and systems resources.

Modeling of mobility events can be viewed as analogous to modeling of the Flexible Manufacturing System (FMS) as both exhibit a series of sequential operations. Zuberek et al. [ZK99] model and analyze the simple schedules for manufacturing cells. Similar techniques can be applied to conduct performance analysis for the mobility systems model. I use Deterministic Time Petri net [Mur89] to model the mobility event and derive the relevant optimized models by applying the appropriate concurrent and proactive mechanisms.

4.5 Petri net primitives

In this section, I define some fundamentals of Petri net and describe some of the primitives of Petri net that are essential for mobility analysis.

Formally a Petri net can be defined as follows:

PN = (P, T, I, O, M_0) where

1. $P = \{P_1, P_2, \ldots .. P_m\}$ is a finite set of places
2. $T = \{t_1, t_2, \ldots t_m\}$ is a finite set of transitions, $P \cup T \neq \emptyset$, $P \cap T = \emptyset$
3. $I : (P \times T) \rightarrow N$ is an input function that defines the directed arcs from places to transitions, where N is a set of non-negative integers.
4. $O : (P \times T) \rightarrow N$ is an output function that defines directed arcs from transitions to places, and
5. M_0: $P \rightarrow N$ is the initial marking

A *marking* is an assignment of *tokens* to the places of a Petri net. Tokens are assigned to and can be thought to reside in the places of a Petri net. The number and position of tokens may change during the execution of a Petri net. The tokens are used to define the execution of a Petri net.

If I (p,t) = k (O (p,t) = k), then there exist k directed (parallel) arcs connecting place P to transition t (transition t to place P). If I (p,t) = 0, (O(p,t) = 0), then there exist no directed arcs connecting p to t (t to p).

In Petri net modeling, a transition is enabled after a token is made available from the places representing shared resources. Once an operation completes, these tokens are returned to the places that represent the shared resources, thus making the resources available for other operations. Following are the firing rules for any transition within a Petri net.

1. A transition t is said to be enabled if each input place P of t at least contains the number of tokens equal to the weight of the directed arc connecting p to t.
2. An enabled transition t may or may not fire depending on the additional dependency.

3. A firing of an enabled transition t removes from each input place p the number of tokens equal to the weight of the directed arc connecting p to t. It also deposits in each output place P the number of tokens equal to the weight of the directed arc connecting t to P.

Zhou and Robbi [ZR94] discuss the basic operations of Petri nets, their precedent, concurrent and conflicting relationships. Figure 4.1 shows some of the fundamental operations of Petri net that can be applied while scheduling the set of handoff operations. I give a brief description of these Petri net-based operations below.

1. **Sequential:** If one operation follows the other, then the places and transitions representing these should form a cascade or sequential relation in Petri nets. Figure 4.1 illustrates that transition t_2 can fire only after the firing of t_1. This imposes the constraint "t_2 after t_1". Such precedence constraints are typical of the execution of the parts in DEDS. Also, this Petri net construct models the causal relationship among activities.

2. **Concurrent:** If two operations are initiated by an event, they form a parallel structure starting with a transition where two places are two outputs of the same transition. In Figure 4.1 (c), the transitions t_1 and t_2 are concurrent. It is to be noted that necessary condition for transitions to be concurrent is the existing of a forking transition that deposits a token in two or more output places.

3. **Conflicting:** If either of the two operations can follow an operation, then two transitions form two outputs from the same place. In Figure 4.1 (b), t_1 and t_2 are in conflict. Both are enabled, but the firing of any transition leads to the disabling of another transition. The resulting conflict may be resolved in a purely non-deterministic way or in a probabilistic way, by assigning appropriate probabilities to the conflicting transitions.

4. **Cyclic:** If a sequence of operations follows one after another and completion of the last one initiates the first one, then the cyclic structure is formed among these operations. Figure 4.1 (i) shows the cyclic nature of Petri Net.

5. **Synchronization:** Figure 4.1 (d) shows an example of data dependency or synchronization among events. In this example, t_1 is enabled only when both p1 and p2 receive a token. Arrival of tokens in each of the two places could be the result of availability of data resulting out of some other complex operations elsewhere.

6. **Mutually exclusive:** Two processes are mutually exclusive if they cannot be performed at the same time due to constraints on the usage of a shared resource. Two types of mutual exclusiveness, namely parallel mutual exclusion and sequential mutual exclusion are discussed by Zhou and DiCesare [ZD91]. Figure 4.1 (g) shows an example of how a resource may be shared by two processes thus not being able to be completed at the same time.

7. **Inhibitor arcs or priority transitions:** Modeling of priorities can be achieved by introducing an inhibitor arc. An example of Petri net with an inhibitor arc is shown in Figure 4.1 (h). Transition t_1 is enabled if place p1 contains a token, while t_2 is enabled if p2 contains a token and p1 has no token. This gives priority to transition $t1$ over $t2$.

Many of the above scenarios can be applied while designing the Petri net model for handoff.

4.6 Petri net-based modeling methodologies

It is useful to have a set of methodologies while designing handoff using Petri nets. In order to model the mobility event using Petri nets, it is required to follow the systematic approach. A general modeling method can be summarized into the following steps [ZV99].

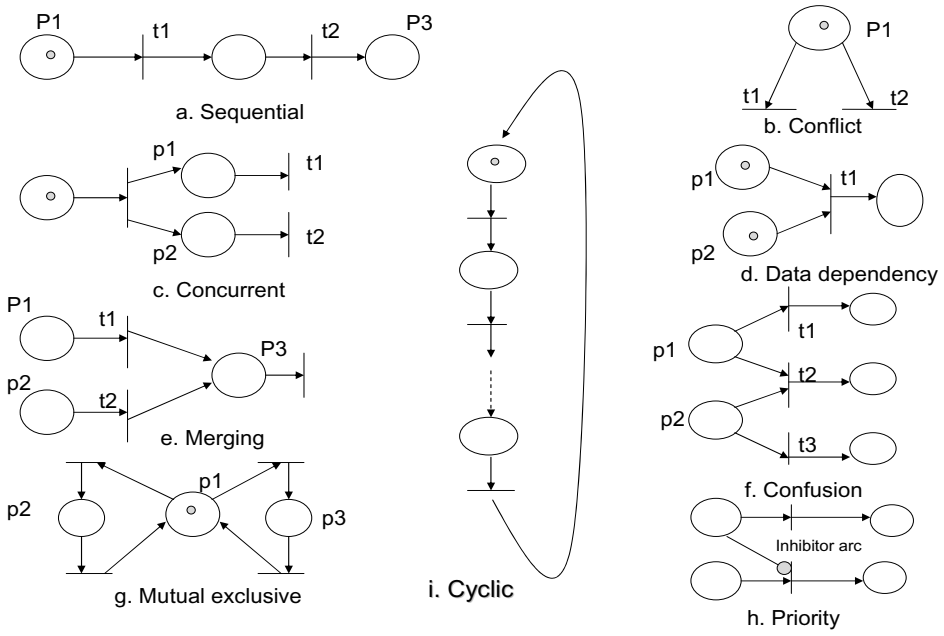


Figure 4.1: Petri net primitive operations

1. **Identification of operations and resources**: Given a system description, major events, operations, processes, resources, and conditions need to be identified.

2. **Identification of relations**: The relationships among the identified events, operations, and processes have to be determined. The resources should be divided into different classes and appropriate allocation policies for shared ones should be determined. Then an initial net structure can roughly be decided.

3. **Petri net modeling principles**: Following are sequence of rules for Petri net that are useful for designing handoff system.

   (a) Design and label the places and transitions that represent events, operations and processes.

   (b) Arrange the places and transitions according to the relationship identified during the identification process

   (c) Designate and label places that model the status of the resources and conditions.

(d) Insert necessary places and transitions such that no two places can link to each other neither two transitions.

(e) For each transition, draw an input arc to it from a place if enabling it requires the resource(s), truth of the condition, or the completion of the operation(s) represented in the place; draw an output arc from it to a place if firing it releases resource(s), changes the condition(s), or signals the initiation of the operation in the place.

(f) The number of input arcs from a place to the transition should equal the required quantity of the tokens (often implying resources) in the place to enable the transitions. The number of output arcs from the transition to a place should equal the quantity of the tokens to the place to be produced due to the transition's firing.

(g) Determine the initial number of tokens over all places according to the system's initial state.

(h) Associate other characteristics, e.g., timing, with places, transitions, arcs, if needed, for example for a Deterministic Timed Petri Net, timing information should be added to the transition.

In the following sections, I describe how I have applied some of the above methodologies in formulating the Petri net model for handoff.

4.7 Resource utilization during handoff

In Petri net terminology, a resource can be represented as a structural implicit place. Structural implicit means if we have an arbitrarily large number of resources (i.e., the number of tokens in places representing the resources is arbitrarily large), then the marking of these places does not limit the concurrent processing. However, in most cases most of the opera-

tions take place under some resource constraints.

Taking into account the constraints in any handoff system, the resources can be classified as dedicated to specific task and shared ones. The dedicated resource can be represented as a place with single input and single output arc only and the shared resource is represented as a place with multiple-input and multiple-output arcs. The same kind of resources may be represented by a place with the number of tokens corresponding to the amount of resources. Initiation of an operation requires often several kinds of conditions and available resources, modeled as a transition with several input places. Completion of an operation may release some resources and change the status of the conditions modeled as a transition with several output places. When a set of processes are executed concurrently, they can share a set of common resources. These resources can be represented as places (e.g., R_1, R_2, R_3, R_4, R_5....R_n) to model the availability of resources.

Figure 4.2 shows an example of shared resource place R, that is being shared by two processes, A and B. Assuming that process A starts first, process B cannot start until sequence of operations for process A is complete and the resource token is released back to resource place R after transition "t4" fires.

Several types of resources are utilized to complete the handoff related operations during a mobility event. For the sake of analysis, I categorize these resources into four different types, namely battery power (P_T), memory (P_M), bandwidth(P_B) and CPU processing cycles(P_C). These resources can be consumed by different parts of the network namely mobile node, access point, access router, network server and correspondent host during each of these handoff operations. I describe three of these resources in detail that I have modeled.

Battery power is consumed to take care of several operations, namely transmit, receive, display operations and reading and writing operations on the disk of any system. Based on the type of systems, energy consumption will vary for different types of operation a mobile system has to perform [JSAC01]. For example, a laptop will consume a lot more battery

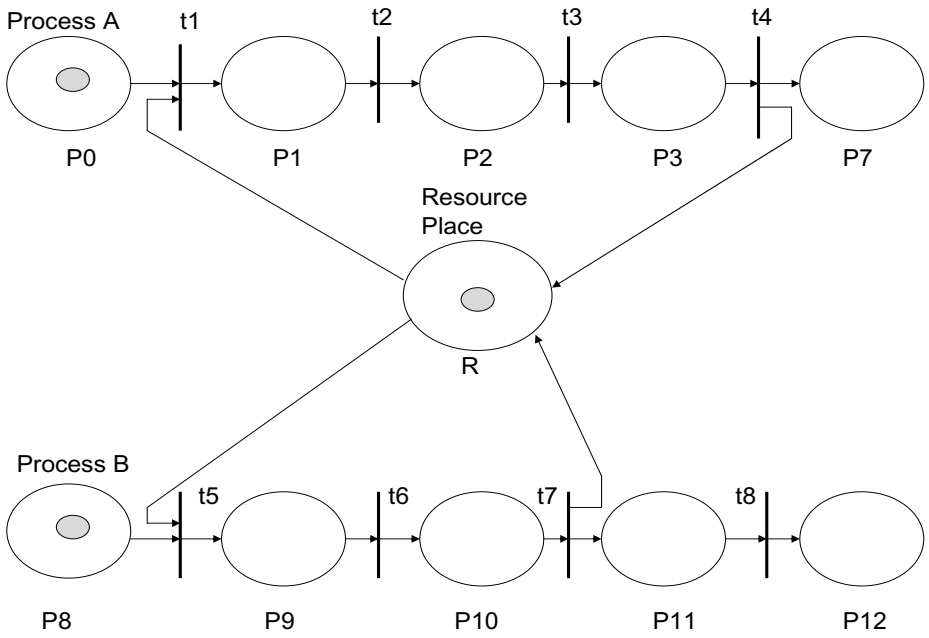


Figure 4.2: Petri net sequence of operations with shared resource

for display on the monitor compared to other operations. However, since we are interested in the handoff operations that involve signaling only, I do consider only signaling related operations related to communication. Shnayder et al. [SHC+04] classify battery power usage for different types of systems operations for the wireless sensor system [HTM+05]. That paper has demonstrated that a wireless sensor system consumes most of the power during transmit operation and receive operation. Pering et al. [PAGW06] demonstrate that wireless interfaces consume approximately seventy percent of the total power. Thus, in my analysis, I have considered energy expensed due to transmit and receive operations for these signaling messages for resource sharing purposes. Caching consumes memory and processing power at the mobile node and correspondent node but does not use the bandwidth.

Per user effective bandwidth during a communication depends on several multiplexing points in the network, namely 802.11 access network (e.g., 802.11), routers in the core of the network, associated air access, and system defined allocations of bandwidth (via priority mechanisms, class-based queuing, weighted fair queuing) providing segregation

of signaling traffic from bearers. Handoff operations involve a set of round trip signaling messages between the network components (e.g., mobile and server). Each of these signaling messages shares transmission resources on the mobile's network interface, channel resources in the access network (e.g., 802.11) and bandwidth associated with the router in the core of the network.

Access to channel resources will vary for different access techniques. For example, in IEEE 802.11-based network, when one mobile has access to the channel resources and is transmitting data, other mobile nodes will have deferred access to the channel resources. Thus, channel resources in the access network get shared by different mobile nodes. However, if there is only one mobile node in the network, per user bandwidth gets shared by different applications (e.g. web, ftp). Kravets and Krishnan [KK00] discuss application driven power management for mobile communication. That paper discusses how scheduling of transmission of applications can help save energy. Handoff related signaling could be considered as one type of application and it is important to investigate what fraction of per-user effective bandwidth would be utilized to take care of handoff related signaling. In modern computer networks, although bandwidth may not be an issue, handoff related signaling may be affected if a large number of mobiles attempt to handoff to a target cell at the same time or a mobile attempts to handoff to a cell that is heavily overloaded resulting in MAC layer retransmission.

CPU cycles are consumed due to processing of the signaling messages at the mobile, router, correspondent host and to perform the operations such as encryption, tunneling, encapsulation and forwarding. Potlapally et al. [PRRJ03] analyze energy consumption characteristics of different cryptographic algorithms for various types of security related operations such as key setup, encryption and decryption. That paper also shows the overhead associated with authentication.

Number of signaling messages for each of the handoff operations varies, each signaling message consists of different amount of bytes and these signaling messages span different

components in the network. Thus, the amount of bandwidth required for each of these operations is different and depends upon the number of signaling messages and bytes per message. For example, since the discovery process involves scanning of the neighboring channels it involves additional number of signaling messages over the air compared to the authentication process that involves messages in the core of the network. On the other hand, authentication process involves more processing power and CPU cycles for key derivation. Similarly, resource usage for each of the other handoff components can be found accordingly. Depending upon the sequence of handoff operations, these resources may need to be shared when several of the handoff operations occur concurrently.

In Section 4.8, I will illustrate number of bytes exchanged for different handoff related signaling messages and the amount of resources utilized to support various types of handoff related transitions. I model the handoff operations using three different types of resources (e.g., battery power, effective user bandwidth and user processing power) and set of data places along with a set of transitions between these places.

4.8 Data dependency analysis for handoff

Analysis of the data dependencies and task dependencies among the required operations of the handoff event can determine the extent of parallelism among the operations or the types of proactive operations that maybe possible. In this section, I introduce the concept of data and task dependency that can be applied to model handoff operations using Petri net. By applying these task dependencies, data dependencies, and resource availability on Petri net model it will be possible to determine the schedule of sequence of events during handoff operations.

4.8.1 Petri net-based data dependency

Filho et al. [CFMBdI00] provide an example of how Petri nets can be used for analyzing data dependency. Maciel et al. [MCFB01] propose a method of estimating number of functional units taking into account the timing constraints and consider the data dependency as input to the estimation process. Belhe and Kusiak [BK93] discuss how a Timed Petri net can be used for performance evaluation of design process and illustrates how a task dependency can be used to generate a Petri net. Before I introduce the data dependency analysis for the handoff event, I give some examples of how a data dependency between the events maps to a Petri net model and the associated resource constraints.

Figure 4.3 shows an example of data dependency relationship among activities for a specific phase of a design process. At each stage of the design process, decisions are made based on the data or information available at that stage which is provided by the preceding stages. Thus, a network of data-dependency relationships among design activities can be established. An arrow can be drawn from design activity *i* to activity *j*, indicating that activity *j* depends upon activity *i*. In Figure 4.3, activity 2 and 3 depend upon the data from activity 1, activity 4 depends on the data from activity 2, activity 5 depends on the data from activity 5, and activity 6 depends on the data from both activities 4 and 5.

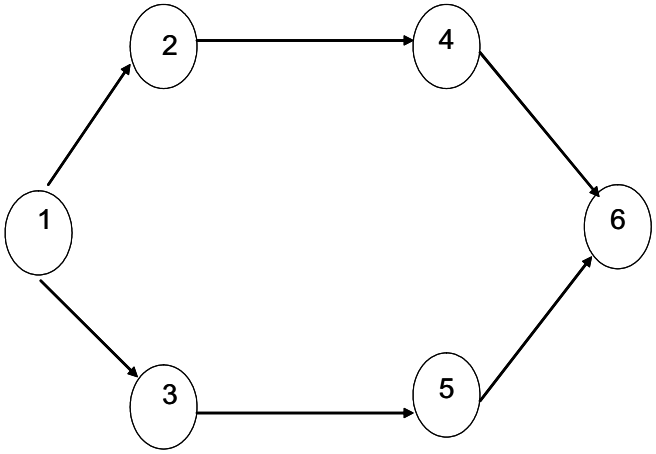


Figure 4.3: Data dependency relationship

Figure 4.4 shows how an equivalent Petri net can be formed based on the data dependency graph shown in Figure 4.3.

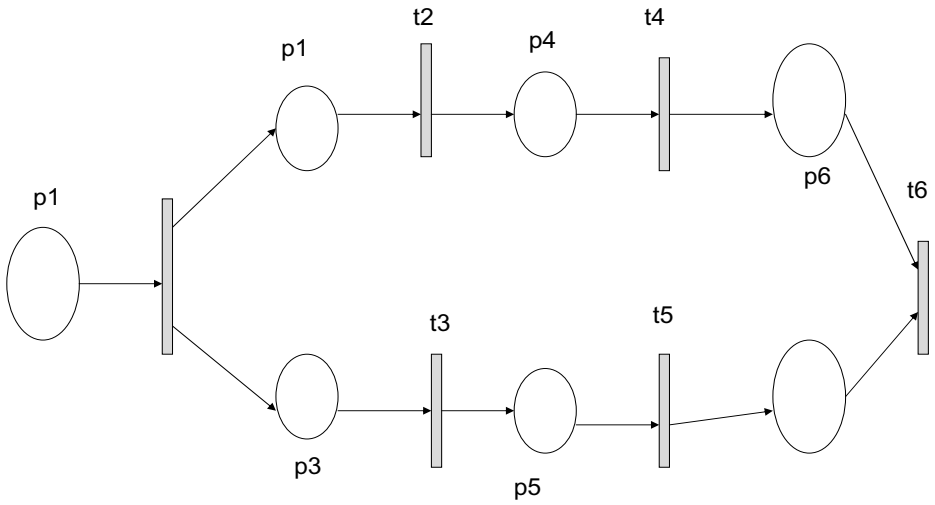


Figure 4.4: Petri net model for handoff based on data dependency

Figure 4.5 adds resources to the Petri net model that is based on the dependency graph. Places P7, P8, P9, and P10 represent the resource places. Each of the resources places is equipped with a certain amount of resources shown as tokens. The resource requirements for different activities are expressed by capacities on the arcs. Availability of a specific data from the previous states and availability of required resources enable a specific task to go forward. Similar task dependency methods can be applied to hadnoff process.

4.8.2 Analysis of data dependency during handoff process

This section describes the dependency among several components of the handoff process. It is helpful to analyze the sequence of tasks among these handoff components and investigate the data that a specific task may depend upon. Completion of one task will result in data that may be needed for execution of another task. Thus, one task may depend upon another task.

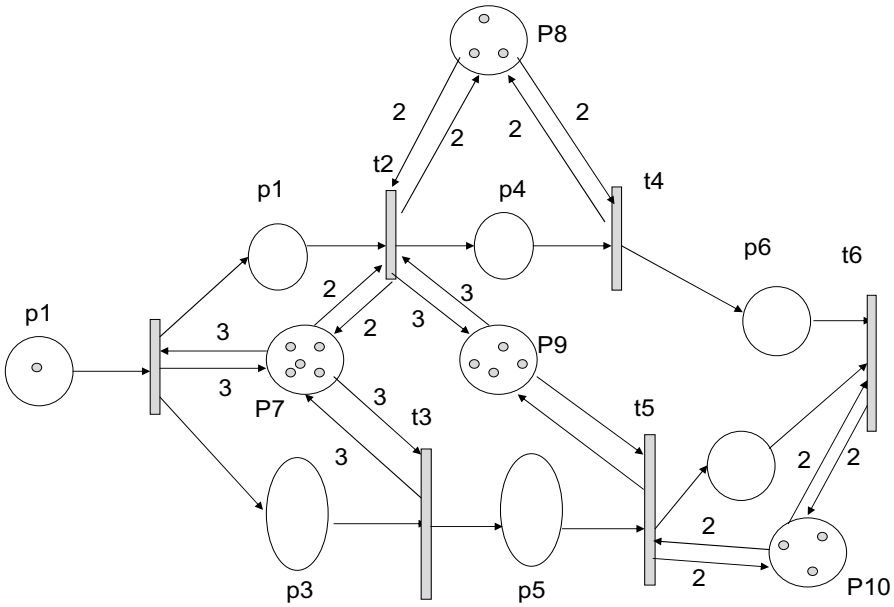


Figure 4.5: Timed Petri net model with resource constraints

Table 4.1 shows several handoff processes and the associated data dependencies among the set of operations based on Figure 3.2 in Chapter 3. I briefly discuss the data dependency for each of these operations.

4.8.2.1 Network resource discovery

Network and resource discovery is the very first task that takes place once the mobile initiates the handoff procedure. Resource discovery takes place at several layers such as channel discovery in layer 2, subnet discovery in layer 3 and server discovery in application layer. I describe below the dependency among the sub-processes that are part of network discovery process according to Figure 3.3.

1. Channel discovery

   Channel discovery of the mobile node depends upon successful receipt of the beacons or pilot signals from the neighboring access points or base stations. For example, in normal handover scenario for 802.11 networks, the mobile needs to be in the

Table 4.1: Petri net based data dependency for handoff

Handoff process	Precedence relationship for operations	Data it depends on
P_{11} – Channel discovery	P_{00}	Signal-to-Noise Ratio measurement Distance from AP
P_{12} – Subnet discovery	P_{21},P_{22}	Layer 2 beacon ID L3 router advertisement
P_{13} – Server discovery	P_{12}	Subnet address Default router address
P_{21}- Layer 2 association	P_{11}	Channel number MAC address Authentication key
P_{22}- Router solicitation	P_{21}, P_{12}	Layer 2 binding
P_{23}- Domain advertisement	P_{13}	Server configuration Router advertisement
P_{31} – Identifier acquisition	P_{23},P_{12}	Default gateway Subnet address Server address
P_{32} – Duplicate address Detection	P_{31}	ARP Router advertisement
P_{33} – Address resolution	P_{32}, P_{31}	New identifier
P_{41} – Authentication	P_{13}	Address of authenticator
P_{42} – Key Derivation	P_{41}	PMK (Pairwise Master Key)
P_{51} – Identifier update	P_{31},P_{52}	L3 Address Uniqueness of L3 address
P_{52} – Identifier verification	P_{31}	Completion of COTI
P_{53} – Identifier mapping	P_{51}	Updated MN address at CN and HA
P_{54} – Binding cache	P_{53}	New Care-of-address mapping
P_{61} – Tunneling	P_{51}	Tunnel end-point address Identifier address
P_{62} – Forwarding	P_{51}, P_{53}	New address of the mobile
P_{63} – Buffering	P_{62}, P_{51}	New identifier acquisition
P_{64} – Multicasting/bicasting	P_{51}	New identifier acquisition

coverage area of the target network to be able to receive the SSID from the broadcast channel. Thus, the ability to receive the beacons over a broadcast address is a precedence operation for the channel discovery operation to start. However, this precedence condition may not be necessary if the mobile uses a proactive discovery technique such as 802.21-based information service [DDF+05] where a mobile can receive the channel information of the neighboring networks based on the current position of the mobile. Thus, a channel discovery process can stay out of the critical path of the handoff operation if the target channel is discovered prior to mobile's handoff.

2. Subnet discovery

   Since subnet discovery is a layer 3 process, the mobile needs to complete the layer 2 association process, P_{21}, before it can receive router advertisements to discover that it has indeed moved to a new subnet. The mobile depends upon the prefix from the router advertisement before it can discover a new subnet. Although router advertisement takes place at a predefined rate based on the router configuration, the mobile can always send a router solicitation message to trigger the router advertisement. This helps to discover the layer 3 point of attachment in the absence of missing or delayed router advertisement. In most cases, the lower layer events such as layer 2 association and network attachment event notification such as "Link Up" triggers the router solicitation from the mobile that in turn triggers the router advertisement.

3. Server discovery

   Server discovery process can take place only after subnet discovery is complete. Server discovery process usually includes discovery of DNS server, outbound SIP server that provide additional services such as domain look up, routing of signaling messages, respectively. Server discovery process P_{13} depends upon the completion of subnet discovery P_{12} and identifier acquisition process P_{31}. The mobile needs to

have acquired the default router address and subnet prefix before it can complete the server discovery process.

4.8.2.2 Network attachment

Next phase of the handoff operation is network attachment. Network attachment process involves association with the new point of attachment at different layers. Layer 2 association, router solicitation and domain advertisement are three different tasks that are part of this process.

1. Layer 2 association

   In order to complete a successful layer 2 association, the mobile will need to have access to the channel number, MAC address of the access point and security keys for authentication. While channel number can be obtained during the discovery or scanning process, the mobile node needs to have access to WEP-based security key for shared key authentication and needs to acquire PSK for using EAP authentication method in case of 802.11i-based authentication. In case of open system authentication, the WEP keys can be used to encrypt the data.

2. Router Solicitation

   Router solicitation message from the mobile triggers the layer 3 attachment process by soliciting a router advertisement from the router. However, this operation takes place only after the mobile has established the layer 2 binding with the access point, so that the mobile can communicate with the access point using MAC address.

3. Domain advertisement

   Similarly, the domain advertisement process helps the network attachment at the application layer. This process helps the mobile to determine the administrative domain it belongs to. The mobile needs to have completed server configuration or router ad-

vertisement. The mobile will depend upon the data from server configuration or router advertisement before it can proceed with the domain advertisement process.

4.8.2.3 Mobile configuration

The next phase of the handoff process is configuration that includes several tasks such as mobile's identifier acquisition, duplicate address detection and address resolution. Following is a description of dependence among the sub-processes of the configuration process.

1. Identifier acquisition

   Mobile node depends upon the following data such as default gateway and subnet prefix in order to complete the identifier acquisition task. These data are obtained as a result of the completion of tasks such as domain advertisement (P_{23}), and subnet prefix discovery (P_{12}).

2. Duplicate address detection

   Mobile node depends upon the completion of acquisition of layer 3 identifier in the new network before it can determine its uniqueness. Thus, duplicate address detection process cannot start until a new identifier has been obtained in the new network. Thus, it is not possible to start process P_{32} before process P_{31} is complete.

3. Address resolution mapping

   Address resolution or mapping of IP address with the MAC address can be completed only after a new address has been obtained. However, duplicate address detection and address resolution could take place in parallel reducing the extent of sequential operation.

4.8.2.4 Security association

The following paragraphs describe the instances of data dependency among the sub-processes that constitute the security association process.

1. Authentication

   A successful authentication operation depends upon successful discovery of the authenticator. After the discovery of the authenticator, the mobile can communicate with the authenticator that in turn communicates with the authentication server (e.g., AAA server) to complete the authentication and authorization process. This process involves an exchange of signaling messages between the mobile node and authenticator. Thus, the data from the operation P13 in Table 4.1 that provides the information of authenticator acts as precedence data for the authentication sub-process P_{41}.

2. Key derivation

   Derivation of an encryption key (e.g., the Pairwise Transient Key)(PTK) that helps to establish security association between the mobile and the authentication server by encrypting the data is possible only after a successful authentication process that generates a Pairwise Master Key (PMK) is complete. Thus, there is a precedence relationship between derivation of PMK by way of EAP-based authentication and generation of PTK. However, the authentication process can always take place before the mobile moves to the target network (proactive authentication), thereby leaving only the generation of PTK to take place after the handover to the new network. This is one way of optimizing the authentication process.

4.8.2.5 Identifier update

Updating the new identifier at the home agent is possible only after the mobile has obtained the new layer 3 address. Thus, data such as new layer 3 address and its uniqueness test are pre-requisite data before the mobile can start the binding update process. In some cases, such as MIPv6, this identifier update needs to be verified against any unlawful binding update before the binding update is complete. This verification is usually done by means of return routability process, whereby MN and CN exchange a pair of keys to authenticate

each other. Similarly, identifier mapping and establishment of binding cache entry depend upon successful completion of the processes, namely identifier update and identifier verification.

4.8.2.6 Data forwarding

Tunneling, forwarding and buffering are the rest of the handoff related operations that take place before the handoff is complete and data is received at the mobile. A tunnel can only be set up after the mobile has learnt the address of the home agent and the home agent has learnt the new care-of-address of the mobile. The tunneling operation can be complete only after binding update is over.

Similarly, forwarding of data can only take place after the mobile has obtained a new care-of-address and has sent a binding update to the local forwarding agent (e.g., Foreign Agent in the previous network) and the home agent. Depending upon where the buffering takes place (i.e., at the edge of the network or at the home agent), forwarding and buffering operations can take place in parallel. Multicasting and bi-casting are some forms of techniques to forward the transient data during the handoff. I describe some of these techniques in Chapter 5.

4.9 Petri net model for handoff

Figure 4.6 shows a high level view of how a set of discrete states associated with a mobility event can be represented in a formal framework by using a Timed Petri net approach [Mur89]. This high level Petri net model shows the handoff operation where a mobile goes from a *connected* state to a *disconnected* state and then comes back to a connected state[1]. Each place (P_i) represents various stages of the mobility event and the transition (t_i) represents the time taken to complete different set of operations between the stages. Each

[1]Connected state and disconnected state are defined in the definition section as part of Appendix C

of these stages can be considered as multiple sub-systems of the system representing the mobility event.

Table 4.2 describes the places that represent various stages of handoff event, shared resources, and transitions associated with Figure 4.6. The transitions represent the respective operations and delays are associated with each of these handoff operations. Although a deterministic delay is assumed for the transitions here, this framework can also be applicable if the transitions follow other types of delay distribution, namely exponential, uniform or finite discrete. The transition time will vary depending upon the processing speed and shared resources available. Shared resources are ideally represented by tokens that become available before a transition is fired, leading to next stage in the handoff process.

In Figure 4.6, each of the places P_0-P_7 represents various stages of the mobility event as described in Chapter 3 and places P_B, P_M and P_P represent shared resources, namely effective user bandwidth, battery, and processing power, respectively, that are used during any mobility event. Effective user bandwidth is a shared resources that is determined by available network bandwidth in the access and core network, whereas battery power and CPU cycles are the shared resources used by the same mobile node for multiple operations within the mobile.

The path shown using the dotted lines in Figure 4.6 representing place P_7 illustrates an example of parallel operation where a hierarchical binding update (t_6) occurs in parallel with a regular binding update (t_7). After a security related operation (t5) is over, both hierarchical binding update (t_6) and regular binding update (t_7) can take place in parallel. P_{5A} and P_{5B} are the intermediary places in this model after transition t5 is fired.

Each of these transition processes may consist of several sub-processes. For example, Table 4.3 shows the sub-transitions for each of the transitions. Transition process t_1 (Network Resource Discovery) consists of several sub-processes that are represented by several sub-transitions, such as t_{11}, t_{12}, t_{13}. These sub-transitions define layer 2 channel discovery, layer 3 subnet discovery, and server discovery, respectively.

The hierarchical property of Petri nets [Zub00] can be applied to each of these transition processes to study the interaction at the sub-process level. For example, as shown in Figure 4.7, transition process t_1 (network resource discovery) consists of several sub-processes (places) such as channel discovery (p_{11}), subnet discovery (p_{12}) and server discovery (p_{13}) that are connected by several sub-transitions, such as t_{11}, t_{12}, t_{13}. Similarly, Figure 4.7 also shows how the hierarchical nature of Petri net is applied to the network attachment (t_2) and the configuration processes (t_3). Although only three processes have been highlighted in Figure 4.7, the hierarchical property of Petri nets can be applied to other operations as shown in Table 4.3 as well.

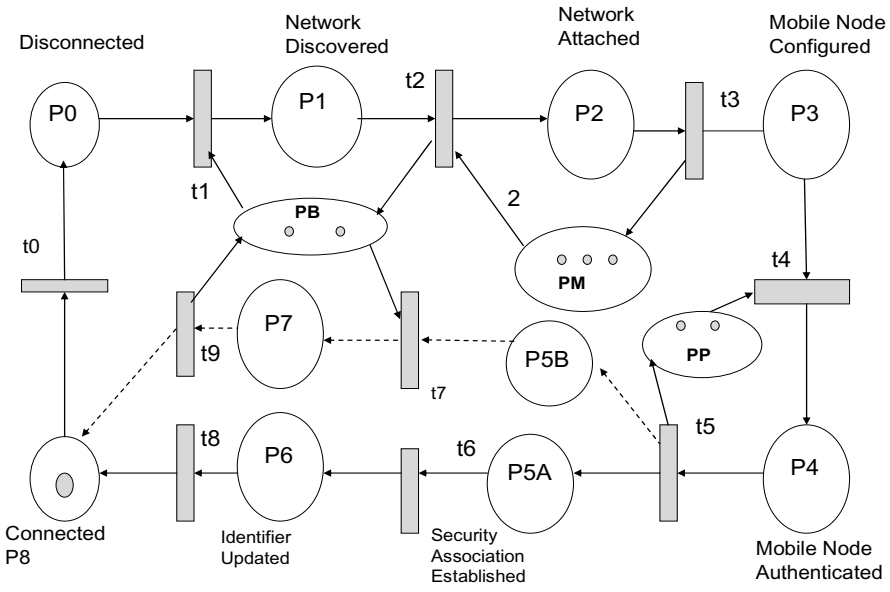


Figure 4.6: A generalized high level Timed Petri net model for handoff

Figure 4.8 shows a subnet level view of a mobility systems model as represented by modular nature of Petri net. It includes the state transitions at each layer and illustrates the component level interaction in each layer and across layers. Several subprocesses in each of the layers, such as discovery, layer 2 authentication, layer 3 address acquisition, layer 3 authentication, security association, and binding update are illustrated here. The reduction of the number of transitions, minimization of time during each of the state transitions,

Table 4.2: Description of places and transitions for handoff

Place	**Description**
P_0	Mobile is in disconnected state
P_1	Network and resources discovered
P_2	Target network selected
P_3	Mobile node is configured and registered
P_4	Mobile node is authenticated
P_{5A},P_{5B}	Security association is established
P_6	Binding update is complete
P_7	Intra-domain binding update is complete
P_8	Mobile is in connected state
P_B	Bandwidth resources
P_M	Battery resources
P_P	CPU resources
Transition	**Description**
t_0	Mobile node gets disconnect trigger
t_1	Mobile node discovers the network and resources at the new PoA
t_2	Mobile node selects the network
t_3	Mobile node goes through configuration and registration
t_4	Mobile node goes through authentication process
t_5	Mobile node goes through key derivation and security association process
t_6	Mobile goes through binding update process
t_7	Mobile node goes through hierarchical binding update
t_8	Data gets redirected to the mobile node
t_9	Network buffering during handoff

parallelization of many of the state transitions within each layer and across layers, and reduction of number of states will contribute to the overall optimization of the handoff process.

Table 4.4 illustrates how three different types of resources (e.g., bytes transferred, CPU cycles and battery power) are used for different handoff operations as described in Chapter 3. These values are based on the experimental results in the mobility testbed explained in Chapter 5. Bytes exchanged refer to the total number of bytes exchanged during the transmit and receipt of signaling messages at the mobile during a specific handoff operation. CPU cycle samples are taken from Oprofile [LE05] analysis in the mobile for different operations during the handoff. Battery power consumption is based on the energy expensed that is based on the bytes received and transmitted on the mobile. Freeney and Nilsson

Table 4.3: Atomic operations during handoff

Transition	Handoff operation	Sub transitions	Sub-operations
t_0	Disconnect trigger	t_{00}	Layer 2 un-reachability test
		t_{01}	Layer 3 unreachability
t_1	Network discovery	t_{01}	Discover layer 2 channel
		t_{12}	Discover layer 3 subnet
		t_{13}	Discover server
t_2	Network Attachment	t_{21}	Layer 2 association
		t_{22}	Router solicitation
		t_{23}	Domain advertisement
t_3	Mobile Configuration	t_{31}	Identifier acquisition
		t_{32}	Duplicate address detection
		t_{33}	Address resolution
t_4	Authentication	t_{41}	Layer 2 open authentication
		t_{42}	Layer 2 EAP
t_5	Security association	t_{51}	Master key derivation
		t_{52}	Session Key derivation
t_6	Binding update	t_{61}	Identifier update
		t_{62}	Identifier verification
		t_{63}	Identifier mapping
		t_{64}	Binding cache
t_7	Hierarchical binding update	t_{71}	Fast binding update
		t_{72}	Local caching
t_8	Media redirection	t_{81}	Tunneling
		t_{82}	Forwarding
		t_{83}	Buffering
t_9	Local data redirection	t_{91}	Local id mapping
		t_{92}	Multicasting/bicasting

Table 4.4: Resource assignment for each of the sub-operations

Transitions	Operations	Resource Consumption		
		Bytes	CPU cycles	Battery (nJ)
t_{00}	Layer 2 unreachability	43	5	51600
t_{01}	Layer 3 unreachability	86	3	103200
t_{11}	Discover layer 2 channel	109	3	130800
t_{12}	Discover layer 3 subnet	110	4	132000
t_{13}	Discover server	450	5	540000
t_{21}	Layer 2 association	99	2	118800
t_{22}	Router solicitation	70	4	84000
t_{23}	Domain advertisement	226	4	271200
t_{31}	Identifier acquisition	1426	5	1711200
t_{32}	Duplicate address detection	164	6	196800
t_{33}	Address resolution	60	3	72000
t_{41}	Layer 2 open authentication	94	3	112800
t_{42}	Layer 2 EAP	2842	6	3410400
t_{43}	Four-way handshake	504	4	604800
t_{51}	Master key derivation (PMK)	0	10	0
t_{52}	Session key derivation (PTK)	0	6	0
t_{61}	Identifier update	352	4	422400
t_{62}	Identifier verification	148	6	177600
t_{63}	Identifier mapping	0	8	0
t_{64}	Binding cache	0	3	0
t_{71}	Fast binding update	110	3	132000
t_{72}	Local caching	0	6	0
t_{81}	Tunneling	60	2	72000
t_{82}	Forwarding	100	2	120000
t_{83}	Buffering	120	3	144000
t_{91}	Local id mapping	40	4	48000
t_{92}	Multicasting/bicasting	192	2	230400

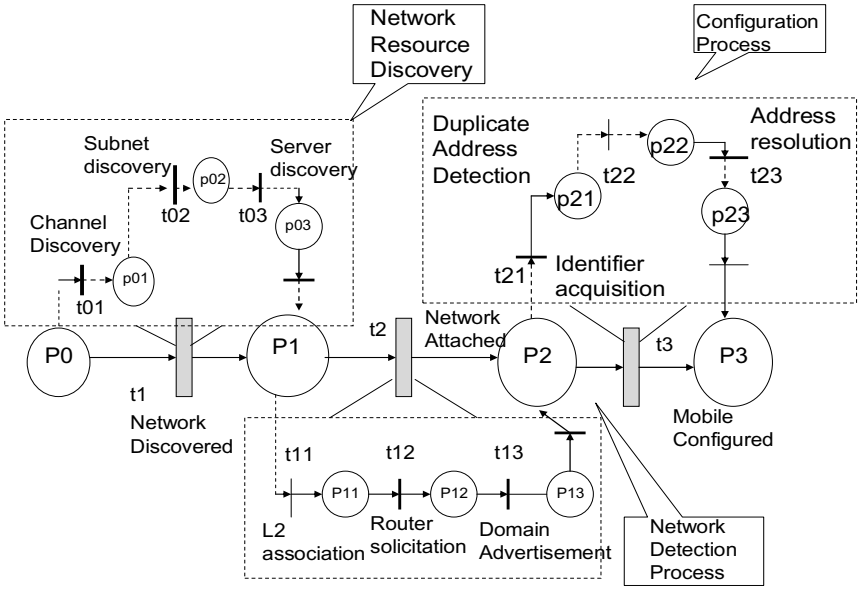


Figure 4.7: Hierarchical decomposition of Petri net-based handoff model

[FN01] model per-packet energy consumption that consists of a fixed component due to channel access and a variable component that is proportional to the size of the packet transmitted and received. That paper also shows how different amount of energy is needed for transmit and receive operations. Thus expensed energy can be described using the linear equation as follows.

$$Energy = m \times size + b \tag{4.1}$$

Where m is the amount energy spent per byte message and b is the fixed size of energy per channel access. It is important to note that the value of m will vary for both transmit and receive operations and energy cost per channel access will also vary for transmit and receive operations. Rima and Izhak [KBR09] cite the examples of energy spent for transmitting and receiving a bit for 802.11-based network. According to that paper, at 50 mW transmit power and 11 Mb/s transmission rate, it takes about 1.3 micro Joule of energy to transmit a byte of data and it takes about 1 micro Joule of energy to receive a byte of data at that rate. If the radio interface works on a slower rate, it takes more energy to transmit a byte. For example, the mobile will consume about 7 micro Joule per byte when the radio operates at 2

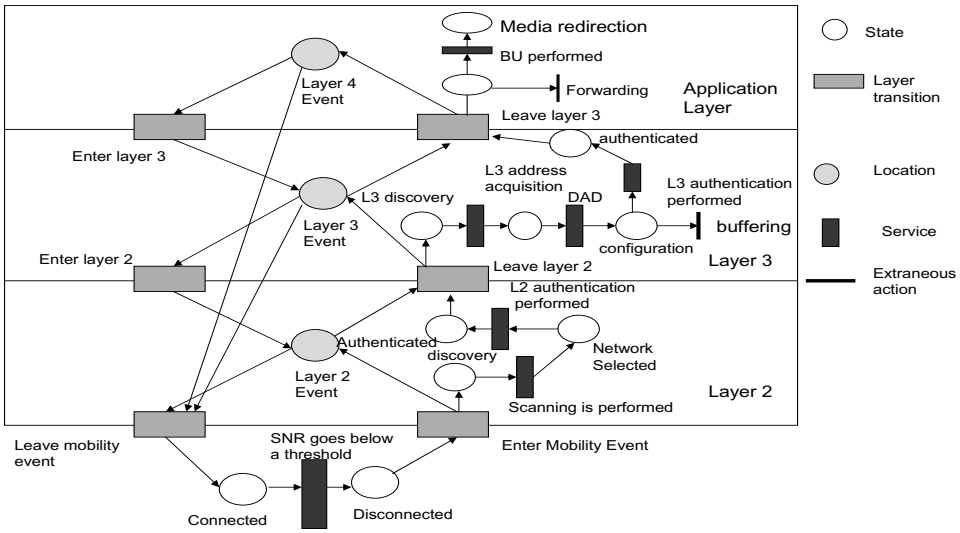


Figure 4.8: Layered modeling with Petri net

Mb/s. According to Sierra wireless card's CDMA 1X EvDo interface, energy consumption rate at 200 mW transmit power is about 1.4 Watt. Thus it takes about 6 micro Joule to transmit one byte at 1.8 Mb/s uplink speed for a CDMA 1X-EvDO interface.

In order to determine what percentage of energy is used for handoff related operations, it is important to find out total amount of data and voice a mobile node transmits and receives on a daily basis and the average number of handoffs a mobile is subjected to. From the experimental calculation, I found out that a mobile node transmits and receives about 7 Kbytes of data when MIPv6 is used as the mobility protocol. This amount will vary if a different mobility protocol (e.g., SIP or ProxyMIP) is used instead. For example, SIP-based mobility protocol will exchange additional data for binding update resulting in about 10 Kbytes of data, but Proxy MIPv6 will require less amount of data to carry out the binding update operation and will result in about 6 Kbytes of data exchange. Thus, the total amount of data transfer and number of times a mobile will need to access a channel will also vary. From a recent online survey, it is shown that a PDA user is subjected to 100 Mbytes of data over a month period and FCC data shows that a mobile is subjected to about 5 Mbytes of voice traffic over month period. Thus, the mobile is subjected to about 3.5 Mbytes of voice

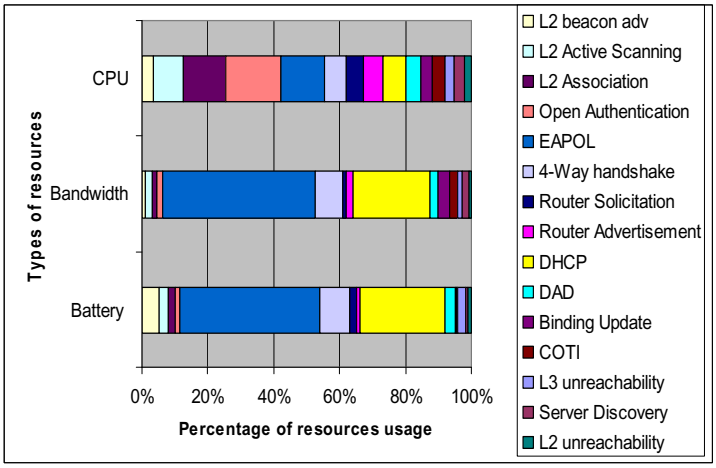


Figure 4.9: Relative resource usage during handoff

and data transfer per day. The number of times a PDA is subjected to handover depends upon the mobility pattern of the type of user (e.g., salesperson, researcher) and the access characteristics (e.g., CDMA and 802.11). From the experimental study [TP06] it is shown that a PDA is subjected to average 10 handoffs during a day in a micro-cellular environment amounting to about 70 Kbytes of data transfer due to handoff when FMIPv6 is used as the mobility protocol in IEEE 802.11-based environment. Thus, handoff will contribute to about 2 percentage of total energy spent in a micro-cellular environment. Handoff rate in a cellular environment will be smaller than the micro-cellular environment. I give an example of approximate handoff rate for a daily commuter in USA. Average cell size for cellular network varies from 5 miles to 25 miles and average one way commute distance for a mobile subscriber in USA is about 20 miles a day. Thus, a commuting mobile subscriber will be subjected to an average of 5 to 6 handoffs per day. However, this value does not include the scenario where the mobile subscriber with dual mode radio can also switch over to a hotspot access (IEEE 802.11) or there are a lot of overlapping cells in a highly population dense urban area.

It is assumed that the tokens in the resource places represent different amounts of handoff resources (e.g., 1 bandwidth token = 100 Kbits, 1 battery power token = 50 mjoules, 1

CPU token = 2 cycles). I give examples of Petri net models for various handoff operations.

Figure 4.9 shows the relative resource usage during a handoff operation based on the values in Table 4.4.

Based on the resource requirement for each of these sub-operations, I model four specific operations, namely *discovery, attachment, authentication, and configuration* processes along with the shared resources needed for these operations.

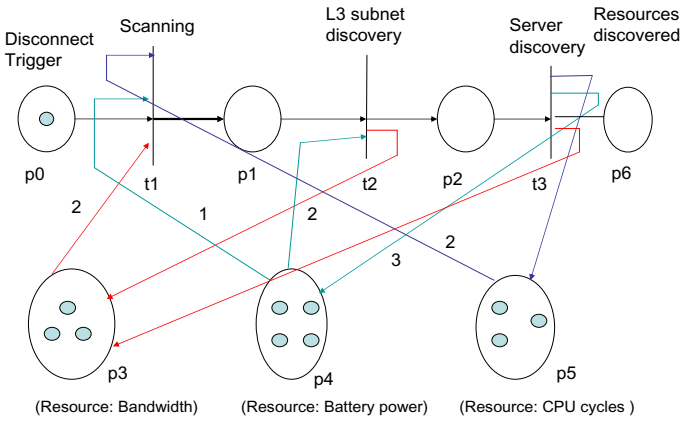


Figure 4.10: Petri net model for discovery

Figure 4.10 models the discovery process with a limited number of tokens that are available in resource places. The resource places are P_3, P_4, and P_5 that represent bandwidth (network capacity), CPU cycles and processing power, respectively.

As soon as the mobile gets a disconnect trigger (e.g., SNR goes below a certain threshold), the mobile is ready to perform the scanning operation. In order to complete the scanning operation, it needs to get the disconnect trigger indication from place P_0 that is considered as a data token, 2 tokens from resource place P_3, 1 token from resource place P_4 and 2 tokens from resource place P_5. Thus, in order for scanning operation to complete, the resource places P_3, P_4 and P_5 should have enough tokens available leading to the firing of transition t1. Once the transition t1 is fired, the resources are released for subsequent operations. As the mobile moves to the next operation (e.g., layer 3 subnet discovery) it may need additional resources of a specific type and release other types of resources. For exam-

ple, in this specific case, layer 3 subnet discovery needs additional amount of battery power since the mobile needs to take care of additional transmittal and receipt of the messages. After layer 3 subnet operation is over, it releases 1 token back to the place P3, bandwidth place. Server discovery is the last operation of the discovery process and it releases 1 resource token of CPU, 3 resource tokens of bandwidth and 1 resource token of power. Thus, after server discovery process is over, the resource places are put back to the initial stage and the mobile moves to next state P6 that has data token. This resource data can be used as an input to the next stage of handoff operation *network attachment*.

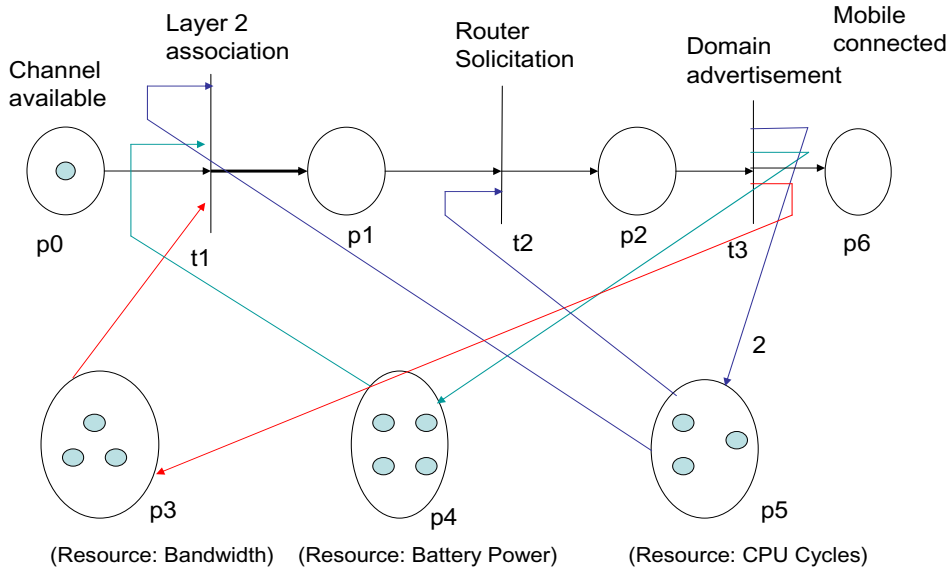


Figure 4.11: Petri net model for network attachment

Similarly, Figures 4.11, 4.12 and 4.14 illustrate the Petri-net models for other subsequent operations such as network attachment, configuration and authentication, respectively.

Figure 4.13 shows the Petri net models for the configuration sub-tasks, namely *identifier acquisition*, *duplicate address detection* and *address resolution*. If any of these operations are done in parallel, then additional resources will be needed to complete these operations during that period.

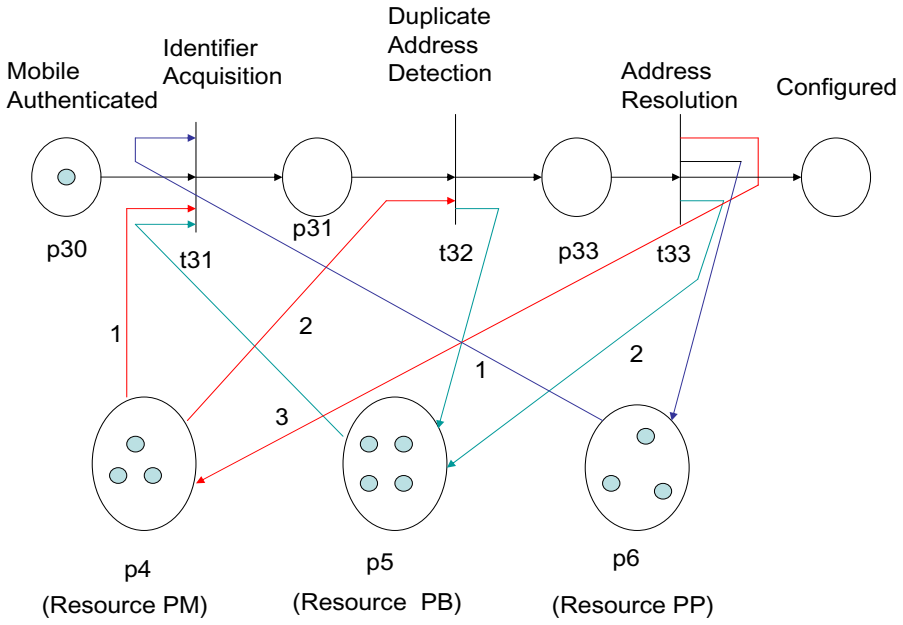


Figure 4.12: Petri net based model for configuration

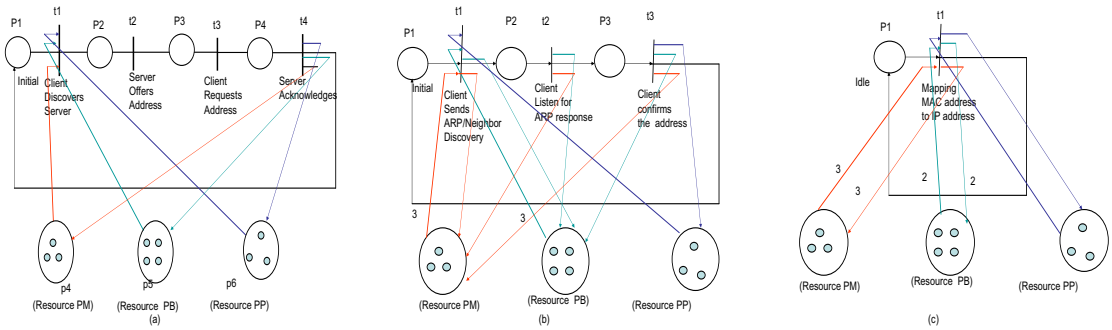


Figure 4.13: Petri net model for configuration subtasks

Figure 4.15 shows the model when these four operations (e.g., discovery, attachment, configuration and authentication) work together to complete parts of the handoff operation. While running operations in parallel may reduce the amount of time needed to complete the handoff, resource sharing and non-availability of data may lead to a deadlock situation where a specific operation cannot complete because of lack of resources.

Figure 4.16 shows the Petri net model based on the data dependency relationship shown in Table 4.1 when there are no concurrent handoff operations. Figures 4.17 and 4.18 illustrate the dependency graphs of two different sequence of operations that involve some level

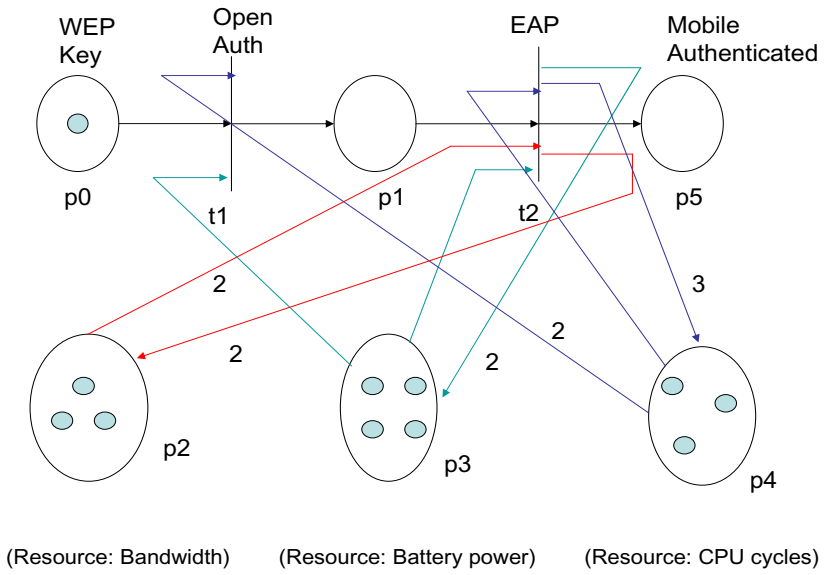


Figure 4.14: Petri net model for authentication

of concurrency. Figure 4.17 illustrates the dependency graph when security association takes place during scanning and Figure 4.18 shows an example when both security association and subnet discovery take place during scanning. Figures 4.19 and 4.20 show the corresponding Petri net models in terms of places and transitions. These models are shown without the resources in the data dependency graphs. It clearly shows how specific data dependency might affect the sequence of execution of handoff operations. A Petri net-based model can be used to analyze various types of mobility events, namely intra-subnet, intra-technology, inter-subnet, and inter-technology handover involving various mobility protocols. This analysis can be carried out by many existing Petri net-based software tools as mentioned in Section 4.3.

Out of the several tools available, I have used two specific tools to model the Petri nets for the handoff, namely TimeNet and MATLAB-based tool. Figure 4.21 shows how to model MIPv6 [JPA04] using TimeNet software. It takes into account the hierarchical feature of Petri net [Zub00] and shows the interaction between different states within a mobility event at a granular level. Hierarchical feature of Petri net provides the ability to decom-

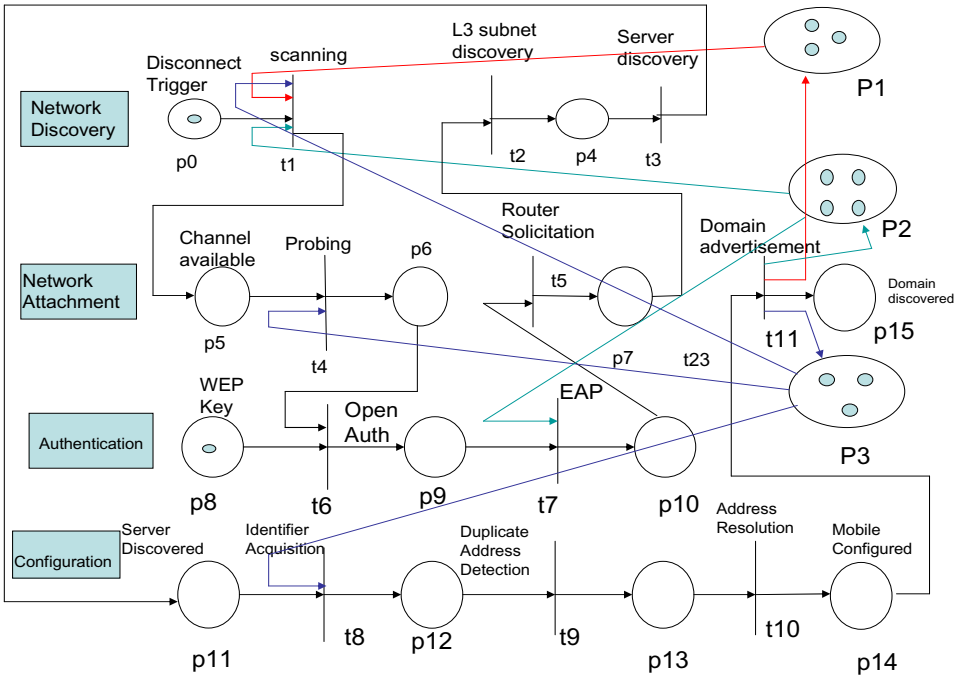


Figure 4.15: Petri net model with combined operations

pose a specific system event into smaller tasks. However, I have used a MATLAB-based Petri net tool [MMP03] to model different handoff components, associated optimization techniques and analyze the associated behavioral properties. This toolbox is equipped with a user-friendly graphical interface. Its functions cover the key topics of analysis such as coverability tree, structural properties (including invariants), time-dependent performance indices, max-plus state-space representations. Some of the MATLAB-based results for different handoff optimization techniques are explained in Chapter 9.

4.10 Petri net-based analysis of handoff event

Many of the behavioral properties of a handoff system, such as reachability or deadlocks leading to allowable sequence of transitions during the handoff event can be verified using Petri net analysis methods. There are two types of Petri net analysis methods: *reachability tree analysis* and *matrix equations*. I analyze deadlocks and the effectiveness of reachability analysis and matrix equations to detect the deadlocks and determine the sequence of

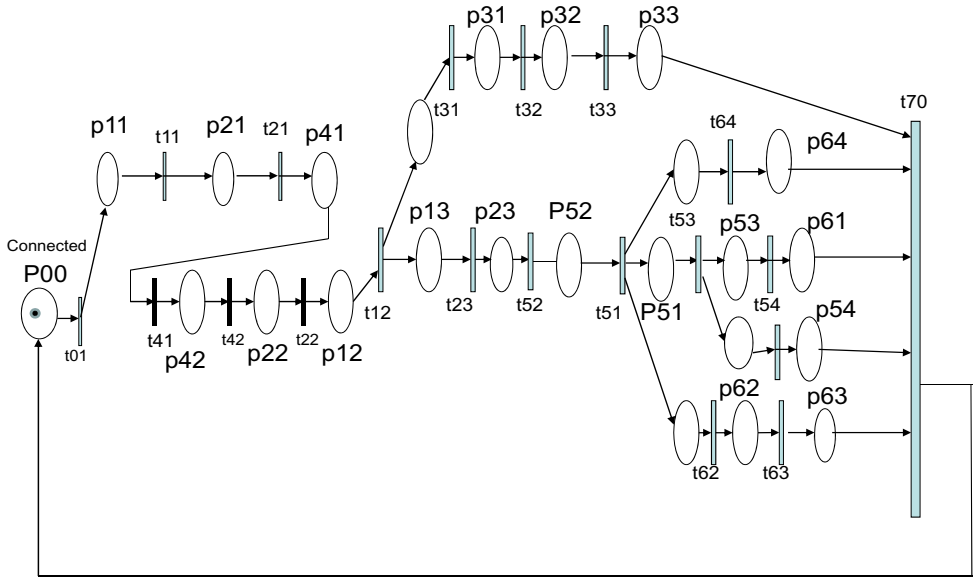


Figure 4.16: Petri net model based on data dependency

transitions that could be possible during the handoff event.

4.10.1 Analysis of deadlocks in handoff

Deadlock is a highly unfavorable situation in which a set of tasks request the resources held by other tasks at the same time. A deadlock situation involving a set of tasks can easily propagate to other tasks, eventually crippling the whole system that remains blocked indefinitely, so deadlock can bring to a system unnecessary costs such as long unavailability of the system and low use of some critical and expensive resources. The concept of liveness in Petri net is closely related to the deadlock situation that has been studied extensively in the context of operating system. Coffman et al. [CES71] discuss the following four necessary conditions that must be held for the deadlock to occur.

1. Mutual exclusion: A resource is either available or allocated to a process that has an exclusive access to this resource.

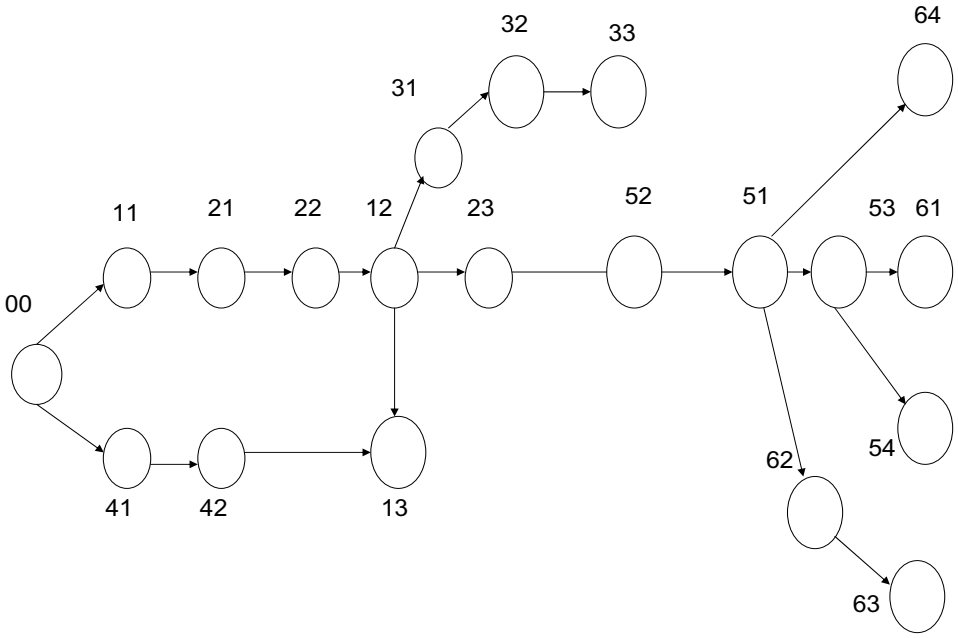


Figure 4.17: Dependency graph: security association during scanning

2. Hold and wait: A process is allowed to hold resource (s) while requesting more resources.

3. No preemption: A resource allocated to a process cannot be removed from the process until it is released by the process itself.

4. Circular wait: Two or more processes are arranged in a chain in which each process waits for resources held by the process next in the chain.

A deadlock in a Petri net is a transition (or a set of transitions) that cannot fire. A transition is live if it is not deadlocked. A transition t_j of a Petri net P is potentially fireable in a marking M_i if there exists a marking $M_j \in R(P,M_i)$ and t_j is enabled in M_j. A transition is live in a marking M_i if it is potentially fireable in every marking in $R(P,M_i)$.

I explain a simple example that can illustrate the deadlock problem in Figure 4.22. Consider a system with two different resources q and r and two processes a and b. If both the processes need both the resources, it will be necessary for the resource to be shared. To accomplish this one requires each process to request a resource and later release it. Figure

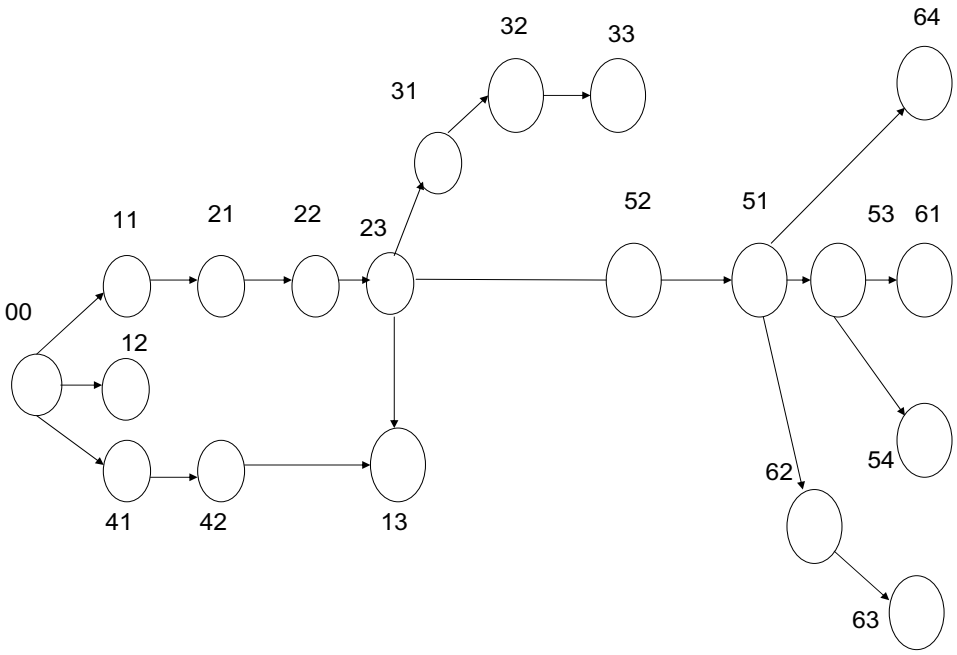


Figure 4.18: Dependency graph: security association and discovery during scanning

4.22 illustrates these two processes and resource allocation with a Petri net. Initial marking indicates that the resources $q(p4)$ and $r(p5)$ are available and processes a and b are ready. By inspecting the Petri net, it is evident that several sequences of transitions are possible.

By doing a reachability analysis, it is evident that the execution sequences, $t_1t_2t_3t_4t_5t_6$ and $t_4t_5t_6t_1t_2t_3$ do not result in deadlock. However, if a specific execution sequence consists of t_1t_4, where process a has q and wants r; process b has r and wants q, then the system is deadlocked and neither process can proceed.

These conditions for deadlock can also be applied to the system model representing a handoff event. In case of a Petri net model representing a handoff event, deadlock may happen due to absence of data from a preceding event that may be needed to successfully enable a transition. Lack of resources such as power, bandwidth, CPU cycles or memory due to resource sharing will also result in deadlock. Thus, either the absence of data from the preceding event or lack of resources may prevent a transition to fire and will result in deadlock. A specific deadlock scenario can be determined by performing a reachability analysis. By investigating the reachability markings one can determine which specific

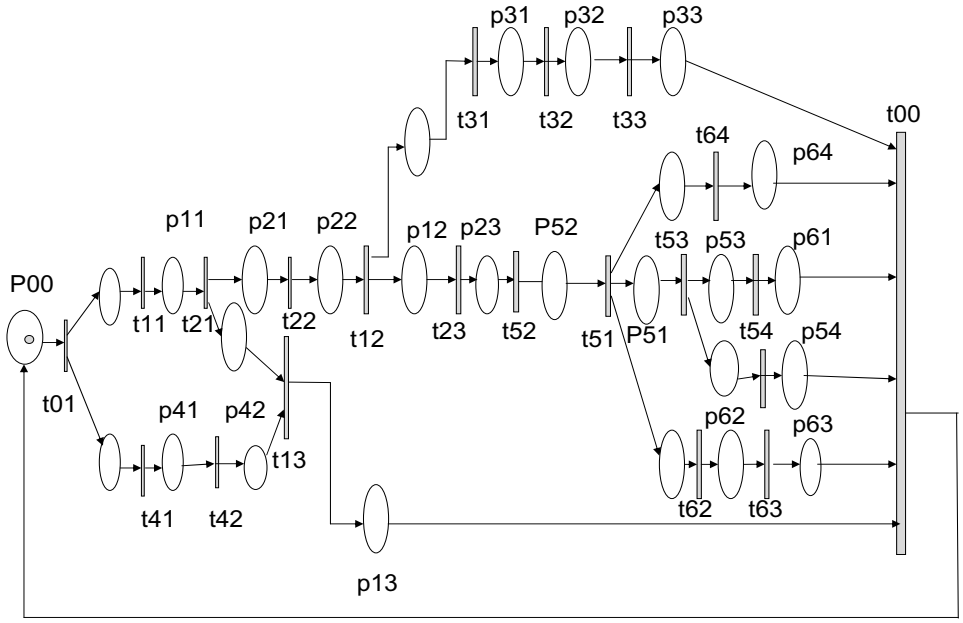


Figure 4.19: Petri net model without resources: security association during scanning

sequence of handoff operations may lead to deadlock situation.

Chapter 9 illustrates some examples of specific sequence of schedules that may lead to blocking event and how these can be detected by using a reachability analysis.

4.10.2 Reachability analysis

An important issue of designing a handoff systems model is to determine whether the system can reach a specific state or exhibit a particular functional behavior during its transition from one state to another state. In order to find out whether the modeled system can reach a specific state as a result of a required functional behavior, it is necessary to find such a sequence of firings that would transform a marking M_0 to M_i where the sequence of firings represents the required functional behavior. A real system may reach a given state as a result of exhibiting different permissible patterns of the functional behavior that would transform M_0 to the required M_i. A marking M_i is said to be reachable from a marking M_0 if there exists a sequence of transitions firing that transforms a marking M_0 to M_i. A marking M_i is said to be immediately reachable from M_0 if firing an enabled transition in

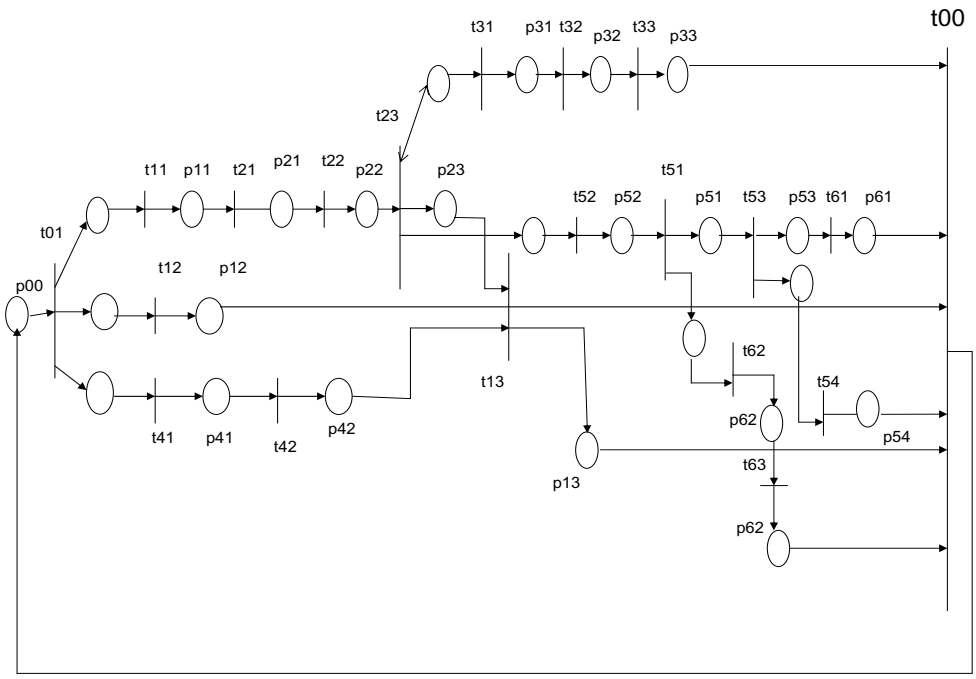


Figure 4.20: Petri net model: security association and subnet discovery during scanning

M_0 results in M_i.

The list of reachable states from initial state (initial marking) can be found out in various ways. Reachability analysis is one way of finding out if the mobile can reach a specific state after going through a set of transitions. A reachability analysis graph of Timed Petri net clearly indicates the sequence of transitions that are possible from an initial marking M_0 and the ones that are not reachable from a specific marking M_i. Reachability analysis of a Petri net leads to a construction of coverability tree. Given a Petri Net N, and its initial marking M_0, one can obtain as many new markings as the number of the enabled transitions. From each new marking one can obtain more markings. Repeating the procedure over and above results in a tree representation of markings. Nodes represent markings generated from M_0 and its successors, and each arc represents a transition firing that transforms one marking to another.

Figure 4.24 is an example of coverability tree that illustrates different markings of the sample Petri net model shown in Figure 4.23. As different transitions fire, it clearly shows the sequence of transitions that leads one state to another state.

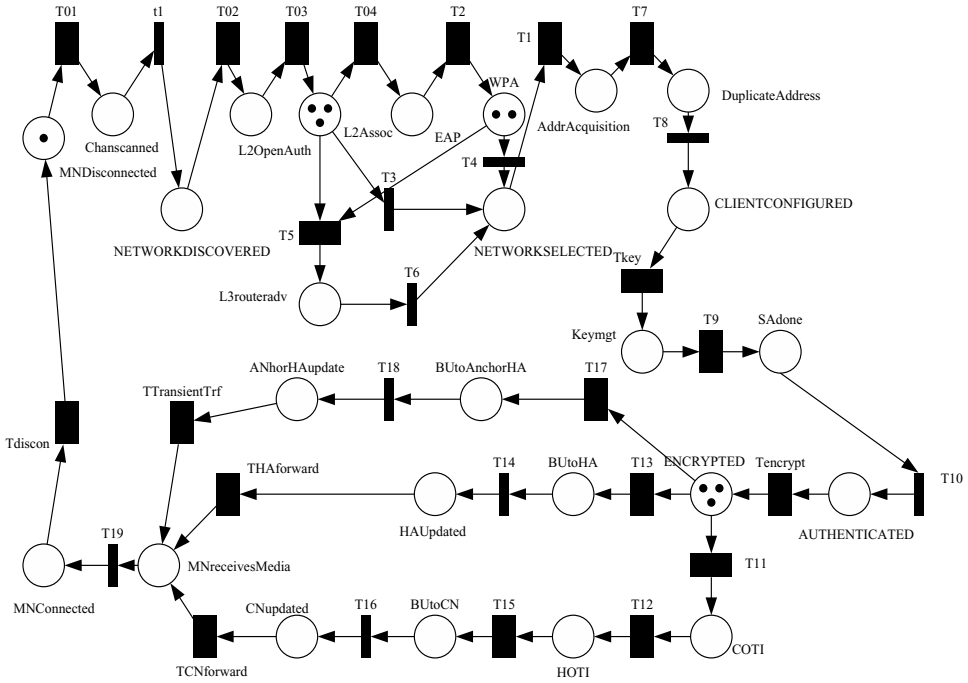


Figure 4.21: Petri net representation of Mobile IPv6 using Time Net

Reachability analysis of timed petri nets representing a mobility event can provide some insights into which specific sequences of transitions are possible during the handoff process when the mobile goes from *Connected* to *Disconnected* and then to *Connected* state again. It can also determine if specific target transition is attainable from a given transition. For example, it is possible to find out if a mobile that is in a configuration stage can proceed to the binding update stage if it follows a specific sequence of operations.

By looking at the coverability tree, one can easily determine a specific sequence of executions that should be avoided. For example, a specific marking is a deadlock if no further transition is possible from that marking. It is desirable to prevent deadlocks in the system by avoiding the transitions that give rise to these deadlocks.

4.10.3 Matrix equations

A second approach to analyze the Petri net-based handoff system is based on matrix view of Petri nets. Matrix-based analysis can provide the answer such as whether a specific

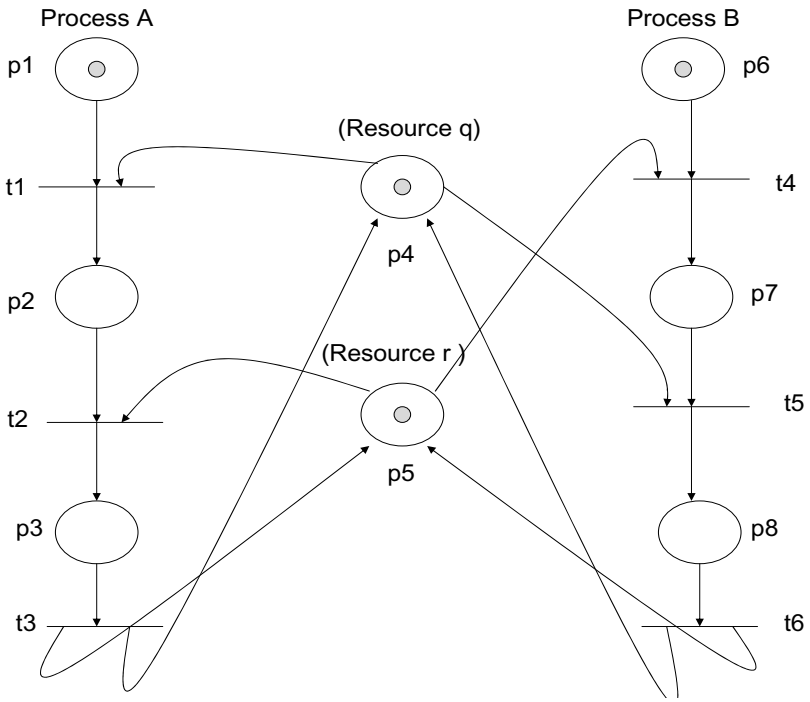


Figure 4.22: Example of deadlock from resource sharing

sequence of transition is attainable or not from a given marking. A Petri net (P,T,I,O) can have two matrices D^- and D^+ that represent the input and output functions, respectively. Each of these matrices can be represented as m rows by n columns, where m rows represent transitions and n columns represent places. D^- defines the inputs to the transitions and D^+ defines the outputs. As per Peterson [Pet81] Matrix D can be defined as

$$D = D^+ - D^- \tag{4.2}$$

The development of the matrix-based analysis for Petri net theory provides a useful tool for solving the reachability problem. Assume that a marking M_j is reachable from a marking M_i, then there exists a sequence σ of transition firings that will lead from M_i to M_j. This means that $f(\sigma)$ is a solution, in nonnegative integers, for x in the following matrix equation.

$$M_j = M_i + x.D \tag{4.3}$$

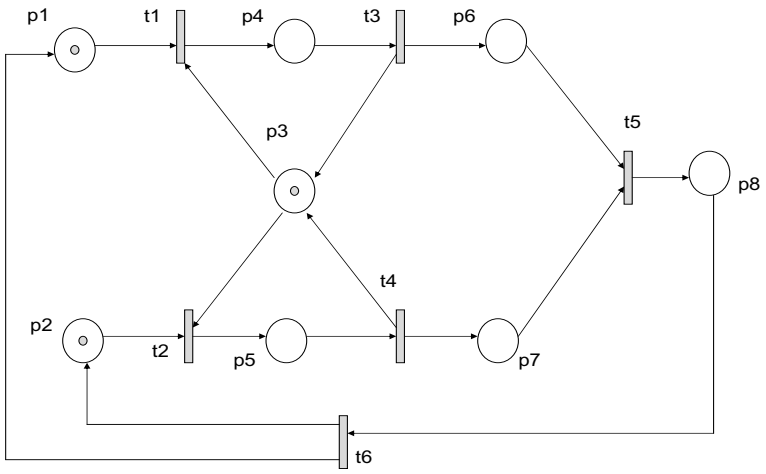


Figure 4.23: A Petri net with shared resources

If M_j is reachable from M_i then the equation has a solution in nonnegative integers; if the equation has no solution, then M_j is not reachable from M_i.

I describe a specific example, the discovery operation to illustrate the usefulness of Matrix-based equations to determine a sequence of operations.

Below are the matrix equations for analysis of the discovery operation shown in Figure 4.10. Equations 4.4, 4.5, and 4.6 are the input matrix, output matrix and incidence matrix, respectively.

$$\mathbf{D}^{-} = \begin{bmatrix} 1 & 0 & 0 & 2 & 2 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.4}$$

$$\mathbf{D}^{+} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.5}$$

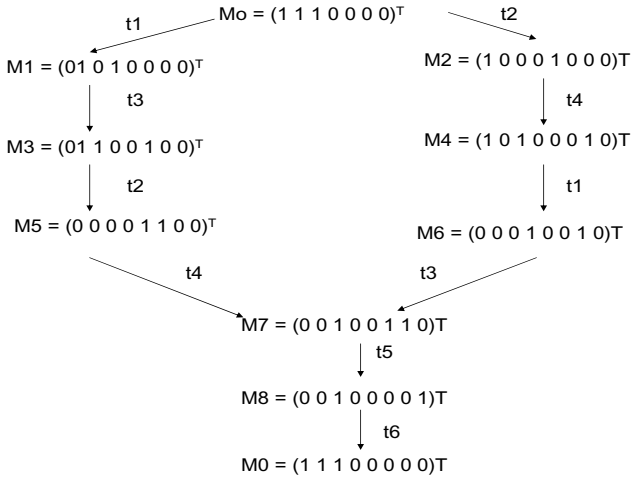


Figure 4.24: Coverability tree

$$\mathbf{D} = \mathbf{D}^{+} - \mathbf{D}^{-} = \begin{bmatrix} -1 & 1 & 0 & -2 & -2 & -2 & 0 \\ 0 & -1 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 3 & 2 & 1 \end{bmatrix} \tag{4.6}$$

Given a firing sequence $\sigma = t_1 t_2 t_3$, it translates to a firing vector f(σ) = (1,1,1). Applying equation 4.3, it would be possible to find the final marking with the firing vector f(σ)=(1,1,1).

M_j = (1,0,0,3,4,3,0)+(1,1,1).D produces the final marking M_j = (0,0,0,3,4,3,1).

Thus, it is evident that M_j is reachable from the initial marking M_i with a sequence of transition $t_1 t_2 t_3$ that corresponds to firing vector (1,1,1). A firing vector determines the number of times a specific transition is fired on. Similar matrix analysis can be done for other handoff related operations such as configuration, authentication and attachment to determine the reachability.

4.11 Evaluation of systems performance using Petri nets

The systems performance of a Timed Petri net model representing a mobility event can be verified in several ways as described in [HV87] [RH80], [Zub80], [Mur85]. I briefly describe the analysis by Ramamoorthy and Ho [RH80] for DTTPN (Deterministic Timed Transition Petri Net) models. I then explain three approaches to study the systems performance of a mobility model using Petri net. First approach illustrates that minimum cycle time can be obtained from the Petri net model using the number of directed circuits, number of transitions, and timing associated with each transition. Minimum cycle time is an indicator of maximum performance. The second approach involves computing the shortest distance by using Floyd's algorithm [Flo62]. The third approach is based on a scalar measure called the resource-time product (RTP) for timed petri net models. The resource-time product is defined as the product of the number of tokens (resources) and the length of the time these tokens are reserved for firing an enabled transition. Using the concept of the resource-time product, one can derive a lower bound on the minimum period at which intervals each transition can initiate its firing.

Here are some of the related definitions as described in [Wan98].

Definition 1 In a Petri net, *a directed path* is a sequence of places and transitions, $p_1t_1p_2t_2...p_k$ if transition t_i is both an output transition of place p_i and input transition of place p_{i+1} for $1 \leq i \leq k\text{-}1$.

Definition 2 In a Petri net, *a directed circuit* sequence of places and transitions, $p_1t_1p_2t_2...p_k$ if $p_1t_1p_2...p_k$ is a directed path and p_1 equals p_k.

Definition 3 A Petri net *is consistent* if and only if there exists a nonzero integer assignment to its transitions such that at every place, the sum of integers assigned to its input transitions equals the sum of integers assigned to its output transitions.

Definition 4 A Petri net *is decision-free* if and only if, for each place in the net, there is exactly one input arc and exactly one output arc.

4.11.1 Cycle time-based approach

The Petri net model representing the general mobility systems is actually a decision free Petri net, where a minimum cycle time is an indicator of maximum performance. The cycle time is represented as $C = \max T_k/N_k{:}k = 1, 2, 3...q$, where T_k = sum of the execution times of the transmissions in circuit k and (N_k) is the total number of tokens in the places in circuit k and q is the number of circuits in the net. In case of a system model representing a mobility event, these values can vary depending upon the number of transitions and sequence of transitions involved during a cycle.

Table 4.5 shows how the overall cycle time using directed circuits approach is affected when some of the optimization techniques, such as hierarchical binding update, proactive discovery, configuration, and anchor-based security association are applied to a generalized Petri net system mobility model shown in Figure 4.6. These optimization techniques are explained in details in Chapter 5. In this table, time duration for each of the transition is represented in terms of "t" unit of time.

Table 4.5: Cycle time for Petri net with mobility optimizations

Type of Optimization	Loops in the state transition path	T_i	N_i	T_i/N_i
No optimization	p0t1p1t2p2t3p3t4p4t5p5t6p6t7p10	24t	1	24t
Hierarchical binding update	p0t1p1t2p2t3p3t4p4t5p5t8p7t9p10	19t	1	19t
Proactive	p0t9p10	2t	1	2t
Maintain security binding	p0t1p1t2p2t3p5t6p6t7p10	19t	1	19t

Floyd algorithm

```
/* Assume a function edgeCost(i,j) which returns the cost
   of the edge from i to j (infinity if there is none).
   Also assume that n is the number of vertices and
   edgeCost(i,i)=0
*/ \\
 int path[][];\\

 /* A 2-dimensional matrix. At each step in the algorithm,
  path[i][j] is the shortest path from i to j using
  intermediate vertices (1..k-1). Each path[i][j] is initialized
  to edgeCost(i,j) or infinity if there is no edge between
  i and j. */

procedure FloydWarshall ()\\
  for k: = 1 to n \\
    for each (i,j) in (1..n)\\
      path[i][j] = min ( path[i][j], path[i][k]+path[k][j]);\\
```

4.11.2 Floyd algorithm-based approach

In order to verify the systems performance of a mobility event that demonstrates a specific optimization methodology and to determine if the system satisfies the desired threshold for cycle time one can generate a Place matrix P, Transition matrix Q. Entry (A,B) in the matrix P equals x if there are x tokens in place A, and place A is connected directly to place B by a transition. Entry (A,B) in the transition matrix "Q" equals t_i if A is an input place of transition t_i, and B is one of its output places. Entry (A,B) contains symbol "w" if A and B are not connected directly. Given a threshold value of cycle time C, one can generate a distance matrix CP-Q. Then using the Floyd [Flo62] algorithm, one can then determine matrix S. By inspecting the values of the diagonal elements of matrix S, it is possible to determine if the system satisfies the desired system performance.

A sample Pseudo code for Floyd algorithm is given below.

Using Floyd algorithm, there are three possible cases to figure out the systems performance:

1. If all diagonal entries of matrix S are positive (i.e., CN_k-T_k > 0 for all circuits), the system performance is higher than the given requirement.

2. If some diagonal entries of matrix S are zeros and rest are positive (i.e., CN_k-T_k = 0 for some circuits and CN_k-T_k > 0 for the other circuits) the systems performance just meets the given requirement

3. If some diagonal entries of matrix S are negative (i.e., CN_k-T_k < 0 for some circuits), the system performance is lower than the given requirement.

A set of matrices that represent a sample Petri net model consisting of four places with a given value of C = 3 are shown in equation 4.7 and 4.8. In this specific example, all the diagonal entries of matrix S are non-negative. Thus, the system performance just meets the threshold requirement of C = 3. By way of this analysis, one can easily determine if a specific Petri net model representing a handoff model can meet the desired system performance.

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} w & 2 & w & w \\ w & w & 3 & w \\ w & w & w & 4 \\ 3 & w & w & w \end{bmatrix} \tag{4.7}$$

$$\mathbf{CP} - \mathbf{Q} = \begin{bmatrix} \infty & 1 & \infty & \infty \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 1 \\ 0 & \infty & \infty & \infty \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \tag{4.8}$$

In Section 9, I compare various optimization techniques using cycle time-based approach and Floyd algorithmic based approach.

4.11.3 Resource-Time Product approach

The resource-time product is defined as the product of the number of tokens (resources) and the length of time these tokens are reserved for firing an enabled transition. It is assumed that each transition in the timed Petri net model has a given time delay in completing its firing. Using the concept of resource-time product, one can derive a lower bound on the minimum period at which intervals each transition can initiate firing. Since the Resource-Time Product is based on S-invariant and T-invariant, I describe details of these two invariants as explained originally by Murata in [Mur85].

4.11.3.1 Invariant analysis

I provide the details of S-invariant and T-invariant and describe how these are calculated.

An integer solution x of the homogeneous equation

$$A^T x = 0 \tag{4.9}$$

is called a T-invariant. A is the incidence matrix. The non-zero entries in a T-invariant represent the firing counts of the corresponding transitions that belongs to a firing sequence transforming a marking M_0 back to M_0.

Similarly, an integer solution y of the transported homogeneous equation

$$Ay = 0 \tag{4.10}$$

is called an S-invariant. The non-zero entries in a S-invariant represent weights associated with the corresponding places so that the weighted sum of tokens on these places is constant for all markings reachable from an initial marking.

If it is assumed that a timed Petri net N under consideration is periodically functioning with a period of τ units of time, during which time the initial marking is reproduced and transition i fires z_i times, where $z_i > 0$ for each i. Thus, we have Z_i=cX_i or in terms of

n-vectors, $z = [z_i]$ and $x = [x_i]$,

$$z = cx \geq x \tag{4.11}$$

where c is a positive integer and x is a minimal T-invariant of N such that $x_i > 0$ for each i. τ_m is called the minimum cycle time in a timed Petri net, where the time delay of each transition firing is given.

When a transition i is enabled, a_{ij}^- tokens will be reserved in place j for at least d_i units of time. Thus, the resource-time product due to these a_{ij}^- tokens reserved is $a^- * d_i$. Since transition i fires z_i times during one period τ units of time, the resource-time product for firing transition i is $a^i * d_i * z_i$. Applying this to all the transitions in N, one gets the total resource-time product for place j as following,

$\sum_{i=1}^{n} a_{ij}^- d_{ij} z_i$.

Thus, resource time product for all m places are given by a column vector expressed in the following matrix product.

$(A^-)^T$Dz, where D is the diagonal matrix of delays d_i's.

In the above resource-time products, we consider only tokens that are reserved for firing.

As per [Mur85], on further analysis it is found that,

$$\tau \geq y^T (A^-)^T Dx / y^T M(t_o) \tag{4.12}$$

Therefore, a lower bound of the minimum period maybe given by

$$\tau_{min} = maxyT(A^-)^T Dx / y^T M(t_o) \tag{4.13}$$

Where the maximum is taken over all independent minimal support S-invariants, $y_k \geq 0$. For a timed marked graphs, each directed circuit C_k yields a minimal-support S-invariant y_k. Thus, equation 4.9 can be reduced to τ_{min} = max *Total delay in* $C_k / M_0(c_k)$ which is

equivalent to circuit analysis approach.

I describe an example of how a cycle time is derived using RTP approach. Figure 4.25 describes a Petri net of a readers/writers system, where k tokens in place $p1$ represents k processes which may read and write in a shared memory represented by p3. As per the Petri net model, up to k processes maybe reading concurrently, but when one process is writing, no other processes can be reading or writing. Equation 4.14 shows the incidence matrix A and forward incidence matrix A^-.

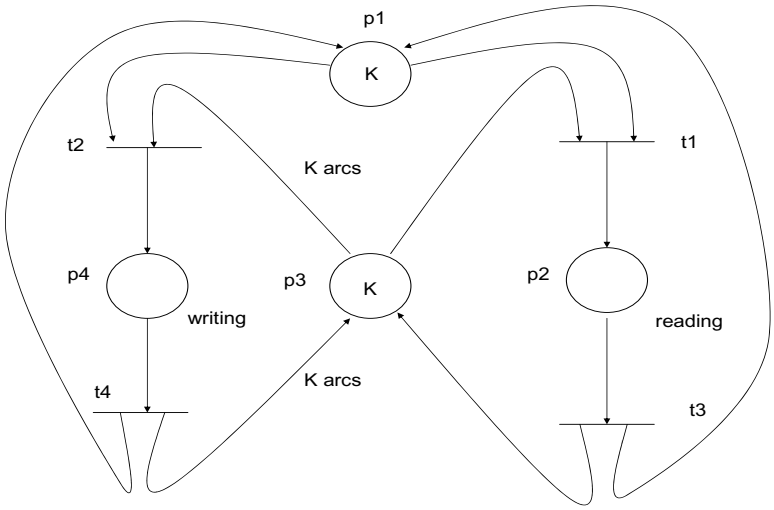


Figure 4.25: Petri net for resource time product calculation

System performance or cycle time for a handoff event can be measured by using any one of the above approaches. While determining cycle time using directed circuit method is a reasonable approach as it gives a visual solution, it has other shortcomings since one needs to enumerate a number of circuits for a system containing a large number of places and transitions. Floyd algorithm-based approach or RTP-based approach is preferred when the number of places and transitions is large. In order to determine the cycle time, one needs to determine the S-invariants and T-invariants.

$$\mathbf{A} = \begin{bmatrix} -1 & 1 & -1 & 0 \\ -1 & 0 & -k & 1 \\ 1 & -1 & 1 & 0 \\ 1 & 0 & k & -1 \end{bmatrix}, \mathbf{A}^{-} = \begin{bmatrix} -1 & 1 & -1 & 0 \\ -1 & 0 & -k & 1 \\ 1 & -1 & 1 & 0 \\ 1 & 0 & k & -1 \end{bmatrix} \tag{4.14}$$

In this specific example as shown in Figure 4.25, there are two independent S-invariants, $y_1 = (1101)^T$ and $y_2 = (011k)^T$ and two independent T-invariants $x_1 = (1010)^T$ and $x_2 = (0101)^T$. A firing count vector of a firing sequence leading back to the initial marking after firing each transition at least once is given by $x = x_1 + x_2 = (1111)^T$. Let the time delay of transition i be the d_i, i = 1,2,3,4. Since the initial marking is given by $M(t_o)= (k0k0)^T$, evaluating Eq. (4.11) for the two S-invariants, y_1 and y_2, we get

τ_m = Max $(d_1+d_2+d_3+d_4)$/k, $(d_1+\text{k}d_2+d_3+\text{k}d_4)$/k = $d_2+d_4+(d_1+d_3)$/k.

That means that transitions t_1, t_2, t_3 and t_4 can initiate at most once every $[d_2+d_4+(d_1+d_3)/\text{k}]$ or units of time, since x = $(1111)^T$ is the minimal T-invariant such that $x_i > 0$ for each i in this timed Petri net.

In Chapter 9, I show how some of these approaches can be used to model the handoff systems.

4.12 Opportunity for optimization

In a Petri net model representing a handover process, a cycle time is a representative of the systems performance. Ordering of execution of the handoff processes and the resources in the system play an important role in determining the cycle time. Figure 4.26 illustrates Petri net representation of possible sequence of execution for a pair of processes, such as Pa and Pb that are part of a mobility event. Thus, sequence of execution of sub-tasks during a mobility event affects the overall cycle time although the amount of system resources expensed may vary because of parallel execution of events. Amount of resources utilized

during a parallel operation increases the amount of peak resources utilized.

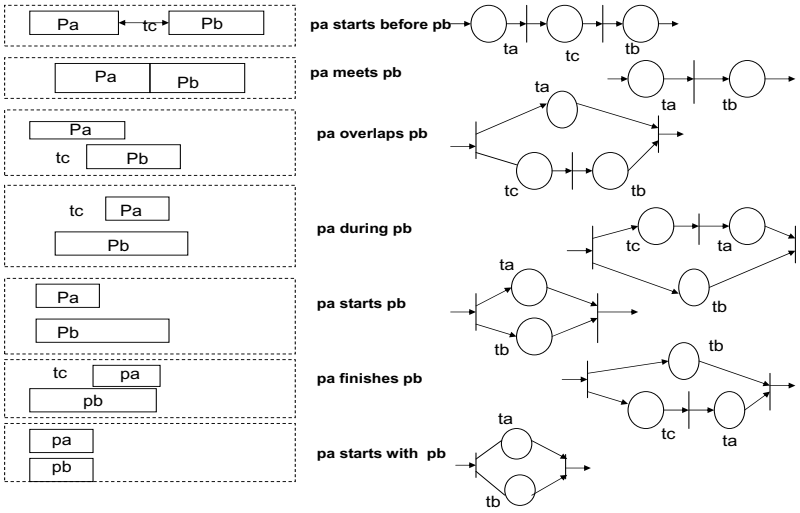


Figure 4.26: Petri net mapping for sequence of events

Thus, several optimization techniques can then be applied to the generalized mobility model to obtain the corresponding optimized models. These techniques can also be applied to the sub-processes within the overall system or in a hierarchical manner for each of these sub-processes. Performance evaluation can then be performed for each of these optimized models and be compared with the non-optimized models. Most of the optimization techniques that I will describe in Chapter 5 can primarily be mapped into sequential, concurrent, and proactive modes of operation using Petri net model. In Figure 4.26 a proactive mode of operation can be equated to a variation of case where operation *Pa* happens during operation *Pb*. It implies that some parts of operation *Pb* start before operation *Pa*. In the following Section, I describe few ways of optimizing the handoff events using different sequence of operations.

4.12.1 Analysis of parallelism in handoff operations

If there are two operations that are not mutually exclusive, then these operations can be done in parallel. Two operations can be done in parallel, only if there is no resource con-

straint or starting of one operation does not depend upon data of the other operation. However, in some resource constrained environment, these operations may need to share some common resources. In those situations, there maybe deadlocks due to lack of resources.

Analysis of sequence of events and their precedence relationship among these operations help to analyze the opportunity for parallelism. In some situations, when execution of a specific event does not depend upon the data from another event, these two events do not need to occur in sequence and there is an opportunity for parallelism. Here are some examples of parallelism that are possible during handoff operations.

Example 1: A layer 3 PoA (Point of Attachment) can be discovered during a layer 2 association if some of the router advertisement information (e.g., prefix of the RA, default gateway) can be passed as part of layer 2 beacon. One way this can be achieved by beacon-stuffing approach [CPRW07] whereby layer 3 information can be passed as part in the layer 2 beacon advertisement. This approach results in parallel operation leading to simultaneous discovery of layer 2 and layer 3 points of attachment. Thus, it reduces the time taken compared to if those operations would have been done in sequence.

Example 2: In normal situation, duplicate address detection is performed only after an identifier is obtained. However, a proactive look-up operation for the uniqueness of the address before it is assigned to the mobile can help speed up the identifier configuration process. Alternatively, address uniqueness can be done during the process of address acquisition.

Example 3: Another possible opportunity for parallelism could be execution of authentication and key derivation process being performed in parallel. Authentication is usually performed using EAP over L2 or EAP over L3 that generates PMK (Pairwise Master Key). Generation of PTK (Pairwise Transient Key) is followed by regular authentication process. If these two operations can be done in parallel, handoff delay due to authentication will be reduced.

Example 4: Forwarding, buffering and multicasting can all be done in parallel to reduce

the loss of data at the expense of additional bandwidth resources.

4.12.2 Opportunity for proactive operation

In some cases, where parallel operations among the handoff processes are not possible, many of the operations can be completed proactively. Proactive operations means, some of the handoff related operations can be done in the current network before the mobile moves to the new network. Network discovery, IP address acquisition process, and authentication processes are some of the handoff related operations that can be performed ahead of time in the serving network before the mobile moves to the new network. However, additional resources are needed to support these proactive operations. I will describe many of the proactive mechanisms for handoff in Chapter 5.

4.13 Concluding remarks

Based on the data dependency, resource analysis for different handoff operations it is possible to build a mobility model using Timed Transition Petri net for both optimized and un-optimized handoff systems. Using the hierarchical approach of Petri net, each of the handoff components can be modeled independently based on the primitive operations associated with the respective handoff component. For example, handoff component such as layer 3 configuration is based on several layer 3 primitive operations namely, IP address acquisition, duplicate address detection and address resolution. Petri net models for each of these handoff components (e.g., discovery, configuration) can be synthesized to build the complete handoff system. Depending upon the types of handoff and whether one would like to investigate the systems performance (e.g., cycle time) or check the anomaly of the handoff system (e.g., deadlocks), one can apply a specific Petri net based methodology. For example, in order to verify the systems performance one can use Floyd algorithm approach, in order to determine the exact cycle time, one can use circuit-timed based approach.

On the other hand if one needs to determine the system anomaly such as deadlocks, one would need to determine that using reachability analysis. By investigating the Petri net model for un-optimized handoff system one can determine the opportunity for parallelism whereby some of the handoff related operations could run in parallel, but this parallel operations will need some modifications of the behavior of the current systems. For example, having the ability to discover layer 3 point of attachment (default router) during the discovery of layer 2 point of attachment (access point), one would need to modify the layer 2 beacon so that it can carry router's address or the subnet prefix. Thus, a mobility model can help analyze the performance of the existing handoff systems while it also offers the ability to predict the behavior of the new systems that can be designed based on the available resources and expected systems performance.

Chapter 5

Mobility optimization techniques

In this chapter, I describe the key techniques that I have developed to optimize several basic operations of a mobility event at different layers and highlight the associated key principles for those optimization techniques. These techniques are based on a few fundamental principles, such as reduction of signaling messages during the basic operation, minimizing the traversal distance of the data, reduction of data and signaling overhead, minimization of look-up cost, caching, parallelization of sequential handoff operations, proactive operations, cross layer triggers, and localization of binding updates. I explain these techniques as those are applied to optimize different handoff components described in Chapter 3 and describe the experimental results from some of these optimization techniques.

In Chapter 9, I apply these optimization techniques to Petri net-based mobility model and then experiment with a few of these proactive techniques to evaluate the overall systems performance.

5.1 Summary of key contribution and indicative results

I summarize below key contributions of some of my proposed optimization mechanisms for different handoff components that I analyzed in Chapter 3. In this section, I highlight the key technical problems my proposed mechanisms address, briefly describe the proposed

mechanisms and highlight some of the key advantages.

I describe the detailed approach, related work and experimental results of each of these mechanisms in Sections 5.3 through 5.11.

5.1.1 Discovery

Discovery of the network elements and resources during handover in a heterogeneous access network depends upon the respective layer 2 discovery mechanism. These mechanisms introduce delays that are dependent on the underlying access characteristics. Currently, there is no existing mechanism that can discover the network elements in the neighboring networks in an access independent manner prior to mobile's handover.

I designed an application layer discovery mechanism that enables the discovery of network parameters and resources of the target network in an access independent manner prior to mobile's handover to the target network. The mobile then caches the addresses of the discovered network elements, namely access point's MAC address, channel number, router's IP address, and IP address of the authentication server so that it can perform many of the handoff related operations namely authentication, security association and configuration ahead of time.

The following are some of the key benefits of my application layer discovery mechanism.

- ◇ Using my proposed mechanism, a mobile can discover the network elements at all layers of the target network (e.g., access Point, routers, AAA servers) without depending upon the underlying access mechanism. While the lower layer discovery mechanisms such as IEEE 802.11u [Gas05] allows the mobile to discover the higher layer network parameters, 802.11u mechanisms are limited to 802.11 networks only. Gloserv [AS04] is a prior application layer discovery mechanism that discovers different types of services but it does not discover the network elements. My proposed

mechanism can discover the network parameters in an access independent manner (e.g., 802.11, CDMA) using application layer protocols.

- My proposed mechanism reduces the network discovery delays by proactively discovering the network elements and caching these locally in the client. While there are access dependent optimization techniques by Shin et al. [SSFR04], Montavont et al. [MMN05], and Velayos et al., [VK04] to reduce the discovery latency of 802.11 networks during mobile's handover, my proposed mechanism is access independent and optimizes the network discovery operation for both homogeneous handover and heterogeneous handover.

I describe the related work, details of my proposed mechanisms and experimental results in Section 5.3.

5.1.2 Authentication

In general, layer 2 authentication is dependent upon respective access mechanisms (e.g., CDMA, 802.11) and is performed after the mobile hands over to the new network. Although some of the existing layer 2 access (e.g., 802.11) support pre-authentication mechanism (e.g., 802.11i), where part of the authentication process (e.g., derivation of pre-shared keys) is performed before the mobile moves to the target network, this pre-authentication support is limited to intra-subnet handoff only. Thus, a mobile will still be subjected to authentication delay during handover between subnets.

My proposed network layer assisted layer 2 pre-authentication mechanism bootstraps layer 2 authentication process in the neighboring networks and completes most of the handoff related operations before the mobile moves to the new network. Thus the layer 2 authentication delay during handoff is reduced to the time needed to complete 4-way handshake operation only, that generates the pair-wise transient key (PTK).

These are the following two key advantages of my proposed techniques:

- ◇ My proposed mechanism takes care of the shortcomings of the existing layer 2 pre-authentication mechanism (e.g., 802.11i) by supporting pre-authentication across subnets and administrative domains while providing equivalent performance as IEEE 802.11i. My proposed mechanism reduces the authentication delay to only 4-way handshake delay resulting in an average of 16 ms authentication delay. This delay is comparable to the performance offered by IEEE 802.11i-based pre-authentication.

- ◇ By supporting pre-authentication across heterogeneous access networks, my proposed mechanism eliminates the dependence on layer 2 access for authentication. By pre-authenticating with the target network using the current interface, the mobile does not need to turn on the secondary interface for authentication and thus saves battery power.

I describe the related work, details of my proposed mechanisms and experimental results in Section 5.4.

5.1.3 Layer 3 configuration

IP address acquisition process and duplicate address detection process are two main components that contribute to a mobile's layer 3 configuration delays in a new network. IP address acquisition process involves signaling exchange between the mobile and a DHCP server and the mobile waits for a random period of time to complete the duplicate address detection before the mobile can assign the new address to its interface.

I describe below two of my proposed mechanisms that reduce the IP address acquisition delays and duplicate address detection delays using *proactive* and *reactive* techniques, respectively.

1. By using the proposed *proactive* discovery mechanisms, the mobile discovers the DHCP server or router in the target network and communicates with it to obtain the IP address from the target network over a secured handover tunnel that is set up

between the mobile and the target router. The mobile then checks the uniqueness of the IP address and caches the IP address of the target network locally until it moves to the new network where the mobile assigns the address to its interface.

2. As part of optimized reactive address configuration, I proposed a router assisted duplicate address detection mechanism where the router multicasts ARP-cache in a periodic interval so that the mobile does not need to initiate address resolution process to determine the address uniqueness.

My proposed proactive layer 3 configuration mechanism reduces the signaling exchange between the client and the server completely by obtaining the IP address over a secured proactive tunnel before the mobile hands over to the target network. This proactive IP address configuration process can work in conjunction with the pre-authentication mechanism to securely obtain the IP address. My proposed proactive mechanism reduces the layer 3 configuration delay to about 100 ms that is equivalent to time taken to assign an IP address statically to a mobile's interface. Compared to existing mechanisms (e.g., FMIPv6) my proposed mechanism is client assisted, works for inter-domain mobility and pre-configures the IP address securely.

In situations, when the proactive layer 3 configuration is not possible, my proposed reactive technique can be used to reduce the delay for duplicate address detection. Compared to other available mechanisms that reduce the duplicate address detection [FSS06], my proposed reactive mechanism does not need any additional element in the network to detect the uniqueness of IP address. The proposed mechanism reduces the duplicate address detection delay from 4 seconds to a few hundred milliseconds that is dependent upon the router advertisement interval.

I describe the related work, details of my proposed mechanisms and experimental results in Section 5.5.

5.1.4 Layer 3 security association

When the IP address of either of the communicating end points changes, a new security context needs to be established that requires generation of new keys. This process results in additional signaling exchanges giving rise to additional handoff delays and media interruption during layer 3 security association.

I proposed two different mechanisms that can reduce the delays due to layer 3 security association during handover. These mechanisms can be categorized as *reactive* and *proactive*.

1. My proposed *reactive* mechanism maintains the network layer identifier address of the mobile by introducing an additional anchor agent (e.g., home agent) in the network. This allows the security context to be maintained by avoiding the re-keying process when mobile's IP address changes. Compared to traditional non-optimized versions, my mechanism reduces the handoff delay and packet loss during security association.

2. My proposed *proactive* mechanism uses a pre-registration technique to establish the security association in the target network. Using pre-registration with the outbound SIP proxy server in the target network and home subscriber server, the mobile establishes the security context ahead of time by generating the cipher key (CK) and integrity keys (IK) proactively during the AKA (Authentication and Key Agreement) phase.

Layer 3 security binding between two communicating hosts can either be maintained by way of establishing security context proactively or by hiding the IP address change in a reactive manner with the help of an additional network element in situations such as mobile VPN (Virtual Private Network), when proactive optimization is not feasible. While the mechanism proposed by Miu et al. [MB01] cannot operate under heterogeneous access network, my proposed reactive mechanism can reduce the packet loss to zero and still

works for both homogeneous and heterogeneous handover. Unlike other proactive proposal by Bargh et al. [SHE+04], my proposed proactive technique reduces the security risks by avoiding the domino effect [HA07] that results when the security context is transferred between the end points. My proposed proactive mechanism reduces the layer 3 security association delay to zero but it depends upon the mobile's ability to discover the outbound proxy server in the neighboring network.

I describe the related work, details of my proposed mechanisms and experimental results in Section 5.6.

5.1.5 Binding update

Longer distance between the mobile node and correspondent node or home agent delays the binding update resulting in overall handoff delay and packet loss since the media keeps getting forwarded to the previous network until the biding update is complete.

I proposed two mechanisms namely, *hierarchical binding update* and *proactive binding update* that reduce the binding update delay and as a result minimize the packet loss.

1. My proposed hierarchical binding update mechanism is a reactive mechanism that uses two level hierarchy of addresses (e.g., local care of address and global care-of-address) and introduces an anchor agent called mobility agent (MA) in the network to limit the global signaling update during mobile's mobility within a domain. This mechanism reduces the global signaling update by 70 percent for a network with 10 subnets per domain.

2. My proposed proactive binding update mechanism sends the binding update to the home agent and correspondent node before the layer 2 handover over a secured tunnel and uses the IP address that is obtained as part of proactive configuration from the target network as the new care-of-address. Using this mechanism the mobile eliminates the binding update delay completely after its move to the network.

Compared to MIP-RR [Per02b] which was developed around the same time, my proposed *reactive* mechanism works for both network layer (e.g., MIP) and application layer mobility (e.g., SIP) protocols and supports dynamic load balancing and fast-handoff. My proposed *proactive* mechanism reduces the binding update completely as the mobile does not need to send a new binding update after the handover.

I describe the related work, details of my proposed mechanisms and results in Section 5.7.

5.1.6 Media rerouting

During media rerouting process transient data may get lost due to handoff operations at several layers or get delayed due to operations such as encapsulation, de-capsulation, tunneling and buffering in the network.

I summarize below my proposed mechanisms that reduce the data traversal delay by optimizing the media rerouting process.

1. *Reactive forwarding of data from previous network:* This proposed mechanism uses reactive forwarding mechanism to redirect the in-flight data from previous network using application layer mobility proxy in case of longer binding update delay.

2. *Proactive multicasting:* This proposed mechanism proactively multicasts the in-flight data to the neighboring candidate networks and reduces in-flight packet loss.

3. *Mobile controlled proactive buffering:* My proposed mobile controlled buffering mechanism provides a per-mobile packet buffer at the edge router that controls the buffering period dynamically based on handoff duration during proactive handoff. My proposed techniques can be categorized into two categories, namely, *time-limited buffering* and *explicit buffering*.

My proposed *reactive* mechanism is the first of its kind forwarding technique that uses application layer mobility proxy to forward in-flight data from the previous network. My

proposed *proactive* multicasting mechanism avoids any additional network element in the network in the neighboring networks and proactively multicasts data to the neighboring networks where the mobile is impending to move. Unlike the existing scheme by Tan et al. [TLP99], this mechanism uses a single multicast address for all the mobiles that are under one MA's domain where the MA encapsulates the unicast data destined to a mobile with the appropriate multicast address.

Compared to the existing buffering techniques by Khalil et al. [KAQ+99] and Moore et al. [Moo04] that depend upon extension of mobile IPv4 and mobile IPv6, respectively, my proposed mechanism is an independent protocol and can be used with any type of mobility protocol (e.g., SIP and Mobile IP). For example, I have experimented with my proposed buffering control protocol for both MIPv6 and SIP-based mobility protocols without making any changes to these protocols. Both the approaches when applied to proactive handover mechanisms reduce the packet loss to zero.

I describe the details of reactive forwarding techniques, proactive multicasting techniques and proactive buffering techniques, related work and the experimental results in Section 5.8 and 5.9, respectively.

5.1.7 Route optimization

End-to-end transport delay of media traffic affects the performance of real-time communication. Media transport delay is also affected when signaling is delayed due to long route between the mobile node, home agent and correspondent node. Several mobility related operations such as triangular routing, encapsulation and decapsulation processes add further delays to the media transport. Thus, it is essential to optimize the route between the correspondent node and the mobile node so that both media transport delay and signaling transport delay are minimized after handover. Although MIPv6 [Car00] does support route optimization techniques, Mobile IPv4 and its variants such as MIP-LR [JRY+99] and Proxy MIPv6 [GLD+08] do suffer from route optimization problem.

I have designed the following few route optimization techniques that minimize the signaling route and media route between the mobile node and correspondent node.

1. I designed an interceptor-assisted packet modifier that is used at the end-hosts and help maintain direct media path between the mobile node and correspondent node by modifying the source and destination address as needed. This technique can be applied to both types of mobility protocols, namely, MIP-LR [JRY+99] and MIPv4 [Per02c]. SIP-based mobility management [SW00] is an existing approach that provides route optimization using direct media path, but it can support real-time traffic only.

2. I designed a route optimization technique that uses a packet interceptor and a forwarding module at the mobile's outbound SIP proxy. This proposed technique takes care of routing indirection of SIP signaling caused by underlying network layer mobility protocol mobile IP in an IMS (IP Multimedia Subsystem environment).

3. I designed a binding-cache-based technique that uses proxy binding update to establish the mapping at the local anchor agent (Media Access Gateway) and routes packets locally instead of routing it via LMA.

The interceptor-assisted packet modifier is an application layer technique and reduces the end-to-end delay of the media traffic by 60 percent for large size packets (e.g., 1024 bytes). Its benefit becomes more effective when the mobile is further away from the home network. In an IMS environment, my proposed intercepting proxy assisted mechanism reduces SIP registration delays by 20 percent and SIP INVITE delays by 30 percent. By using binding-cache-based route optimization technique, end-to-end media delay is minimized and does not change even if the distance between the mobile and the local mobility agent (LMA) increases.

I describe the details of these route optimization mechanisms, related work and experimental results in Section 5.10.

5.1.8 Media independent cross layer triggers

Post handoff detection mechanisms at layer 2 (e.g., access point) and layer 3 (e.g., router) points of attachment work independent of each other causing additional delays during handover. Handoff related functions are spread across different layers of the protocol stack and are executed independently. There is also no existing mechanism that allows to exchange control information across layers. However, for efficient network communication it is essential for a protocol layer to utilize cross layer information. Thus, it is useful to have a set of abstract primitives that can be used to pass on the information across layers in order to expedite the handoff operations.

I proposed a set of abstract primitives that can pass information across layers and work independent of the access mechanisms (e.g., CDMA, 802.11). Some of these abstract primitives were used to develop media independent handover functions that have recently been standardized in IEEE 802.21 standards. Unlike other proposals, these primitives can be applied to support handover among heterogeneous access networks such as 802.11 and CDMA. These triggers can be categorized as *Information Service*, *Command service* and *Event service*. Using these primitives, the mobile can quickly detect the new networks and loss of old networks. Section 5.11 discusses the detailed description of these triggers and different implementation steps. As part of my proposed mechanisms, I have designed the following cross layer triggers that expedite the network detection process and triggers the upper layer handoff operations such as IP address acquisition, binding update.

1. Proactive Triggers: In order to prepare for an impending handoff and perform some of the handoff operations ahead of time I have developed cross layer triggers such as 'MIH_Link_Going_Down', 'MIH_Link_Handover_Imminent', 'MIH_Link_Parameters _Report' that will trigger many of the handover related upper layer operations such as application layer discovery and network layer assisted layer 2 pre-authentication.

2. Reactive Triggers: I developed link layer cross layer triggers such as 'MIH_Link_

Up', 'MIH_Link_Down', 'Link _Detected', 'MIH_Link_Get_Parameters' events to expedite upper layer handoff operations in an access independent manner such as handoff between 802.11 and CDMA networks. Unlike other event triggers, my proposed cross layer triggers work across different access mechanisms.

3. Cross Layer Triggers: Using these mechanisms layer 3 related information (e.g., subnet prefix, default router address) are passed during access point discovery. This is accomplished by modifying layer 2 access point beacon and stuffing layer 3 information.

I used these triggers to build a media independent proactive handoff system as described in Chapter 9. I describe the related work, details of my proposed mechanisms and experimental results in Section 5.11.

In the rest of the chapter, I describe the details of my proposed mechanisms for each of the handover components that I summarized, related work and results from experimental prototype that I built using these optimization techniques.

5.2 Introduction

In order to experiment with the key optimization techniques, I have implemented an Internet mobile multimedia testbed and the associated functional components where I demonstrated several of the mobility functions. In particular, I have implemented a configuration agent, signaling agent, mobility agent, home agent, authentication agent, and authorization agent using the IETF-based protocols, namely DHCP [Dro97], SIP [RSC+02], MIP, PANA [JLO08], and Diameter [CLG+03] over heterogeneous access networks including IEEE 802.11 and CDMA. I describe the details of the implementation of the multimedia testbed in [DAD+04]. In the rest of the chapter, I describe the optimization techniques associated with some of the primitive operations of the handoff event as described in Chapter 3, namely discovery, authentication, security association, configuration, media delivery, and

buffering. In addition, I explain how cross layer triggers can help expedite the handoff related operations and reduce the delay.

In the following sections, for each of the handoff components, I follow a systematic approach to describe the performance parameter (e.g., handoff delay, packet loss) that is optimized, highlight the fundamental principles and techniques that are used to optimize these parameters, demonstrate the experimental system that validates these techniques and compare the results by applying these core techniques with non-optimized version.

5.3 Discovery

As discussed in Chapter 3, experimental results show that network discovery and resource discovery processes in IEEE 802.11 networks contribute to a large amount of delay during handoff. During a handoff between heterogeneous access networks involving WiFi and cellular networks, discovering a cellular network such as GSM also takes time [Rah93], [SLGW01] depending upon the channel assignment strategies and types of handover scenarios as described in Chapter 2. In this section, I propose an application layer network discovery mechanism that can discover the network elements and resources in the neighboring networks independent of the underlying access technology. Using this discovery technique, the mobile can pro-actively discover many of the layer 2 and layer 3 network resources, namely channel number, default router's address, and authenticator in the target network. This proactive operation will help to reduce the handoff delay as many of the discovery related operations, namely layer 2 scanning, router solicitation and server discovery do not need to be performed after the handoff.

In this section, I first describe the general principles that are needed to optimize the delay contributed by discovery operations at several layers. Then, I cite the related papers that have attempted to optimize the discovery related delay at the expense of other systems resources such as network bandwidth and CPU cycles. I then introduce the proposed ap-

plication layer discovery technique and describe its advantages over the existing discovery mechanisms. Finally, I illustrate the experimental results in a testbed environment.

5.3.1 Key principles

Following are the key principles that govern the optimization of the discovery process. This optimization process aims to optimize the delay during discovery with respect to other network resources such as processing power at the end hosts and network bandwidth.

1. Limit the number of signaling exchanges between the mobile and centralized server needed to discover the network resources.

2. In case of passive scanning, increase in the rate of beacon advertisement reduces the time to discover the new point of attachment at the expense of additional network bandwidth and processing at the end hosts.

3. Caching of neighboring network resource parameters before the mobile moves to the new network.

4. Use of a media independent application layer discovery protocol to discover network resources to support handover in heterogeneous access networks without depending upon any access specific technology.

5.3.2 Related work

In cellular networks such as GSM and CDMA, the pilot signals of the mobile, namely BCCH (Broadcasting Channel) and Sync channels, respectively report the details of the neighboring networks to the serving MSC (Mobile Switching Center). Serving MSCs use this information to decide the target networks for the mobile. Recently, for IP-based networks, some efforts have been underway to design discovery protocols that provide service

discovery and network discovery at different layers. I highlight some of the related work in the area of network discovery and their optimization techniques.

There are a few task groups within the IEEE 802.11 standard groups that propose network discovery mechanisms at layer 2 and application layer. The IEEE 802.11u [Gas05] working group proposes methods of network selection along with other external networks such as cellular networks. The IEEE 802.11k [Sta04] working group proposes methods that enable the APs to query mobile devices for location and neighbor information. It proposes a few new request/response measurement mechanisms, namely measurement pilot, neighbor report, link measurement, station statistics, and location configuration information so that the mobile can obtain information about its neighbors and make appropriate decisions for fast transition. However, IEEE 802.11k-based discovery mechanism is limited to 802.11 access networks only and works on layer 2 within the same ESS (Extended Service Set).

Several service discovery protocols and architectures exist today including SLP (Service Location Protocol)[GPVD99], JINI [Wal99], UPnP [Plu], Salutation [MP00], and LDAP (Lightweight Directory Access Protocol)[JC98]. However, these focus mostly on how a user retrieves service-related information assuming that the information is already available in the databases. The service-related information and hence the servers that host the information can be organized into a hierarchy, for example, in a way similar to the Internet Domain Name System (DNS). The service-related information can either be preconfigured or provisioned dynamically on the servers. The information can then be updated either by human administrators or automatically by the servers themselves exchanging updates with each other. However, none of these protocols provide support for discovering information about the neighboring networks at a higher layer, dynamic construction of the discovery databases and determining what information to collect and provide to mobiles. Instead, the existing service discovery mechanisms focus on how to retrieve information already existing in the databases. These mechanisms rely on all local network providers to implement service information servers, that are usually not deployed in the public networks.

Recently, the IEEE 802.21 working group has finalized to use Information Service mechanism that provides information discovery at application layer. Some of the techniques such as application layer discovery mechanisms using RDF (Resource Discovery Framework) [LS+99a] developed as part of this thesis have contributed to the development of Information Server (IS) components of IEEE 802.21. I describe the details of these mechanisms in Section 5.3.3.

A representative example of discovery protocol at layer 3 is the Candidate Access Router Discovery (CARD) protocol [LSCF05] that provides network discovery mechanism at layer 3. A candidate access router is an access router in a neighboring network to which the mobile device may move into. CARD is designed to be used by a mobile device to discover a candidate access router, before the mobile performs IP-layer handoff into the neighboring network. With CARD, a mobile listens to layer-2 identifiers such as IEEE 802.11 BSSIDs broadcast from the radio Access Points (APs) in neighboring networks prior to making a decision about IP-layer handoff. The mobile then sends these layer-2 identifiers to the access router in its current network, which will in turn map the layer-2 identifiers to the IP addresses of the candidate access routers in the neighboring network and then send the candidate router addresses back to the mobile. In order to use CARD to support network neighborhood discovery, routers in the network need upgrade. This also needs security and trust between the neighboring routers and thus may not work if it involves handoff between two administrative domains.

There are a few related papers that attempt to reduce the network discovery time in IEEE 802.11 environment. Shin et al. [SSFR04] adopt a selective scanning and caching strategy to reduce the 802.11 handover latency. However, this method is more applicable to an environment where the mobile has associated with the neighboring APs in the past, and cannot be applied if the target access point is a new AP. Montavont et al. [MMN05] propose a periodic scanning method, where the mobile does scan different channel periodically and builds up a list of neighborhood APs. However, this mechanism generates more traffic, and

as a result consumes more energy. Velayos et al. [VK04] provide techniques to reduce the layer 2 discovery process by reducing the beacon interval time and performing the search phase in parallel with data transmission. Brik et al. [BMB05] propose to use a second interface to scan while communicating with the first interface, thereby avoiding scanning delay during communication. Most recently, Forte and Schulzrinne [FS07] have developed discovery mechanisms using cooperative roaming techniques that are suitable to work in an infrastructure less environment.

5.3.3 Application layer discovery

As part of my work on minimizing handoff delay due to discovery component of the handoff process, I have developed an access independent information server-based application layer discovery mechanism that helps to discover the network parameters and resources of the target network [DMZ+06]. Unlike the existing network discovery mechanisms, the application layer discovery mechanism does not depend on any access specific discovery technique such as IEEE 802.11u.

This discovery technique can be applied to both infrastructure-assisted and end-system assisted scenarios. I have analyzed how this discovery mechanism can be effective in a collaborative environment using an *end-system assisted approach* [ZMD+05] where each end system can act as the source of information about the neighborhood information. As part of *infrastructure-assisted scheme*, the information server stores the details of the networks and the associated resources in a generic format in an access independent manner that can be queried by the mobile client at any time. The client communicates with the information server and discovers the neighboring network elements, such as access router, authentication agent (IEEE 802.11i authenticator), configuration agent (e.g., DHCP server) and authorization agent (e.g., AAA server), and communicates with these entities prior to its handover to these networks. By discovering the details of the target access points prior to handoff, the mobile keeps the MAC address and channel number of the access point in

its cache and avoids some parts of the scanning procedure, such as channel probing during 802.11-based handover. I have described how prior discovery of routers and authentication servers helps to complete other handoff related functions, such as authentication and configuration prior to handoff [DDF+06]. The proposed information server-assisted discovery technique has been adopted as one of the discovery mechanisms for the Media Independent Information Server (MIIS) function of IEEE 802.21. Evaluation of a complete system using network assisted discovery scheme is described in Chapter 9. I describe the details of the architecture and proposed schema below.

Currently, no database querying mechanism allows one to obtain detailed information of a neighboring network given a certain property such as network type, GPS coordinate of the mobile. Such detailed information can be the MAC addresses of the neighboring APs (Access Point), channel number associated with the AP, IP address of DHCP server, router, and AAA server. Currently, DHCP provides DHCP option mechanism [Dro99] whereby a client can discover a specific server and the geo-coordinates of the nearby access points [PSL04]. However, the DHCP server usually stores the information specific to a subnet and cannot provide services to the mobile that are not located in the same subnet without the help of a relay agent, namely DHCP relay agent. Thus, the DHCP-based discovery mechanism is limited to a specific subnet and cannot span over multiple networks. The query mechanism should also be extensible and should accommodate proprietary vendor definitions. Thus, it is desirable to design a query mechanism which can support a schema-based (or sub-schema) access and can cover the networks beyond a specific subnet.

My proposed approach is based on a new architecture called Application-layer Information Service (AIS) that supports network discovery including methods to solve the discovery database construction problem and methods for mobiles to discover information regarding neighboring networks. AIS is designed to be extensible enough to support current and future types of network information that may be needed by the mobile nodes. AIS leverages existing protocols as much as possible. Although information about the network

elements can have multiple usages, I focus on how the mobile can use this discovery information to support secured and proactive handoff. Some of the key design factors that need to be looked into while designing the discovery architecture include constructing the information, retrieving the information, and format of the information stored in the information server.

I describe below a sample implementation of the query and response processes that are part of the network discovery mechanism. To query information related to a specific network interface (e.g., 802.11, CDMA), a mobile needs to first know which information attributes are supported by a network interface. Thus, a query-response mechanism may use two steps: the first query provides the meta-data information (i.e., the attributes names) and the second query provides values of the attributes the mobile is interested in or require.

The information on the Information Server should be stored in a standard and easy to access manner. I have used RDF (Resource Description Framework) [LS+99a] based schema to describe and store the information regarding networking elements and their characteristics on the Information Server. RDF is a framework that describes a language for representing information about WWW resources. It is intended for representing meta-data, such as title, author, and modification date of a WWW page, and copyright about WWW resources.

RDF provides a common framework for expressing the information so that it can be exchanged between applications without loss of meaning. It is intended for describing information that needs to be processed by applications, rather than being only displayed to people. Therefore, RDF-based query and response mechanisms provide a suitable way for the mobiles to report to and retrieve information from the application information server. It allows a mobile to query specific information elements about a network by providing the characteristics of the information elements in a granular manner.

The characteristics of these network information elements can be SSID (Service Set Identifier), location-info (geo-coordinate), or layer-2 (L2) security information. The RDF schema defines the structure of the information elements as well as the relationship be-

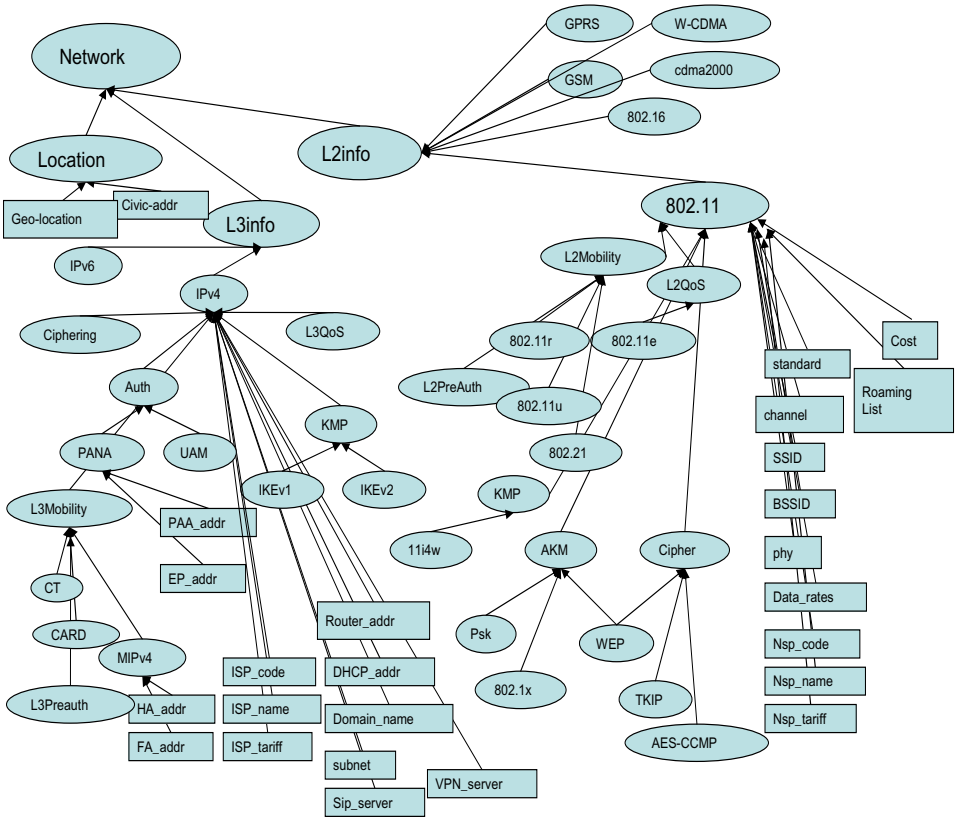


Figure 5.1: Inter dependency chart for network elements

tween the information elements. The RDF schema is usually partitioned into two schema types; *basic schema* and *extended schema*. The basic schema is static and includes media independent classes and properties. The extended schema includes the properties that are dynamic in nature, such as bandwidth.

Figure 5.1 shows a very simple view of the RDF-based tree illustrating how these network entities are constructed in a hierarchical manner. It shows the network elements in the neighborhood networks and the inter-dependency and shows how location, L3 info, L2 info, network types are constructed in a hierarchical manner.

I explain the schema for information service in Appendix A. Here I briefly present the architecture and describe the functional components used in information query and update process. At the information server end, I used Joseki [KD73] to interpret the RDQL

[Sea04] and send appropriate responses to the client. I have used Jena [McB02] for forming RDQL. Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS [McB04] and OWL (Web Ontology Language) [MVH+04], including a rule-based inference engine. The implementation in Jena is coupled to relational database storage so that an optimized query is performed over the data held in a Jena relational persistence store. I have used Joseki server for publishing RDF models on the web. These models are represented by URLs and can be accessed by query using HTTP GET.

5.3.4 Experimental results and analysis

Figure 5.2 shows a possible deployment architecture where this information discovery scheme can be useful. Initially, the mobile is in network 1, and is connected to access point AP1. Network 2, Network 3 and Network 4 are the neighboring networks. The Information Server stores the information about these networks and the associated network elements, namely authentication server, configuration server, access point identifier.

I have implemented both the database population mechanism by the end clients and the network discovery process during the handover. Although there are several ways an information server can be populated as I have explained in [DMZ+06], I implemented the end system assisted population scheme in the current experiment. As the mobile moves from one network to another network, it populates the information server with the several network parameters (e.g., router, access point, channel numbers) of the network it had just visited. Thus, the next mobile can query the required information from the information server. Figure 5.3 shows how a mobile node populates the information in the database and subsequently, how it communicates with the information servers to discover the networks and resources.

However, I only describe the network discovery part here. When a mobile decides to handover to one of the neighboring networks, it makes an RDF-based query to the informa-

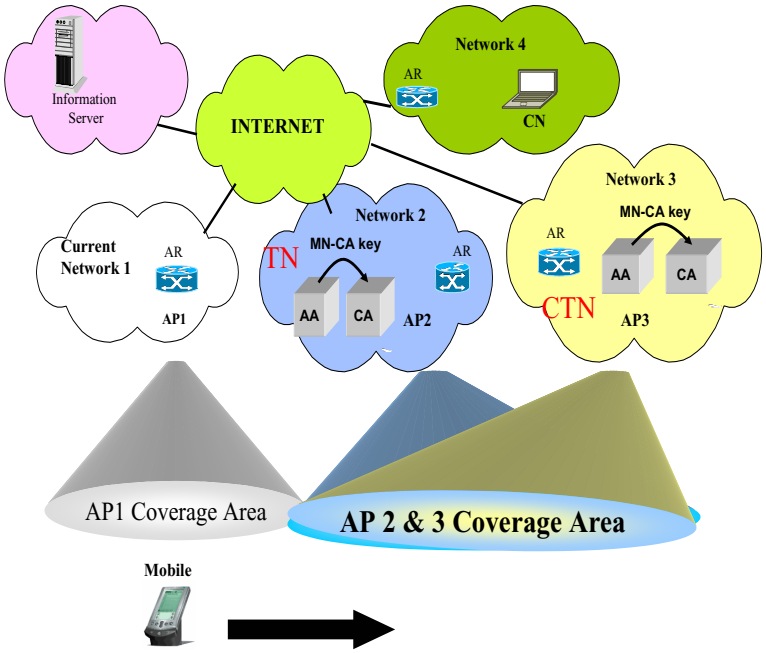


Figure 5.2: Deployment of application layer information discovery

tion server that has been populated with the network information beforehand and gets the meta information about the neighboring networks such as types of networks (e.g., 802.11, CDMA). Once it has the information about the available types of networks, based on certain policy such as the cost of the network; it queries the information server again to get other detailed information about the network elements for a specific type of networks. Figure 5.4 shows the functional components on the client and server that are involved during the query processing.

Table 5.1 shows the results that include the time for both the initial query for the meta data and subsequent queries to obtain the values of specific network elements. It also reports the amount of time needed to process the query at different parts of the network. These values show average of five runs in the experimental testbed consisting of two neighboring networks, information server, mobile and two access routers. I have described the details of this testbed in [DMZ+06].

In Table 5.1 API (Application Programming Interface) delay represents the delay in-

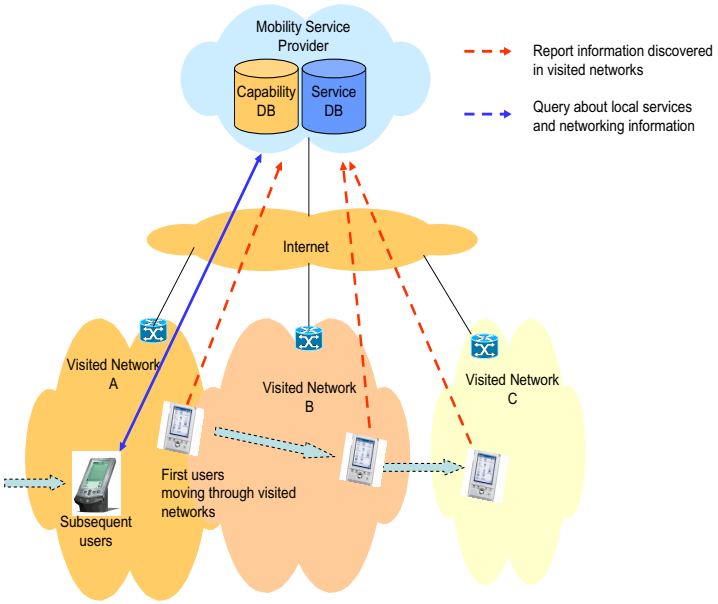


Figure 5.3: Information population and query process

curred during interaction with the query application at the mobile and server. API delay includes interaction with the database and is dependent upon the implementation language such as JAVA or C. Network layer delay includes delay due to TCP layer transaction. Processing delay at the client and server includes the time spent for HTTP processing at each of these nodes. In this experiment, during the transaction of query 1, the mobile sends 1288 bytes of data and gets 1684 bytes as response. Query 2 involves 1713 bytes of data sent by the client and 1335 bytes of data sent by the server. Query and Responses are carried back and forth in chunks to accommodate the maximum segment size (MSS) thus adding to the delay. Since these query and responses are carried as part of HTTP messages, TCP is the chosen transport method. Transport delays could be significantly reduced if UDP is used as a transport protocol instead. During the database update procedure by the client, I observed that the average update delay by the client to be 353 ms with a standard deviation of 153 ms. This information update time will of course depend upon the amount of data

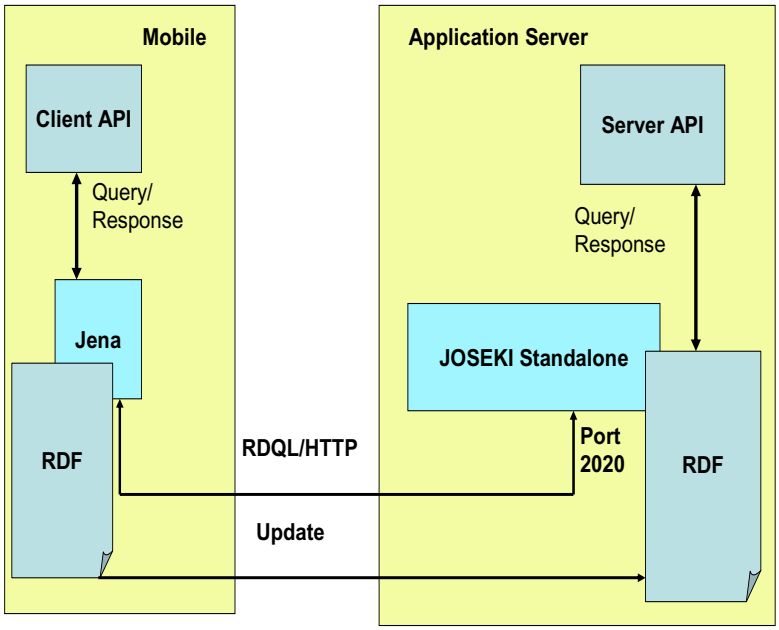


Figure 5.4: Interaction among the functional components

being updated and the network bandwidth. However, query update time is not critical for handover decision.

Optimizing the query delay is important for the mobiles with high mobility rate as the mobile needs to make a decision ahead of time based on the query delay. Processing power at the end clients and transport methods will help optimize the delay associated with the query and response. I observed that the delay due to second query to be less than the initial query because of the additional ARP (Address Resolution Protocol) performed during query 1. Response to first query is the meta data, where the mobile finds out the relevant networks that are of type 802.11, and using tariff as the policy it chooses a specific network type and decides to get more information about other network elements such as access router, PANA server, and DHCP server. A more detailed breakdown of stack level delays for each of the IS primitives is shown in Chapter 9.

The IEEE 802.21 working group has included both XML (Extensible Markup Language) and TLV (Time-length-value) format as part of media independent information ser-

Table 5.1: Information service query processing

Query Type	Response	Processing delay (ms)	
Current PoA: AP **Query 1:** Provide list of 802.11 type neighboring networks and their with associated tariff values	Neighbor 0 PoA: ID:00:20:A6:53:B2:5E, Network Tariff:20 Neighbor 1 PoA : ID:01:23:45:67:8:AB, Network Tariff:50	Total	2292
		API	1291
		Network	919
		Server processing	18
		Client processing	64
Neighbor 0 selected **Query 2:** Provide list of network elements for Neighbor 0	Target Network Channel: 10 SSID: ITSUMO newpoa1 Router address: 10.10.10.52 Router MAC: 00:00:39:e6:8b:ee Subnet: 255.255.255.0 DHCP Server: 10.10.10.52	Total	1473
		API	991
		Network	451
		Server processing	13
		Client processing	18

vice. My proposed mechanism contributed to the XML format that was included. RDQL uses XML format for query and response. I did a preliminary performance comparison between XML and TLV format. The sizes of query and response obtained via RDQL are much larger than the size of TLV query and response. However, if basic schema is changed to a flatter structure, then the size of query and response will be reduced. On the other hand, the XML-based query provides more extensibility and flexibility in terms of its ability to query a specific network element. Since the number of bytes going over the air is a concern, I have also used compressed version of XML for the query response. By using the compressed version of XML during information query, it reduces the overall discovery time. Another approach to reduce the query time is to use a combination of XML and TLV, where a mobile makes an XML query but obtains the information in TLV format.

5.4 Authentication

In Chapter 3, I have defined the authentication and authorization processes that are needed during a mobility. I have also illustrated how the during a mobility event, authentication and authorization processes add to the disruption in communication and packet loss. Figure 5.5 shows a basic Internet roaming scenario where two different administrative domains that are managed by different wireless service providers establish business agreements between them in order to provide roaming services for their customers. In particular, these business relationships allow users who belong to one domain (Home Domain) to access the network and services in other domains (e.g., Roaming Domain A or Roaming Domain B in Figure 5.3). A domain here is defined as an administrative domain. There may be several subnets within an administrative domain. These agreements are enforced by means of the deployed AAA infrastructure (e.g., AAAv (AAA visited), AAAh (AAA home)) in each domain. In nutshell, the home AAA domain (where the user belongs to) is equipped with a home AAA (AAAh) and each roaming AAA domain is equipped with an AAA proxy (AAAv) that contacts the home AAA infrastructure in order to verify the roaming user's credentials. Figure 5.5 also highlights three different types of movement: intra-subnet, inter-subnet and inter-AAA-domain (or inter-domain hereafter). Link-layer handoff is the common scenario in this roaming architecture. Thus, certain optimization on the establishment of security during link-layer handoff deserves some attention.

In general, authentication and authorization take place in the target network after the mobile moves to the new network. For example, in IEEE 802.11-based networks, the authentication mechanism requires an IEEE 802.1X message exchange with the authenticator in the target network, such as an access point that can initiate an EAP (Extensible Authentication Protocol) [ABV+04] exchange with the authentication server. Following a successful authentication, a four-way handshake with the wireless access point derives a new set of the session keys to encrypt the data. The handover latency introduced by this authentication mechanism has proved to be larger than what is acceptable for some handover

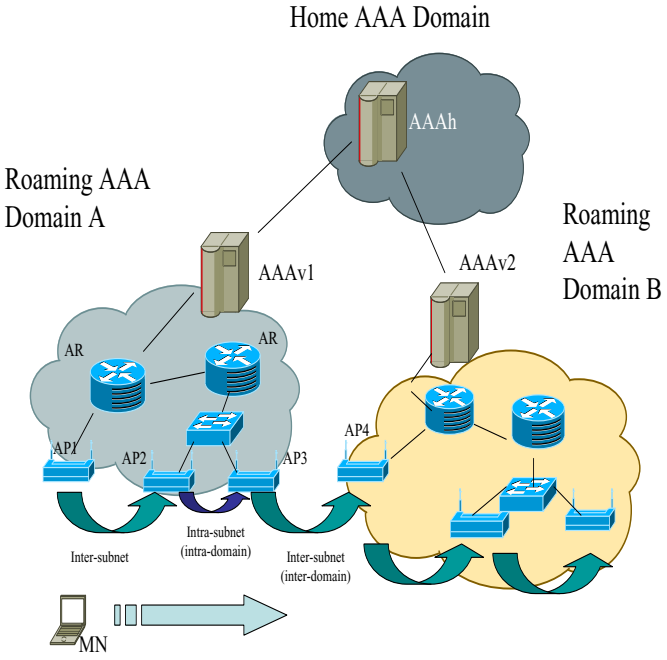


Figure 5.5: Illustration of roaming environment

scenarios involving inter-domain handover. Hence, improving the handover latency due to authentication procedures is a necessary objective for such scenarios. Various standard organizations such as IEEE 802.11i and 802.11r working groups, 3GPP and WiMAX forums are developing access specific techniques to reduce the authentication delay. However, these mechanisms are designed to work at link-layer level that entails some implications and limitations for inter-technology and inter-subnet handoff, such as the inability to pre-authenticate.

In this section, I first describe the key principles that should be considered while optimizing authentication delay against other network resources, namely bandwidth, processing power and battery power. I then describe some of the related work that have tried to reduce the authentication delay. Then, I describe my proposed authentication mechanism and highlight its key differences compared to the existing techniques. Finally, I describe the experimental testbed and analyze the measurements that validate my proposed optimization mechanism.

5.4.1 Key principles

Following are the key principles that need to be considered to optimize delay and processing power during an authentication operation.

1. Minimize the time needed to authenticate and authorize a mobile after each handoff during the re-authentication procedure.

2. Reduce the number of signaling message exchange between the mobile node and authenticator needed to generate a shared secret key.

3. Use of an appropriate key generation algorithm that will reduce the processing load on the end hosts.

4. Placement of authenticator and authentication server closer to the mobile.

5. Reduction of installation time of the Pre-shared keys (e.g., PSK) on the authenticator in case of proactive authentication.

6. Proactive caching of security context on the neighboring access points prior to handoff either by proactive authentication or context transfer.

5.4.2 Related work

IEEE 802.11i and IEEE 802.11r [O'H04] provide link-layer handoff optimization mechanisms that attempt to reduce the delay due to link-layer authentication during a node's mobility. IEEE 802.11i was conceived to provide stronger security to IEEE 802.11 WLAN. It relies on IEEE 802.1X for the authentication and access control of IEEE 802.11 stations (STA)[1]. As part of 802.1X, a successful authentication allows both the STA (mobile node) and AP (Access Point) to generate a pairwise master key (PMK). Typically, the AP relies on a backend authentication server (AS) such as an AAA server acting as a termination

[1] STA and mobile node are used interchangeably

point of an EAP (Extensible Authentication Protocol) authentication method, in order to verify authentication credentials of the peer and deliver the PMK to the AP, after the verification is successful. In case of pre-shared Key (PSK) mode, STA and AP pre-share a 256 bit key that is used as PMK. Therefore, no EAP authentication is needed. Moreover, a 4-way handshake protocol uses the PMK for mutually authenticating STA and AP and establishing fresh pairwise transient keys (PTKs) to protect link-layer frames. However, IEEE 802.1X authentication can last from several hundred milliseconds to several seconds [SHE+04]. Hence, each time a STA moves from one AP to another, this delay and associated packet loss during the handoff affect the real-time application such as VoIP. In order to overcome this problem, IEEE 802.11i introduces a mechanism of pre-authentication, where the STA starts a new EAP authentication with the target AP where it is likely to hand off, through its currently associated AP. After the EAP authentication has completed successfully, the generated PMK is properly stored at the target AP. When STA finally roams to the target AP, both parties engage the 4-way handshake by using the specific PMK. Therefore, EAP authentication is not performed after the handoff. By decoupling the authentication and network access control operations from the handoff, IEEE 802.11i pre-authentication reduces the handoff delay. However, 802.11i has also some drawbacks and limitations that are worth mentioning:

1. Each IEEE 802.11i pre-authentication involves a full EAP authentication. Consequently, it implies a lot of signaling with the authentication server (AS) during each movement.

2. The mechanism does not work when the involved APs belong to different distribution systems (DS), where a distribution system is used to interconnect a set of basic service sets and integrated local area networks (LANs) to create an Extended Service Set (ESS). For example, inter-subnet and inter-domain pre-authentication is not possible.

3. The full association and 4-way handshake are still required to be finished after the movement.

IEEE 802.11r overcomes most of these problems by introducing a three level key hierarchy (started either from a master session key (MSK) generated during an EAP authentication or a PSK) and a supporting architecture that allows the STA to perform fast transition between the APs within the same mobility domains (MD) without the need to run EAP authentication during each movement. Additionally, IEEE 802.11r allows to perform part of the 4-way handshake and some resource reservation at the target AP before STA moves. When STA finally hands off, it only needs to re-associate with the target AP to complete the handoff. Thus, IEEE 802.11r reduces the handoff delay compared to IEEE 802.11i. However, both IEEE 802.11i and IEEE 802.11r mechanisms do not work when the involved APs belong to different distribution systems (DS), which is the case for inter-subnet and inter-domain handoffs. Basically, the reason is that 802.11i and 802.11r handover optimization mechanisms are based on link-layer frames, which cannot operate across different subnets.

The IEEE 802.11f, a trial use recommended practice has defined context transfer and caching mechanism to transfer some of the 802.11i keying related information between the neighboring APs. It uses Inter Access Point Protocol (IAPP) to transfer the keys between the access points. However, IEEE 802.11f has been administratively withdrawn since 2006 because of security concern due to communication between the access points.

The problem of applying link-layer handoff optimization mechanisms between different subnets has also been addressed by the research community. However, most of the solutions are based on context transfer mechanisms [SHE+04, Geo04, DDG04]. The optimization is achieved by transferring the security context (keys and related parameters) created by STA and previous AP to new AP between subnets. Consequently, STA does not need to run full EAP authentication to create a new PMK and only the 4-way handshake is required after the handoff. For example, Bargh et al. [SHE+04] explain how to transfer IEEE 802.11i context between two APs under different networks by using a combination

of Context Transfer Protocol (CxTP) [LNPK05] and Candidate Access Router Discovery (CARD) [LSCF05]. Georgiades [Geo04] extends Cellular IP to signal a context transfer between two base stations (BS) under two different gateways (GW). New GW contacts the previous GW to recover security context from previous BS. Duong et al. [DDG04] also propose an optimized solution based on CxTP and CARD by pro-actively transferring a context when MNs move is imminent. From the security perspective, it is not always a good idea to transfer the cryptographic keys between different network entities. For example, Housley and Aboba have raised a warning on security context transfer in [HA07]. Additionally, to achieve a secure context transfer, one needs to have certain security associations and strong trust relationships between the policy enforcement points such as APs that are not always possible. Finally, it only allows handoff between the same technologies such as 802.11(homogeneous handoff).

Mishra et al. [MSPJ+04] and Pack et al. [PC02] completely avoid the use of context transfer by pre-installing keys into APs before the STA moves to the target network. In general, they are based on algorithms that steer the key installation process based on the movement of mobile node (MN). These solutions assume that an AAA server or trusted third party is in charge of pre-distributing keys to different APs where MN could potentially associate. It implies that AAA server has the knowledge about the location of the APs. This may work when a single wireless service provider is considered. However, in case of roaming scenarios, the home AAA server needs to know the location of the APs in the visited domain. Unfortunately, this is not always possible since, usually, the visited domain shall not want to reveal details about its internal network deployment for privacy purposes, even when roaming agreement has been defined. Additionally, the assumption that an AAA server is able to store the key after EAP authentication is not always true (e.g., RADIUS). Ruckforth et al. [RL04] propose a different approach where a combination of Fast Mobile IPv6 [Koo05] and IEEE 802.11i frames are used to inform user's home domain AAA server about next IPv6 router and next AP where STA may move. With this precise information,

AAA server creates a new PMK and sends it to the AP and AR. However, the solution is restricted to IPv6 networks because of the MIPv6 related messages between the access routers. Forte et al. [FS07] propose a cooperative roaming approach to authenticate the mobile, but its usage is limited to a domain only.

5.4.3 Network layer assisted pre-authentication

In order to take care of limitation of the existing mechanisms, I have proposed network-layer assisted link-layer pre-authentication mechanism [DOF+10], [LDOS07] that can take care of many of the drawbacks of the existing approaches. These mechanisms propose to reduce link-layer handoff latency when existing link-layer handoff optimization mechanisms cannot be applied for cases involving inter-domain and inter-access technologies. It uses pre-authentication at network-layer to assist link-layer handoff optimization techniques by allowing a fast transition even when the APs involved in the handoff do not share same link layer. Although this mechanism can work independent of link-layer access technologies, I focus my study and experiments on 802.11-based access networks. The proposed mechanism also preserves the security criterion raised in the IETF by not allowing context transfer between the APs. In this section, I describe the architecture of this mechanism, provide experimental results from a testbed implementation, and compare these with IEEE 802.11i pre-authentication.

In an inter-domain mobility scenario, an authentication process is followed by an authorization process. In addition to reducing the delay due to layer 3 related authentication and authorization, these proposed mechanisms can reduce authentication delay at link-layer when existing pre-authentication mechanisms (e.g., 802.11i-based pre-authentication) cannot be applied to take care of handoff involving inter-domain, inter-subnet and inter-access technologies. A successful authentication prior to handoff results in proactive configuration and establishment of security association between the mobile and network elements in the target network. I have discussed two types of pre-authentication , namely *direct*

pre-authentication and *indirect pre-authentication* in the pre-authentication problem statement draft [OQG09] that is being discussed in HOKEY (Handover Keying) working group within the IETF. In case of direct pre-authentication, the serving authenticator forwards the EAP pre-authentication traffic as it would do for any other data traffic or there may be no serving authenticator at all in the serving access network. In indirect pre-authentication, it is assumed that a trust relationship exists between the serving network (or serving AAA domain) and candidate network (or candidate AAA domain). Indirect pre-authentication is needed if the peer cannot discover the candidate authenticator's IP address or if IP communication is not available due to security or network topology reasons.

Figure 5.6 illustrates the protocol interaction among the network components when IEEE 802.11i-based pre-authentication is used and Figure 5.7 shows the protocol interaction among the network components for network-layer assisted pre-authentication.

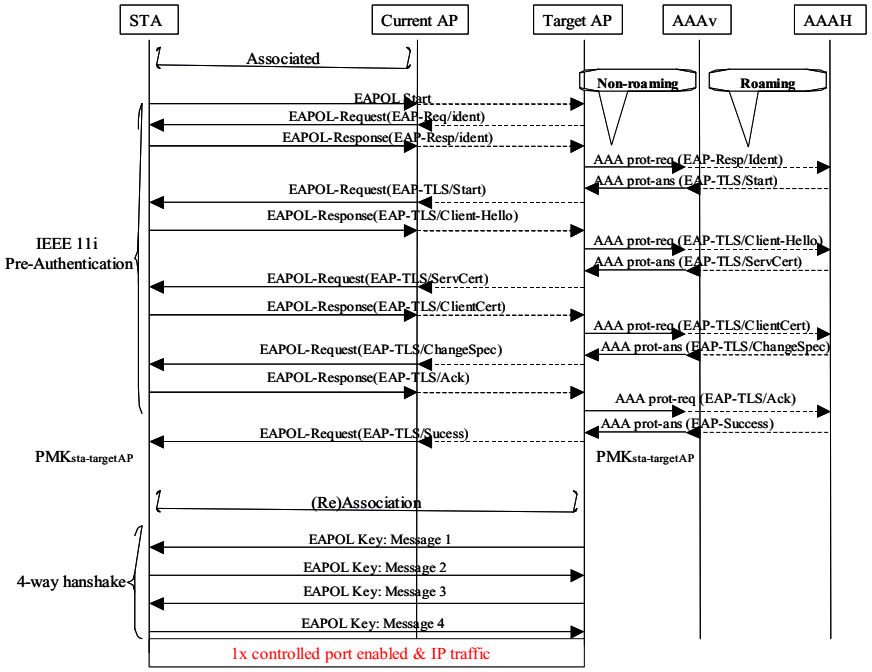


Figure 5.6: Protocol flow for IEEE 802.11i-based pre-authentication

Both roaming and non-roaming cases are illustrated in Figure 5.5. Initially, during the discovery phase, MN discovers through some means (e.g., 802.21 information service) the target AP and PAA's (PANA-based Authentication Agent) IP address that man-

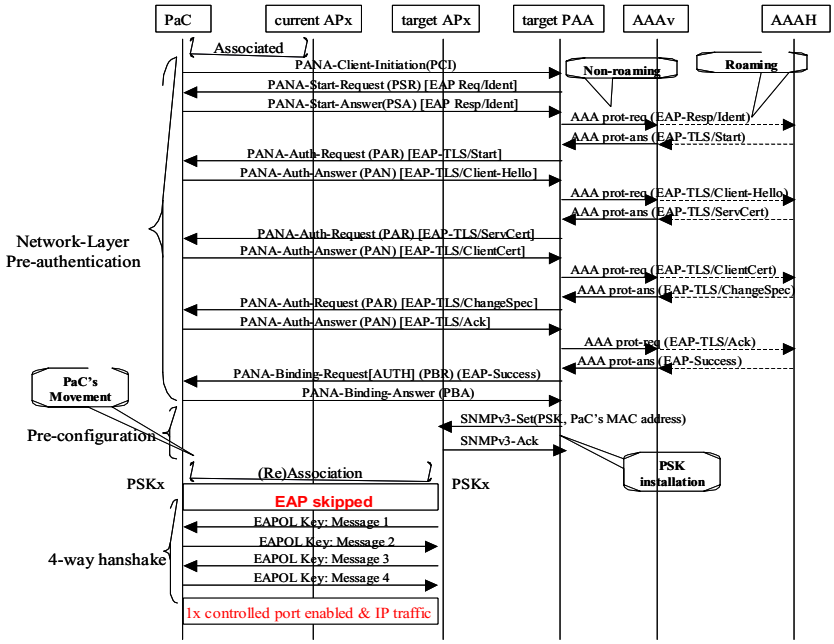


Figure 5.7: Protocol flow for network layer assisted layer 2 pre-authentication

ages the target AP. Then the MN pre-establishes a PANA Security Association (SA) (pre-authentication phase) with the candidate target network (CTN), via its serving network, by performing an EAP exchange between MN and PAA. In the example shown, EAP-TLS [AS99] is used as EAP method for the authentication. The PAA can rely on a backend AAA server to carry out an EAP authentication method. From MSK generated during the EAP authentication method, PAA can derive a distinct PSK (pre shared key) per AP. PAA installs these keys in those APs (pre-configuration phase), and provides the MN with the required information (e.g., APs' MAC addresses) to generate the same PSKs. Then the MN moves to the new AP, and after association, runs a 4-way handshake by using the specific PSK_{ap} generated during PANA pre-authentication. At this point, the handoff is complete. Thus, by pre-authenticating and pre-configuring the link, the security association establishment during handoff reduces only to 4-way handshake.

In comparing IEEE 802.11i pre-authentication presented in Figure 5.6 with the PANA-based network-layer pre-authentication shown in Figure 5.7, one may notice that both the schemes reduce the delay invoked by the authentication process during the handover be-

tween access points. In particular, the delay is reduced to the time for the 4-way handshake required to establish a security association between the PaC (PANA client resident on the mobile node) and the Target AP in both cases. Therefore, in terms of handoff delay, both schemes result in comparable values. However, the proposed mechanism obtains the same reduction even when the APs belong to different subnets, that may be part of different administrative domains. Thus, it takes care of the limitation imposed by the regular IEEE 802.11i pre-authentication mechanism. Another interesting advantage in the current proposal is that a PAA (Pana Authentication Agent) can control and distribute PSKs to several APs through a single EAP authentication, the one performed during the pre-authentication shown in Figure 5.7. This means that, although two messages are required for key installation, when the mobile running PaC (PANA client) moves between the APs covered by the same PAAs area, it avoids additional EAP authentication. As depicted in Figure 5.6, EAP authentication typically involves several round trips to the backend AAA infrastructure. Thus, the proposed scheme avoids a full EAP authentication in contrast with 802.11i pre-authentication where a full EAP authentication is performed during each handoff.

Figure 5.8 compares key derivation methods among three of these mechanisms, namely, 802.11i-based re-authentication, 802.11i-based pre-authentication and network layer assisted layer 2 pre-authentication. However, I describe below only the key derivation and key installation procedures that are part of this proposed network layer assisted pre-authentication mechanism.

5.4.3.1 Pre Shared Key (PSK) derivation

During PANA-based pre-authentication, a master session key (MSK) is generated after EAP authentication. The MSK is used to derive a PaC-EP-Master-Key, specific for both the AP and mobile node. In turn, the PaC-EP-Master-Key is used to derive the PSK. Since the PSK is dynamically derived from PaC-EP-Master-Key, it has an associated lifetime. In PANA, the PaC-EP-Master-Key lifetime (and thus the PSK lifetime) is bounded by the

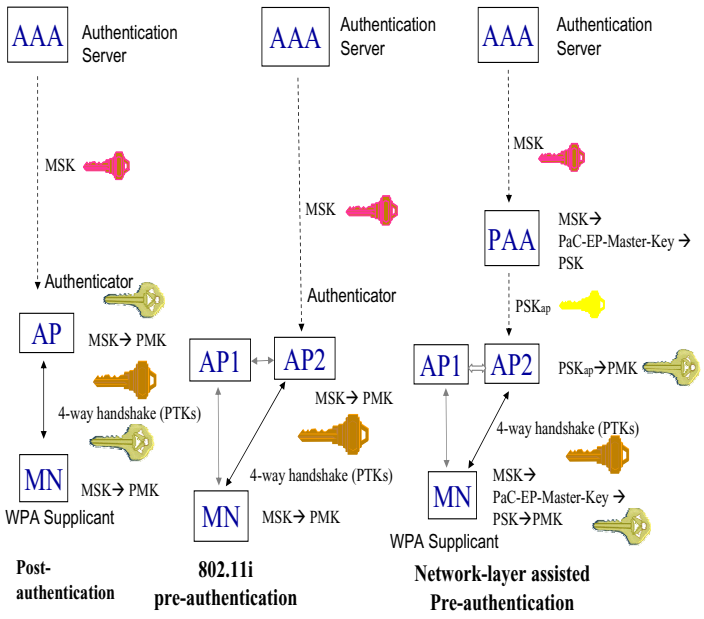


Figure 5.8: Key generation mechanisms in various authentication schemes

PANA security association lifetime which, in turn, is bounded by the MSK lifetime. Since each EAP re-authentication generates a new MSK, new PaC-EP-Master-Key and PSK are derived. For security reasons, when a new PSK is installed in the AP, the 4-way handshake must be run subsequently. It allows to generate new fresh PTKs from the new PSK. It is worth mentioning that, in general, PaC-EP-Master-Key can be used for bootstrapping link-layer security at policy enforcement points (PEP) of any link-layer types (e.g., either 802.11 or CDMA), which allows MN to roam among multiple PEPs of different link-layer types without additional EAP execution if the PEPs are controlled by the same PAA.

5.4.3.2 Key installation process

The PAA (PANA Authentication Agent) installs the PSK on the target access points. I consider two key installation methods, namely, *pre-emptive* and *on-demand*. As part of pre-emptive installation process, the PAA installs PSKs in a pre-emptive way in all target APs. However, this introduces scalability and resource consumption problems when many APs are the under the control of one PAA or many MNs are connected to APs served by one

PAA. Since it provides the needed PSK for a particular MN and AP before MN is attached, it reduces the time to start 4-way handshake.

Alternatively, an AP may inform the PAA when the MN is associated with it. This mechanism is on-demand key installation for the AP. Although this mechanism can save systems resources, it introduces a delay to gain network access because both MN and AP need to wait for the PSK provisioning.

In order to take advantage of both the methods and minimize some of their disadvantages, algorithms such as those proposed by Mishra et al. [MSPJ+04] and Pack et al. [PC02] could be used. These algorithms determine the most probable APs where MN may move to, so that PAA can install PSKs only at those APs selected by the algorithm, as part of preemptive key installation. However, if the prediction fails and MN finally moves to another AP where PSK has not been installed, on-demand key installation may be used instead. Depending on the number of APs and the number of users, a wireless service provider may decide to use one or another technique or even a combination of both.

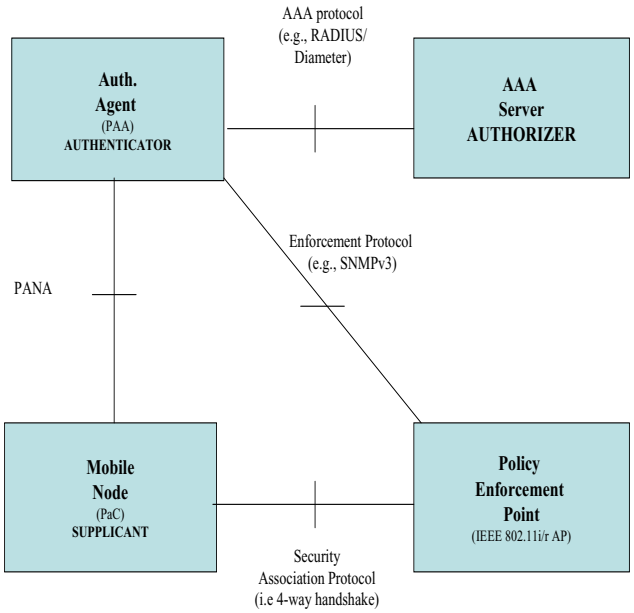


Figure 5.9: Interaction among the functional components

5.4.4 Experimental results and analysis

I have implemented the proposed network layer assisted pre-authentication mechanism in a testbed as shown in Figure 5.10. I illustrate different scenarios and demonstrate how network-layer assisted pre-authentication can provide link-layer handoff optimization. In particular, I apply the pre-authentication mechanism over IEEE 802.11 networks and compare the results with the existing pre-authentication mechanism for IEEE 802.11i. Figure 5.9 shows the interaction among several functional components and the protocols used between each pair of these components.

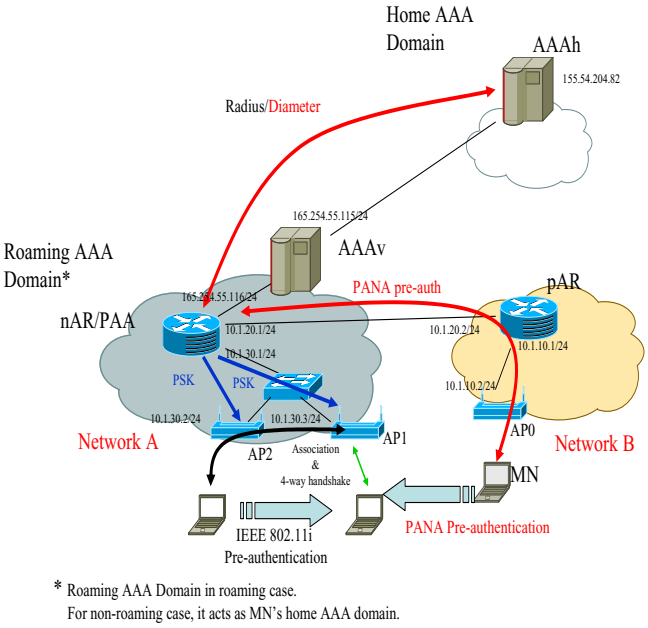


Figure 5.10: Experimental testbed for pre-authentication

In this experimental testbed I have used hostapd software [Hos] and madwifi driver [MAD] and have configured three Linux systems to act as access points. Two of these access points (AP1 and AP2) work as IEEE 802.11i APs. Both of these APs may work in either PSK (when network-layer pre-authentication is used) or 1X EAP mode. There is also inbuilt RADIUS client functionality within the AP (for the cases where network-layer pre-authentication is not enabled). Each AP implements a SNMPv3 agent (Simple Net-

work Management Protocol) that allows it to set PSKs and associated parameters such as key lifetimes. Finally, the last access point (AP0) is configured with open authentication. The MN is a laptop equipped with WPA supplicant software [MAD] that provides 802.11i functionality, madwifi driver, and Open Diameter's PANA client implementation [Ope]. PANA agent is based on open Diameter implementation that also provides inbuilt Diameter client. I have used open Diameter [Ope] and Free Radius [Fre] as the AAA protocol implementations.

I have experimented with three types of movement scenarios involving both roaming and non-roaming cases. In the roaming case, mobile node is visiting in an administrative domain that is different than its home domain. Consequently, the AAAh, which is placed in a different continent (e.g., in university of Murcia, Spain) in our experiment, needs to be contacted. For the non-roaming case, I assume the MN is moving within its home domain and only local AAA server (AAAv) is contacted.

The first scenario does not involve any pre-authentication. The MN is initially connected to AP0 and moves to AP1. Because neither network-layer authentication is enabled nor IEEE 802.11i pre-authentication is used, MN needs to engage in a full EAP authentication with AP1 to gain access to the network after the move (post-authentication). This experiment shows the effect of delay when there is no pre-authentication.

The second scenario involves 802.11i pre-authentication and involves movement between AP1 and AP2. MN is initially connected to AP2, and starts IEEE 802.11i pre-authentication with the target access point AP1. This is an ideal scenario to compare the values obtained from 802.11i pre-authentication with that of proposed network-layer assisted pre-authentication. Both the first and the second scenarios use RADIUS as AAA protocol with the APs implementing a RADIUS client.

The third scenario takes advantage of the proposed network layer assisted link-layer pre-authentication. It involves movement between two APs (e.g., between AP0 and AP1) that belong to two different subnets where 802.11i pre-authentication is not possible. Here,

Diameter is used as AAA protocol where PAA (PANA authentication agent) implements a Diameter client.

In this third movement scenario, MN is initially connected to AP0 in Figure 5.10. Mobile node starts PANA pre-authentication with the PAA which is co-located on the AR in the new candidate target network (nAR in network A) from the current associated network (network B). After authentication, PAA installs two pre-shared keys, PSK_{ap1} and PSK_{ap2} in both AP1 and AP2 respectively by using a preemptive key installation method. Finally, because PSK_{ap1} is already installed, AP1 starts immediately the 4-way handshake upon the mobile's arrival in network A.

As illustrated, I have used the same target access point AP1 to perform the handover for all the three scenarios. Therefore the 4-way handshake time measurement is always taken at this access point (e.g., AP1). For the first scenario, the mobile node (MN) is initially attached to AP0 because we try to demonstrate the case when 802.11i pre-authentication cannot be executed since both the access points are connected to two different subnets. This happens when the target AP (AP1) is not placed in the same DS (Distribution System) as current AP (AP0). For the second scenario, both AP1 and AP2 are configured with 802.11i support, so that one can simulate 802.11i-based network protection. Therefore, in order to initiate a handoff to AP1, the MN starts the test attached to AP2 after running an initial EAP authentication. Finally, for the third scenario, the MN is initially attached to AP0 and the handoff is performed to AP1. In this case, I simulate the scenario so that layer 2-based 802.11i pre-authentication cannot be performed and network-layer pre-authentication can be used instead.

MN uses application layer discovery mechanism discussed in Section 5.2 to discover PAA's (PANA authentication agent) IP address and all required information about the target APs, namely AP1 and AP2 (e.g., channel, security-related parameters, MAC address) at some point before the handoff. This avoids scanning during link-layer handoff. Because the focus is on reducing the time spent on authentication part during handoff, I do not

discuss the details of how I reduce the layer 2 scanning time. I have described the details of how scanning is optimized in [DZO+05]. It can also use any of the existing techniques that reduces layer 2 scanning as described in Section 5.3.

Table 5.2 shows the average timing (rounded off to the most significant number) associated with some of the handoff operations that I have measured in the testbed. I briefly explain each of the timings below.

Tauth refers to the execution of EAP-TLS authentication procedures. This time does not distinguish whether this authentication was performed during pre-authentication or a typical post-authentication.

Tconf refers to time spent during PSK generation and installation after EAP authentication is complete. When network-layer pre-authentication is not used, this time is not considered.

***Tassociation*+4way** refers to the time dedicated to the completion of association and the 4-way handshake with the target AP after the handoff.

I show the total time during the process by adding these components. Finally, I also highlight the time that affects the handoff in each case.

Each of these timings may safely be considered as independent per each experiment. Thus, the authentication phase, the configuration phase, and the association or 4-way handshake can be considered as independent events. In fact, *Tassoc+4way* time seems to be similar in value regardless of the movement scenario. Also, independent of whether PANA was run on roaming or non-roaming case, value of *Tconf* remains same.

The first two columns in Table 5.2 show the results for *non-roaming* and *roaming cases*, respectively, when no pre-authentication is used. The second and third columns depict the same cases when IEEE 802.11i pre-authentication is used. Finally, the last two columns show when network-layer pre-authentication was used. When pre-authentication is used, only the Tassoc+4way affects the handoff time. When no pre-authentication is used, the time affecting the handoff includes *Tauth* (the complete EAP-TLS authentication) plus *Tas-*

soc+4way. These results illustrate how network layer assisted layer 2 preauthentication can provide comparable results with 802.11i-based pre-authentication and at the same time can support inter subnet and inter-domain mobility that cannot be supported by IEEE 802.11i.

In Chapter 9, I illustrate how the proposed pre-authentication mechanism can interwork with other handoff related operations and uses application layer and network layer mobility protocols to build a complete handoff system.

Table 5.2: Experimental results for pre-authentication

Types of authentication	Post authentication		802.11i pre-authentication		Network layer assisted pre-authentication	
Types of movement	Non roaming	Roaming	Non roaming	Roaming	Non roaming	Roaming
T*authentication*	61 ms	599 ms	98 ms	638 ms	177 ms	831 ms
T*configuration* 2 AP	_	_	_	_	16 ms	17 ms
T*association*+ 4 way handshake	18 ms	17 ms	16 ms	17 ms	15 ms	17 ms
Total	79 ms	616 ms	114 ms	655 ms	208 ms	865 ms
Time affecting handover	79 ms	616 ms	16 ms	17 ms	15 ms	17 ms

5.5 Layer 3 configuration

In Chapter 3, I have defined a mobile's configuration processes during a mobility event. I have also illustrated how the mobile's layer 3 configuration processes affect the handoff delay and contributes to the packet loss. During layer 3 configuration, the mobile acquires IP address and assigns to its interface so that the mobile can communicate using the newly obtained IP address. Before assigning the IP address, the client usually performs duplicate address detection (DAD) by way of ARP (Address Resolution Protocol) or Neighbor Discovery in IPv4 and IPv6 networks, respectively. For example, for an IPv4-based network, this detection procedure may take up to 4 seconds to 15 seconds [VM98]. DAD-related delay for stateless address configuration of IPv6 address identifier may take up to 1500

ms and depends on the random value that determines the Neighbor Solicitation interval [NNS98].

In this section, I first analyze the effect of layer 3 configuration on handoff delay for both IPv4 and IPv6 networks. Then, I describe the key principles that need to be considered in order to optimize the delay due to configuration. I introduce a few related work that have optimized the configuration related delay. Then, I describe my proposed techniques that expedite part of layer 3 configuration process at the expense of additional signaling messages. Finally, I highlight the experimental results from the testbed.

As part of my investigation into layer 3 configuration optimization techniques using DHCPv4 and MIP, I have verified that with ARP enabled, the IP address acquisition took an average of 15 seconds, but when the ARP is suppressed average time taken for IP address acquisition was 436 ms. I conducted several experiments to analyze the effect of two factors on IP address acquisition, namely duplicate address detection (DAD) [TN98] and router selection on the disruption of real-time voice traffic over IPv6 network. I have experimented with both Mobile IPv6 from USAGI [TP01] and SIP-based terminal mobility [WS99].

DAD confirms the uniqueness of the IPv6 address on the link. During the DAD process, the new address is called a tentative address. According to RFC 2462 [TN98], a tentative address is not allowed to be used by a node. This means that the MN cannot send packets with a tentative address as a source IPv6 address and has to discard all inbound packets to a tentative address during DAD phase. This imposes an additional delay on any mobility binding update such as a re-INVITE in case of SIP. With default values as described in [TN98] the average delay caused by DAD is 1500 ms.

Router selection also plays an important role during handoff. According to RFC 2461 [NNS98], a host needs to perform certain steps before switching to another access router. These additional steps such as routing table update and neighbor unreachability detection (NUD) processes contribute to the delay for router selection process. I describe these two processes below.

5.5.0.1 Routing table update

To perform rapid handoff, hosts in an IPv6 environment should attach to the new access router whose RA (Router Advertisement) is most recent. However, commonly used Linux hosts do not always select the new access router quickly. If routing table has other routes, a host may select a different router. In this case, an IPv6 host performs NUD against old router to confirm unreachability, and after confirmation of unreachability, a host is allowed to switch to another router to connect.

5.5.0.2 Neighbor unreachability detection (NUD)

Neighbor unreachability detection verifies that two-way communication with a neighbor node exists. The host sends a neighbor solicitation to a node and waits for a solicited neighbor advertisement. If a solicited neighbor advertisement is received, the node is considered reachable. During a handoff operation, an IPv6 host must confirm unreachability to an old access router before switching to a new access router by using NUD mechanism in the absence of aggressive router selection mechanism. My study shows that NUD with default values can impose more than 8 seconds delay on the configuration without a mechanism like aggressive router selection.

I briefly explain how NUD contributes to the configuration delay. In NUD, each neighbor has a reachability state. When a host confirms that a neighbor is reachable, the reachability state of that neighbor is called REACHABLE. Then the host waits for REACHABLE TIME (ms) before the state goes to STALE. By receiving an RA it can also go to the STALE state. During the STALE state, nothing happens until a host sends new packets. After a host sends a packet, active reachability confirmation starts in the state of DELAY. During this state, the host waits for another DELAY FIRST PROBE TIME (seconds) and goes to PROBE state. In this state, the host uses Neighbor Solicitation to confirm reachability with predefined number of retransmissions (MAX UNICAST SOLICIT). The host does not get any Neighbor Advertisement from the target neighbor, the reachability state of the neighbor

goes to NULL. The amount of delay introduced by NUD process depends upon the time when a host gets a new RA and the NUD state of the old access router at that moment. If a host detects unreachability to an old access router before getting a new RA, NUD operation may not introduce any additional delay to the configuration process.

In order to study the effect of DAD and NUD on the configuration delay, I modified the Linux kernel to avoid the DAD and and I enabled the aggressive router selection procedure in the kernel module that helps the mobile to communicate with the new router quickly enough without doing a neighbor unreachability detection (NUD)[NNS98].

Figure 5.11 shows the IPv6 testbed where I have experimented with SIP-based mobility to study the effect of handoff delay due to DAD and NUD. This IPv6 testbed has one home network (N1) and two visited networks (N2) and (N3). The experiments involve three movement scenarios: i) movement between home network and visited network (from N1 to N2), ii) movement between visited networks N2 and N3 and iii) movement between visited network N3 and home network N1. The experimental results shown in Table 5.3 demonstrate how DAD- and NUD-related delays affect both the signaling and media redirection delays in case of SIP-based terminal mobility.

The delays shown in Table 5.3 are not inclusive of layer 2 access delays such as 802.11 scanning delays. Two different scenarios have been considered: (a) SIP mobility without aggressive router selection; (b) SIP mobility with aggressive router selection. Both the handoff related signaling delay and media delays are shown when the mobile moves between the home network and two visited networks, namely visited 1 and visited 2. H12 denotes when the mobile moves from home to visited network 1, H23 denotes when the mobile moves from visited network 1 to visited network 2, and H31 denotes when the mobile moves from visited network 2 to back to home network. The values demonstrate how by avoiding DAD and adopting aggressive router selection technique to reduce the effect of NUD, I could reduce the signaling delays to 200 ms and media interruption to less than 500 ms for SIP-based mobility [WS99]. I have published the details of this experiment in

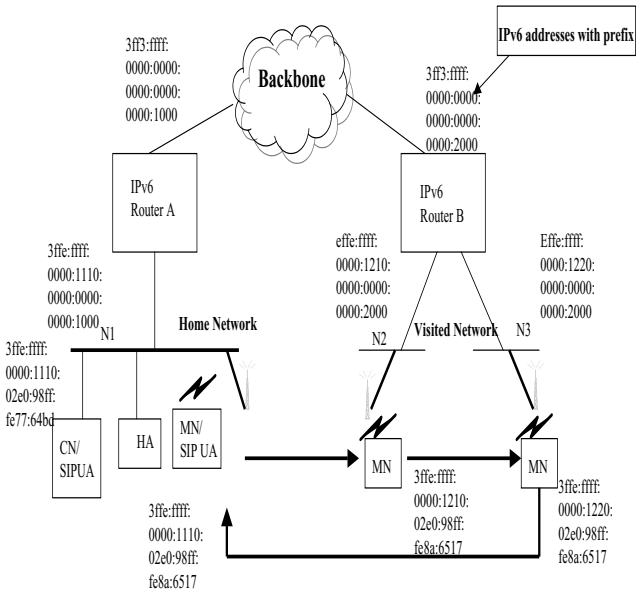


Figure 5.11: Experimental IPv6 testbed for handoff

[NDDS03].

5.5.1 Key principles

Following are some of the key principles that will help optimize the time taken for IP address acquisition during layer 3 configuration process.

1. Reduce the number of signaling message exchange between the mobile node and the DHCP server during the stateful IP address acquisition.

2. Minimize the time taken to verify the uniqueness of IP address of the mobile.

3. Perform the address uniqueness checking ahead of layer 3 handoff.

4. Pre-fetching and caching the new IP address reduces the time taken for IP address acquisition after the handoff.

Table 5.3: Effect of duplicate address detection (IPv6) on handoff

Handoff Case	Signaling Delay (ms)		Media Delay (ms)	
	SIP w/ DAD and NUD	SIP w/o DAD and NUD	SIP w/ DAD and NUD	SIP w/o DAD and NUD
H12 (Home-Visited 1)	3829	171	3854	421
H23 (Visited 1 – Visited 2)	3932	161	4188	419
H31 (Visited 2 – Home)	1935	161	1949	408

5. Perform the address resolution by mapping between IP address of the target router and MAC address before the mobile has moved to the new network.

5.5.2 Related work

For IPv6 networks, Moore et al. [Moo06], Han et al. [H+03] propose some of the optimization techniques needed to carry out DAD (Duplicate Address Detection) optimization for the IPv6 clients. Optimistic DAD [Moo06] ensures that the probability of address collision is not increased and thus improves the resolution mechanisms for address collisions. There are a few proposals in the IETF, such as Passive DAD [FSS06] and DHCP rapid commit option [PKB05] that try to expedite the IP address acquisition for IPv4 networks. I have proposed and implemented two optimization techniques that help expedite the IP address configuration process, namely router assisted duplicate address detection and proactive IP address configuration. Compared to the existing techniques, the first approach does not need any additional agent in the network and the router assists in reducing the time taken for IP address acquisition. The second approach reduces the delay at the expense of additional resources such as tunnels between the target router and mobile and additional network bandwidth. Below I describe these two methods in details.

5.5.3 Router assisted duplicate address detection

I have designed and implemented a router assisted duplicate IP address detection mechanism that reduces the layer 3 configuration time by expediting the duplicate IP address detection. It adopts the general principle of network doing the duplicate address detection instead of the mobile itself. In this mechanism, an upstream router keeps the list of IP addresses configured in a specific subnet in its neighbor-cache. A router in each subnet acts like a reporting agent and sends a list of IP addresses that are currently in use via a scope-based multicast address. A scope-based multicast address could be a multicast address with some TTL (Time-to-Live) value that can work over a range of subnets. An upstream router can send the list of the IP addresses in use in the neighboring subnets periodically using a scoped multicast address. A TTL (Time To Live) scoped multicast address can be used to limit the number of subnetworks the router can cover. For example, a router in a subnet can use a TTL of one whereas an upstream router can use a TTL that is higher than one and can cover multiple subnets. Figure 5.12 shows how the mobile obtains the list of used IP addresses that could be used in its own subnet or in the neighboring subnets from the router and thus does not need to perform an ARP before it assigns the address.

Thus, a mobile can obtain the list of addresses that are currently in use within its own subnet or in the neighboring subnets from the router without performing an ARP and having to wait for the ARP reply. Unlike other approaches, the proposed approach does not need any new element in the network and does not need changes in the DHCP server. However, there is a trade-off between the frequency of router advertisement and the load on the network. This technique also needs some modification on the router and the neighbor-cache entry in the router needs to be rebuilt in case of a power failure. I have explained the details of the proposed duplicate address detection mechanism in [DMCS06].

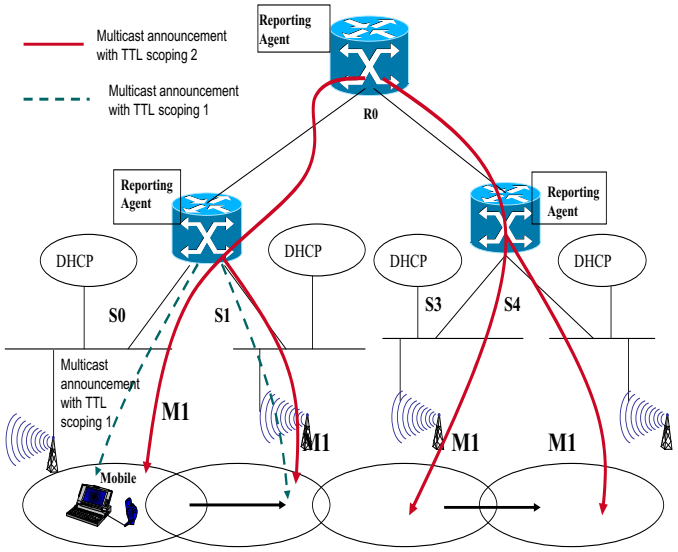


Figure 5.12: Router assisted duplicate address detection

5.5.4 Proactive IP address configuration

I have designed a proactive configuration technique that can work independently or in conjunction with pre-authentication mechanism described earlier. It adopts the general principle of proactive caching. The proactive configuration mechanism consists of several steps, namely, proactive address acquisition, proactive duplicate address detection and proactive address resolution. Below I have explained these steps in details.

5.5.4.1 Proactive IP address acquisition

Although FMIPv6 [JPA04] can pro-actively acquire an IP address, by obtaining the router prefix from the next access router, it expects that the adjacent routers need to cooperate and discover each other. Thus, FMIPv6-based fast handoff mechanism does not work for inter-domain mobility. My proposed technique is client assisted, and can be applied to both intra-domain and inter-domain mobility scenarios. In the proposed technique, the client obtains the IP address of the target network while the mobile is still in the current serving

network. The client assigns this pro-actively obtained address to a virtual interface and performs a subsequent proactive binding update to the home agent or correspondent node. Alternatively, the mobile can store it in the local cache and assign the address later on. This avoids the delay due to signaling messages needed during the address acquisition process after the handover.

5.5.4.2 Proactive duplicate address detection

When the DHCP server dispenses an IP address, it updates its lease table, so that this same address is not assigned to another client for that specific period of time. At the same time the client also keeps a lease table locally so that it can renew when needed. In some cases, where a network consists of both DHCP and non-DHCP enabled clients, there is a possibility that another client in the LAN may have been configured with an IP address from the DHCP address pool. In such scenario, the server detects a duplicate address based on ARP (Address Resolution Protocol) and IPv6 Neighbor Discovery for IPv4 and IPv6 networks, respectively before assigning the IP address. This detection procedure may take from 4 sec to 15 sec [VM98] and will thus contribute to a larger handover delay.

In my proposed method, the mobile node performs the duplicate address detection ahead of time, while it is still in the previous network, thus reducing the IP address acquisition time. This is performed by DHCP relay that co-locates with the next target router. In case of stateless address configuration, the proactive duplicate address detection (DAD) over the candidate target network is performed by the previous access router (PAR) on behalf of the mobile at the time of proactive handover tunnel establishment since duplicate address detection over a tunnel is not always performed.

5.5.4.3 Address resolution

Address resolution process has been defined in Chapter 3. Through address resolution process, one can obtain the mapping between the MAC address and IP address. Having prior

knowledge of IP address-to-MAC address mapping, both the neighboring first hop router and the mobile do not need to discover each other at layer 2 after the mobile moves to the new network. For example, if the MAC-to-IP address mappings are known to the mobile ahead of time, the mobile can communicate with nodes in the target network after attaching to the target network without waiting for an ARP broadcast or neighbor solicitation process. Mobile communicates with the access router, authentication agent, configuration agent and correspondent node after the handover.

I describe below several possible ways of pro-actively performing address resolution to obtain MAC-IP address mapping.

1. Use an information service mechanism (e.g., IEEE 802.21) to resolve the MAC addresses of the nodes. This requires that each node's network information (e.g., IP address, channel address, authentication scheme) are populated in the information server database. These information can be entered using the approaches discussed in [DMZ+06]

2. Authentication protocol that helps to pre-authenticate a mobile or the configuration protocol that is used for pre-configuration can piggyback the MAC address of the network entities during pre-authentication or pre-configuration process. The mobile can thus keep this MAC address in its cache and will avoid the address resolution process after it is handed over to the target network. For example, if PANA is used as the authentication protocol for pre-authentication, PANA messages may carry AVPs (Attribute Value Pairs) that can be used to carry the MAC address. In this case, the PANA authentication agent in the target network may perform address resolution on behalf of the mobile node and carry the related network parameters to the mobile node before the handover.

When the mobile node attaches to the target network, it installs the pro-actively obtained address resolution mappings without necessarily performing address resolution queries for

the nodes in the target network. On the other hand, the nodes that reside in the target network and are communicating with the mobile node should also update their address resolution mappings for the mobile node as soon as the mobile node attaches to the target network. The above proactive address resolution methods could also be used for those nodes to pro-actively resolve the MAC address of the mobile node before the mobile node attaches to the target network.

In order to expedite the address resolution process, a mobile could trigger the address resolution process as soon as it detects new network. This is based on mobile gratuitously performing address resolution [Per02c], [JPA04] in which the mobile node sends an ARP request or an ARP reply in the case of IPv4 or a neighbor advertisement (NA) in the case of IPv6 immediately after the mobile node attaches to the new network so that the nodes in the target network can quickly update the address resolution mapping for the mobile node.

5.5.5 Experimental results and analysis

I have demonstrated proactive address acquisition for both IPv4 and IPv6 networks using PANA. Independent of pre-authentication mechanism, I have also used stand-alone protocols, such as GIST (General Internet Signaling Transport) [SH08] and IKEv2 [SE06] to configure the mobile pro-actively. I briefly describe these experiments. I have described the details of these techniques in [DZO+05].

These experiments verify that the number of message exchange between the client and the network nodes (e.g., router, server) during IP address acquisition, processing time at the end systems, and network load are some of the key factors that contribute to the layer 3 configuration delay. Proactive caching of the IP address at the client and router or server assisted proactive duplicate address detection technique reduce the layer 3 address acquisition delay at the expense of additional resource usage at the mobile.

5.6 Layer 3 security association

In Chapter 3, I have defined the security association and have illustrated how re-establishment of security association affects the handoff delay and packet loss during a mobility event. The security association between two communicating nodes can exist at multiple layers. IPSec [KA98a] provides security association at layer 3. A layer 3 security association is uniquely identified by an SPI (Security Parameters Index), destination IP address and ESP (Encapsulating Security Payload). Thus, when the IP address of any one of the communicating hosts changes, a new security association needs to be re-established between the pair of nodes. During the mobile's repeated handoff, security association between the mobile and communicating host over the secured channel needs to be re-established when the end-point identifier (e.g., IP address) changes. The process of re-establishing the security association requires exchange of messages to derive the new key and processing at the end hosts and thus, contributes to the added delay during handoff.

In this section, I describe the key principles that need to be considered while optimizing the delay due to re-establishment of security association. I then describe the related work that attempt to optimize the delay due to security association during handoff. I describe the proposed techniques that optimize the delay due to security association at the cost of additional resources such as an additional home agent in the network and additional tunneling operations. Finally, I illustrate the experimental results in a testbed.

5.6.1 Key principles

Following are the key principles that can be considered to minimize the delay due to security association.

1. Maintain the security binding between the two communicating end points.

2. Avoid signaling exchanges between the peers in order to generate the encryption keys.

3. Maintain security context by way of reactive or proactive context transfer.

4. Maintain constant end-point connection identifiers.

5. Hide the change of IP address of the end-points by using additional home agent.

I have designed optimization technique that is based on few of the above principles and have experimented with it in a testbed. I have published the details of this optimization technique and the results in [DMDL07, DZM+05].

5.6.2 Related work

Miu et al. [MB01] describe an architecture that helps to maintain the security association when the mobile moves between public Internet and private enterprise networks. However, this solution is limited to movement between homogeneous networks (e.g., 802.11b). Rodriguez et al. [RCC+04] introduce the concept of mobile router where the end clients with multiple access technologies connect to the mobile router's down-link interface. In this case, the end clients do not change their IP addresses, rather the mobile router keeps on changing the external IP addresses as it moves around and connects to different access networks, such as GPRS, CDMA and 802.11b. The router uses NAT (Network Address Translator) functionality to shield the clients from re-initiating the sessions.

5.6.3 Anchor assisted security association

In this section, I describe my proposed mechanism that optimizes the handoff delay due to security association at the cost of an additional home agent in the network. While handoff delay is reduced, it introduces tunneling overhead because of the additional home agent in the network. This optimization technique is based on the key principles numbered 1,2 and 4 described in Section 5.6.1. In Chapter 9, as part of systems evaluation discussion, I demonstrate handoff optimization in IMS (IP Multimedia Subsystem) that uses the principle numbered 3 in conjunction with other optimization techniques.

The proposed mechanism uses an anchor agent that acts as a home agent and maintains the security association during the handoff thereby reducing the handoff delay and packet loss [DZM+05]. The key principle introduced by this technique is to maintain the security association with the end client even when the end-point identifier changes. This avoids the delay due to re-establishment of the security association during the mobile's handoff. I have experimented with this technique using both network layer mobility and application layer mobility protocols. In the experiment, a mobile with two interfaces moves back and forth between an enterprise network equipped with 802.11, a cellular network with CDMA1XRTT access, and a hotspot equipped with 802.11. By introducing an anchor such as a secondary home agent in the network, one can achieve secured seamless communication without the need to re-establish the IPSec [KA98a] tunnels during each subnet move.

Figure 5.13 illustrates a scenario of how security re-association is avoided by introducing an additional home agent x-HA. By using the external home agent x-HA, the mobile does not need to set up new IPSec association as it moves between subnets or domains. An internal home agent (denoted by i-HA) inside the intranet supports mobility inside the intranet. The external home agent (denoted by x-HA) in the DMZ (Demilitarized Zone)[2] handles a mobile's mobility outside the enterprise and ensures that a security association with the mobile does not break when the mobile changes its IP address.

Figure 5.14 illustrates how the mobile IP tunnels and VPN (IPSec) tunnels are set up during the mobile's movement from an enterprise network to an external network. If the mobile uses mobile IP's reverse tunneling, the data from the mobile will flow to the correspondent host in the reverse direction of the path shown in Figure 5.14. These tunnels are the additional systems resources expended while handoff delay is reduced by avoiding re-establishment of security association.

I describe the details of the architecture, implementation, and experimental verification in [DZM+04]. I briefly describe the techniques and associated results here.

[2]DMZ has been defined in Appendix C

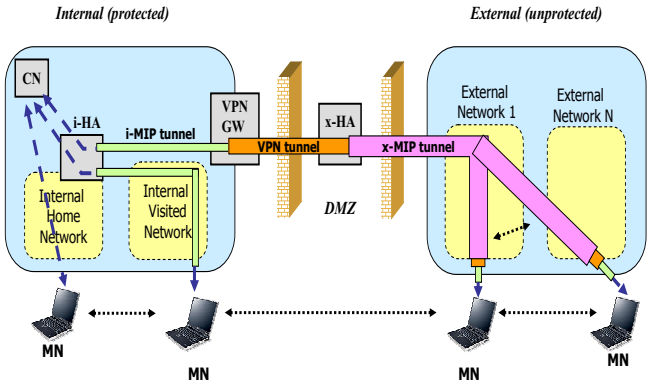


Figure 5.13: Anchor agent-assisted security association

The i-HA and x-HA collectively ensure that the packets received by the i-HA can be forwarded to the mobile currently on an external network. A mobile has two MIP home addresses: an internal home address i-HoA in the mobile's internal home agent and an external home address x-HoA in the external home agent. The mobile's care-of address registered with its i-HA is referred to as its internal care-of address and will be denoted by i-CoA. The mobile's care-of address registered with the x-HA is referred to as its external care-of address and will be denoted by x-CoA. The instance of MIP running between the mobile and its i-HA is referred to as internal MIP or i-MIP. The instance of MIP running between a mobile and the x-HA will be referred to as external MIP or x-MIP. After a successful VPN establishment (e.g., after a successful IPSec security association), the mobile obtains an address from the VPN gateway (VPN-GW) that is denoted as TIA (Tunnel Inner Address).

When a mobile moves into a cellular network, setting up the connection with a cellular network can take a long time. For example, from the experiment, I routinely experienced 10-15 second delays in setting up PPP connections to a commercial CDMA2000 1xRTT network. In addition, establishing an IPSec connection to the mobile's enterprise network could also lead to excessive delay. To enable seamless handoff, handoff delays need to be

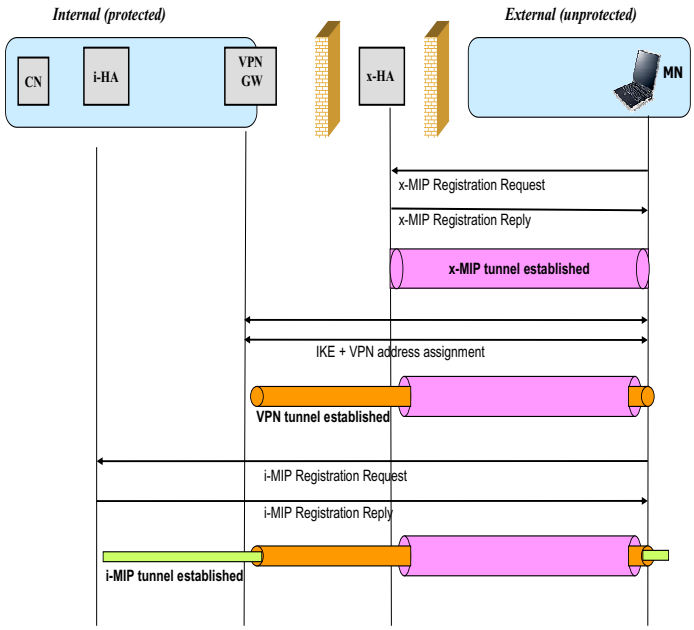


Figure 5.14: Mobile IP and VPN tunnels

significantly reduced.

Therefore, I applied handoff pre-processing and make-before-break techniques to reduce the handoff delay. In particular, a mobile anticipates the needs to move out of a currently used network, based on, for example, the signal-to-noise ratio in the networks. When the mobile believes that it will soon need or want to switch to a new network, it will start to prepare the connectivity to the target network while it still has good radio connectivity to the current network and the user traffic is still going over the current network. Such preparation may include the following steps:

1. Activate the target interface if the interface is not already on (e.g., a mobile may not keep its cellular interface always on if it is charged by connection time.

2. Obtain IP address and other IP-layer configuration information (e.g., default router address) from the target network.

3. Perform required authentication with the target network.

4. Establish the network connections needed to communicate over the target network (e.g., PPP connection over a CDMA2000 network).

Although both the interfaces are turned on at the same time, the decision to switch over from one interface to another will depend upon local policy that can be client-controlled or server-controlled. However, in this case, the handover anticipation is purely based on signal-to-noise ratio (SNR) of the 802.11 interface. But this handoff decision could be based on any other specific cost factor. When the mobile decides that it is time to switch its application traffic to the target interface, it takes the following steps:

1. It registers its new care-of address acquired from the target network with the x-HA.

2. It establishes a VPN tunnel (IPSec association) between its x-HoA and the VPN gateway inside the DMZ of its enterprise network.

3. It registers the gateway end of the VPN tunnel address as its care-of address with the i-HA. This will cause the i-HA to tunnel packets sent to the mobile's home address to the VPN gateway, which will then tunnel the packets through VPN tunnel and the x-MIP tunnel (Mobile IP tunnel with the external home agent) to the mobile.

4. When the mobile moves back to the enterprise network, the VPN and the MIP tunnels will be torn down. Tearing down the VPN tunnel takes up to a few seconds due to negotiation between the end points. Thus, some in-flight packets may get lost or may arrive at a later time leading to out-of-order packet delivery. Most of today's applications are capable of reordering of the out-of-sequence packets (e.g., out-of-sequence RTP packets).

5. When the mobile moves from one external network to another external network and acquires a new local care-of address (x-CoA), the mobile's x-HoA remains the same. Therefore, the mobile's existing security association does not break. The mobile only

needs to register its new local care-of address with the x-HA so that the x-HA will tunnel the VPN packets to the mobile's new location.

5.6.4 Experimental results and analysis

I have experimented with the proposed technique for both CBR (Constant Bit Rate) traffic (audio) and VBR (Variable Bit Rate) traffic (video) and have analyzed the packet loss, delay, and jitter during the handoff. Figure 5.15 shows the experimental testbed where I have conducted this experiment. It shows the enterprise network, two home agents (external home agent and internal home agent), VPN gateway, cellular network and another external WiFi network. The mobile moves back and forth between the enterprise network, cellular network and the WiFi hotspot. The mobile sets up IPSec connection with the VPN gateway.

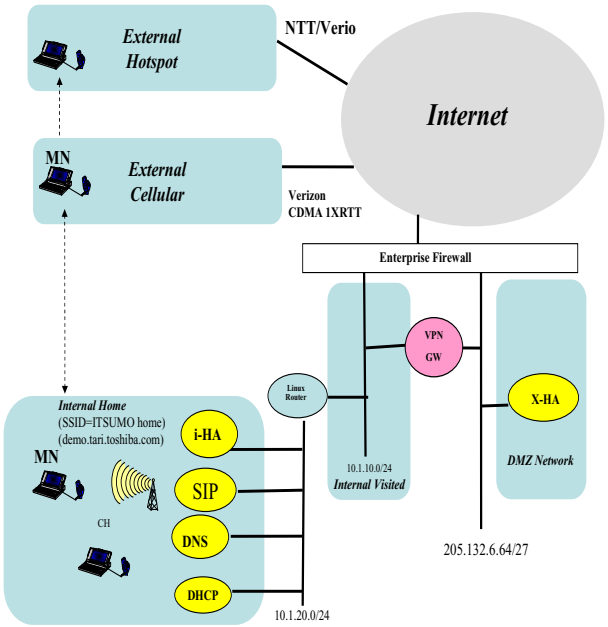


Figure 5.15: Experimental testbed for security association

In the absence of the proposed optimized technique, the mobile experiences packet loss due to the delay associated with IPSec tunnel setup and tear-down every time it changes its point of attachment. Without any optimization, layer 2 configuration took about 10 seconds

in a CDMA network, layer 3 address acquisition took about 3 seconds in 802.11 network. Both the binding updates, namely both external and internal MIP registrations, took about 300 ms and 400 ms, respectively to complete. The IPSec-based security association took about 6 seconds. As the mobile moved back to the home network, it took around 200 ms for mobile IP de-registration. These signaling exchanges result in degradation of real-time services due to the associated delay and packet loss. However, using a combination of anchor-based security association technique that helps to maintain the security binding and a make-before-break technique, one can obtain zero packet loss during the handover from 802.11 network to cellular network and vice-versa. Although there was no packet loss in the optimized case, the mobile received a few out of order packets during its movement back from cellular network to 802.11 network as the transit packets on the slow cellular link arrived later than the initial packets that arrived via the 802.11 interface.

Figure 5.16 shows the interaction between different network components (e.g., CN, MN, i-HA, x-HA and VPN-GW) during the mobile's movement from 802.11 to a CDMA network and vice versa when the optimization technique is deployed. Figure 5.16a shows how a mobile receives the data traffic while in the 802.11 network and it prepares to hand over to CDMA network at a certain threshold signal value S1, by setting up the PPP connections and establishing the xMIP and IPSec tunnels. Figure 5.16b shows how at an SNR value of S2, the mobile updates the internal home agent with the tunnel inner address. At this point the data flows directly to the CDMA interface using triple encapsulated tunnels. Figure 5.16c shows the signaling sequence when the mobile goes back from CDMA network to the 802.11 network.

Figure 5.17 shows the results of packet loss with and without optimization for security association. Figure 5.15(a) shows the packet loss due to re-establishment of security association. Figure 5.15(b) shows how the packet loss is avoided by introducing the additional home agent as the anchor point. Although no packet loss was observed, the mobile received out-of-order packets when it moved from cellular to WiFi network. When the mobile is in

the cellular network, the slope of RTP traffic is less inclined meaning that the packets are subjected to buffering delay in CDMA base station and output rate is less than the input rate. If the packet after the handoff is delayed beyond a certain threshold (e.g., inter packet delay between the last packet before handoff and first packet after the handoff is larger than 300 ms), then the packet may be considered lost for certain application such as VoIP.

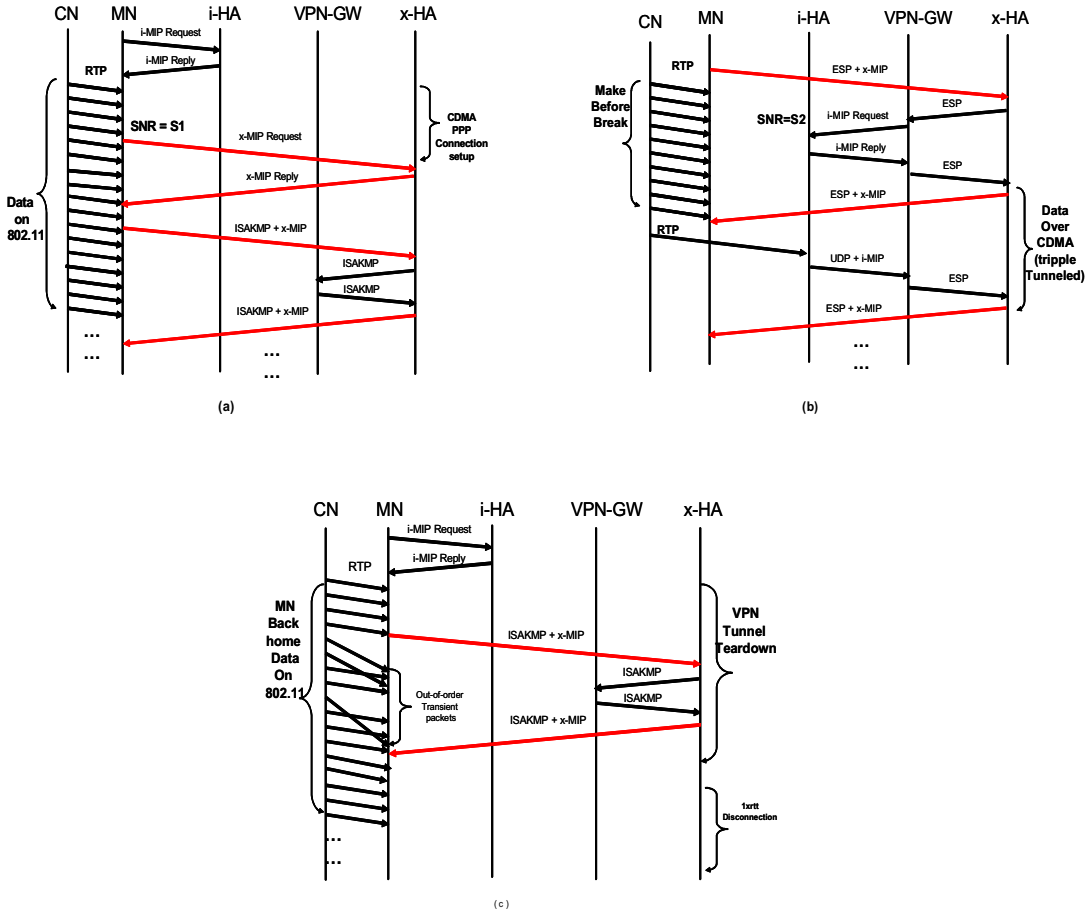


Figure 5.16: Interaction between network components during handoff

Figure 5.18 illustrates packet transport delay in 802.11 and CDMA networks and jitter introduced during handoff between 802.11 network and CDMA network for CBR traffic such as VoIP. I used audio application RAT (Robust Audio Tool) [UCL] to generate the audio traffic. Figure 5.19 illustrates packet transport delay both in 802.11 and CDMA networks and jitter introduced during handoff between 802.11 network and CDMA network for VBR traffic such as video over IP. I used the video conferencing application VIC (Video

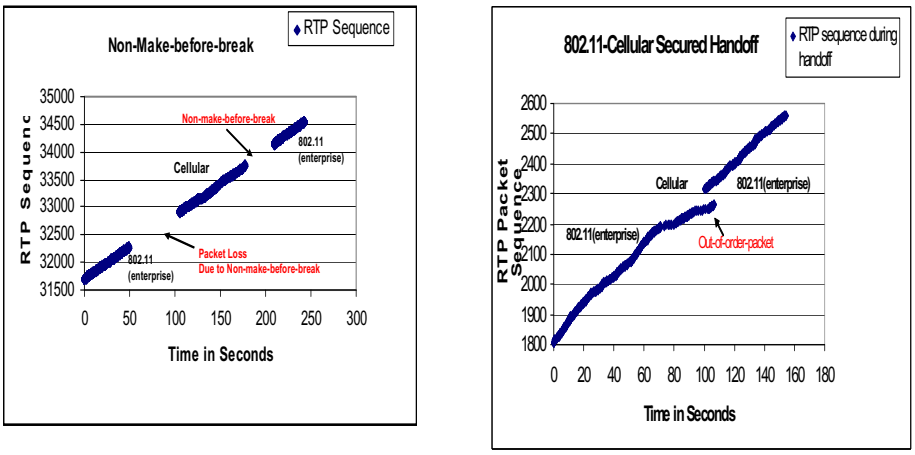


Figure 5.17: Effect of security rebinding and its optimization

Conferencing tool) [UCB] to generate the video traffic. Both RAT and VIC are open source software.

5.7 Binding update

I have explained binding update procedure and its effect on handoff delay in Chapter 3. Distance between the mobile node and the correspondent node or the home agent (HA) contributes to the binding update delay resulting in overall handoff delay and data loss. In this section, I propose several optimization techniques to optimize the binding update delay and reduce the effect of binding update delay. These techniques for binding update can be categorized as *hierarchical binding update*, *proactive binding update*.

I first describe the key principles that can be considered to optimize the binding update delay or reduce the effect of binding update. I describe some of the related work that have attempted to reduce the binding update delay. Then I introduce my proposed techniques that use some of these principles in optimizing binding update delay at the cost of additional resources in the network. I validate binding update optimization techniques using

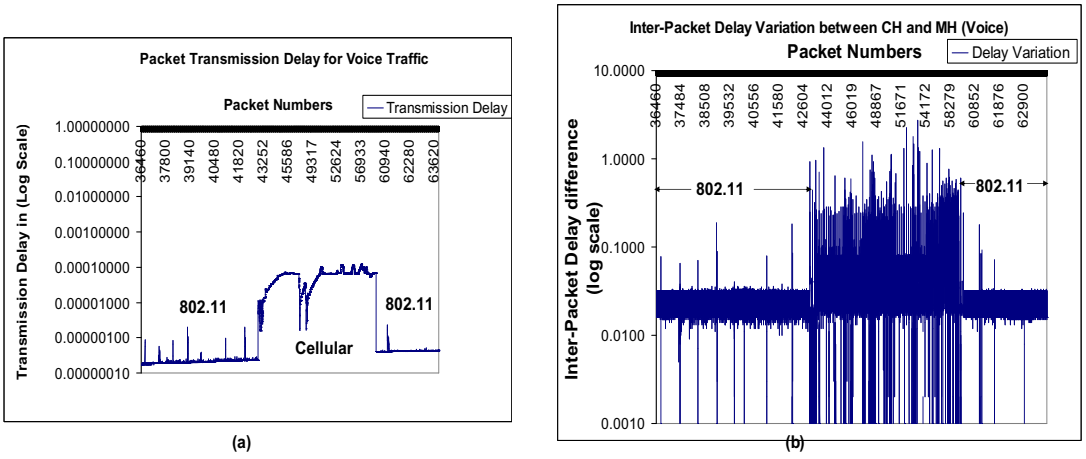


Figure 5.18: Delay and jitter effect on VoIP traffic

experiments.

5.7.1 Key principles

The following are some of the key principles that are taken into account for optimizing binding update delay.

1. Limit the traversal of binding update closer to the mobile after every handoff.

2. Use of two levels of binding update by using an anchor agent between the home agent and the mobile node.

3. Apply the binding update pro-actively in the previous network before the mobile has moved to the new network.

4. Simulcast the data to help reduce the data loss due to longer binding update delay. This can probably be achieved by using localized multicast approach.

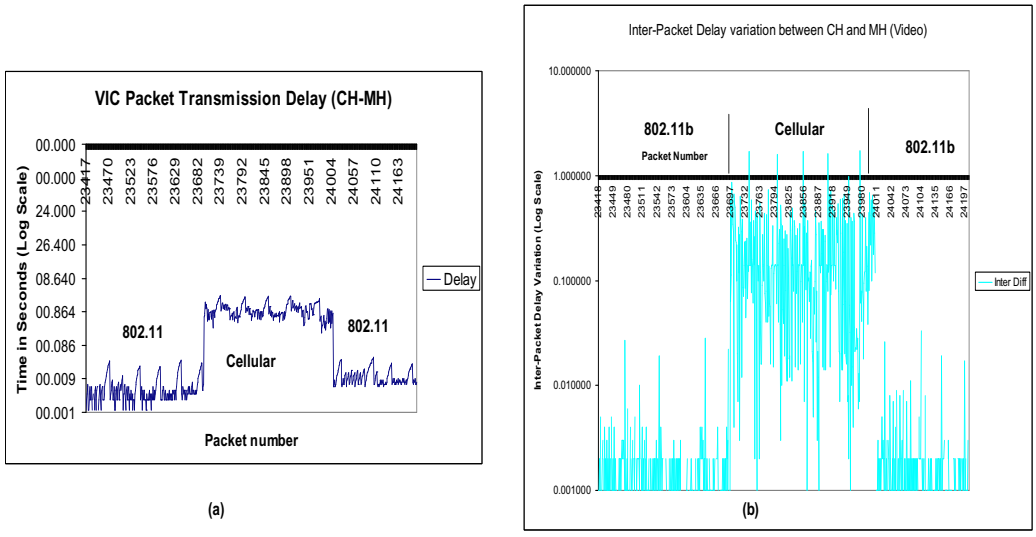


Figure 5.19: Delay and jitter effect on video Traffic

5.7.2 Related work

There are enhancements to layer 3-based mobility protocols to reduce the binding update delay when CN (Correspondent Node) and MN are far apart. MIP regional registration [FJP07] provides hierarchical mobile IP registration for IPv4. HMIPv6 [SCeMB06] introduces an agent called MAP (Mobility Anchor Point) to localize the intra-domain mobility management.

Proactive binding updates allow the mobile to send a binding update before the mobile has moved to the new network. It helps to eliminate the delay due to binding update after the handoff. FMIPv6 [Koo05] adopts a fast binding update (FBU) technique where it sends the binding update to the previous access routers so that the in-flight packets during handoff can be forwarded from the previous access router (PAR) to the mobile. However, this requires additional signaling between the neighboring routers to forward the data. Malki et al. [Mal07] also describe techniques to provide low latency handoff in MIPv4 environment where the transient packets are forwarded from the previous foreign agent.

In the following sections, I describe my proposed techniques.

5.7.3 Hierarchical binding update

I have developed and demonstrated hierarchical binding update techniques for both network layer mobility and application layer mobility protocols. My proposed techniques introduce an anchor point in the network that helps to limit the binding update when the mobile's movement is confined to a domain, where a domain is defined to be a set of subnetworks that are controlled by the mobility agent. This technique helps to optimize binding update delay, reduces the network load at the expense of additional network element such as mobility agent. I have applied these techniques to both the application layer mobility and network layer mobility protocol.

I have explained the details of the implementation and experimental analysis for the above two cases in [DDM+02] and [DMC+04], respectively. These techniques were developed around the same time with the other related hierarchical mobility management techniques. I describe these two techniques and experimental results below.

5.7.3.1 Network layer mobility agent assisted

I have designed a network layer-based intra-domain mobility management [DDM+02] protocol by adopting a similar approach where a mobility agent acts like an anchor point. Figure 5.20 shows how an anchor agent called mobility agent (MA) can be used to provide hierarchical binding update when the mobile moves within a domain.

The mobile assigns two addresses: *local care-of-address* and *global-care-of-address*. The first time the mobile moves to a domain, it sends two binding updates, one to the mobility agent with local care-of-address and one to its home agent with the address of the mobility agent which is same as the global care-of address. Thus, any packet from the home agent gets intercepted by the local mobility agent first. The local mobility agent first decapsulates the original packet, then encapsulates it again with local-care-of-address and then sends it to the mobile. For every subsequent move within the domain, the local binding update is sent to the anchor agent only and is not propagated to the home agent. Although

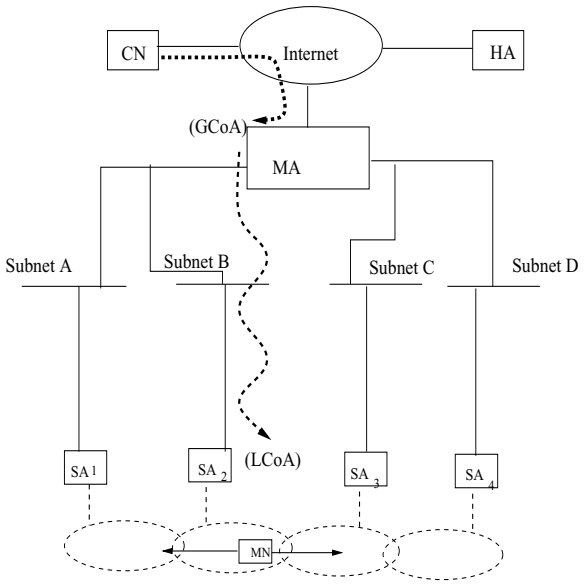


Figure 5.20: Functional architecture for hierarchical mobility agent

this technique reduces the delay due to binding update, the traffic will be subjected to additional processing delay due to encapsulation and decapsulation at the mobility agent. Figure 5.21 shows the call flow when the MN first moves into a new domain managed by a mobility agent. Figure 5.22 shows the call flow during subsequent intra-domain movement.

According to the flow shown in Figure 5.21, when the MN first moves into a domain, it obtains a local care-of address (this LCoA is SA_2's address) by performing a subnet-specific registration. IDMP allows the serving SA (SA_2 in this case) to dynamically assign the MN a Mobility Agent (MA) during this subnet-specific registration process. The MN then performs an intra-domain location update by communicating its current LCoA to the designated MA. The MA includes either its address or a separate GCoA in the intra-domain location update reply. Subsequently, the mobile node is responsible for generating a global location update (registration) to the necessary remote nodes (e.g., HA if Mobile IP is used for global mobility management or Registrar (LR) if SIP is used); this is however independent of the IDMP specifications.

After the initial intra-domain registration process, IDMP now allows the MN to retain its global care-of address as long it stays within the same domain. Whenever MN changes subnets within this domain, it performs a new subnet-specific registration with the new SA.

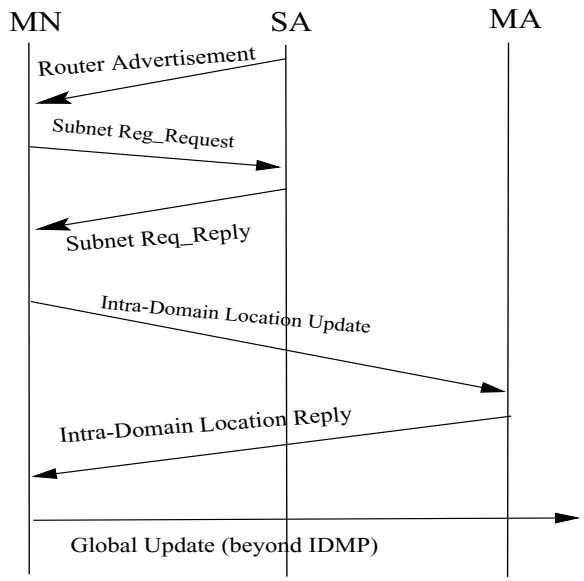


Figure 5.21: Initial intra-domain location update

Since the MN indicates that it has an existing valid registration, the SA does not allocate it a new MA address in this case. The MN then performs a new intra-domain location update and informs its MA of its new local care-of address. No global messages are generated in this case, since the global care-of address remains unchanged. As with other hierarchical mobility management schemes, the localization of intra-domain mobility significantly reduces the latency of handoffs across subnets within the same domain and also decreases the frequency of global signaling traffic.

Figure 5.22 shows call flow during subsequent intra-domain movement.

5.7.3.2 Application layer anchoring agent assisted

In case of an application layer mobility protocol, I use a back-to-back SIP user agent (B2BUA) as the anchor point, possibly closer to the mobile node. A B2BUA consists of two SIP user agents where one user agent receives a SIP request, possibly transforms it and then has the other part of the B2BUA re-issue the request. A B2BUA in each domain needs to be addressed by the MH in the visited domain. The B2BUA issues a new request to the CH containing its own address as the media destination and then forwards the packets, via RTP translation or NAT, to the MH. I have described the details of how a

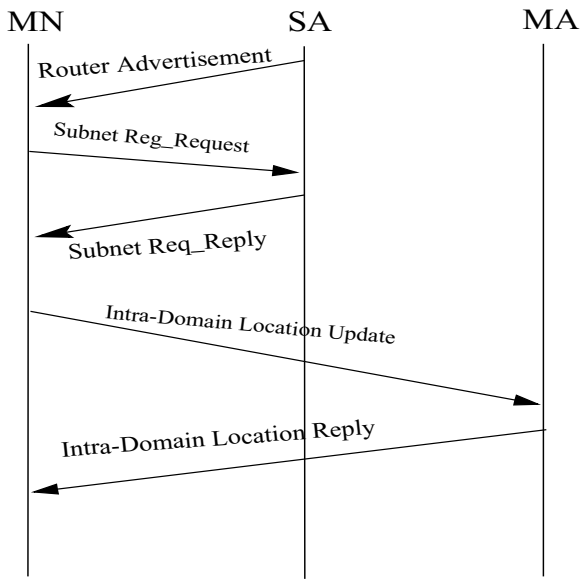


Figure 5.22: Call flow during subsequent intra-domain handoff

B2BUA can be used as an anchor point to reduce the binding update delay in [DMC+04]. Figure 5.23 illustrates the functional architecture of how B2BUA can be used to reduce the binding update delay. Figure 5.24 shows the detailed protocol flow for the B2BUA-based binding update.

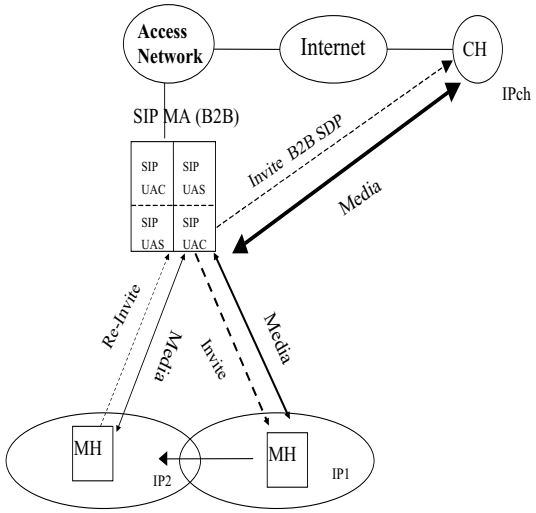


Figure 5.23: B2BUA-based hierarchical binding update

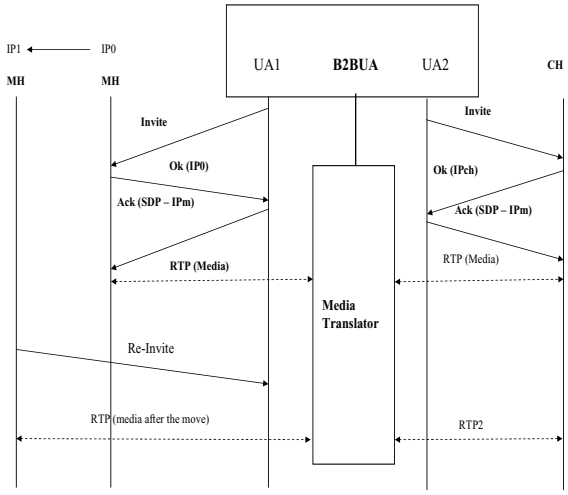


Figure 5.24: B2BUA-based flow for hierarchical binding update

5.7.4 Experimental results and analysis

This section describes basic prototype implementation of IDMP. I first describe the functional components of the implementation. The Mobility Agent (MA) handles local registration requests from MNs that are currently in its domain, and provides temporary bindings to the MNs as long as they remain in the domain. As far as the handling of such registration (or location update) requests is concerned, there is little functional difference between HA and MA. Unlike the HA, which has a permanent list of mobility bindings for each MN associated with its home network, the MA maintains a dynamic list of mobility bindings for currently registered MNs. The major functional difference between HA and MA is in terms of packet forwarding to the MN. When the MN is away from the home network, the HA is responsible for collecting all the packets directed at the MN's permanent IP address and tunneling the packets to the global care-of address (which is also the IP address of the MA interface). The task of the MA is simpler; it receives the packets automatically, and after decapsulating the packets, redirects the inner IP packet to the MN's local care-of-address.

In fact, the HA is potentially unaware of the use of IDMP and the presence of the

MA. As in conventional Mobile IP, it simply has to intercept all packets intended for the MN from the home network, encapsulate them and forward them to the care-of address specified in the MN-HA registration message. I have used [BZCS96] in the experimental testbed. The registration request and reply message formats for global registrations are, in fact, identical to Mobile IP used in MosquitoNet [BZCS96] with a single exception: the reserved bit in flags field is now used to indicate whether the MN is operating with in cooperation with a mobility agent.

Figure 5.25 shows the experimental network testbed used for validating this mechanism. It shows the functional components and the associated IP addresses. I considered a single MN served by its HA (Durga=192.4.20.44) in its home network (10.10.5.0), with home IP address 10.10.5.10. The home interface address of Durga is 10.10.5.1. Two MAs, e.g., MA_1 (Lakshmi=192.4.20.43) and MA_2 (Saraswati=192.4.20.45) are connected to routers serving subnets 10.10.1.0 and 10.10.2.0, respectively. I assume that the mobility domain comprises both subnets 10.10.1.0 and 10.10.2.0. Accordingly, both the hosts *Lakshmi* and *Saraswati* can serve as mobility agents for the MN as long as it stays within this domain.

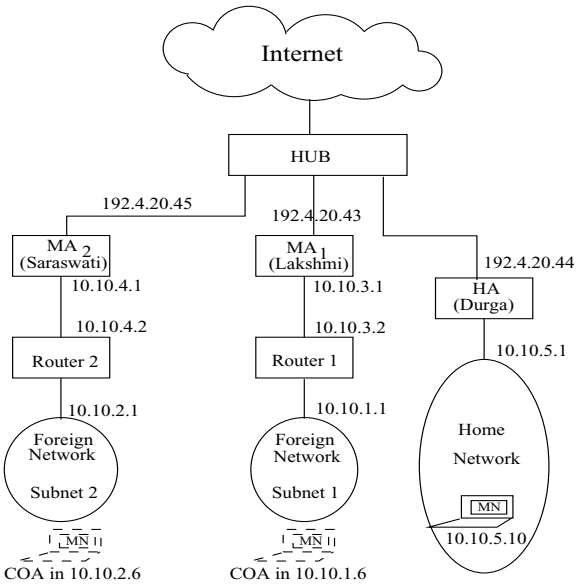


Figure 5.25: Experimental testbed for hierarchical mobility

As the MN enters into the subnet 10.10.1.0, it receives a locally scoped co-located address 10.10.1.6 and the IP address of (192.4.20.43) as its global care-of address. The

MN accordingly first informs MA_1 of its local care-of address (10.10.1.6) and subsequently registers with the HA using 192.4.20.43 as its care-of address. Afterwards, the MN roams into the subnet 10.10.2.0 and gets a new local care-of address 10.10.2.6. Since MA_1 is still its MA, the MN simply performs an intra-domain location update, informing MA_1 of its new local care-of address.

To test the case of inter-domain (global) mobility, I subsequently configured the DHCP server to provide a new MA address, say (Saraswati=192.4.20.45), to the MN. In this case, the MN performs both the intra-domain and inter-domain registrations.

5.7.4.1 Analysis of results

In this section, I compare the signaling overhead associated with MA assisted mobility management with that of base Mobile IP. I use the following parameters to express the signaling overhead.

L_g = 46: Size of global registration packet (in bytes).

L_l= 50: Size of local registration packet (in bytes).

(Note that $L_g \leq L_l$, since the global registration request does not contain the local care-of address field.)

T_s: Average duration for which MN remains in a subnet (secs/subnet).

T_d: Average duration for which MN remains in a domain (secs/domain).

N: Average number of subnets in a domain.

N_{MA} =2: Average number of hops from MN to MA when the MN is in foreign network.

N_{HA} =5: Average number of hops from MN to HA when the MN is in foreign network.

(2 and 5 are arbitrary numbers)

Clearly, T_s and T_d depend on the network and topology and the mobility pattern of the MN. For the sake of simplicity, in my analysis I assume $T_d = N \times T_s$. Table 5.4 displays the expressions for signaling overhead in basic Mobile IP and under hierarchical mobility

management involving MA. In each expression, the factor of 2 is due to the fact that each registration attempt involves exchange of a registration request and a corresponding reply message.

Table 5.4: Expressions for signaling overhead

Architecture	Signaling overhead (bytes/sec)		
	Local per hop	Global per hop	Total in Network
Mobile IP	0	$2L_g/T_s$	$2N_{HA}L_g/T_s$
Mobility Agent	$2L_l/T_s$	$2L_g/T_d$	$2N_{HA}L_g/T_d$+$2N_{MA}L_t/T_s$

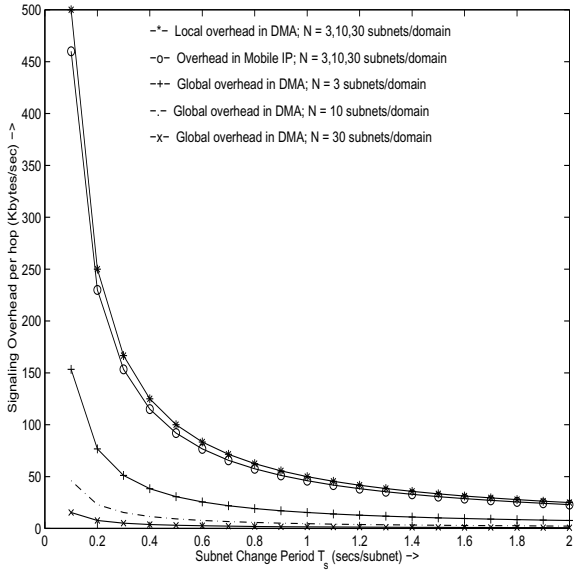


Figure 5.26: Global and local signaling overhead for IDMP

The global and local signaling overhead per hop in the MA-assisted hierarchical mobility architecture (shown as DMA (dynamic mobility agent) in the figure)) against T_s for different values of N (3, 10, and 30) are plotted in Figure 5.26. These numbers are chosen arbitrarily in order to reflect the increase in number of subnets per domain. As expected, global signaling overhead in the proposed architecture is significantly less than the corresponding local overhead. Also the signaling overhead goes down as the MN stays longer in a subnet (and domain). As the number of subnets in a domain increases, the global signaling overhead reduces whereas the local signaling overhead remains unchanged. In other

words, global signaling overhead in basic mobile IP and local overhead with hierarchical MA does not depend on N. Since global signaling messages travel over a larger number of hops (and hence consume a larger portion of network resources), hierarchical mobility management has advantage over mobile IP in terms of total network capacity (aggregated over all hops). From the plots in Figure 5.26, it is clear that hierarchical mobility agent results in a significant reduction in the network signaling overhead, especially when mobiles change subnets more frequently and when a larger number of subnets form a single domain. As N_{HA} increases, the reduction in signaling overhead in the proposed scheme becomes more significant. For example, if I use hierarchical mobility management (DMA) in a 30 subnets/domain network instead of 3 subnets/domain network, the percentage gain in terms of signaling overhead will be approximately 14 keeping the subnet mobility rate constant.

5.7.5 Proactive binding update

I have designed a proactive binding update technique that works in conjunction with the pre-authentication and proactive configuration techniques described earlier in Section 5.3 and Section 5.5, respectively. As part of the proposed technique, the mobile uses the proactive handover tunnel and sends the binding update pro-actively target the cached IP address that the mobile has obtained from the target network. Thus, any packet destined to the new address is picked up by the next access router and is tunneled to the mobile before it moves to the new point of attachment. After the mobile moves to the new network, the IP-IP tunnel is deleted and the new IP address is assigned to the physical interface, but no new binding update is necessary [DOF+10]. This effectively removes the need for sending another binding update after the mobile moves to the new network. Thus, this technique eliminates the delay associated with the binding update completely. This technique uses the general optimization principle of reducing the amount of signaling messages after the handoff in establishing the new identifier. However, it introduces the additional complexity

of managing the transient tunnel between the mobile and the router in the target network.

I have experimented with the proposed proactive binding update technique for both application layer mobility protocol based on SIP and network layer mobility protocol, namely MIPv6. In Chapter 9, I will describe how proactive binding update mechanism works in conjunction with other handoff optimization techniques, namely, pre-authentication, proactive IP address acquisition for both network layer and application layer mobility protocols.

5.8 Media Rerouting

In Chapter 3, I have defined media routing as one of the handoff components. Media rerouting is the final step in the handover process before the data path is re-established between the communicating nodes. Media rerouting process may include several elementary operations, such as encapsulation, decapsulation, tunneling, buffering, and store-and-forward. There is certain overhead associated with encapsulation, decapsulation and tunneling operations. During the media re-routing process, transient data may get lost or may get delayed because of these operations. However, the in-flight data can be captured and redirected to the new point of attachment. Redirecting the transient media during handoff reduces the in-handoff packet loss. There are several ways the transient data can get forwarded from the CN to the mobile. I describe few candidate protocols or mechanisms that can be used to forward the traffic so that in-flight data loss is minimized.

In this section, I describe the key principles that should be considered while designing the optimization techniques to reduce the media delivery delay and packet loss. I then highlight some related work that have optimized the media delivery to reduce the in-flight media delay and packet loss. I describe few redirection techniques that I have developed as part of my thesis based on these key principles. These are 1) *Data redirection using forwarding agent*, 2) *mobility proxy-assisted time-bound data redirection*, and 3) *time-bound multicasting*. These techniques help mitigate the effect of binding update delay by forwarding the

in-flight data to the new point of attachment. Compared to the related work, the proposed techniques do not need any changes in the existing networking infrastructure. Finally, I demonstrate these techniques with some experimental results in the testbed.

5.8.1 Key principles

The following are some of the key principles that help reduce the packet loss due to the redirection of in-flight data during handoff. I have applied some of these key principles while designing the optimization techniques.

1. Simultaneous binding of both the care-of-addresses (care-of-address at the current network and target network) to the home agent or CN reduces the data loss due to media redirection.

2. Forwarding of in-flight data from the previous network during handoff reduces the packet loss. Forwarding from previous network can be done either using reactive tunnels[3] or application layer forwarding technique. Forwarding technique helps to forward those packets from the previous network that would have been lost due to delay in binding update. However, it cannot avoid the packet loss completely (i.e. those due to L2 and L3 handoff delay). Although, it could be useful for low-latency application, as there is no network buffer delay.

3. A combination of buffering and forwarding can be applied without doing simultaneous binding update. Buffering techniques can be applied to any parts of the network, such as the edge of the network, core of the network or at the source.

4. Bi-casting or localized multicasting of data at the edge of the network helps to reduce the data loss.

[3]Reactive tunnel has been defined in the definition section

5.8.2 Related work

There is a relatively small amount of related work, namely RFC 4881 [Mal07], [PW99], and [CHK+00] that help reduce the transient data loss during the handoff by redirecting the data from the previous network. Koodli et al. [Koo05] propose reactive and proactive handover mechanisms that allow the in-flight data to be forwarded from the previous network and buffered in the target router, respectively. I have experimented with those techniques and have presented the results in Chapter 3. Vakil et al. [VFBF01] designed a virtual soft-handoff method for CDMA-based wireless IP networks using localized multicasting technique. However, this scheme works for CDMA network only and does not provide a generalized solution suitable for other type of access network, such as 802.11. Tan et al. [TLP99] propose a fast handoff scheme for wireless networks that use hierarchical mobility approach and use multicasting technique to reduce packet loss during intra-domain handoff. However, this approach requires that each mobile node is assigned a dedicated multicast address.

I have developed a few mechanisms that help to reduce the packet loss due to binding update delay during handoff. These techniques are protocol independent and could be applicable to both network layer and application layer mobility protocols. These mechanisms are dependent upon packet forwarding techniques and localized multicasting techniques. Unlike the mechanisms proposed by Koodli et al. [Koo05], these mechanisms do not need any cooperation between the previous access router and next access router and work across the administrative domains. Unlike the mechanisms proposed by Vakil et al. [VFBF01] and Tan et al. [TLP99] the proposed localized multicasting technique is access independent and does not need to assign multicast address to each mobile node. I explain these techniques in details in the following sections.

5.8.3 Data redirection using forwarding agent

The forwarding agent takes care of capturing the in-flight data during handoff and sends it to the new destination. Placement of the forwarding agent in the network determines the amount of in-flight data that can be forwarded during the handoff. In most ideal scenario, it is better to place the forwarding agent closer to the access networks.

Figure 5.27 shows a scenario where the mobile moves from network 1 to network 2. Forwarding of data from the current network (e.g., network 1) to target network (e.g., network 2) can be established reactively by setting up a transient tunnel between the router in the current network 1 and the mobile in the new network or by applying any application layer forwarding technique. Similarly, proactive forwarding of data from network 2 and network 1 is established by setting up a transient tunnel between router 2 and the mobile in the previous network. I categorize these types of transient tunnels into two basic categories, namely *proactive* and *reactive*. Depending upon the nature of tunnel and type of data forwarding, these tunnels can be defined as proactive handover tunnel or reactive handover tunnel. I define the functionalities of these tunnels in more details.

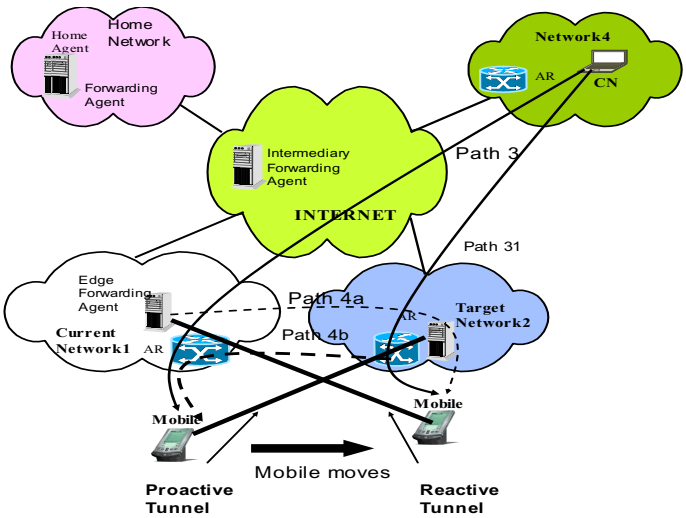


Figure 5.27: Forwarding agent for data redirection

5.8.3.1 Reactive Handover Tunnel (RHT)

According to Figure 5.27, the reactive handover tunnel is established between the mobile and router 1 in the previous network after the mobile has moved to the new access network. Reactive handover tunnel helps to forward the in-flight data traffic from the previous network, until the new path is established between the mobile and correspondent node. Path 4a in Figure 5.27 shows how the in-flight data that get redirected from network 1 to network 2 until path 5 is established between the mobile and the correspondent node. In-flight data is sent over this tunnel and is received by the mobile in network 2.

5.8.3.2 Proactive Handover Tunnel (PHT)

Proactive handover tunnel is established between the mobile in network 1 and the router in network 2 before the handover. Path 4b in Figure 5.27 shows the data that get redirected from network 2 to network 1 after path 3a is established with the new network before the handover. This mechanism is useful during proactive handover. In case of proactive handoff, data path 3a is established before the mobile has moved to network 2. In this case, data is forwarded from network 2 to network 1 while the mobile is still in network 1 and data gets buffered at router 2 during mobile's handoff from network 1 to network 2. After an IP address is pro-actively acquired from the DHCP server or via stateless auto-configuration from the candidate target network, a proactive handover tunnel is established between the mobile node and the access router in the target network. The mobile node uses the acquired IP address as the tunnel's inner address. In case of proactive handover, the media is sent to target network ahead of time when the mobile is still in network 1 using path 3a. The media is then tunneled from the target network to the mobile node over the proactive tunnel. However, in this case, the in-flight data during handover is buffered in the target network for the duration of handoff and gets delivered after the mobile attaches to the new network.

5.8.4 Mobility-proxy assisted time-bound data redirection

This technique depends on the general principle of packet interception and forwarding that uses a mobility-proxy to capture the in-flight data in the previous network and forward it to the new address of the mobile in the target network. I have implemented this technique in a SIP-based environment. In case of intra-domain mobility, each visited domain may consist of several subnets. For SIP-based mobility, every move to a new subnet within a domain causes the MH (Mobile Host) to send a re-INVITE to the CH containing its new care-of-address. If the re-INVITE request gets delayed due to path length or congestion, transient media packets will continue to be directed to the old address and thus get lost. These proposed techniques reduce the in-flight data loss resulting out of continuous hand-offs within a domain and thus minimize the effect of delay contributed during application layer rebinding. In-flight packets can be redirected to a unicast or multicast address based on the movement pattern of the mobiles and usage scenario. I experiment with SIP registrar and RTP translator or NAT, the outbound proxy, and a mobility proxy to implement these mechanisms.

I provide the details about the fast-handoff mechanism in [DMC+03] and [HDS03]. I briefly describe below how I have applied these two techniques.

Figure 5.28 shows the basic framework for mobility proxy assisted media redirection.

In this specific framework, the visited network has an outbound proxy. I enhance this proxy with the ability to temporarily register visitors [Sch01]. The mobile node in the visited network obtains a temporary, random identity from the visited network and uses it as its new address-of-record to register with the registrar in the visited network. The hierarchical registration speeds up the registration, but does not address the “delayed binding update” issue using SIP’s re-INVITE feature if the CH is very far. I have taken care of the effect of delayed binding update using a mobility proxy assisted technique.

In the experiment, each subnet within a domain is equipped with a mobility proxy that has the ability to intercept the packet destined to mobile’s old address and forward it to the

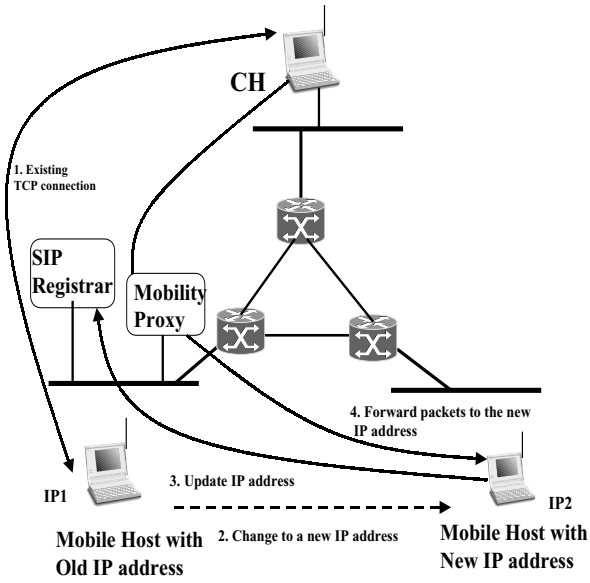


Figure 5.28: Media redirection using SIP-based mobility proxy

new destination. RTP translator [SC], [SCFJ03] provides an application-layer forwarding technique that can forward the RTP packets for a given address and UDP port to another network destination. SIP requests typically traverse a SIP proxy in the visited network, the *outbound proxy*. As the mobile moves to a new network, it sends both re-REGISTER and re-INVITE messages via the outbound proxy. This *outbound proxy* can be configured as visited registrar. Thus, the visited-network registrar receives the registration updates from the MH that has just moved, and immediately sends a request to the mobility proxy in the network that the MH just left. The request causes the mobility proxy to intercept the packets and the RTP translator forwards any incoming packets to the new address of the MH. After a set time interval or after no media packets are received by the RTP translator, the mobility proxy relinquishes this old address and removes the forwarding table entry, assuming that the re-INVITE has reached the CH.

Alternatively, the outbound proxy can use the data in the MH-to-CH re-INVITE to configure the mobility proxy in the previous network. The advantage of this approach is that the outbound proxy usually has access to the Session Description Protocol (SDP)

information containing the MH media address and port, thus simplifying the configuration of the translator or NAT. On the other hand, this outbound proxy has to remember the INVITE information for an unbounded amount of time and become call stateful, since it needs the old information when a new re-INVITE is issued by the MH. I have verified the mobility proxy-based technique by using two different tools, namely rtptrans [Sch] and Linux iptables [Her00] that help direct the transient traffic from the previous subnet to the new one. Figure 5.28 shows how SIP registrar and mobility proxy interact with each other to forward the in-flight packets to the new network.

5.8.4.1 Experimental analysis

In the experimental testbed as shown in 5.29, RTP translators are associated with the mobility proxies in the respective subnets. Mobility proxy in each of these subnets intercepts the traffic meant for the mobile host and RTP translator forwards it to the new address of the mobile host after capturing it. This is achieved by a combination of SIP-CGI (Common Gateway Interface) and SIP REGISTER [LS99b].

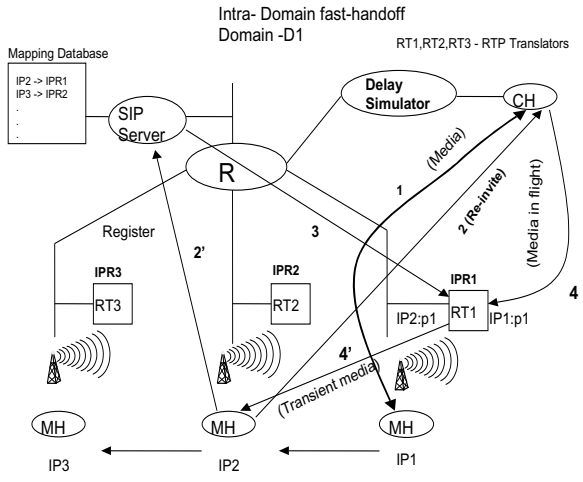


Figure 5.29: Experimental testbed for mobility proxy

I provide an analysis of how packet loss is minimized by means of forwarding mech-

anism using RTP translator. This analysis is based on the experimental testbed shown in Figure 5.29. The time taken for complete subnet movement including IP address acquisition and layer 2 movement is T_s. Time taken for Re-INVITE to reach CH is T_i (mostly decided by the distance factor). Time taken to process Re-INVITE at CH is T_p, time taken to register at SIP proxy is T_g, time taken for SIP registrar to forward the packet after capturing is T_f. Packet generation rate at CH is P_r packets per second. Thus, total number of packets lost during handoff using SIP registration and RTPTrans is $P_{rt} = (T_s+T_g+T_f)\times P_r$.

As part of the experiment, I delayed re-INVITE signal to simulate distance between CH and MH after the mobile has moved to the new network. Both VIC and RAT tools [SHK+95] were used to measure the delay performance of audio and video streaming traffic, respectively. I delayed the traversal of re-INVITE signals by 100 ms, 200 ms, 500 ms, 1 sec, 2 sec and 3 sec to emulate the distance between the visited network and home network. This technique also shows how RTP translator helps the media redirection and mitigates the packet loss during mobile's movement.

In an earlier experiment [DVC+01], I have measured processing time for re-INVITE at the CH to be about 100 ms. Complete SIP registration takes about 150 ms. It takes about 200 ms to complete the subnet movement and IP address acquisition including the layer 2 detection. In the current experiment, I measured the packet forwarding delay due to redirection at the registrar to be less than 1 ms when iptables-based NAT approach was used, whereas RTP translator approach added 4 ms of delay. The additional delay is due to application layer redirection used by RTPtrans. In the current 802.11-based experimental environment, the mobile lost about 15 packets due to layer 2 delay, IP address acquisition delay, re-INVITE processing delay, registration and packet forwarding delay.

Figure 5.30 compares the efficiency of SIP-based optimized handoff approach using a combination of mobility proxy and RTP translator with a SIP-based mobility protocol without fast-handoff technique. As the figure shows relative packet gain at the mobile node increases as the distance measured in number of hops increases between CH and MH for a

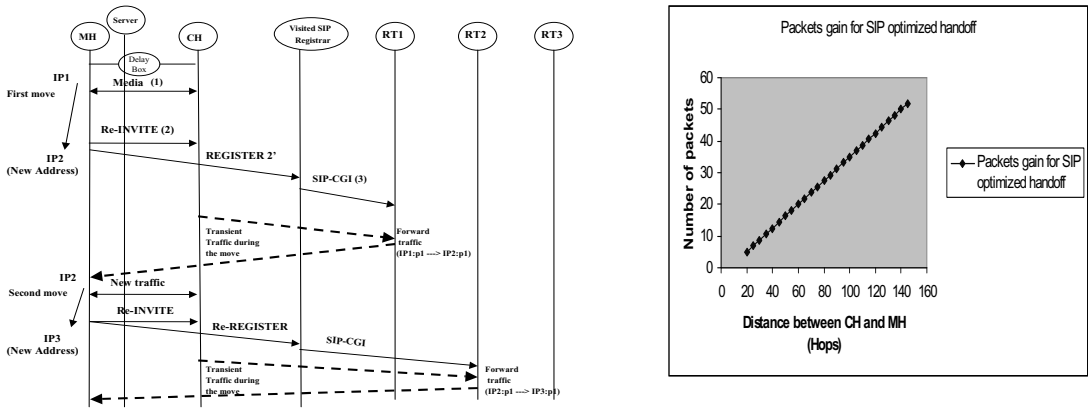


Figure 5.30: Reducing packet loss with localized media redirection

given packet generation rate.

5.8.5 Time-bound localized multicasting

Locally scoped multicast technique allows to multicast in-flight data to the neighboring networks during handoff. It helps to avoid packet loss when the MH can predict that it is about to move to one of the new subnets within the neighboring network. I have applied this mechanism to reduce the packet loss during media delivery for both network layer and application layer mobility protocols. These mechanisms are described below.

5.8.5.1 Network layer-based

I have applied my proposed technique to reduce packet loss for IDMP and have described the details in [DDM+02].

Figure 5.31 shows the architecture where I have applied this time-bound multicasting mechanism to support fast-handoff for network layer mobility.

In this case, the mobility agent (MA) encapsulates the unicast packets in a multicast address and sends it to the neighboring base stations pro-actively. The base stations buffer these packets and upon mobile's arrival, the unicast packets are delivered to the client. I

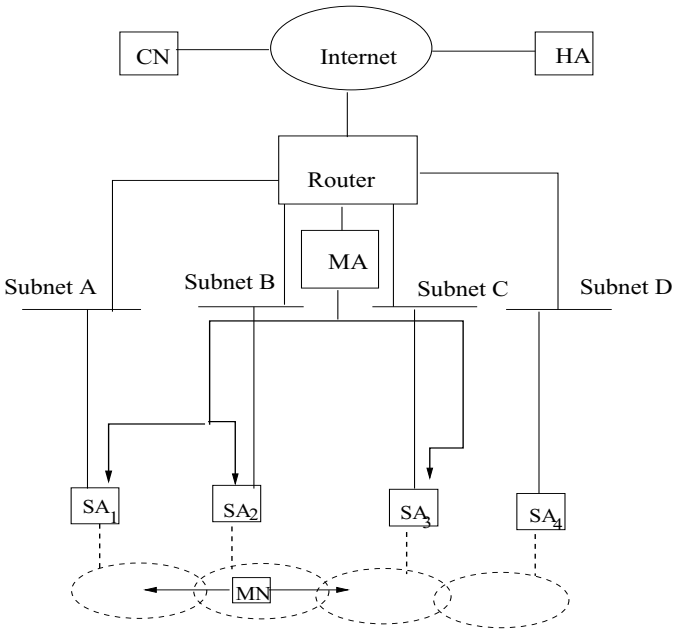


Figure 5.31: Time-bound multicasting for IDMP

briefly describe below how fast handoff is achieved by applying these optimization techniques in the architecture shown in Figure 5.31.

It is assumed that a layer-2 trigger will be available (either to the MN or to the old BS) indicating an imminent change in connectivity. Layer 2 trigger is an indication from the lower layer to trigger an action at layer 3. As shown in Figure 5.32, an MN moves from SA_2 to SA_3. To minimize the service interruption during the handoff process, the mobile node or the old SA (SA_2) generates a *MovementImminent* message to the MA serving the MN. Upon reception of this message, the MA multicasts all inbound packets to the entire set of neighboring SAs (SA_3 and SA_1) in this case). Each of these candidate SAs buffers such arriving packets in per-MN buffers, thus minimizing the loss of in-flight packets during the handoff transient. When the MN subsequently performs a subnet-level registration with (SA_3) this subnet agent (SA_3) can immediately buffer all such buffered packets over the wireless interface without waiting for the MA to receive the corresponding Intra-domain Location Update.

I highlight some key benefits of the proposed techniques compared to other existing localized multicasting techniques.

The proposed technique uses network-controlled, network or mobile initiated handoff technique. It is the MA which decides the set of target BSs to which in-flight packets are multicast. This is especially useful in scenarios where the MN may be in contact with multiple BSs and is unable to specify the future point of attachment exactly. While current cellular networks use a network-controlled handoff technique (where the base station controller (BSC) determines the candidate BS based on link-layer measurements supplied by the MN or BS), the IP mobility model is typically MN-driven, with the MN selecting an FA from a list announced via agent advertisements. The proposed technique preserves the network-controlled handoff model for future IP-based cellular networks, without compromising the MN's ability to select such fast handoff support.

Unlike other multicasting based fast-handoff approaches, the proposed multicasting scheme prevents unnecessary wastage of wireless bandwidth, since a base station does not unilaterally transmit all arriving multicast packets over the wireless interface. Such pro-actively multicast packets are temporarily buffered by a BS in per-user buffers and forwarded to the MN over the wireless interface only if the mobile happens to register at that BS. In case the MN does not register at a particular SA, the buffered packets are discarded after a specified maximum time interval.

I briefly describe here the implementation approaches. I have described the details of the mechanism and its pros and cons with other approaches in [MDD+02]. The use of locally scoped multicast is only effective if the MH can quickly acquire a multicast address and there is a multicast infrastructure available. Additional encapsulation overhead is an associated trade-off.

5.8.5.2 Fast-handoff implementation

For a prototype implementation, I use IP multicast to pro-actively distribute such packets to possible points of attachment. This mechanism requires only one multicast group per neighbor set; all the BSs that are neighbors of a specific BS are members of this multicast

group. Since a single BS can be a neighbor of multiple BSs, each BS can indeed be a member of multiple multicast groups. This approach does not require the establishment of dynamic multicast groups for individual MNs. The membership of the neighborhood set is also not dynamic: given a fixed network topology, the set of neighboring BSs stays constant. Each BS is thus permanently subscribed to one or more multicast groups, each of which always has a well-defined distribution tree. Accordingly, the fast handoff scheme does not require a BS to dynamically join or leave a group, and hence, does not suffer from any transient tree-establishment latencies.

Figure 5.32 shows the sequence of protocol flow among network components and how the packets are encapsulated and decapsulated as the mobile moves from SA_{old} to SA_{new}.

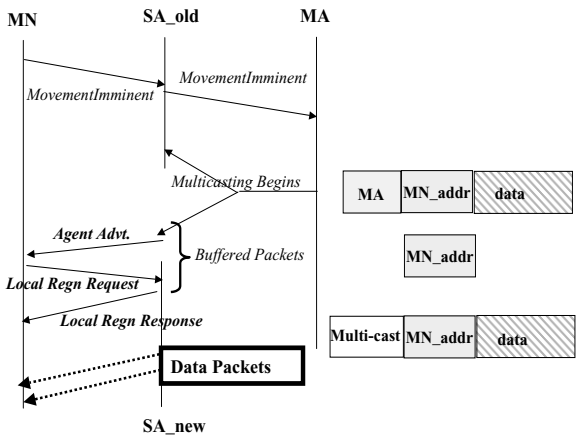


Figure 5.32: Scope-based multicast flow data redirection

On receiving a *Movement Imminent message*, the MA encapsulates an in-flight packet and then tunnels it to the appropriate multicast address. (For such multicast forwarding, the MA does not perform the conventional tunneling towards the current LCoA). On receiving such a tunneled multicast packet, each SA will first decapsulate the outer-most header. It then buffers the decapsulated packet in a per-user buffer, using the destination address in the inner-header (which is unique to a specific MN) as an index. When a mobile node

subsequently performs a subnet-specific registration with an SA (say SA3 in Figure 5.31), the SA can then forward any cached packets to the MN before the intra-domain location update process is complete. Simple calculations indicate that even a small user buffer is effective in reducing the loss of in-flight packets. For example, if the intra-domain update latency (L) is 200 ms, and the incoming traffic rate (R) is 144 Kb/s, then a buffer size of (L*R) 3.6 Kbytes is able to protect against buffer overflow due to multicast packets transmitted during the handoff.

5.8.5.3 Application layer-based approach

In case of application layer mobility, the mobile informs the visited registrar or B2BUA of a temporary multicast address as its contact address in the SDP. Once the MH has arrived in its new subnet, it updates the registrar or B2BUA with its new unicast address, while it continues to receive the in-flight data over the multicast address. I have described the details of how locally scoped multicast address can be used to reduce packet loss during handoff in [DMC+04]. Multicast agent may co-locate with the first-hop router or can co-exist with the B2BUA or SIP proxy. Using scoped multicast is only effective if the MH can quickly acquire a multicast address that can be used as part of SDP to update the back-to-back-UA (B2BUA). Figure 5.33 shows how this forwarding technique can be applied to support fast handoff using application layer mobility. In this case, the mobile sends a Re-INVITE to the B2BUA with the local scoped multicast address in the SDP when the handoff is imminent. Thus, media traffic is multicast to both the neighboring subnets. After the handover to the new network, the mobile sends a new Re-INVITE with the unicast address in the SDP. This unicast address is the care-of-address of the mobile after it has moved to the new network. Thus, the effect of binding update delay is minimized by redirection of the in-flight packets during handoff by way of reducing the packet loss.

There is a likelihood that duplicate packets are received during the *mobile′s* movement between the subnets. RTP packets have their own sequence numbers associated and thus

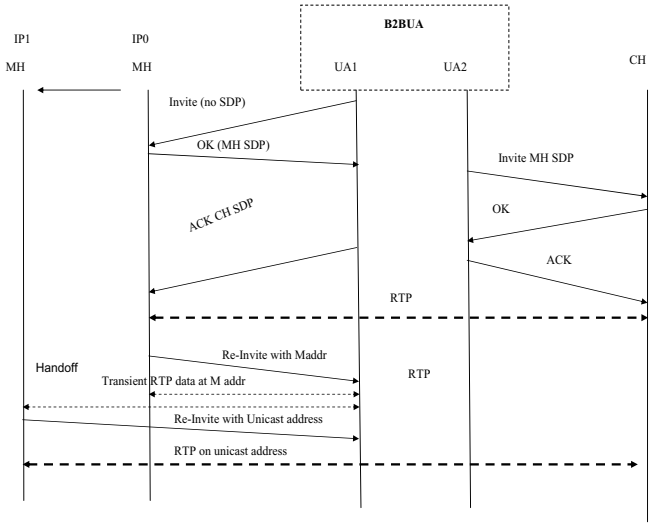


Figure 5.33: Data redirection using multicast agent

these packets can be reordered. Although mechanisms similar to described by Perkins and Wang [PW99] can be adopted to take care of duplicate non-RTP-based traffic.

5.9 Media buffering

In Chapter 3, I have introduced buffering as one of the sub-processes of media routing process during handoff that help to reduce the packet loss at the cost of added packet loss. Although media redirection techniques help redirect the in-flight packets caused due to delayed binding update, some packets may be lost during link-layer handover. Thus, it is essential to mitigate the effect of link layer handover delay by reducing the packet loss. Bicasting or buffering the transient packets at different parts of the network can be applied to minimize or eliminate the packet loss. However, bicasting alone cannot eliminate packet loss if link-layer handover is not seamless. Although buffering can mitigate the effect of layer 2 handoff by reducing or eliminating the packet loss, it introduces an additional one-way delay for the in-flight packets. While this additional end-to-end delay may not affect the streaming traffic, interactive traffic, such as VoIP application cannot tolerate the jitter resulting out of variable one-way delay. Ability to control the buffer dynamically provides

a reasonable trade-off between the delay and packet loss that is within the threshold limit to support real-time communication. As part of my thesis, I have developed a dynamic buffer control protocol (BCP) [DvF+06] that can provide a dynamic buffering mechanism based on the duration of handoff and placement of buffers at the edges of the networks.

The proposed technique introduces a solution beyond the application end points by providing a per-mobile packet buffer at an access router or network entity (Buffering Node) near the edge of the network where the mobile is moving away from or moving towards. Packets that are in flight during the handoff period get buffered in the Buffering Node (BN). When handoff completes, the buffered packets are flushed out and forwarded to the MN in its new location. This approach provides zero packet loss for all packets destined for the MN that have reached the BN. The solution also describes a buffering scheme that enables the MN to have control over the behavior of the BN to help reduce the overall handoff delay. Outgoing packets sent by the MN during the handoff period can also be lost during the handoff process. In such a case, a BN can also be implemented on the MN itself to provide buffering for the egress packets during the handoff period. Having a buffering node functionality both in the MN and the network edge provides bi-directional buffering during handoff and will reduce packet loss in both the directions.

The BN may also be located within the access point specifically to assist an MN that performs active scanning. During active scanning on channels different from the currently associated access point, the mobile can no longer receive packets from that access point. In the current implementation, the MN uses power saving mode to signal the access point and allows it to start buffering on behalf of the MN. Implementing buffering functionality on the access point itself also provides the same functionality with better control on the buffering period and buffer size.

In this section, I describe some principles that can be considered while designing buffering protocol to support handoff. I highlight some related papers that introduce buffering techniques to take care of packet loss. I then introduce my proposed optimization tech-

niques and elaborate on the dynamic buffer control protocol that reduces the packet loss at the expense of added delay. Finally, I demonstrate the experimental results using two different buffer control approaches that I proposed.

5.9.1 Key principles

These are the key principles that need to be considered while designing buffer control protocol for mobile's handoff.

1. Buffering in-flight packet during mobile's handoff can eliminate the packet loss inclusive of layer 2 handoff delay.

2. Added delay due to network buffering may not be suitable for low latency applications, as "delayed packet" beyond certain threshold is considered lost packet. Buffering can also help TCP type traffic without compromising the data throughput and streaming traffic (e.g., IPTV).

3. Buffering period can be adjusted based on the handoff interval.

4. Media can be buffered at any part of the network, such as at the source, edge routers, core network or at the mobile.

5. In most cases, buffering is useful for proactive handoff, where the packets are buffered before the handoff begins and are flushed after the handoff is over.

6. Packet generation rate at the source, handoff period, time taken to signal the buffer to flush, packet transmission time are some of the parameters that affect the optimal buffer length at the edge router.

7. While the overall buffering period is influenced by the handoff delay, buffering affects end-to-end delay, number of packets delayed, and the jitter.

8. Jitter observed due to buffering of packets at the router node can be compensated by proper playout buffer at the mobile.

5.9.2 Related work

Moore et al. [Moo04] and Krishnamurthi et al. [KCP01] have developed buffering techniques for mobile IPv6. Khalil et al. [KAQ+99] describe a mobile IPv4 buffering protocol that resembles the method proposed in this thesis. These proposals define extensions to mobile IPv4 and mobile IPv6 protocol, respectively to support buffering in the network during a handover period. Moore et al. describe the use of adding a P-bit in the mobility header of BU (Binding Update) and LBU (Local Binding Update) messages. Proposal by Krishnamurthi et al. is very similar to the proposal by Khalil et al. [KAQ+99]. However, that technique adds discovery feature for buffering capability and takes advantage of IPv6 router advertisement to check the buffering capability of a network.

There are alternate mechanisms that reduce packet loss without the use of any buffer management protocol but depend heavily on the cooperation of the end clients. Most multimedia applications resort to playout buffers, FEC (Forward Error Correction) [RS99], RTCP-based feedback [OWS+06] and other stream repair techniques [PH98] in order to minimize the effects of packet loss or to reduce the jitter. However, existing end-system assisted solutions may not be appropriate in a wireless medium where the bandwidth is limited and the end hosts are separated by a long distance. These mechanisms have not been applied to take care of packet loss during handoff.

In layer 2, there is an existing method that uses the power management functionality of IEEE 802.11 to avoid packet loss while the MN (Mobile Node) is actively scanning [RL03] the channels. In this method, the MN signals the current access point that it is entering into sleep mode and the access point attempts to buffer packets for the MN until the MN wakes up. However, this method cannot be used for buffering packets during a handover because the method assumes that the MN continues to be associated with the access point

after it wakes up to stop buffering and the applicability is limited because the method does not carry additional information such as traffic flow identification information, buffer size and buffering period which might be required to meet particular QoS requirements of the mobile.

The existing proposals are tightly coupled with specific mobility management protocols such as Mobile IPv4 and Mobile IPv6. In contrast, my proposed buffering method can work with any mobility management protocol by allowing the buffering control mechanism to be defined as a separate protocol. In the existing proposals, location of buffering node is limited to mobility agents such as home agent and mobility anchor point. In contrast, the proposed method provides more flexibility on location of buffering node. In the existing proposals, forwarding of buffered packets to the mobile node after completion of the handover period depends on the forwarding behavior of the mobility agent that is part of the mobility protocol. In contrast, the proposed technique defines its own tunnel establishment mechanism used for forwarding buffered packets to the mobile node to provide perfect independence of mobility management protocols. The proposed methods also define detailed queuing and forwarding mechanisms for the buffering packets as well as detailed behavior in erroneous situations are defined, while such details are missing in the existing proposals.

5.9.3 Protocol for edge buffering

I have designed and implemented a dynamic buffering scheme [DvF+06] that ensures zero packet loss at the expense of additional end-to-end delay that can be controlled dynamically. Figure 5.34 shows four different scenarios that illustrate how buffering techniques can be applied at different parts of the network. The figure shows how the buffering can be applied to previous access router, next access router, at the source or at the destination. The buffering scheme is used in conjunction with the existing mobility protocols, or can be used as an independent network or link layer access mechanism.

Ability to control the buffer dynamically provides a reasonable trade-off between delay

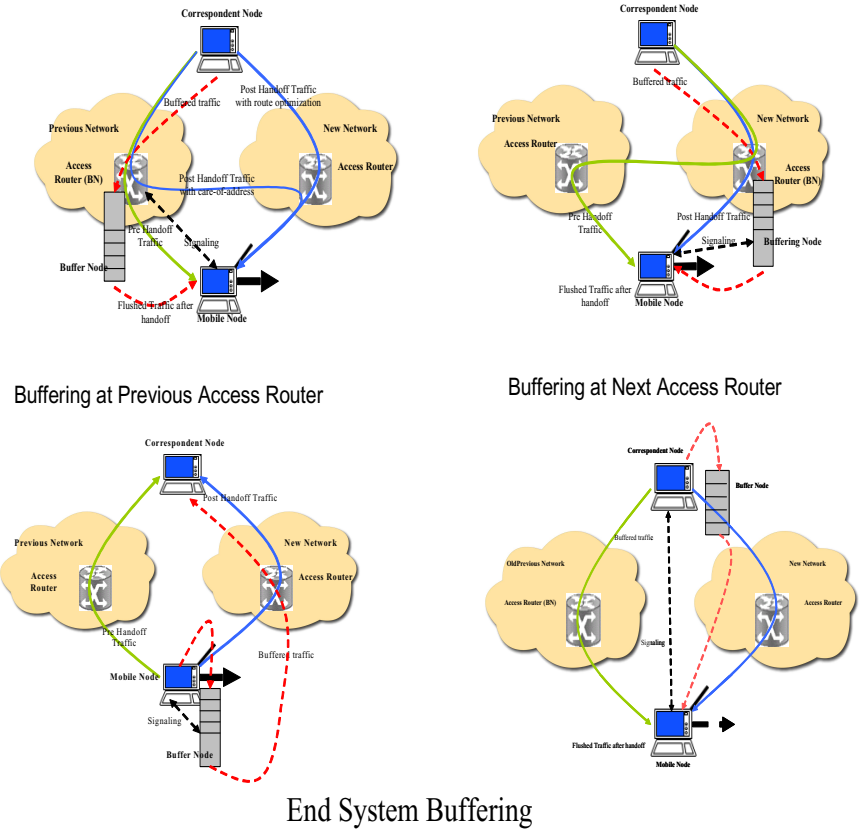


Figure 5.34: Buffering alternatives

and packet loss within the threshold limit for real-time communication. I have experimented with two kinds of buffering schemes, namely *time limited buffering* and *explicit signaling buffering*. In case of *time limited buffering technique*, the mobile node and buffering node can negotiate a buffering period that is conveyed to the buffering node during the initial setup signal. The buffering node buffers the packets for the duration of the buffering period as defined in the initial control message. In case of *explicit signal buffering*, the buffering period is equivalent to the total handoff period and additional time taken to flush the buffer after the handoff has taken place. I have experimented with both types of buffering schemes using different traffic rates and buffering periods in conjunction with media independent pre-authentication mechanism. Average inter-packet delay during handoff,

packet loss and average number of packets buffered were calculated for each case. In case of explicit signaling, the number of packets buffered were dependent upon handoff delay and packet generation rate. Time limited buffering on the other hand introduces higher probability of packet loss resulting out of buffer overflow.

5.9.3.1 Protocol details

In this section, I briefly describe the protocol details of the dynamic buffering mechanism. The BCP (Buffer Control Protocol) is used by the MN to request buffering services at the BN (Buffering Node). It is a simple and reliable messaging system composed of request and answer signal pairs. The BCP may be defined as a new protocol or as extensions to existing protocols such as PANA, SIP and Mobile IPv4 or Mobile IPv6 or link layer protocols. For example, in Mobile IPv6, it may be possible to define a new mobility option in binding update or acknowledgement message exchange that carries the BCP in TLV format. In PANA [FOP+08], it is possible to define BCP AVPs that can be appended to the PUR (PANA Update Request) and PUA (PANA Update Response) message exchange. Other methods may be employed as long as the requirements of the BCP signaling can be accommodated. In all these cases, delivery and encoding of BCP signals may become specific to each protocol that carries BCP.

Signaling messages for BCP

In this section, I describe some of the signaling messages that are used to take care of buffering. As a rule, request signals are sent from the MN to the BN and answer signals are sent by BN to MN in response to a request signal. The BN should never generate a request signal. Request signals carry parameters regarding the request and answer signal contains result codes. Reliability is supported by using transmission timeouts, re-transmission and error handling behavior. I describe below some of these signals.

BReq[initial] and BAns[initial]

These signals are initially exchanged between MN and BN. It is used to establish the

buffering service. These signals have the following format.

BReq[initial] = id, bp, tc, bsz, p

BAns[initial] = id, bp, bsz, rcode

These parameters are described below.

```
id: MN Id used to uniquely identify the MN to the BN. This
can be the source address or MAC address of the MN.
bp: Buffering period
tc: Application specific traffic to be classified and
buffered
bsz: Suggested buffer size to be allocated
p: FP for EOS, valid values are drop,forward
or drop with signal
flag {m}: Request flags
m: if set bsz is mandatory and cannot be negotiated
rcode  Result code provided by BN
```

BReq[ext] and BAns[ext]

These signals are exchanged after establishing buffering service and before or after the *MN*'s handoff period. They are used to extend the parameters of the buffering service. Here are some of the parameters that are used to provide this buffering service.

BReq[ext] = id, seq, bp, bsz, p, coa

```
id - MN id sent in the BReq[initial]
seq - Signal sequence number
bp - Additional buffering period, maybe zero (0)
bsz - Additional buffer size, maybe zero (0)
p - new FP for EOS, valid values are drop, forward or
drop with signal
coa - current CoA of the MN
```

BAns[ext] = id, seq, bp, bsz, rcode

BReq[stop] and BAns[stop]

These signals are exchanged to stop the buffering service. Following are some of the parameters

BReq[stop] = id, p, coa

BAns[stop] = id, rcode

```
id - MN id sent in the BReq[initial]
seq - Signal sequence number, must match BReq[ext]
hp - New buffering period for this service
bsz - New buffer size allocated for this service
rcode - Result code provided by BN


id - MN id sent in the BReq[initial]
p - Termination FP for EOS, valid values are drop, forward or
   drop with signal
coa - current CoA of the MN
```

Service Attributes

The BCP also creates service attributes (state information) within the BN. These attributes include the following:

1. MN Id (id)
2. Buffering period (bp)
3. Negotiated buffer size (bsz)
4. Traffic classification (tc) parameter
5. FP, current EOS flushing policy
6. Last extension request sequence number
7. Current MN CoA
8. Previous MN CoA

The attributes should be allocated during the request phase. The values of the attributes are updated by the BN upon receiving valid request signals or other local events. The

```
id  MN id sent in the BReq[initial]
rcode  Result code provided by BN
```

attributes lifetime is limited to the duration of the service. If a positive or negative EOS is met, the BN should release resource occupied by these attributes.

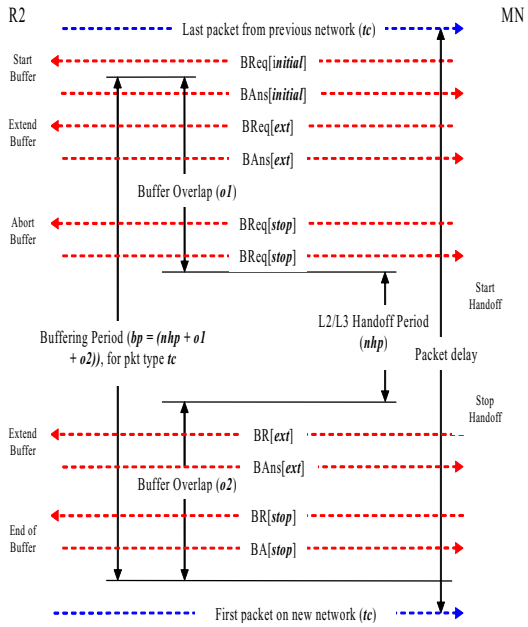


Figure 5.35: Protocol flow for buffer control protocol

The protocol flow shown in Figure 5.35 provides a general sequence of the signal exchanges as it relates to the MN's handoff, traffic classification and buffering period. The variables $o1$ and $o2$ define the buffering overlap period before and after the handoff although MN and BN may still have the connectivity. This will happen if the packets get buffered before the handoff starts and keep getting buffered throughout the handoff period and after the mobile's handoff.

5.9.4 Experimental results and analysis

In this section, I describe the experimental results for both types of techniques: *Time limited buffering* and *Explicit buffering*. I have used this mechanism in conjunction with media independent pre-authentication technique where the target access router is used as the buffering node (BN). Figure 5.36 shows a typical scenario where I have experimented this

technique with media independent pre-authentication. Figure 5.37 shows the protocol flow when PANA is used as buffer control protocol.

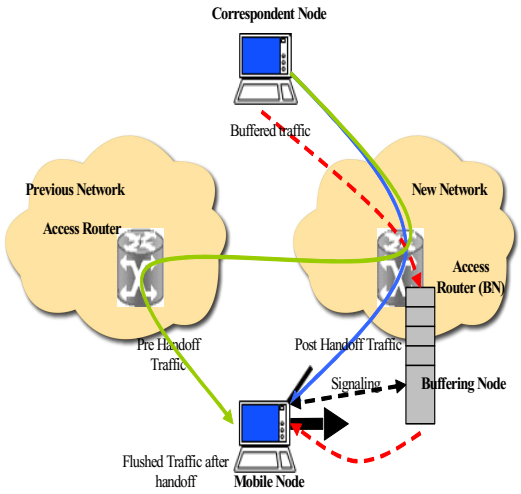


Figure 5.36: Buffering with media independent pre-authentication

Table 5.5 and Table 5.6 show values of average packet delay, average packet loss during handoff and average number of packets buffered for time limited and explicit buffering techniques, respectively. Average packet delay is defined as the delay between the last packet before the handoff and first packet after the handoff.

Table 5.5: Time limited buffering

Traffic rate (pkts/sec)	X (ms)	Y (ms)	Packet delay (ms)	Packet loss	Average packet buffered
70	0	N/A	37	0	2.5
80	0	N/A	42	0	3
90	0	N/A	44	0	3.5
100	0	N/A	45	0	4

The current solution is implemented using kernel queue module that hooks into linux netfilter's QUEUE handler. The new module is called *ipmparb* (IPv4 MPA router buffer). This module has the following advantages:

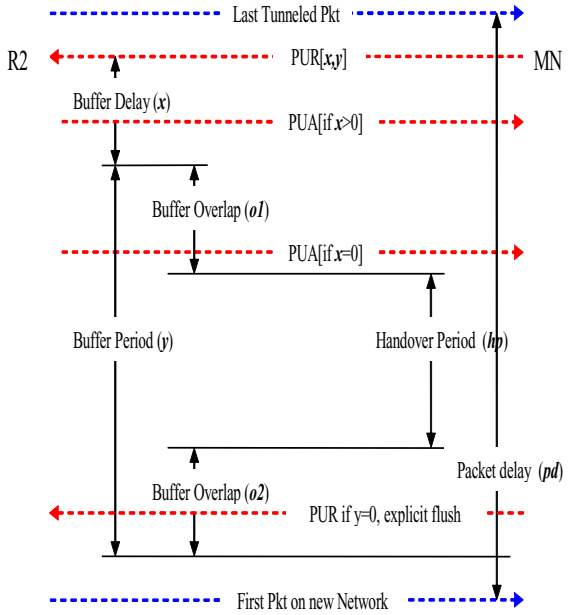


Figure 5.37: Protocol flow using PANA as BCP

1. Packet classification is done by iptables so the module is much simpler. It will simply rely on iptable's packet classifier with the *ipmparb* as the target.

2. Implementation is efficient since packets are routed to the module in *skbuff* objects so no copying is done. *ipmparb* simply queues the *skbuffs* without modification.

3. Implementation is very fast since *ipmparb* is a kernel level module that becomes part of the ip routing stack. No additional socket mechanism is required.

4. Easily meets the requirement of maintaining packet sequence since all packets that must be buffered have to pass through this module. So when buffered packets need to be flushed, they can be transmitted first prior to allowing newly arrived packets to be transmitted.

5. *ipmparb* can use any queuing discipline we require. At the moment, a simple FIFO queue is used.

Table 5.6: Explicit buffering

Traffic rate (pkts/sec)	X (ms)	Y (ms)	Packet delay (ms)	Packet loss	Packets buffered
70	0	30	27	0	19
	0	20	29	0	3
	0	10	12	0	1
80	0	30	28	0	3
	0	20	33	0	3
	0	10	15	0	1
90	0	30	69	0	3
	0	20	46	0	3
	0	10	11	3	1
100	0	30	69	0	10
	0	20	46	0	4
	0	10	11	3	1

User level interaction is limited to simple control events so one can use existing user level commands that can pass control events to kernel modules. The most ideal scenario is to reduce overlapping period $o1$ and $o2$ to zero though this is not possible for all practical purposes. A negative value for $o1$ and/or $o2$ will result in packet loss.

This means that hp (handover period) is not encompassed within y. Based on the experiments, $o1$ and $o2$ can be fine-tuned using x and y where y is based on hp (handover period) and x is based on average round trip time. In addition, another alternative is the use of an explicit flush message instead of fine-tuning y (noted as PUR if y=0 in Figure 5.37). The experimental results are based on packet generation rate of 70, 80, 90 and 100 packets/sec. The rate is based on a value that is greater than the codec rate of RAT (Robust Audio Tool) used in the MN e.g.,60 packets/sec. Also, switchover process to router R2, e.g., deleting the tunnel, updating the ARP cache etc. always occurs during o1, immediately after R2 begins buffering. This switchover period is very small (average about 0.300 to 0.500 ms) so it is not considered in the figure above.

Table 5.5 summarizes the average results of first four samples that use madwifi driver and IEEE 802.11 netgear card with *time-limited buffering* approach. Modification involves

only layer 2 optimization that avoids scanning with no other functional changes. Four different packet generation rates are experimented. The average hp (handover period) is 10-16 ms. This includes L2 and L3 related delays that happen in sequence. Average L2 delay is 4-8 ms followed by average L3 delay of 6-8 ms. Since the IP address is obtained beforehand, the layer 3 related delay is the delay associated with assigning the previously obtained IP address to its physical interface. Bulk of the handoff delay is avoided because of many of the handoff operations done pro-actively. The buffering node in the experiment is the next access router that is discovered before the mobile moves to the new network. Because the hp value is very small, y can be fine-tuned to its minimum of 10 ms without incurring packet loss. Value of x has been kept to zero as it introduces additional delay. When we used RAT (Robust Audio Tool) as the media agent, it generates an average 60 pkts/sec, there is no discernible loss in the audio sample and almost always have no packet loss.

Table 5.6 shows the experimental results under the same environment (madwifi driver and IEEE 802.11 netgear card) but using *explicit buffering* approach. The average hp is 10-16 ms. Using explicit signaling between MN and R2 to flush the buffer, it is guaranteed that no packet loss will occur compared to using a value of y that has the possibility of having $o2 < 0$ resulting in packet loss. The price is additional delay. As an example, when using an ideal y value at 70 pkts/sec the average delay is only 12 ms as compared to explicit signaling which is 36 ms. Similar to the case of time-limited buffering, experimental results using RAT as a media did not produce any discernible loss in the audio and is guaranteed to have no packet loss.

I describe the detailed break-down of handoff delay for an experiment with 100 packets/sec traffic rate, 1024 bytes of packet size and explicit signaling method. Total handoff (handover) delay is about 12.5 seconds. Layer 2 (L2) delay with madwifi driver is about 4.8 ms, and L3 configuration time is about 0.5 ms (L3). Processing delay (a) for buffer request at PAA =5.699 ms, switch over period at PAA (b) is 0.46 ms that includes tunnel setup and

ioctl calls. Processing delay at MN (c) to send stop request is 6.788 ms. Processing delay (d) to flush the packets at the buffer is 4.626 ms. Flushing period (e) at PAA is 0.205 ms. Thus the total handover delay (hp) D = (L2/L3/c)+d +e. Thus, it appears from the above that the handover period is a fraction of the total packet delay incurred. Explicit signaling method adds to the total packet delay because of the delay associated with the flushing where as time limited signaling increase the probability of packet loss.

5.9.5 Tradeoff analysis between buffering delay and packet loss

End-to-end delay of any specific packet and the delay between last packet in the previous point of attachment and first packet in the new point of attachment are most important. Buffering mechanism while reduces the packet loss, it also introduces additional delay to both end-to-end packet delay for the in-flight packets and handoff delay. A packet during handoff that would have got lost otherwise gets buffered in the buffering node for a certain period of time that is determined by the handoff delay and time taken to flush the packets from the queue at the new point of attachment. Yemini [Yem83] provides a trade-off analysis between delay and packet loss. This paper also stresses the fact that as soon as the buffer threshold is exceeded any newly arriving packets cause the first packet of the queue to be lost. Although this paper focuses on the queue at the sender side it could generally be applicable to the general theory of the buffering protocol described in this paper. In case of *explicit signal buffering*, buffering period is equivalent to the total handoff period and additional time taken to flush the buffer after the handoff has taken place. Total number of packets stored in the network buffer depend on the buffer length, transmission time and packet generation rate at the source. Packets arrive in the buffer at a regular interval. But when these packets are flushed out of the buffer after the handoff, all the buffered packets are flushed out at the same time without any inter-packet gap. Although this avoids packet loss, the mobile is subjected to a spike since these packets arrive on the mobile almost instantaneously. The in-handoff packets that are buffered in the edge router are subjected

to an increased amount of delay compared to pre-handoff and post-handoff packets. But each consecutive in-handoff packet is subjected to a different amount of delay since these packets spend different amount of time in the buffer. Later packets are subjected to lesser amount of delay compared to the packets that got in first.

End-to-end delay for in-handoff packets, delay between the last packet in the old network and first packet in the new network and total number of packets affected due to handoff are the important parameters that need to be considered to support real-time communication. Packet generation rate at the source, handoff period, time taken to signal the buffer to flush, packet transmission time are some of the parameters that determine the optimal buffer length at the router. While the overall buffering period is influenced by the handoff delay, it affects the end-to-end delay, number of packets delayed, and the jitter. However, the jitter observed due to buffering at the router node can be compensated by the play-out buffer at the mobile. Figure 5.38 shows how the packets during handoff get affected because of the buffering at the edge router. It shows that the packets during handoff are subjected to jitter because of increased end-to-end delay due to buffering.

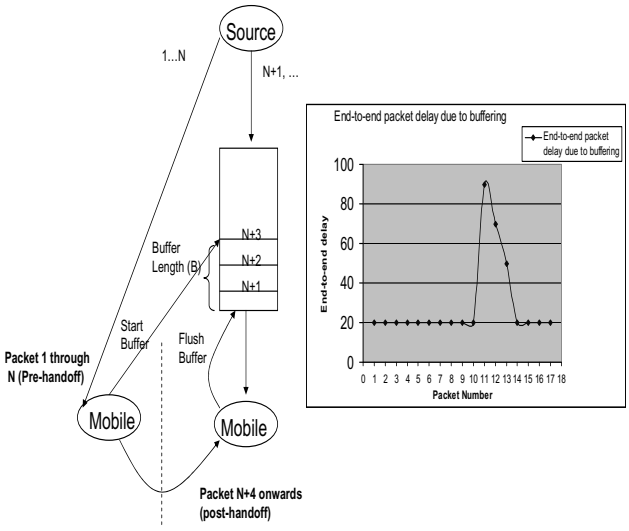


Figure 5.38: Effect of edge buffering on in-handoff packets

5.10 Route optimization

In Chapter 3, I have defined different processes that are part of media re-routing process and illustrated how these affect the handoff delay and packet loss during a mobility event. Triangular routing, encapsulation and decapsulation processes associated with any mobility protocol affect the performance of real time traffic due to associated transport delay for signaling, data, and encapsulation overhead. Route optimization is the process of optimizing the route between the communicating hosts by eliminating encapsulation and decapsulation processes and maintaining the direct route.

In this section, I first describe the key principles that need to be considered for route optimization while optimizing packet transport delay and data overhead. Then, I describe the related work that describe the route optimization techniques for different mobility protocols. I then describe four route optimization mechanisms that I have developed based on some of these principles and illustrate the experimental results. These mechanisms optimize the signaling and data traversal by maintaining the direct path between the communicating hosts and avoid any associated encapsulation overhead. Since Mobile IP inherently suffers from this route optimization problem, I experimented with a few of these optimization techniques and then compare the results with that of MIPv4.

5.10.1 Key principles

The following are some of the key principles that can be applied to optimize the data path between the end hosts contributing to reduced end-to-end delay.

1. Maintain the direct path between the communicating hosts. If a protocol allows the mobile hosts to update each other's identifier directly without the help of any anchor agent in the middle of the network it reduces the data traversal path after the handoff.

2. Limit the media traversal between the communicating hosts within the local domain when both the communicating hosts are away from home and are visiting in the same

domain.

3. Split the media and signaling path to avoid the data transmission via the home network. Only the mobility signaling such as binding updates are sent to the home agent while media traversal is limited to the local domain.

4. Modifying the source and destination addresses at the end host before the data is passed onto the application at the end host. This helps to maintain the direct path.

5. Dual anchoring mechanism that allows the mobile to use different address for signaling and media. This method can be applied in case of IPv6 networks and are applicable when both the communicating nodes are away from home. Although signaling needs to travel to the home network, it is important to confine the traversal of media in the visited network by avoiding the longer path traversal to the home network.

5.10.2 Related work

The IETF has addressed these issues by proposing route optimization support for MIPv6 [JPA04]. However, route optimization for MIPv4 never got standardized but various forms of route optimization have been proposed by others [WCL+02]. Recently, the NETEXT (Network-based mobility Extensions) working group within the IETF has included route optimization as one of its working group charter items and is defining the problem statement and solutions for route optimization for Proxy MIPv6.

I describe below a few of the route optimization techniques based on the key principles explained earlier and demonstrate the associated experimental results.

5.10.3 Maintain a direct path by application layer mobility

SIP-based terminal mobility [WS99] performs binding updates by application layer signaling. It reduces one-way packet delay by avoiding the triangular routing that is inherently

present in Mobile IPv4 [Per02c]. Both SIP-based mobility and MIPv4 have been briefly introduced in Chapter 2 as part of introduction to mobility protocols. I augmented the SIP-based terminal mobility with a complete set of hand-off operations to support subnet and domain mobility [DVC+01] and compared the effect of triangular routing on the packet transmission.

Initially, I compared the latency for SIP-based mobility and mobile IP for different packet sizes during the subnet handoff. By using SIP-based mobility protocol, I could obtain 50 percent one way latency improvement for real-time (RTP/UDP) traffic thus providing a reduction in latency from a baseline of 27 ms to 16 ms for large packets and a 35 percent utilization increase due to avoidance of additional IP-in-IP encapsulation. Figure 5.39 shows the reduction in end-to-end packet delay for SIP while compared to MIP for various data packet sizes as obtained from NS2-based simulation [Sim05] and lab experiments. The simulation results demonstrate how a direct signaling path between CH (Correspondent Host) and MH (Mobile Host) minimizes end-to-end delay of the data packet and reduces data overhead. These experimental results demonstrate that by maintaining the direct signaling path between MH and CH, the end-to-end delay of the data packet and packet loss can be optimized for secured inter-domain mobility.

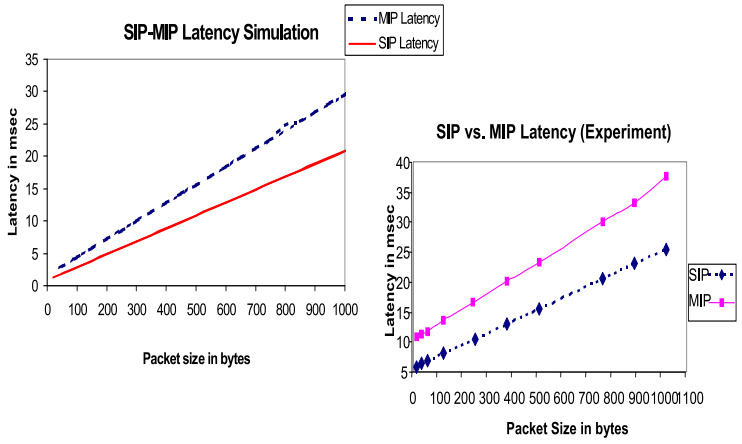


Figure 5.39: One way data transfer delay for SIP and MIP-based protocols

5.10.4 Interceptor-assisted packet modifier at the end point

Mobile IP with location register (MIP-LR) [JRY+99] allows the mobile node to register with multiple location registers (LRs) and avoids triangular routing for the data. I have augmented the original MIP-LR with application layer modules, namely packet interceptor and packet modifier. The packet interceptor and modifier modules at both sender and receiver side cooperate with each other to provide route optimization by sending the data directly to the mobile node and reduces the data overhead by avoiding the tunnels. Interceptor modules intercept the outgoing packets and the packet modifier modules change the destination address of the packet at the sender side before it is transmitted. The packet modifier module at the receiver side changes the destination address back to the permanent address of the mobile before sending it to the application. This way, the application is not aware of the underlying IP address change and the packets from the correspondent host are sent directly to the mobile's new point of attachment. From the experimental results in the testbed [DBJ+05], I have verified that one can attain up to 50 percent reduction in management overhead and up to 40 percent improvement on end-to-end delay compared to standard mobile IP in co-located mode. Figure 5.40 shows how the packet interceptor module and packet modifier are implemented in MIP-LR. Figure 5.41 shows the experimental testbed where I have carried out this experiment. The mobile host (mh) moves between the two 802.11 access points. *Delay*1 and *delay*2 are emulated delays between CH and HA. These delays were introduced using NIST delay simulator. *Lr*1 and *lr*2 are the location registers in each visited network. Table 5.7 shows the round trip time delay comparison between the interceptor assisted MIP-LR and mobile IP. *Ha* is the home agent.

Table 5.7 shows a comparison of RTT (Round Trip Time) between CH and MH for both MIP-LR and MIP for two fixed payload size e.g., 64 bytes and 1024 bytes respectively. These results show that MIP-LR outperforms the MIP as the payload size increases. As the delay factor *delay*1 was varied simulating increase in distance between CH and HA (Home

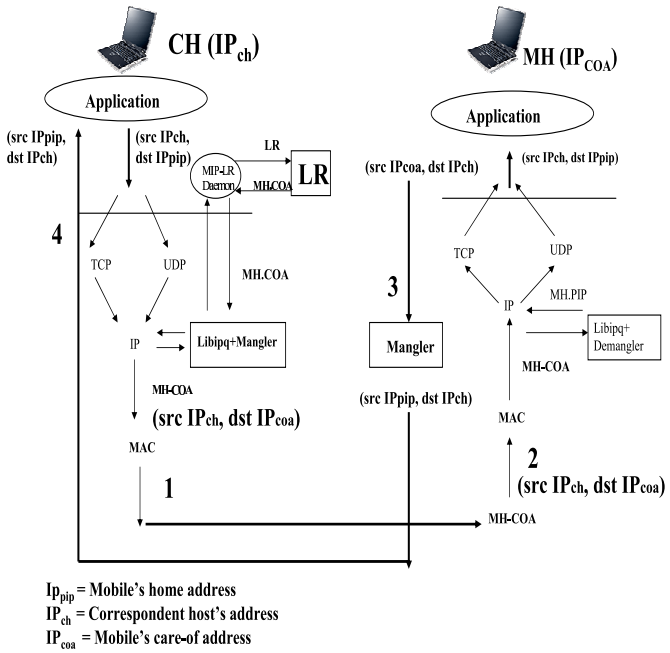


Figure 5.40: Packet interception technique for MIP-LR

Agent), MIP-LR's RTT is not affected because the packets to MH do not have to traverse via home agent as a result of the direct binding update from the mobile to the CH. These results demonstrate the effect of proposed techniques that provide direct path between the CH and MH. Even if the network delay between the home network and visited network increases, there is no effect on end-to-end packet delay in case of mobile IP.

5.10.5 Intercepting proxy-assisted route optimization

In some cases, forwarding of application layer signaling is delayed due to the routing indirection caused by the underlying network layer mobility mechanism when mobile IP is used as the mobility protocol. For example, in an IMS-based environment, when the mobile is in a visited network, mobile's SIP signaling goes via the outbound SIP proxy server, P-CSCF (Proxy Call Session Control Function). However, due to underlying network layer mobility protocol mobile IP, SIP signaling gets routed through the home agent even if the SIP proxy server is located very close to the mobile. This will delay SIP re-registration procedure and SIP Re-invite process after the mobile has handed over to the new network.

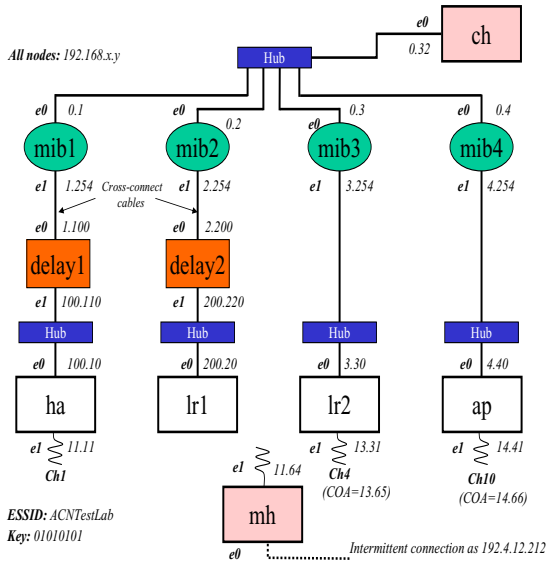


Figure 5.41: Experimental testbed for MIP-LR

For example, in MIPv4, reverse tunneling at FA (Foreign Agent) forces the packet from a mobile node to route via the home agent giving rise to the additional route traversal. At the application layer, SIP signaling is usually routed via the outbound proxy (e.g., SIP server) that is closer to the mobile. However, due to the indirection imposed by the underlying mobility layer, these packets need to travel to the home network before being directed to the application servers in the edge of the network. This additional routing causes signaling delay that affects the handoff.

Figure 5.42 illustrates the experimental testbed where I have applied the route optimization technique in order to reduce the traversal delay in signaling traffic. The solid line shows the optimized path, whereas the dotted line shows the non-optimized path routed via the home agent.

RFC 3024 [(Ed01] specifies a means to make use of the encapsulated delivery style to perform selective reverse tunneling. This is intended to support packet delivery to local resources. Packets meant to be reverse tunneled are sent using encapsulated delivery style (via the MN-FA tunnel) by the MN. The FA must reverse tunnel these packets to the HA.

Table 5.7: Experimental validation of route optimization

Packet Size Bytes	Emulated delay 0 ms		Emulated delay 10 ms		Emulated delay 20 ms		Emulated delay 40 ms	
	MIP (RTT) (ms)	MIP-LR (RTT) (ms)	MIP (RTT) (ms)	MIP-LR (RTT) (ms)	MIP (RTT) (ms)	MIP-LR (RTT) (ms)	MIP (RTT) (ms)	MIP-LR (RTT) (ms)
64	5.9	4.1	25.3	5.4	45.2	5.9	85.3	7.2
128	6.6	4.7	27.4	5.6	46	6.9	86.3	8.6
256	8	5.9	28.1	6.2	48	7.8	88.2	10
512	13.9	10.2	32	10.8	51.9	11.4	92	13.8
1024	19.5	13	39.5	14.2	59.4	15.4	99.7	16.9

Packets not meant to be reverse tunneled are sent using direct delivery style (not encapsulated), the FA will forward these and will not use reverse tunnel to send these to the HA. The MN can send all packets meant for the P-CSCF using normal IP routing and the FA will forward these as regular packets.

This approach solves one part of the trombone routing problem by optimizing the route from the MN to the P-CSCF. However, packets from the P-CSCF to the MN will be still routed via the HA. In addition, this selective reverse tunneling with encapsulated delivery style approach assumes changes to be made in the MIP protocol behavior. Thus, this feature may not be desirable for already installed MIPv4 infrastructure.

In this section, I propose an interceptor-assisted technique to optimize packet delivery from MN to P-CSCF and P-CSCF to MN in the visited network. Thus, the cost is represented as delay. The proposed approach provides an encapsulation technique between FA and P-CSCF. It installs packet interceptors and forwarding modules both in FA and P-CSCF to perform selective tunneling operation in both directions and also establishes an IP-IP tunnel between the P-CSCF and the FA for all the packets destined for the MN from the P-CSCF. Packets received at the FA via the P-CSCF-FA tunnel will be decapsulated at the FA and forwarded to the MN. This is identical to the manner in which encapsulated packets received at the FA via the HA-FA tunnel are processed. In addition, it does not

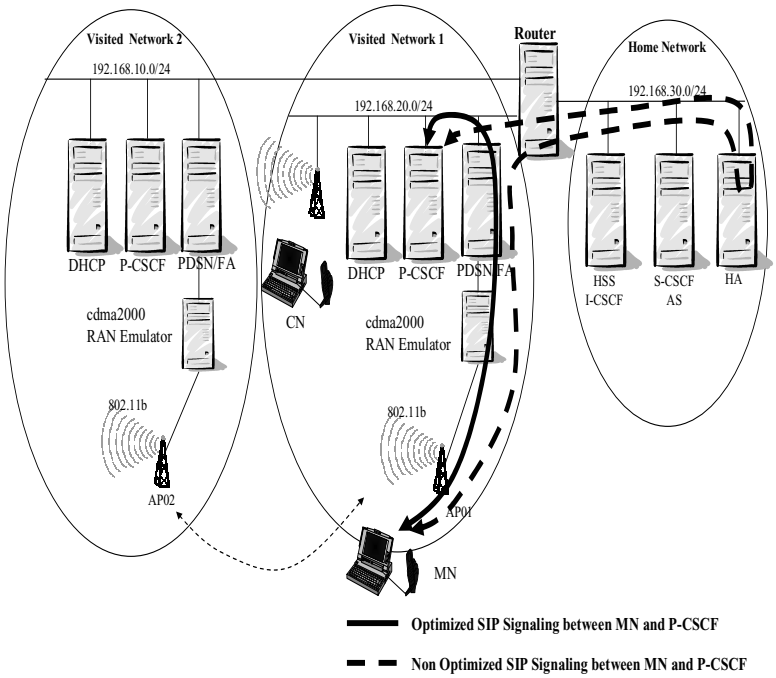


Figure 5.42: Route optimization of signaling traffic

require any changes in MIP functional behavior. Figures 5.43 and 5.44 illustrate the SIP signaling flow without and with route optimization approaches, respectively.

5.10.6 Cost analysis and experimental analysis

I applied this technique to optimize the path for SIP signaling messages, such as re-REGISTER and re-INVITE in a MIPv4-based mobility environment.

In this section, I provide a simple calculation of the delay based on the cost due to traversal of signaling messages and processing cost in the networking nodes. I do not include all the processing costs in each networking node in this analysis. I assume that the communication distance between MN and FA is d_1, between FA and HA is d_2, between HA and P-CSCF is d_3, between P-CSCF and I-CSCF is d_4, and between I-CSCF and S-CSCF is d_5.

The communication distance between P-CSCF and S-CSCF is d_6, and between FA and

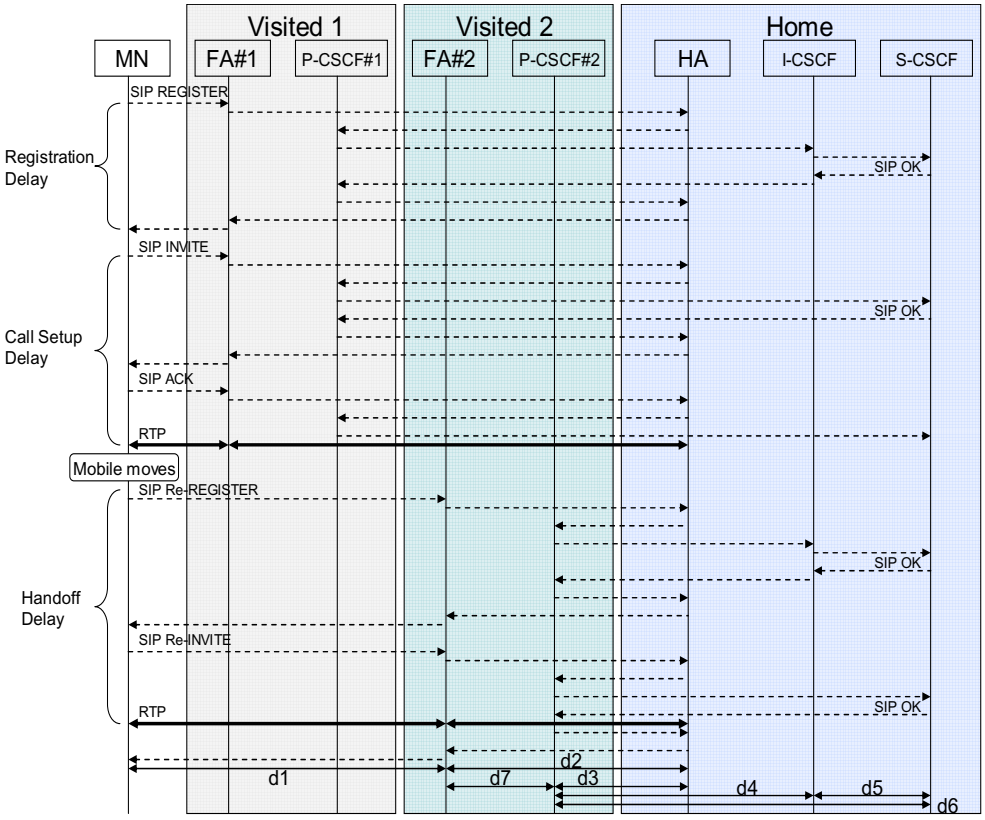


Figure 5.43: SIP signaling flow without route optimization

P-CSCF is d_7. Without loss of generality I assume that d_1, d_5 and d_7 are smaller than the other distances (i.e., entities are close to each other). The associated cost for traversing these communication distances are t_1, t_2, t_3, t_4, t_5, t_6, and t_7, respectively. I now analyze both the cases, without and with route optimization cases. I assume that the processing costs at HA and FA to be P_{HA} and P_{FA}, respectively when the mitigation technique is not applied. On the other hand, when the mitigation technique is applied, there is an additional processing cost due to modifying the packets and the additional look up at FA and P-CSCF. I assume this additional processing cost to be $P_{mitigate}$ for each message. Processing at HA is completely avoided in the optimized mitigation case as the signaling does not pass through HA.

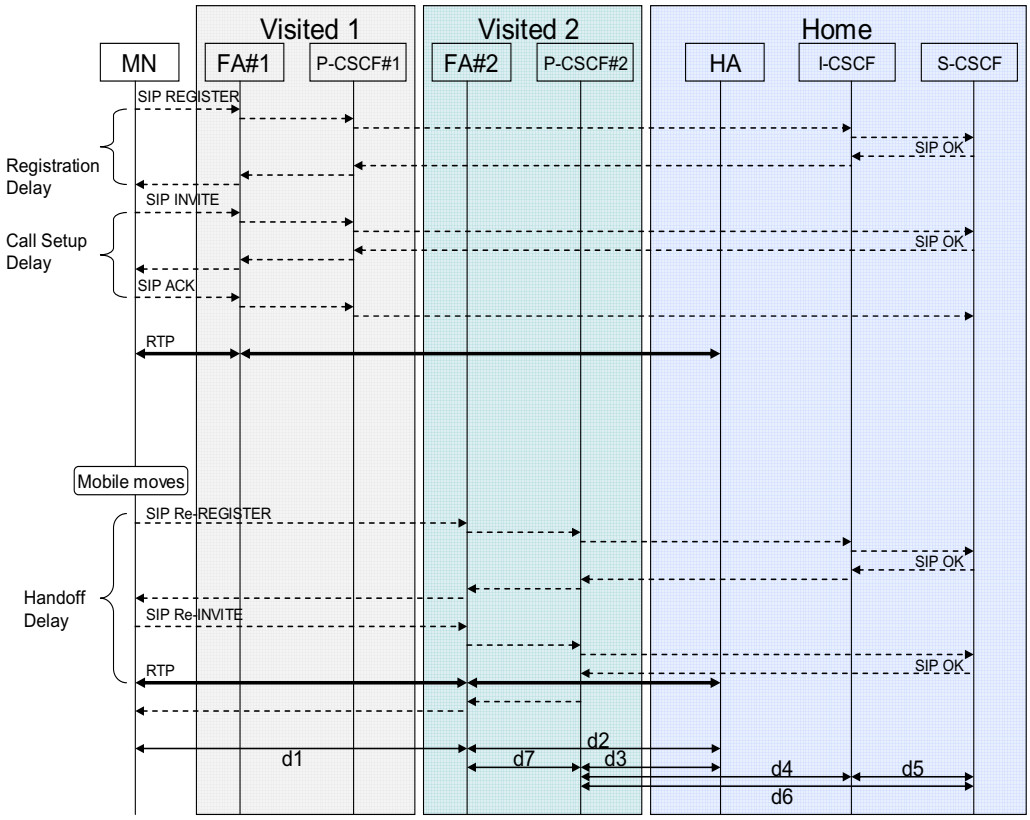


Figure 5.44: SIP signaling flow with route optimization

5.10.6.1 Cost analysis without route optimization

First, I discuss the case where mitigation technique is not applied. Before the move, the MN is in visited network 1 and is subjected to the registration and call setup delays. Referring to Figure 5.43, these delays can be calculated as follows. The SIP registration cost is $2(t_1+t_2+t_3+t_4+t_5)+2(P_{HA}+P_{FA})$. Similarly, call setup consists of three SIP-based signaling, such as INVITE, OK, and ACK. This cost results due to data traversal and processing operation that amounts to $3(t_1+t_2+t_3+t_6)+3(P_{HA}+P_{FA})$. When the MN moves to the visited network 2, it needs to re-register. There are other common set of operations that are part of the handoff, such as PPP (Point-to-Point Protocol) setup and DHCP operation to discover P-CSCF, that are same for both with and without the mitigation technique. Thus, the registration cost in the visited network 2 is the same as that in visited network 1 and amounts

to $2(t_1+t_2+t_3+t_4+t_5)+2(P_{HA}+P_{FA})$. Since the re-INVITE and 200 OK signaling help create the new context in the P-CSCF, that opens up the gate for media at the corresponding FA, these operations are included in the operation.

Call setup cost in the visited network 2 is calculated as $t_1+t_2+t_3+2t_6+P_{HA}+P_{FA}$. Thus, the total handoff delay D_1 when no optimization technique is applied is calculated as $3(t_1+t_2+t_3)+2(t_4+t_5)+2t_6+3(P_{HA}+P_{FA})$.

5.10.6.2 Cost analysis with route optimization

Next, we analyze similar cost when the proposed mitigation technique is applied. Registration cost in the visited network 1 is $2(t_1+t_7+t_4+t_5)+2P_{mitigate}$ and call setup delay is $3(t_1+t_7+t_6)+3P_{mitigate}$. Basically both of these values are smaller than the values when the mitigation technique is not applied. Referring to Figure 5.44, when the MN moves to the visited network 2, the re-registration cost is $2(t_1+t_7+t_4+t_5)+2P_{mitigate}$ and the call setup cost is $t_1+t_7+2t_6+P_{mitigate}$. Thus, the delay D2 during handoff from visited network 1 to visited network 2, in case of mitigation, is $3(t_1+t_7)+2(t_4+t_5)+2t_6+3P_{mitigate}$.

The handoff delay gain due to the mitigation technique is calculated to be $(D_1\text{-}D_2)= 3(t_2+t_3\text{-}t_7)+3(P_{HA}+P_{FA}\text{-}P_{mitigate})$. The effect of the mitigation technique is felt more when the distance between the visited network and home network is greater. When the distance is small, the benefit of the trombone routing mitigation is offset by the additional processing time at FA and P-CSCF during packet capture and encapsulation operation.

I have increased the communication distances d_2, d_3, d_4, and d_6 by an additional 500 ms delay using the NIST delay simulator and measured the performance in the experimental environment. The additional NIST delay between home network and visited network emulates a real deployment scenario. Figure 5.45 shows the handoff time for both without and with the proposed techniques, respectively in the IMS testbed environment.

In particular, Figure 5.45 shows the breakdown of delay due to several operations such as layer 2 handoff, PPP link configuration, mobile IP registration, DHCP INFORM for

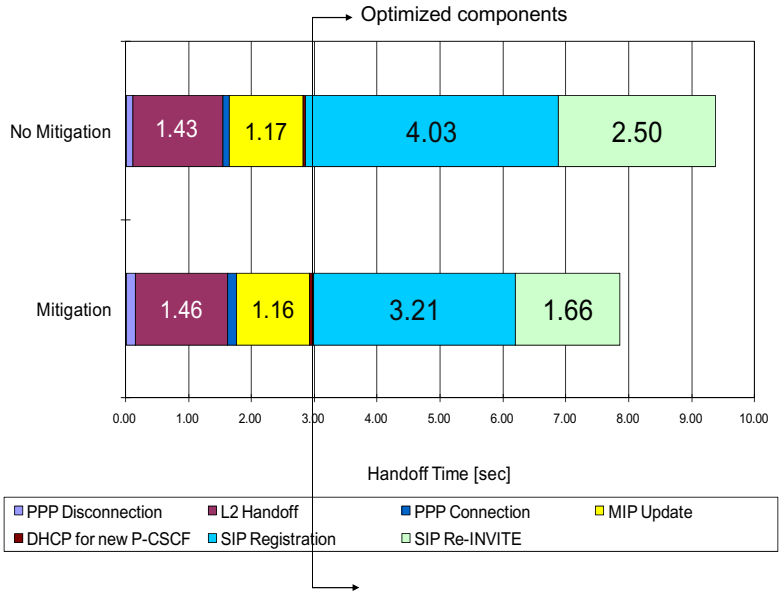


Figure 5.45: Results of route optimization using packet interceptor

P-CSCF discovery and SIP related signaling. It is important to note that the these route optimization techniques do not affect the delays due to non-SIP related operations (e.g., PPP, Layer 2, DHCP) and thus remain same for both the cases, with and without mitigation. Thus, I focus on the comparison on the reduction of handoff delay attributed by SIP related operations only. The amount of delay due to SIP signaling is a large fraction of the total handoff delay. For example, when the proposed mitigation technique is not applied, the delay attributed due to SIP signaling is 6.5 sec out of total handoff delay of 9.4 sec, almost 70 percent of the total delay. When the proposed technique is applied, the delay due to SIP signaling is reduced to 4.9 sec. Using this technique, I demonstrated that the SIP signaling packets are not affected by the packet indirection imposed by the underlying mobility protocol. For example, applying this technique, SIP re-registration took about 3,205 ms compared to 4,025 ms in the absence of this optimization technique. Similarly, SIP Re-INVITE took about 1,660 ms compared to 2,502 ms in the absence of mitigation technique.

However, the proposed technique involves additional amount of processing time at the foreign agent (FA) and P-CSCF because of additional packet capturing and packet modify-

ing operations. Thus, there is a tradeoff between this additional processing time at FA and P-CSCF and reduction in handoff delay due to this proposed technique. Effect of this proposed technique is more pronounced when the SIP server is situated in the visited domain closer to the mobile node and the home network is far from the visited network. I have described the implementation details of these techniques in [CDS07].

5.10.7 Binding cache-based route optimization

In this section, I introduce binding cache-based technique that can be applied to Proxy MIPv6 [GLD+08] to minimize the end-to-end media delay for both intra-domain and inter-domain mobility. This technique uses principle number 1 as defined in Section 5.10.1. Many of the Proxy MIPv6 terms are defined in Appendix C. Figure 5.46 shows the basic network configuration of Proxy MIPv6 architecture that involves an MN's intra-LMA (Local Mobility Agent) movement. A proxy MIPv6 domain is equipped with a specific LMA that acts as a home agent. The communicating nodes can belong to the same PMIPv6 domain or different ones. If the LMA is placed too far from the MAG (Media Access Gateway), then the media delivery between the MNs will be considerably delayed. In this specific intra-LMA scenario, MN#1 is anchored at MAG#1 and MN#2 is anchored at MAG#2. MN#2 establishes communication with MN#1 and then performs handoff to MAG#3. Without any route optimization, before the handoff, data communication between MN#1 and MN#2 goes via MAG#1, LMA, and MAG#2 and after the handoff it goes via MAG#3. Thus, it is desirable to reduce the media route associated with data traversal before and after handoff. For intra-LMA scenario, MN#1 and MN#2 operate under the same LMA and the MN's movement is confined to the MAGs that are under the same LMA. When the route optimization technique is deployed, the communication path is shortened as the packets bypass the LMA. The dotted line shows the non-optimized path through the LMA and the solid lines show the route optimized path that bypasses LMA. Similarly, for inter-LMA movement, data traversal via LMA is avoided and data is forwarded from one

MAG to another MAG directly.

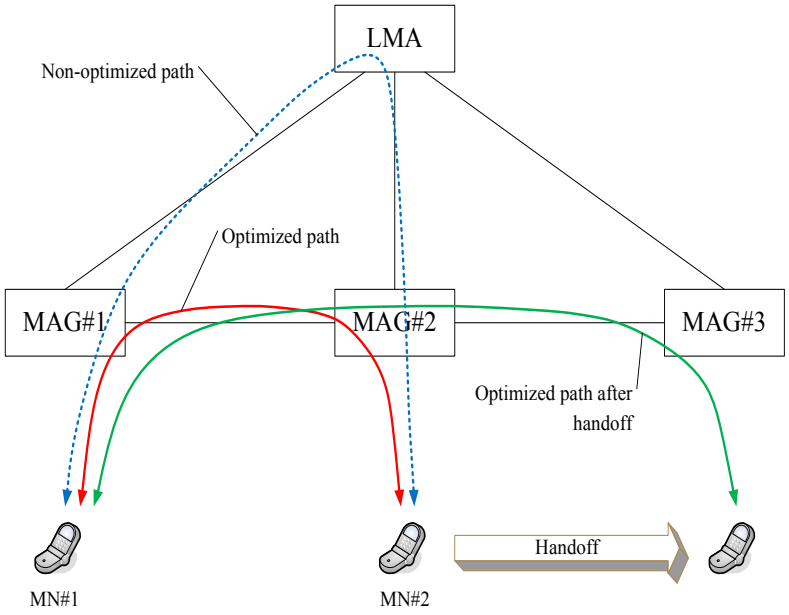


Figure 5.46: Architecture for binding cache-based route optimization

Figure 5.47 shows the basic optimization procedure and flows associated with one of the optimization techniques, that utilizes binding cache entry (BCE) at the LMA and at the MAG. In this figure, I first show that the path optimization from MN#1 to MN#2. Before the handoff, MN#1 attaches to MAG#1 and then MAG#1 sends a PBU (Proxy Binding Update) message to the LMA on behalf of MN#1. Similarly, MN#2 connects to MAG#2, which triggers a PBU (Proxy Binding Update) message to the LMA on behalf of MN#2. The initial packet from MN#1 to MN#2 is tunneled and is sent via the LMA. As soon as the LMA gets this packet, it figures out how to forward the packet to MAG#2. Then, the LMA sends a new message called CBU (Correspondent Binding Update) message to MAG#1 notifying that MAG#2 is the anchoring node for MN#2. After getting this CBU message, MAG#1 keeps a cache that maps MAG#2 with MN#2 and sends a response message called CBA (Correspondent Binding Acknowledge) message to the LMA. Thus, any subsequent packet from MN#1 destined for MN#2 gets intercepted by MAG#1 and is forwarded to MAG#2, instead of being forwarded to the LMA. The tra-

jectory of the route optimized packet thus becomes: MN#1→MAG#1→MAG#2→MN#2 instead of MN#1→MAG#1→LMA→MAG#2→MN#2, thereby optimizing the route of the data packet from MN#1 to MN#2.

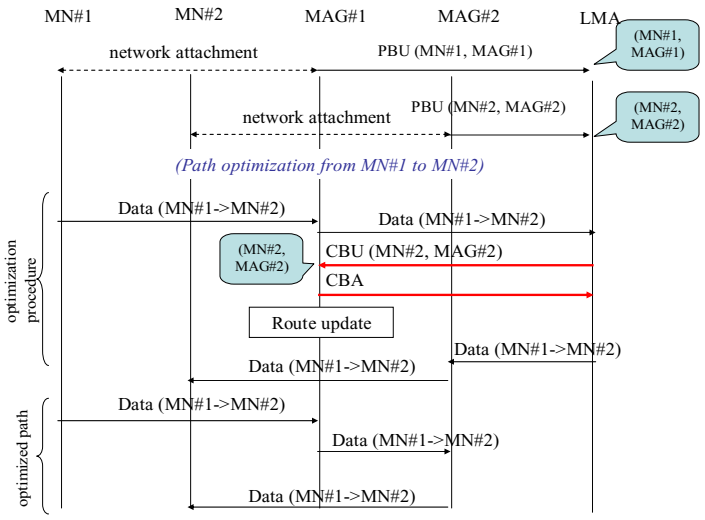


Figure 5.47: Binding cache-based flow

Table 5.8 compares the results of with and without route optimization techniques. In particular, these results show end-to-end media delay between the MNs and SIP signaling delay for the MNs. In the IMS network, all the SIP-related signaling such as REGISTER and INVITE traverse all the way to the home network using the tunnels between the MAG and the LMA. Thus, this route optimization technique does not reduce SIP signaling delay unlike intercepting proxy assisted route optimization described in Section 5.10.5. In the absence of route optimization technique, the media traffic flows via the local mobility agent LMA. However, when route optimization is applied, media traffic bypasses the LMA and flows between the MAGs over the tunnel that is set up between the MAGs. It is evident from Table 5.8 that when route optimization technique is in place, end-to-end media delay does not get affected even if the delay between the LMA and the MAG is increased. Thus, handoff delay will be reduced if route optimization technique is in place after the mobile moves to the new network.

Table 5.8: Results from route optimization using binding cache approach

Additional round trip delay between MAG and LMA	50 ms		100 ms		150 ms		200 ms	
Route optimization	RO	Non RO	RO	Non RO	RO	Non RO	RO	Non RO
End-to-end media delay (ms)	12	71	12	107	15	167	12	213
SIP REGISTER delay (s)	2.34	2.50	2.84	2.82	3.40	3.41	3.90	4.18

I have also explained details of this route optimization technique in the IETF draft [CYD+08].

5.11 Media independent cross-layer triggers

Cross layer triggers are useful hints that can expedite the sequential handoff operations that take place in each layer. These handoff triggers could be applied during several stages in the handoff process, namely during handoff initiation, discovery, and configuration. Lower layer events are generally passed as the triggers to the upper layers so that mobility related functions at upper layers can be expedited. The lower layer triggers prepare the mobile for the impending handover event by performing different phases of the handover operations pro-actively. The layer 2 triggers can assist layer 3 operations that rely exclusively on these indicators to perform specific actions such as detection of network attachment or detachment.

In this section, I first describe the key principles that need to be considered to design handoff mechanisms based on cross layer triggers. I then describe some of the related papers that have used cross layer triggers to expedite the handoff process. I then define the 802.21-based cross layer triggers. I have contributed to the design of IEEE 802.21-based cross layer triggers.

5.11.1 Key principles

Following are some of the principles that are taken into account for designing the cross layer triggers.

1. Handoff related functions are spread across different layers of the protocol stack and are executed independently. For efficient network communication, it is essential for a protocol layer to utilize cross layers' information, such as the from layer 2 triggers.

2. Since each protocol layer is also implemented independently in the current operating systems, it is very hard to exchange control information between protocol layers. Thus, it is helpful to have some abstract set of primitives that can pass on the information across the layers in order to trigger the rest of the handoff operations.

3. Interaction between the events across layers expedite the initiation of a specific event in another layer.

4. Handoff initiation process is expedited when a mobile is made aware of the impending handoff. Prior indication regarding an imminent handoff operation helps the mobile to collect information for the upper layers.

5. Triggers from lower layers help initiate many of the upper layer operations, such as layer 3 discovery, attachment and configuration process.

5.11.2 Related work

There are a few related papers that demonstrate the effect of lower layer triggers during handoff. Teraoka et al. [TGM+08] propose unified Layer 2 (L2) abstractions for Layer 3(L3)-driven fast handovers. Yokota et al. [YIHK02] describe a link layer assisted mobile IP fast handoff method that uses a combination of MAC bridge and 802.11 access point to reduce the handoff period and packet loss during subnet handoff. Tseng et al. [TYCH05] describe a topology-aided cross-layer fast handoff design for IEEE 802.11 and mobile IP

environments. A mobile node can utilize cross-layer topology information, such as the association between 802.11 access points and Mobile IP mobility agents, together with layer-2 triggers, to start layer-3 handoff-related activities, such as agent discovery, address configuration, and registration, in parallel with or prior to layer-2 handoff. However, these triggering techniques are access specific and do not define any abstract primitives. As part of my research, I have contributed to the development of MIHF (Media Independent Handover Function) [DTC+09] that has recently been standardized within the IEEE 802.21 working group. Unlike other proposals, this work develops abstract primitives that can be applied to support handover between heterogeneous access networks such as 802.11 and CDMA. In this section, I describe some of the functional elements of 802.21 services but provide the experimental results involving mobile initiated and network initiated handover in Chapter 9.

5.11.3 Media independent handover functions

I have contributed to the design of cross layer triggers that assist the mobile during its initiation phase or discovery phase by providing the useful triggers about the impending movement, attachment to a new network, disconnection from an old network. Many of these triggers have been standardized as "Information Service", "Command Service" and "Event Service" as part of MIHF (Media Independent Handover Function) that were developed within IEEE 802.21. I explain below some of these primitive services.

Figure 5.48 shows how MIHF can interwork with the mobility protocols using many of the information service, event service and command service primitives.

5.11.3.1 Media independent event service

Media independent event service (MIES) provides services to the upper layers by reporting both local and remote events. Local events take place within a client whereas remote events take place in other components within the network, such as router, access point, server and

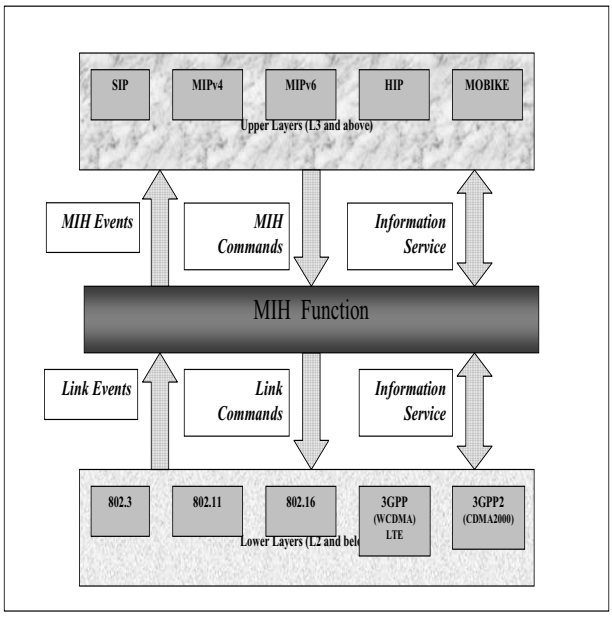


Figure 5.48: Media independent handover functional (MIHF) interaction

communicating host. The event model works according to a subscription and notification procedure. An MIH user (typically upper layer protocols) registers to the lower layers for a certain set of events and gets notified as those events take place. In case of local events, information propagates upward from the MAC layer to the MIH layer and then to the upper layers. In case of remote events, information may propagate from the MIH or Layer 3 Mobility Protocol (L3MP) in one stack to the MIH or L3MP in a remote stack. Some of the common events defined include *Link Up*, *Link Down*, *Link Parameters Change*, *Link Going Down*, *L2 Handover Imminent*. I have described the relevant handover primitives in Table 5.9. After the upper layer of the mobile gets notified about certain events at the lower layers by means of event service, the mobile makes use of the command service to control the links to switch over to a new point of attachment.

5.11.3.2 Media independent command service

The higher layers use the media independent command service (MICS) primitives to control the functions of the lower layers. MICS commands are used to gather information about the status of the connected links, as well as to pass on the higher layer mobility and connectivity decisions to the lower layers. MIH commands can be both local and remote. These include commands from the upper layers to the MIH and from the MIH to the lower layers. Some examples of MICS commands are *MIH Poll*, *MIH Scan*, *MIH Configure*, and *MIH Switch*. The commands instruct an MIH device to poll connected links to learn their most recent status, to scan for newly discovered links, to configure new links and to switch between available links.

5.11.3.3 Media independent information service

Mobiles need to discover available neighboring networks and communicate with the elements within these networks to optimize the handover process. The MIIS defines information elements and corresponding query-response mechanisms that allow an MIHF entity to discover and obtain information of the nearby networks. It provides access to both static and dynamic information, including the names and providers of neighboring networks as well as channel information, MAC addresses, security information, and other information about higher layer services helpful to handover decisions. This information can be made available via both lower and upper layers. In some cases, certain layer 2 information may not be available or sufficient to make intelligent handover decisions. In such scenarios, higher-layer services may be consulted to assist in the mobility decision-making process. The MIIS specifies a common way of representing information by using standard formats such as XML (eXternal Markup Language) and TLV (Type-Length-Value). Having a higher layer mechanism to obtain the information about the neighboring networks of different access technologies alleviates the need for a specific access-dependent discovery method. I have implemented an MIIS based on RDF (Resource Description Frame-

work) for MIIS. Many of the cross layer triggers that are part of "Information Service" help expedite the initiation of handoff process by discovering the network components proactively. Some of the primitives for MIIS that are used to discover the network services are *Get_Info_Request* and *Get_Info_Response*.

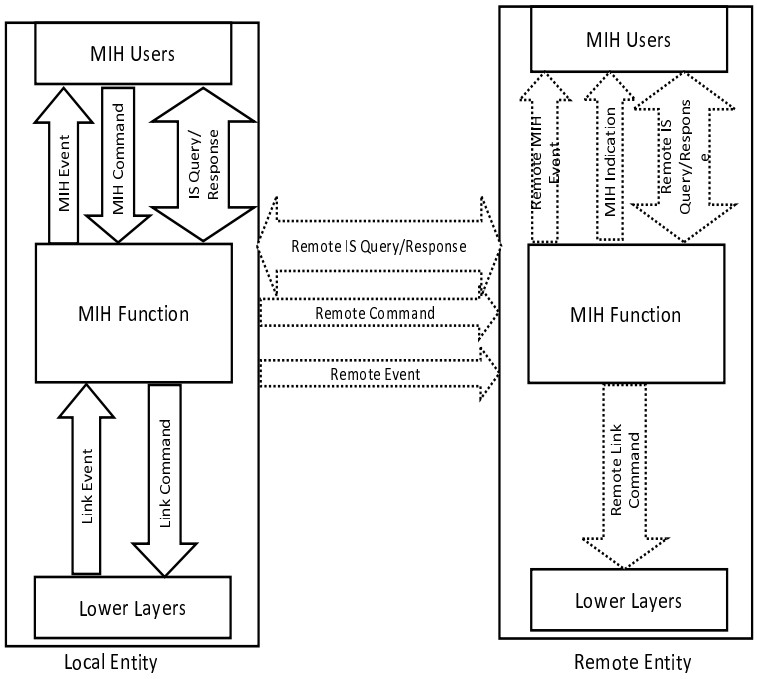


Figure 5.49: Cross layer triggers with MIHF

Figure 5.49 shows how the local and remote MIH functions interact with each other.

Table 5.9 lists several types of the MIH primitives and their interactions. These are categorized as Management, Event service , Command service and Information service. There are several scenarios where these triggers could be useful to expedite the handoff operations. I have experimented with some of these 802.21-based triggers and have demonstrated how these techniques can be used as helpers to many of the existing mobility protocols such as MIPv6, SIP-based mobility and optimization scheme such as Media Independent Preauthentication. I have described those experimental results in Chapter 9.

Table 5.9: Sample MIHF primitives

MIH_SAP Primitives	Service category	Description
MIH_Capability_Discover	Management	Discover list of Events and Commands supported by MIHF
MIH_Register	Management	Register with a remote MIHF
MIH_DeRegister	Management	Deregister from a remote MIHF
MIH_Event_Subscribe	Management	Subscribe for one or more MIH events with a local or remote MIHF
MIH_Event_Unsubscribe	Management	Unsubscribe for one or more MIH events from a local or remote MIHF
Link-Detected	Event	Link of a new access network is detected
MIH_Link_Up	Event	L2 connection is established
MIH_Link_Down	Event	L2 connection is lost
MIH_Link_Going-Down	Event	L2 connectivity is predicted to go down
MIH_Link_Handover_Imminent	Event	L2 handover is imminent
MIH_Link_Handover_Complete	Event	L2 handover link handover to a new access network is complete
MIH_Link_Parameters_Report	Event	Link parameters have crossed specified thresholds
MIH_Link_Get_Parameters	Command	Get the status of the link
MIH_Link_Configure_Thresholds	Command	Configure Link parameter thresholds
MIH_Link_Actions	Command	Control the behavior of set of links
MIH_Net_HO_Candidate_Query	Command	Initiate Handover
MIH_MN_HO_Candidate_Query	Command	Initiate MN query request for candidate network
MIH_N2N_HO_Query_Resources	Command	Query available network resources
MIH_MN_HO_Commit	Command	Notify the serving network of the decided target network information
MIH_Net_HO_Commit	Command	Network has committed to handover
MIH_N2N_HO_Commit	Command	Notify target network that serving network has committed to handover
MIH_MN_HO_Complete	Command	Initiate MN Handover complete notification
MIH_N2N_HO_Complete	Command	Handover has been completed
MIH_Get_Information	Information	Requests to get information from repository
MIH_Push_Information	Information	Notify the mobile node of operator's policies or other information

5.11.3.4 MIHF Implementation

This section describes the software implementations of MIHF. The MIH software implementation includes the MIHF as well as the MIH Information Server. The software is implemented in Java 1.6 and is thus portable across different operating systems.

Different components of MIHF software implementation is shown in Figure 5.50.

It provides the MIH API for the MIH Users. The MIH API embodies the MIH_SAP and supports both local and remote MIH services. Communication for remote services are realized by the MIH Protocol component that implements the MIH protocol. Current version of MIH protocol implementation uses UDP as the MIH transport protocol.

The MIH user manager component is responsible for determining privileges of the MIH users. It enforces coordination between multiple MIH users such that only a particular MIH user is allowed to change the state of network interfaces. This prevents conflicting state changes to be made by different MIH users that employ different handover policies at the

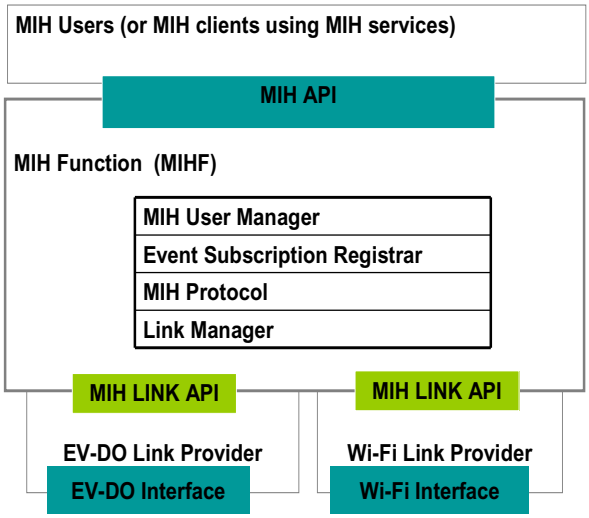


Figure 5.50: MIHF implementation stack

same time. An example could be a network interface that is turned on by one MIH User and then turned off by another MIH User.

Network interfaces are managed by the link manager using the MIH LINK API. The MIH LINK API is implemented by the Link Providers components and embodies the MIH_LINK_SAP. A distinct Link Provider component is defined for each network interface type. The Link Providers are considered the adapters to the network interfaces and can be implemented either inside or outside of network interface drivers. The current Link Providers are implemented outside of the network interface drivers and support MICS and MIES for IEEE 802.11 and cdma2000 EV-DO interfaces in the Linux environment. The Link Providers are implemented in Java with JNI (Java Native Interface) to utilize device specific C calls since most device drivers have C APIs rather than Java APIs.

Link Provider implements Link_Parameter_Report event notification, which generates event notifications when the related interface crossed configured threshold levels. In order to avoid flooding event notification due to frequently changing signal strength, the proposed

Link Provider implements a function to average out the actual signal strength before reporting to the MIHF. On the other hand, this may delay the reaction time on actual threshold crossing. The Event Subscription Registrar component manages local and remote event subscriptions for the link-layer events monitored by the MIHF. It also aggregates multiple event subscriptions by multiple MIH Users of the same MIHF into a single event subscription and delivers notifications to the subscribed MIH Users when event notifications are received.

The IEEE 802.21 Information Server (IS) is implemented as an MIH User that responds to MIIS queries through interaction with the MIH Protocol component. At initialization, the IS registers with its local MIHF to receive IS queries carried in MIH_Get_Information request messages. After the registration, it is ready to respond to queries sent by other MIH Users. Current implementation supports IS queries for Resource Description Framework (RDF) data using SPARQL query language. The IS uses Oracle 11g database to query RDF data.

I show the experimental results associated with the 802.21-based cross layer triggers in Chapter 9.

5.11.4 Faster link down detection scheme

The sooner a mobile detects the loss of connectivity at the lower layer, this information is passed up to the upper layers to complete the handover related information. In this section, I describe an optimized method for determining *link down* indication by the mobile. "Link Down" is an event provided by the link-layer that signifies a state change associated with the interface no longer being capable of communicating data packets. This proposed method uses MAC layer operations for verifying communicability with the access point. These methods can be used to provide fast link down event indication and can help in quickly assisting the upper layer protocols to take actions. This Link Down detection technique can be used in conjunction with 802.21 triggers to expedite the handoff process.

In 802.11-based layer 2 operation, a client that is currently associated with an AP (Accesss Point) may experience sudden disconnection due to device failure in the AP or in the client or perhaps an un-anticipated rapid movement of the client that quickly brings the mobile out of range of the AP. Using signal quality to immediately determine "*link down*" events in the client during these scenarios can be misleading since a client registers the link quality based only on the last received frames. Therefore, link quality only represents historical data and it is reset only after certain number of expected beacon frames have not been received by the mobile. Other implementations verify connectivity using failed transmission events (RTS/CTS) [VK04].

I have designed an optimized link detection technique that uses a combined scheme of passive monitoring of 802.11 frames as well as active probing of the AP at some defined conditions. A combination of these schemes as well as monitoring of independent indicators provide several link-down detection variants that are applicable to different scenarios. By this method, the mobile can rapidly determine the sudden disconnection event and quickly propagates *link down* indications to upper layer protocols, such as mobile IP stack in the mobile. Event triggers, such as *link down*, and *link going down* techniques are useful to optimize the overall handoff operations.

I describe below the details of fast detection algorithm.

The purpose of the fast disconnection algorithm is to provide a definitive measure that complete link loss has occurred within a relatively short period of time. Normally, definitive indicators can be ambiguous for 802.11 networks when considering very short time constraints (in the order of milliseconds) because of the following factors:

1. Signal strength can vary by several dBs when moving just a few meters. The low points of these fluctuation can cause a "false" indication of link loss during a short period of time.

2. Unstable packet loss rate can also cause "false" indications. Most 802.11 links have stable loss rate, though it is expected that there will be bursty periods of loss rate

perhaps due to increase in traffic demand or bursty traffic.

3. Propagation interference from adjacent AP within or near the Non-AP STA transmit channels. This relates to (ii) with packet loss rate at high stable level. This is aggravated by the DCF algorithm of 802.11 MAC [dPPC03] which may cause delay for any frame that requires an ACK.

4. Multipath Fading. Though this can contribute to (iii) it is less likely to occur in more recent DSP/chipset implementations which can compensate for this effect.

I describe below three basic ways a link down detection scheme can be carried out in IEEE 802.11 networks.

5.11.4.1 Passive scan

During passive scanning the access point broadcasts beacons at a regular interval. A mobile counts the number of consecutive beacon losses. A beacon is considered as lost if the mobile does not receive a beacon for a period of time T_B since the receipt of last beacon or loss. A passive scan is said to be failed if the number of consecutive beacon losses reaches a threshold N_B. The link is considered to be down when passive scan fails. Thus, the detection time is $T_B \times N_B$. However, this scheme suffers from few drawbacks. The detection speed is slow if T_B is large, on the other hand if T_B is small, there are too many beacons. There is also an increased probability of false link down detection if N_B is small.

5.11.4.2 Active scan

A mobile unicasts a probe request every interval of time T_P. Active scan is said to be failed if the number of transmissions reaches a threshold N_P and no probe response(s) has been received. If at least one probe response is received before N_P is reached then active scan is has succeeded. The link is considered to be down when active scan fails. Thus the detection time is $T_P \times N_P$. However, there are many issues with active scan. There are too much probe

traffic if T_P is small. Also, there is an increased probability of false link down detection if N_P is small. Detection speed is slow if T_P is large.

5.11.4.3 Hybrid Scan

Hybrid scan performs a combination passive scan and active scan. A mobile would normally perform passive scanning. When passive scan reports a failure, it starts active scanning instead of immediately considering that the link is down. The link is considered to be down if the active scan also fails. If the active scan succeeds the mobile switches back to passive scan. Thus, the detection time would be considered as equivalent to $T_B \times N_B + T_P \times N_P$. This combination of schemes lessen the chance of false detection compared to passive or active scan by themselves, but in a heavily loaded condition, the chance of false detection will increase than lightly loaded condition for the same pair of N_B and N_P values.

5.11.4.4 Independent modifiers

Traffic flows are considered independent modifiers since they directly affect the basic set of algorithms. Addition of these modifiers to the basic algorithms produces variants that uses any sent and received frames as replacement indicators for beacon and probe responses. Following are some of the modifiers.

1. Received frame: The receipt of any frame can be considered as receipt of a beacon frame or probe response. Therefore, a passive or active scan is considered successful if any frame is received from the AP. This takes advantage of heavily loaded conditions where beacon or probe responses can get lost and trigger a false link down event.

2. Transmission failure: Failure to transmit a data frame can be used as an indication of link failure. Under, 802.11 MAC, each data frame sent by the mobile requires an ACK (acknowledgement) from the AP. The mobile will retransmit the data frame if

an ACK is not received within a certain amount of time (normally implementation specific). The link is considered down if the number of retries exceed a configured threshold (also implementation specific.

The independent modifiers are not considered independent solutions since they rely on applications generating the data traffic. However, they can be combined with the basic algorithms, namely active scan, passive scan and hybrid scan to produce several variants. One or more of these variants can be used as actual solutions for fast link down detection based on preference, scalability or implementation considerations. Using these variants helps reduce false link-down detection caused by heavy traffic conditions since it takes advantage of the ongoing data exchanges. Both modifiers can be combined with each basic algorithm as shown below:

Following are some of the proposed fast detection techniques that use a combination of scanning and modifiers.

1. Passive scan combined with modifiers: Passive scanning combined with both modifiers can result in the following variants

   (a) Received frame: A passive scan is successful if any frame is received from the AP within $T_B \times N_B$ otherwise passive scan fails if N_B threshold is reached without receipt of any frame. Any received frame becomes a substitute for an expected beacon frame.

   (b) Transmission failure: A passive scan fails if data transmission fails even if N_B has not yet been reached. Likewise, a passive scan succeeds if data transmission succeeds even if N_B has not yet been reached. If no Non-AP STA application is generating data, the passive scan proceeds as normal.

   (c) Received frame and Transmission failure: A passive scan fails if (a) or (b) of this section fails. Likewise, a passive scan succeeds if (a) or (b) succeeds. The detection time is determined by which failure occurs first (a or b).

2. Active scan combined with modifiers: Active scanning combined with both modifiers can result in the following variants:

   (a) Received frame: An active scan is successful if any frame is received from the AP even before N_P is reached otherwise the active scan fails if N_P threshold is reached without receipt of any frame. Any received frame becomes a substitute to the expected probe response.

   (b) Transmission failure: An active scan fails if data transmission fails even if N_P has not yet been reached. Likewise, an active scan succeeds if data transmission succeeds even if N_P has not yet been reached. If no Non-AP STA application is generating data, the active scan proceeds as normal.

   (c) Received frame and Transmission failure: An active scan fails if (a) or (b) mentioned above fails. Likewise, an active scan succeeds if (a) or (b) succeeds. The detection time is determined by which failure occurs first (a or b).

3. Hybrid scan combined with modifiers: Hybrid scan combined with both the modifiers can result in the following variants:

   (a) Received frame: A hybrid scan is successful if any frame is received from the AP even before the passive AND active scan threshold has been reached. Receipt of any frame constitute a receipt of an expected beacon or probe response. The hybrid scan fails when the passive and subsequent active scan fails without receipt of any frames.

   (b) Transmission failure: A hybrid scan fails if data transmission fails even before the passive AND active scan threshold has been reached. Likewise, a hybrid scan succeeds if data transmission succeeds even before the passive AND active scan threshold has been reached. If no Non-AP STA application is generating data, the hybrid scan proceeds as normal.

(c) Received frame and Transmission failure: A hybrid scan fails if (a) or (b) of this section fails. Likewise, an hybrid scan succeeds if (a) or (b) succeeds. The detection time is determined by which failure occurs first (a or b).

I have used the 802.21-based cross layer triggers in conjunction with faster link down detection scheme that I described in this section to reduce the handover delay and packet loss during movement between 802.11 access network and CDMA network. In this specific scenario, the mobile is connected to the 802.11 network and there is a sudden power failure in the 802.11 access point. Using this faster *Link down* detection mechanism and 802.21 event notification triggers, the handoff delay is reduced. In Chapter 9, I describe how faster link down detection techniques are useful to optimize handovers involving CDMA and 802.11 networks.

5.12 Concluding remarks

My proposed optimization techniques for different handoff components (e.g., discovery, authentication, configuration, security association, binding update, and media rerouting) can primarily be categorized into three types, namely *reactive*, *proactive* and *cross layer*.

My proposed reactive mechanisms are applicable after mobile's handover to the new network. These techniques use additional elements in the network such as anchor agent, mobility proxy, or an external home agent to mitigate the effect of IP address acquisition delay, longer binding update delay, avoid the triangle routing and reduce the delays due to security association.

My proposed proactive mechanisms are applied prior to handover and offer better handoff performance results compared to the reactive techniques at the expense of additional resources in the network such as proactive tunnels with the candidate target networks, multiple media streams due to proactive binding update, caching of IP addresses from the neighboring networks and buffering at the edge routers.

My proposed cross layer mechanisms can expedite the detection of new point of attachment or sudden loss of network. Cross layer triggers are most useful when the mobile performs handover across heterogeneous access networks (e.g., CDMA and 802.11) so that the mobile can prepare many of the upper layer handoff related operations based on layer 2 triggers. Cross layer triggers can be applied both *proactively* and *reactively*. For example proactive cross layer triggers can help initiate an application layer discovery process before the handoff, whereas reactive cross layer triggers can trigger the handoff by passing the lower layer information such as signal to noise ratio (SNR) to initiate the layer 3 handover operations.

Depending upon the type of network environment, performance requirements and resource availability, either reactive, proactive or cross layer techniques could be applied to optimize a specific handoff component.

Chapter 6

Optimization with multi-layer mobility protocols

In this chapter, I describe my proposed multi-layer mobility optimization techniques that use triggers from datalink layer (e.g., layer 2) and application layer, and optimize several handoff operations, namely address configuration, layer 3 binding update, and media traversal. During the discovery process, datalink layer trigger specifies the mobile's movement type based on layer 2 beacon id and layer 3 subnet prefix or domain id. Similarly, application layer trigger specifies the type of application such as TCP-based (e.g., ftp) or RTP-based (e.g., VoIP) on the mobile to trigger the right type of mobility protocol.

My proposed mechanism optimizes handoff delay and reduces packet loss by way of limiting the number of signaling update messages during a mobile's movement within a domain and reduces the end-to-end transport delay of the media by use of direct binding update for real-time application. This mechanism uses a policy-based approach based on mobile's movement pattern and application and decides the mobility protocol that is most appropriate to be used.

6.1 Summary of key contribution and indicative results

Network layer mobility protocol, application layer mobility protocol and local mobility protocols can operate independently without interacting with each other. However, each of these mobility protocols has its own pros and cons. For example, network layer mobility protocol such as MIP [Per02c] needs additional networking element (e.g., home agent) in the home network to support terminal mobility and is thus not optimized; application layer mobility protocol such as SIP [SW00] is best optimized to work for real-time application (e.g., VoIP) but cannot support mobility for TCP-based traffic in its current form. Local mobility protocol such as cellular IP [CGK+00] cannot support mobility across subnets or inter-domain. An integrated mobility management scheme whereby a mobile can use any of these mobility protocols based on certain policy (e.g., type of movement, application type) will enable a mobile to use the best features of each of these protocols and thus will offer optimized handoff performance.

I developed a multilayer mobility management scheme that uses cross layer triggers from data link layers and application layers and optimizes several handoff operations, namely address configuration, layer 3 binding update and media rerouting. My proposed mechanism uses a policy-based approach based on the mobile's movement pattern and type of application and executes the mobility protocol that is most appropriate to be used under a specific network and application environment. This mechanism uses SIP-based application layer mobility to support real-time traffic and MIP-LR-based mobility to support non-real-time traffic during inter-domain movement while it uses the local mobility management protocol (MMP) [WWD+02] to support real-time and non-real-time traffic during intra-domain and intra-subnet movement.

My proposed multilayer mobility mechanisms have the following key advantages.

- My proposed mechanism increases the data throughput by 50 percent under high mobility scenario by reducing the binding update traversal during intra-domain mobility

and uses the lower layer triggers such as information from the layer 2 beacon id to determine intra-domain and inter-domain mobility based on gateway's identifier.

- ◇ My proposed mechanism expedites the discovery operation by discovering layer 3 point of attachment while discovering layer 2 point of attachment using the optimization by way of parallelism and reduces the packet loss during handover.

- ◇ Using the application layer triggers, my proposed mechanism uses the mobility protocol that is optimized for a specific type of application (e.g., SIP for RTP-based traffic and MIP for TCP-based traffic).

After I developed the proposed policy-based mobility management scheme back in 2001, few other integrated mobility management schemes were developed by Politis et al. [PCA+04] and Lee et al. [LLC03] that use SIP for personal mobility and MIP for terminal mobility and carry SIP registration information as part of MIP binding update. However, none of these existing approaches use any cross layer triggers to optimize the handoff performance nor do they provide throughput increase comparable to experimental results from my proposed mechanisms.

In the rest of the chapter, I describe the details of my proposed mechanisms, explain a few integrated mobility management schemes that use a combination of SIP, MIP-LR and micro mobility protocol (MMP), cite some related work and demonstrate the results from the experimental systems that I built.

6.2 Introduction

The proposed multi-layer integrated mobility management scheme is designed keeping in mind the requirements for real-time and non-real-time traffic. Currently, there is no framework that handles global macro mobility as well as micro mobility, but both are important and necessary. I have designed the micro-mobility management protocol (MMP) to han-

dle mobility at layer 2 for the proposed integrated mobility management scheme described in this section. For macro mobility involving subnet and domain movement, I used SIP [RSC+02] to handle mobility for real-time traffic and MIP with Location Registers (MIP-LR) to handle the mobility for non-real-time traffic. In either case, MMP handles micro-mobility to support layer 2 movement. This multi-layer mobility management architecture introduces several novel features as follows:

1. A mechanism that introduces a policy to use SIP to support macro mobility[1] for real-time traffic and MIP-LR to support non-real-time traffic.

2. Use SIP for macro mobility with MMP for micro mobility for real-time traffic.

3. Use of MIP-LR for macro mobility with MMP for micro mobility for non-real-time traffic.

6.3 Key principles

Following are the key principles that are used to design multi-layered mobility optimization scheme.

1. Based on the type of movement at a specific layer such as layer 2 or layer 3, binding update can be confined to a domain by using an anchor agent to optimize the handoff delay.

2. Based on the transport protocol that an application uses (e.g., RTP or TCP), a policy can be applied to invoke either application layer or network layer mobility protocol.

3. Layer 2 trigger and application layer triggers help to determine the type of mobility protocol that is needed to optimize the handoff.

[1]Macro mobility and micro mobility are defined as part of definition in Appendix B

4. A mobile will need to be able to execute a specific type of mobility protocol based on the policy.

6.4 Related work

At the time when this specific work was done and published in [DWB+02], [WDSY03], to the best of my knowledge there was no other prior work that proposed to use a multi-layer mobility scheme based on policy governed by layer 2 layer or application layer triggers. Soon after this work or around the same time, few other papers proposed integration of mobility protocols at multiple layers. Politis et al. [PCA+04] describe a multilayer mobility management involving SIP and MIP along with the enhancement of AAA architecture. Carli et al. [CNP01] describe how Mobile IP and Cellular IP can provide an integrated mobility solution.

Lee et al. [LLC03] describe a mobility management scheme that is based on the integration of Mobile IP and SIP. In this scheme, the client does not register with the SIP registrar even if its IP address changes. Thus, all the data still flow through the home agent. SIP registration is invoked only when personal or service mobility is needed. Thus, the advantage of SIP-based terminal mobility is not realized in this specific scheme. Zeadally et al. [ZSDR04] describe an architecture that integrates SIP and mobile IP to support seamless mobility. That proposal uses MIP to support terminal mobility and SIP to support personal mobility.

Kim et al. [KLPK04] describe how to route SIP messages as part of the Mobile IPv6 binding update message. However, in this mechanism, a home agent (HA) on home subnet acts as a redirect server and a registrar for SIP as well as a home router for Mobile IPv6. Thus, the binding cache in the HA contains the location information for SIP as well as home registration entries for Mobile IPv6. The method requires that the MIPv6 binding update needs to be modified so that it can carry the SIP registrations.

Wong et al. [Won02] provide some architectural alternatives for integrating mobile IP and cellular IP. These alternatives also highlight some of the drawbacks of using Cellular IP with MIP in co-located mode. Thus, the analysis suggests to use other global mobility protocols when mobile uses co-located care-of-address. Most recently, the IETF is considering a network controlled localized mobility protocol [Kem07] that adopts optimization technique similar to cellular IP and helps to reduce handoff delay during a mobile's movement within a domain. It introduces a new network element, the LMA (Local Mobility Agent) that acts as an anchor agent and identifies a specific mobility domain. Based on its movement, the mobile limits the binding update to the LMA domain.

In contrast with all of the proposals, my proposed mechanism does not need changes in the SIP or MIP specification and uses layer 2 and application layer triggers to determine the right mobility protocol based on mobile's movement pattern. To the best of my knowledge, my work is the first to be developed that uses a policy-based approach and takes advantage of global mobility and local mobility protocols to optimize the handoff performance.

6.5 Multi-layer mobility approach

Mobility protocols at each layer are best suited to work for a specific type of application that uses either TCP or RTP as the transport protocol and mobile's movement pattern such as inter-domain and intra-domain mobility. For example, an application layer mobility protocol may be suited to work for interactive traffic such as VoIP and a network layer mobility protocol works well to support TCP-based application. Cross layer optimization techniques help the mobile to reduce the handoff delay by avoiding the layer 3 re-configuration, limiting the binding update and reducing the media traversal by choosing the appropriate mobility protocol. Thus, during intra-domain movement, a local mobility protocol takes care of redirecting the traffic to the new point-of-attachment of the mobile and confines the binding update to the domain itself.

I have designed, prototyped and validated this cross layer mobility optimization scheme [DBW+02] using three mobility protocols at different layers, namely application layer SIP-based [WS99], network layer MIP-LR [DBJ+05] and layer 2-based MMP (Micro Mobility Protocol) [WWD+02] that operate in collaboration with each other. These mobility protocols are triggered by the cross layer information during the handover.

MMP is a modified version of Cellular IP [Val99] that provides additional survivable features in an ad hoc network by adding multiple gateways for each domain. A SIP-based mobility protocol is used for real-time traffic, and MIP-LR is used for non-real-time traffic during a node's movement between two different domains while MMP takes care of the movement within a domain. MMP is designed as a micro-mobility protocol to handle intra-domain mobility and works in conjunction with SIP and MIP-LR. To support real-time communication during the mobile's movement between the domains, the mobile sends a SIP re-INVITE to the CH to keep the session active. Similarly, a MIP-LR UPDATE message is sent to CH for the TCP/IP traffic. However, for any subsequent move within the new domain, re-INVITE or update messages are not sent, since MMP takes care of routing the packets properly within that domain. This helps to limit the binding update from traversing a long distance.

6.5.1 Policy-based mobility protocols: SIP, MIP-LR

For macro mobility, I use both SIP and MIP-LR. Although MIP-LR alone can handle macro mobility for both real-time and non-real-time traffic, I use MIP-LR for non-real-time traffic and use SIP for macro-mobility for real-time-traffic because:

1. SIP is already used for session control signaling for real-time applications, and mobility for these applications can be handled using the same signaling mechanisms.

2. SIP-based terminal mobility integrates well with SIP personal mobility (employing a unique URI for the user and obtaining the assistance of SIP proxies.

3. A SIP-based solution exists for smooth handoffs of real-time traffic streams.

In order to use both SIP and MIP-LR for macro mobility management, I use a policy table. Between the IP level and link layer processing, there is an entity that examines each IP packet and dispatches it to the appropriate handler. The decision is based on the policy table. For example, the MIP-LR software module can capture every IP packet and process every packet that is not related to real-time traffic (i.e., RTP packets or SIP signaling). The real-time traffic passes through untouched, and is redirected by the SIP application when the IP address changes. Figure 6.1 shows how SIP and MIP-LR can both manage mobility at the same time for RTP and TCP packets, respectively. Suppose a voice or video session (carried by RTP) and a file transfer (e.g., using ftp over TCP) are in progress at the same time. The MH starts in domain 1, where it is labeled MH (1st), referring to the first phase of its movement. The MH then moves to domain 2, where it is labeled MH (2nd), referring to the second phase of its movement. The solid arrow shows the movement of the MH between domains. When the MH detects that it is in a new domain (after arriving in domain 2), it performs auto-configuration. MIP-LR then updates the CH and HLR(s) with this new address, so the CH can update the destination IP address of the TCP packets. At the same time, SIP client (on the MH) issues a re-INVITE request and also updates the SIP registrar for location management. The SIP UA on the CH then informs the real-time applications that the address of the MH has changed. Additionally, for real-time traffic a fast handoff scheme could be deployed without affecting MIP-LR-based mobility management.

6.5.2 SIP and MIP-LR integration with MMP

Global update signaling time in SIP, as in MIP, can result in significant handoff latency. It has previously been suggested at a high level by Wedlund and Schulzrinne [WS99] that micro-mobility schemes could be used together with SIP to improve mobile's performance for micro-mobility situations. In this section I describe the details of how these two can coexist based on a specific policy.

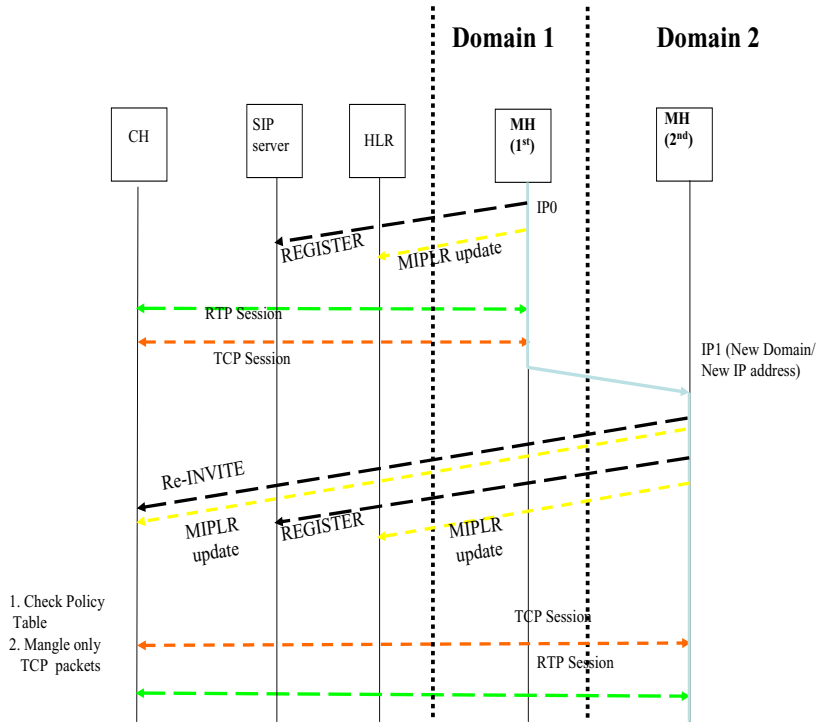


Figure 6.1: Integration of SIP and MIP-LR

6.5.2.1 SIP and MMP

I consider an example scenario in which an MH moves from one domain to another. While in the first domain, it initiates a SIP session with a CH. The MH then moves into the second domain (macro mobility), continuing the session. Within the second domain, the MH moves again (micro mobility), and the session continues. Figure 6.2 shows the signaling flow between the communicating nodes. The solid arrows show the movement of MH between domains and within the domain 2. In general, there might be a number of intermediate nodes with route caches between the MH and the gateway in each domain. Route cache in these intermediate nodes stores the routing information for the traffic between the gateway and the mobile nodes. These are not shown in the figure to reduce clutter. The scenario starts with the MH entering domain 1. The last hop MMP node is configured to send out beacon that contains the address of the gateway it belongs to. From the MMP beacon, it knows it is in a new domain; it auto-configures. There are several ways to do

this, and we illustrate an example later, where a variant of DHCP, namely DRCP (Dynamic Rapid Configuration Protocol) [MMWM01] is used for auto-configuration. DRCP is a light weight version of DHCP and has a lot of similarities with DHCP with rapid commit option [PKB05]. Both of these protocols are designed to reduce the time needed for IP address acquisition. In both the cases, number of signaling message exchange between the client and the server is reduced. However, in case of DRCP, the server sends periodic server advertisement so that the client can detect the presence of a new subnet after the handoff and can initiate the IP discovery process in unicast based on the address of the server received as part of server advertisement. The server can also send unicast-based offer to the client's address directly instead of sending it on a broadcast address. In both of these protocols, the client does not perform a duplicate address detection, instead the server does perform IP address check before it offers the address to the client.

Having obtained a local address from the DRCP server in domain 1, IP1, the client updates the MMP gateway. It should then send one or more SIP REGISTER messages to the appropriate SIP servers (not shown in the figure to reduce the clutter). Some time later, it initiates a SIP session with a CH. After a subsequent move into a new domain (domain 2), the MH listens to the gateway beacon and realizes that it is in a new domain. It auto-configures and sets itself up for micro-mobility management with its new local address. It then sends a SIP re-INVITE to the CH with its new address, so the SIP handoff can be completed with the CH changing the destination address of the packets it sends to the MH. The MH also sends one or more SIP REGISTER messages to appropriate SIP servers, which are not shown for brevity. When the MH moves again, its movement is confined to domain 2. Hence, it listens to the MMP beacon and knows that the move is only a local move. Therefore, it only updates the MMP gateway. SIP is completely uninvolved in the process because the IP address is unchanged. Compared to the inter-domain handoff, this intra-domain handoff incurs very low handoff delay.

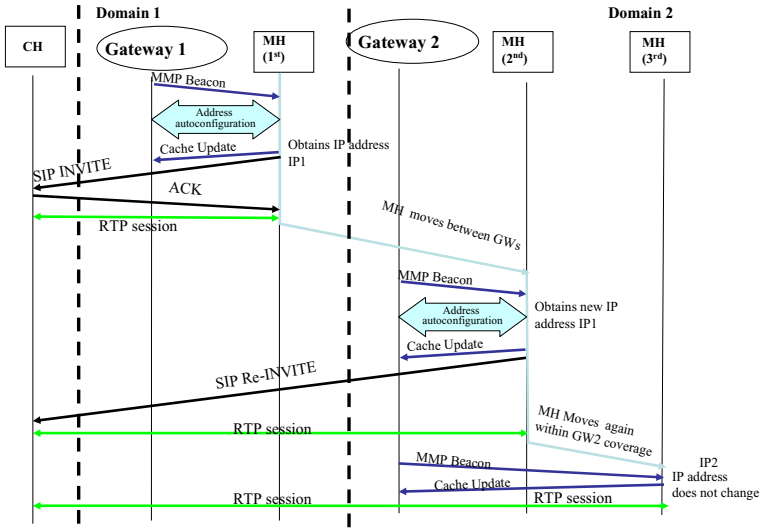


Figure 6.2: Integration of SIP and MMP

6.5.2.2 MIP-LR and MMP

I illustrate the integration of MIP-LR and MMP in Figure 6.3. While in the first domain, the MH initiates a TCP session (e.g., a file transfer) with a CH. In domain 1, the MH sends MIP-LR update messages to appropriate HLRs (not shown for brevity). Then it initiates a file transfer session. After moving into domain 2, the MH hears the gateway beacon, auto-configures, and performs micro-mobility setup signaling. It then sends a MIP-LR UPDATE to the CH with its new address. The MH should also send MIP-LR UPDATE messages to appropriate HLRs.

6.5.3 Integration of global mobility protocol with micro mobility protocol

Figure 6.4 illustrates a policy-based mobility management scenario where macro-mobility and micro-mobility protocols work together based on layer 2, layer 3 and application layer triggers. The figure shows how based on the type of application on the mobile, movement pattern of the mobile (e.g., movement between the subnets, between the cells within a sub-

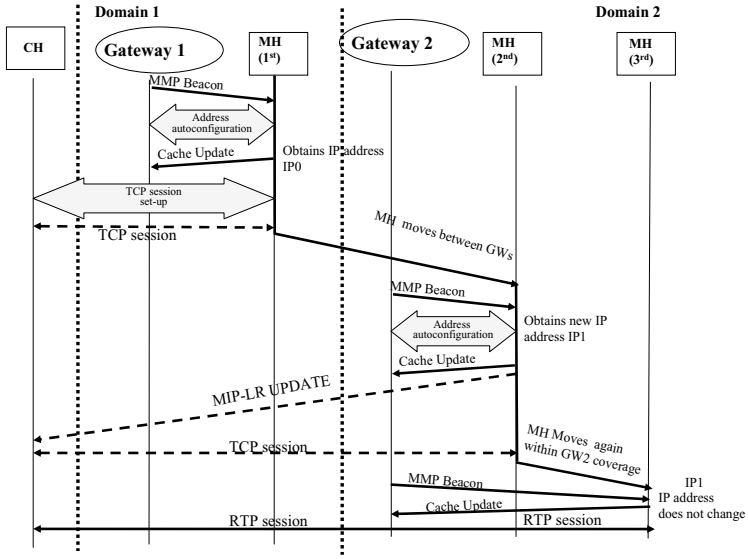


Figure 6.3: MIP-LR-MMP flow

net) a specific mobility protocol is used. The results shown in Figure 6.5 demonstrate how the mobile can obtain higher data throughput when a micro-mobility protocol is used for intra-domain movement compared to a macro mobility protocol because it limits the number of signaling message updates and the traversal updates. I have described the complete implementation of the cross layer mobility management scheme and experimental results in [DWB+02]. Simulation and experimental results for MMP are explained in [WWD+02]. I have described the architectural details of this scheme in [WDB+03].

6.5.4 Implementation of multi-layer mobility protocols

Figure 6.6 shows the setup of our Linux-based laboratory prototype using 802.11 wireless LAN (WLAN) for the wireless links. IP address management (including auto-configuration) is provided by DRCP and DCDP (Dynamic Configuration Distribution Protocol) servers. DRCP is a version of Dynamic Host Configuration Protocol (DHCP) optimized for wireless environments by way of reducing the size of the protocol headers and number of messages between the DHCP client and server. A DRCP server configures a node's interface with an IP address, and provides the addresses of DNS server, SIP server, and other application

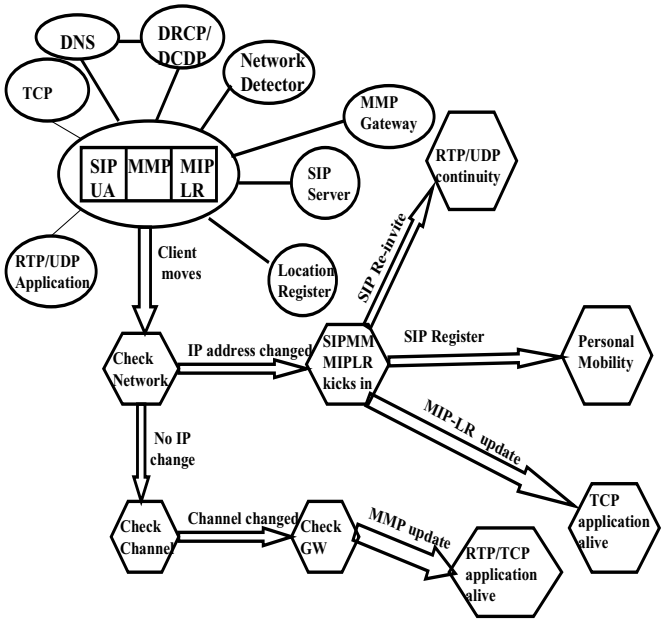


Figure 6.4: Policy-based mobility management

servers. Dynamic Configuration and Distribution Protocol (DCDP) works in conjunction with DRCP that distributes pools of IP addresses to the nodes in a quasi-static network so that these nodes become DRCP servers and dispense IP addresses to the clients. The current implementation of MIP-LR eliminates the tunneling function (and its encapsulation overhead) by using Linux's new *libipq* and *iptables* utilities to modify the packets (change IP header fields) appropriately at the endpoints.

The MH obtains a new IP address once it moves to a new domain, and keeps this IP address as long as it remains within this domain. This is handled automatically by DRCP. As shown in Figure 6.6, as the mobile node moves between the domains it uses SIP or MIP-LR depending on the type of application being supported. But while moving within a domain, mobility management is handled by MMP, where the gateway acts as a DRCP/DCDP server, and one of the MMP nodes acts as a DRCP server. For convenience in this testbed, all access points within a domain use the same WLAN frequency, whereas

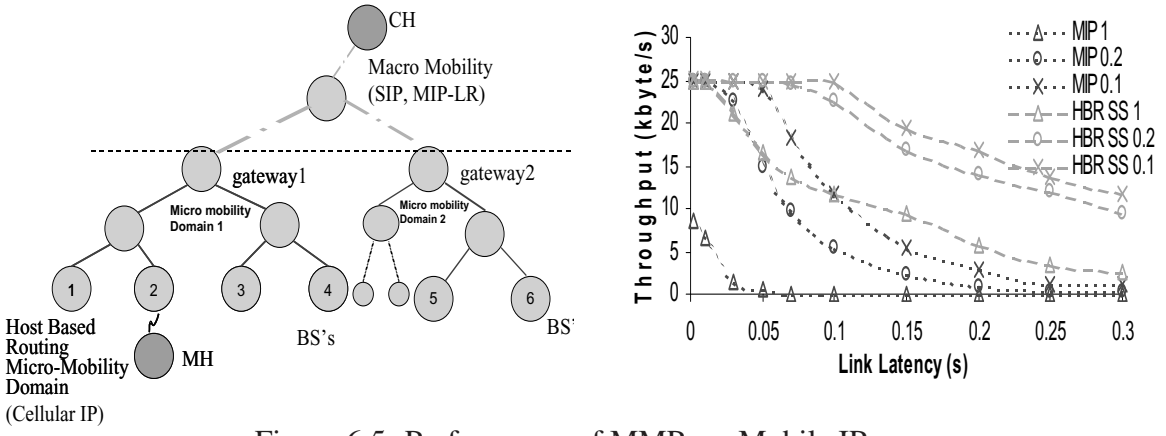


Figure 6.5: Performance of MMP vs. Mobile IP

access points in different domains use different frequencies. It is reasonable for all access points within a domain to use the same frequency, and micro-mobility handoff is optimized in this manner. However, using the same frequency in the adjacent channel may cause interference leading to lower capacity.

The MH acquires a new IP address using DRCP that triggers SIP Re-INVITEs. However, it takes time to change frequencies of the access networks and resume the physical layer connectivity and then to auto-configure with a new IP address. Furthermore, more packets are lost due to longer traversal path of the redirected traffic. The high-rate video traffic is 200 kb/s (whether one-way, high 1 way or in both directions, high 2 way), and the low 1 way (low-rate one-way) is 10 kb/s. The rate of dropped packets increases slowly with the data rates. However, a SIP-based fast handoff mechanism as discussed in Chapter 5 can be used here to reduce packet loss due to longer binding update.

6.5.5 Implementation and performance issues

SIP, MIP-LR, and MIP all provide binding update mechanism that updates mapping between a permanent address and a temporary one. With SIP, this is done with REGISTER (for pre-session mobility) and re-INVITE (for mid-session mobility). With MIP and MIP-

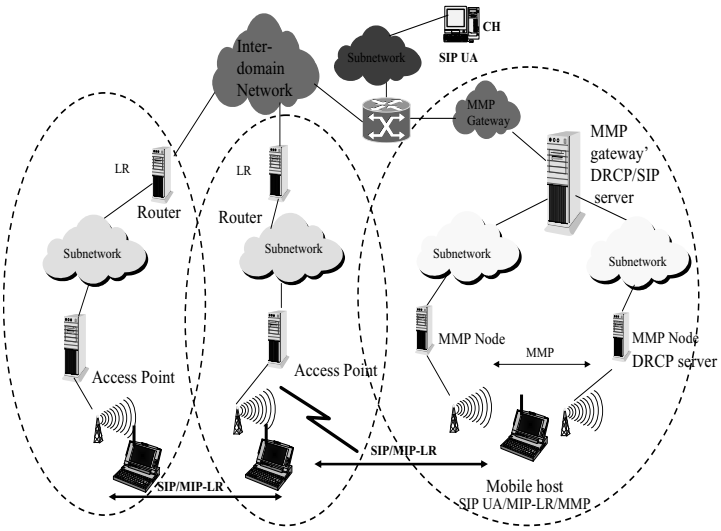


Figure 6.6: Integrated mobility management

LR, this is done with registration (with HAs and HLRs, respectively). MIP (with route optimization), SIP, and MIP-LR all allow binding updates for CHs to route packets directly to the MHs after mid-session mobility. SIP servers and MIP-LR location registers can be replicated for survivability.

How well does the new mobility management scheme meet the requirements stipulated earlier? By virtue of the use of macro-mobility protocols like SIP and MIP-LR, the triangular routing problem of MIP is eliminated. I have found that this significantly increases routing efficiency when the home network of the MH is far from the visited network and the CH is closer to the MH. This scheme has much less overhead than MIP because encapsulation is not used by any of the component protocols, and the use of micro-mobility significantly reduces the global signaling overhead. Avoidance of triangular routing and absence of encapsulation contribute to low latency for both real-time and non-real-time communication. The scheme is survivable by having SIP proxies and multiple HLRs that act like dynamic HAs. In general, the MH maintains a current list of SIP proxies or HLRs that can be contacted prior to a session or during communication.

I investigated the performance of the multilayer mobility management scheme using the

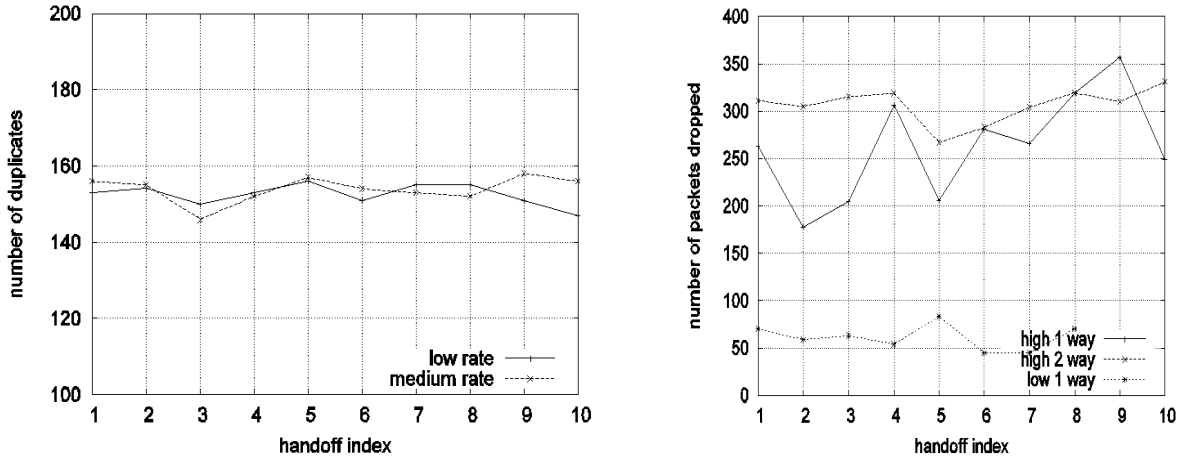


Figure 6.7: Results from integrated mobility management

laboratory testbed as shown in Figure 6.6. I used SIP to initiate a video session between the MH and CH. During the movement of the MH, both micro-mobility and macro-mobility handoffs occurred. For micro-mobility handoffs (within a domain), since the two access points are on the same frequency, the handoff does not require binding to a new frequency. The IP address also remains unchanged. The only difference (for MH transmitting) is that the default gateway and destination medium access control (MAC) address of the outgoing packets are changed to the new access point. This results in practically no disruption in outgoing packets. For incoming packets, since the mobile can receive packets from both access points (same frequency), no dropped packets are observed. However, there was a short time during the handoff when the same packets were transmitted through both access points, resulting in duplicate packets. Figure 6.7a shows the number of duplicate packets measured at different handoffs. By *handoff index* I simply mean a specific handoff number out of all the handoffs considered. The variation of dropped packets is low (less than 5 percent), and this number does not change significantly when the video bit rate doubles from 10 kb/s (low rate) to 20 kb/s (medium rate); This is because when the packet size changes, the packet rate is roughly the same. Duplicate RTP packets should not pose a

problem to most streaming video receivers. However, duplicates could be eliminated by performing a hard handover in the MMP gateway between sending the packets to the old and new access points. However, this may lead to a few dropped packets.

Figure 6.7b shows handoff behavior when the same MH moves across domains. The MH acquires a new IP address using DRCP that triggers SIP Re-INVITEs. However, it takes time to change frequencies of the access networks and resume the physical layer connectivity and then to auto-configure with a new IP address. Furthermore, more packets are lost due to longer traversal path of the redirected traffic. The high-rate video traffic is 200 kb/s (whether one-way, high 1 way or in both directions, high 2 way), and the low 1 way (low rate one-way) is 10 kb/s. The rate of dropped packets increases slowly with the data rates. However, a SIP-based fast handoff mechanism as discussed in Chapter 5 can be used here to reduce packet loss due to longer binding update.

6.5.6 Concluding remarks

In the course of developing and designing the prototype testbed, the following observations were made.

- Care must be taken to be consistent regarding the IP address the MH uses to identify itself in micro mobility zones. The IP address the MH uses to identify itself is the address that is stored in the route caches of the MMP nodes. When this is mobile's home address, we found it works best with FAs collocated with the micro mobility gateway (e.g., MIP-LR can be used with FAs). Otherwise, packets will arrive for the MH addressed to its auto-configured foreign network address, and the route caches need to associate the two addresses (this can be handled by an MMP extension, but is less elegant). Conversely, when identifying itself by its foreign network auto-configured address, it works best without FAs, since the route caches would be set up to forward with the foreign network address in this case.

- ◇ Separation of real-time and non-real-time traffic is becoming practically more reasonable. With standard tools like iptables for Linux 2.4.7-10 and above, it is easy to set policy-based handling of different types of traffic, say, to do MIP-LR processing only for non-RTP packets, bypassing SIP signaling packets, and RTP packets based on the port numbers.

- ◇ There are other significant contributors to macro mobility handoff latency besides MIP signaling latency. We found that the complete auto-configuration process of IP address distribution using DCDP and IP address configuration using DRCP can take a few seconds, including reconfiguration of the wireless interface. In fact, our testbed typically did not have high network latency, but macro mobility handoff latency was still significantly higher than that of micro mobility handoff.

- ◇ Changing the IP address as a result of mobility may require slight application-level changes. For MIP-LR-based macro mobility, applications are unaware of IP address changes with mobility. However, for SIP macro mobility I had to modify our video and audio applications (VIC and RAT, respectively, both available as freeware on Linux), and added modification for interprocess communication with SIP UAs. In general, a mobility-aware RTP stack should be built to adapt itself to IP address change. Some recently built RTP stacks (www.vovida.org) are in fact mobility-aware and adapts itself to the mobility changes.

Chapter 7

Optimizations for simultaneous mobility

In this chapter, I analyze the problem due to non-receipt of binding update that results when both the mobile nodes move simultaneously and propose optimization techniques that increase the successful handover probability under the simultaneous mobility scenario. These optimization techniques could be applied to mobility protocols at several layers - network layer mobility protocols such as MIPv6 [JPA04] and MIP-LR [JRY+99] and application layer mobility protocol such as SIP-based mobility [SW00].

7.1 Summary of key contribution and indicative results

Without any thorough analysis of the simultaneous mobility problem that arises due to non-receipt of binding updates when both the hosts that are in communication move, it is difficult to predict the parameters that affect the simultaneous mobility and propose solutions to mitigate these problems. Prior to my work, there was no comprehensive study that analyzes the simultaneous mobility problem nor there is any existing solution to mitigate these problems in an infrastructure-based mobility environment.

I analyze the simultaneous mobility problem and develop an analytical framework to study the effect of inter-handoff rate of the mobile and binding update latency on the probability of occurrence of simultaneous mobility problem. I proposed timer-based retrans-

mission, forwarding and redirecting mechanisms using binding update and location update proxies and use of simultaneous bindings by the mobile to eliminate the vulnerability of binding update due to simultaneous mobility. I applied these solution mechanisms to several application layer and network layer mobility protocols namely, SIP-based mobility, MIPv6 and MIP-LR.

My proposed analytical framework for simultaneous mobility can predict the probability of simultaneous mobility based on the mobile's inter handoff time and binding update latency. Each of my proposed techniques can be applied either at the sender side or receiver side and work for both network layer and application layer mobility protocols unlike protocol specific mechanisms proposed by Tilak and Ghazaleh [TAG01] and Dreibholz et al. [DJT03] that use TCP migrate and SCTP extensions. Each of my proposed solution mechanisms reduces the vulnerability interval of simultaneous binding update.

In the rest of the chapter, I introduce the simultaneous mobility problem and illustrate this problem for different mobility protocols, develop the analytical framework, prove lemmas covering two cases of simultaneous mobility and propose solution mechanisms that can be applied to few network layer and application layer mobility protocols, namely MIP-LR, MIPv6 and SIP.

7.2 Introduction

Stoica et al. [ZLS+05] propose seven properties that are needed to fully realize the promise of ubiquitous mobility. These properties also include simultaneous mobility. It is expected that non-simultaneous mobility in most scenarios would occur more frequently than simultaneous mobility. Non-simultaneous mobility refers to mobility of one end host while the other remains stationary. Nevertheless, simultaneous mobility would happen once in a while and must be handled properly by the mobility protocols.

Simultaneous mobility problem occurs when two mobile nodes that are part of a com-

munication session in normal state, and they both move such that the binding updates that they send to each other are both lost through belated arrival of binding update, and such that the communication session never returns from interrupted state to normal state. More precisely, simultaneous mobility problem can be defined as the problem of losing a binding update from one mobile node because it is sent to a previous address of the other mobile node that is also moving at around the same time. The disruption caused by the simultaneous mobility problem may far exceed the disruption caused by non-simultaneous mobility.

Thus, the optimization techniques related to simultaneous mobility is a special type of optimization for binding update as discussed in Chapter 5. Any solution for simultaneous mobility should ensure that the end hosts should be able to move simultaneously without breaking an ongoing session between them due to delayed binding update.

7.2.1 Analysis of simultaneous mobility

In this section, I analyze the simultaneous mobility event and describe several concepts associated with simultaneous mobility.

I primarily limit the analysis of simultaneous mobility to layer 3 handoff only, i.e., where IP addresses of the mobile nodes change. A binding update carries the information about the location of the sending mobile host including its new IP address. A binding update is lost if it does not arrive at its intended recipient mobile host. It makes a belated arrival if it arrives at a network where the destination address used to be valid for the intended recipient Mobile Host but it is no longer valid at the moment of arrival. Binding updates do not contain information about future moves of the sending mobile host. While two mobile hosts are in a communication session, they get information on the location of the other Mobile Host only from binding updates. In other words, they do not actively seek the location of the other mobile host, but only passively accept binding updates. Binding updates are sent directly to the most current known address (known by the sender) of the intended recipient mobile host. In general, the latency associated with the binding updates

is assumed to be much smaller than the average inter-handoff time, therefore, it is extremely unlikely that a binding update would be sent and the recipient mobile host would move twice before the binding update arrives at the previous network of the recipient.

The most basic version of simultaneous mobility problem is shown in Figure 7.1. There are two nodes, A and B. Time is in the vertical direction (and flows downward), whereas spatial location is in the horizontal direction. Node A moves from domain A1 to A2 while Node B moves from domain B1 to domain B2. After their respective moves, these two nodes send binding updates to the other node, and both binding updates are lost. Additionally, there are proxies and servers in the network as well.

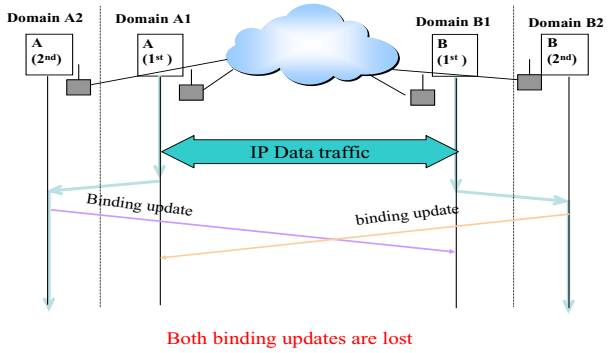


Figure 7.1: Simultaneous mobility scenario

Standard mobility protocols like the original Mobile IP (MIP) handle simultaneous mobility adequately, because of non-mobile home agents. The home agent of the Mobile Host functions as an anchor point for the Mobile Host. No matter where the Mobile Host moves, packets for it always go first to its home network for interception and are tunneled by its home agent. If it turns out that the Correspondent Host is also mobile, it will also have a home agent and packets from the Mobile Host will similarly be intercepted and tunneled to the appropriate network by its home agent. Since both home agents are stationary and can always be reached through IP routing, simultaneous mobility does not present a problem to MIPv4.

However, simultaneous mobility problem occurs for scenarios when the end hosts can send the binding updates directly to each other. Therefore, I analyze the simultaneous

mobility problem for network layer mobility protocols such as MIPv6 and MIP-LR, and application layer mobility protocol such as SIP, and propose a common framework for the solution. It is important to note that the problem of simultaneous mobility is very similar in these protocols because these protocols allow binding updates to be sent to the communicating hosts directly. My proposed solutions are designed to impose minimal changes on the existing protocols while efficiently dealing with the simultaneous mobility problems. I focus on situations where the handoff rate of a mobile node is such that consecutive handoffs of the same mobile node are non-overlapping. I do not focus on the situations of overlapping consecutive handoffs of the same mobile node, where one handoff has not completely finished before the next one begins, e.g., there has not been enough time after the acquisition of an IP address for binding updates to reach their destination networks.

The following are the reasons for assumptions:

- ◇ The problems encountered with overlapping consecutive handoffs are not so much a problem of simultaneous mobility as one of excessive handoff rate. There will be severe problems leading to complete deadlock situations when the mobile node changes its IP address before binding updates for its previous IP address have even arrived at their destinations.

- ◇ For the foreseeable future, the extreme case of handoff rates high enough for overlapping consecutive handoffs is highly improbable. Hence, I assume that consecutive handoffs of the same mobile node are non-overlapping, and I focus on overlap of handoffs of different mobile nodes, i.e., simultaneous mobility.

7.3 Illustration of the simultaneous mobility problem

In this section, I illustrate how simultaneous mobility problems are encountered for SIP-based mobility, MIPv6 and MIP-LR.

Figure 7.1 is easily adapted to illustrate the simultaneous mobility problem with SIP and has been shown in Figure 7.2.

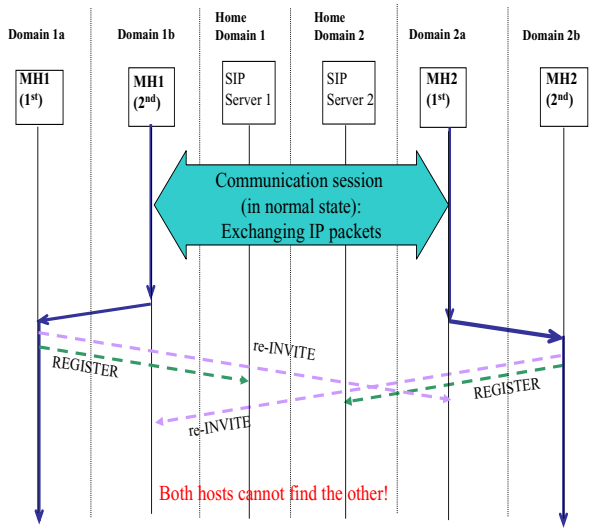


Figure 7.2: Simultaneous mobility in SIP

As shown in Figure 7.2, the binding updates are denoted as re-INVITE messages. The main difference is that there are two additional SIP servers with proxy and redirect functionality in each network, one for each mobile node in its home network. In general, when the re-INVITE messages are lost, the servers do not intervene, but I have proposed a solution that could use these servers and solve the simultaneous mobility problem in SIP-based mobility. These proposed solutions are explained in Section 7.6. Furthermore, Figure 7.2 could also be used to illustrate MIP-LR's simultaneous mobility problem. In MIP-LR, the SIP servers are replaced with MIP-LR home location registers (HLR), and re-INVITE messages are replaced with MIP-LR binding updates.

Similarly, MIPv6 is vulnerable to the simultaneous mobility problem because of the direct binding updates and associated return routablity procedures. The direct binding updates from the mobile node to the correspondent nodes pose a security problem. Thus, the return routability procedure allows the mobile node and correspondent node to set up a shared key in a "reasonably" secure manner. In the return routability procedure, a mobile node sends two messages to the correspondent node, namely the Home Test Init (HTI) and

the Care-of Test Init messages (CTI). These messages are sent through the Home Agent (reverse tunneled to the Home Agent from the mobile node, and then forwarded to the correspondent node) and directly to the correspondent node, respectively. The correspondent node replies by sending two tokens to the mobile node, one directly to the mobile node addressed to its care-of address (the Care-of Test message), and the other with the home address of the mobile node (the Home Test message). The mobile node needs both the tokens to be able to generate the shared key. Thus, the return routability procedure ensures that the mobile node is who it claims to be by testing that it is reachable on both the direct path and through its home address. Subsequently, the correspondent node can accept binding updates directly from the mobile node.

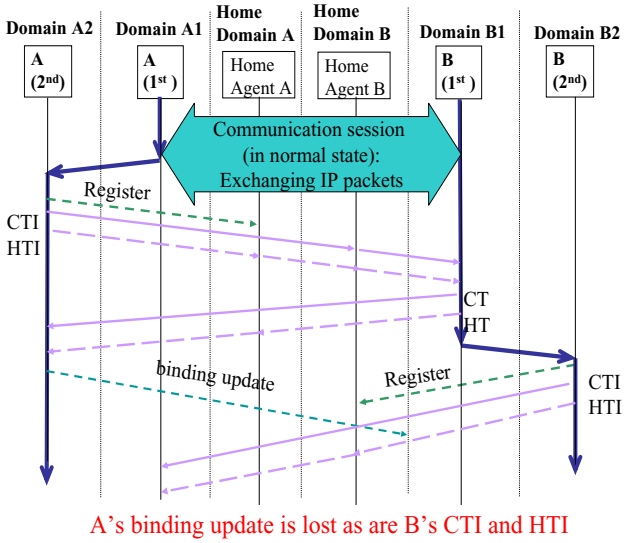


Figure 7.3: Simultaneous mobility for MIPv6

However, the additional message exchange due to return routablity procedure adds to the existing simultaneous mobility problem. Figure 7.3 illustrates the simultaneous mobility problem with Mobile IPv6. Following are three different possible scenarios that could possibly result in simultaneous mobility in MIPv6.

1. Both sides' CTI and HTI messages are lost because of simultaneous mobility. This would look like Figure 7.2 except that CTI and HTI messages are lost instead of re-INVITE (and home agents are used instead of SIP servers).

2. One side actually completes return routability, but then its binding update is lost because the other side moves. This interesting asymmetric scenario is illustrated in Figure 7.3.

3. Both sides complete the return routability checks, but then their binding updates are lost due to simultaneous mobility.

I propose solutions to take care of simultaneous mobility problems for SIP-based mobility, MIP-LR and MIPv6 in Section 7.6.

7.4 Related work

There are only a few papers that discuss about simultaneous mobility. Tilak and Ghazaleh [TAG01] extend the TCP migration mobility protocol [SB00] to handle simultaneous mobility, but there are significant differences between the TCP migration schemes (where mobility is handled at the transport layer) and MIP-related protocols or SIP-based mobility protocols. Dreibholz et al. [DJT03] propose a scheme that handles simultaneous mobility at a layer between the transport and application layer. In that scheme, mobility is handled using stream control transmission protocol (SCTP) extensions. However, no analytical framework or theorems and proofs related to the simultaneous mobility problem for SIP, MIPv6 and MIP-LR have been proposed before.

As part of my research, I have analyzed the simultaneous mobility problems for MIP-LR, SIP and MIPv6 in [WDSY03] and [WD05]. I have also developed some common approaches that could be applied to provide solutions to mobility protocols such as MIPv6, MIP-LR and SIP-based mobility. I have described these results in [DWDSY07].

7.5 Key optimization techniques

Following are some of the key fundamental techniques that should be considered to optimize the handoff event during simultaneous mobility of the communicating hosts.

1. Many of the principles related to optimization for binding update are applicable to simultaneous mobility scenario. However, the handoff rate of the mobile node will determine if any of those techniques can be applied to simultaneous mobility scenario.

2. Reduce the effect of delayed direct binding updates by introducing an anchor point closer to the mobile.

3. Limit the traversal distance of binding updates.

4. Forward the binding updates from the previous network and cache it in a forwarding agent closer to the mobile.

5. Apply retransmission of binding update by the mobile nodes and proxies to complete update.

6. Apply simultaneous binding update by the mobile to reduce the failure probability of reconnection.

7.6 Analytical framework

In this section, I introduce an analytical framework to analyze simultaneous mobility problem. I define some of the fundamental concepts that are used to analyze the simultaneous mobility framework.

7.6.1 Fundamental concepts

In this section, I introduce some fundamental concepts that are used to study the analytical framework associated with simultaneous mobility. In particular, I describe the terms such as handoff sequences and binding updates.

Definition Two mobile nodes are in a communication session if they are actively exchanging data. A communication session may be in a normal state or interrupted state. The session is in a normal state when data from one node is arriving at the right location for the other node, and vice versa. It is in an interrupted state otherwise.

Example A communication session typically is in an interrupted state from the moment a handoff occurs, until data starts arriving again at the new attachment point (e.g., after a binding update is received at the other node). An illustration of this alternation between normal state and interrupted state is shown in Figure 7.4. I explain this figure in more details.

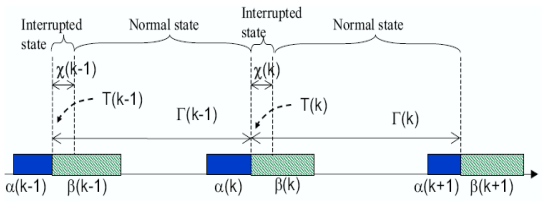


Figure 7.4: Simultaneous mobility framework notation

7.6.2 Handoff sequences

As defined in Appendix C, handoff is a movement of a mobile node from a previous attachment point to a new attachment point. During simultaneous mobility, the handoff time of a handoff instance is the moment in time when it changes from being reachable at the previous attachment point to not reachable at the previous attachment point. Let the handoff time (of a particular handoff instance) be T. Then the node needs time for network configuration, so it becomes reachable (with a valid IP address at the new network) at time T + γ.

If there is a correspondent node, then some time later, T + γ + ζ, it sends a binding update to the correspondent node. The binding update arrives at time T + γ + ζ + Δ. I use these symbols to represent these differential times after a handoff. For convenience I may write X(i) = γ(i) + ζ(i) + Δ(i) as shown in Figure 7.4. So T(i) + X(i) denotes the time when the binding update arrives at the other node. Given that time is continuous, I assume that only one handoff can occur at any given moment in time, i.e., handoff times are unique. It is to be noted that definition of handoff and handoff time may not be applicable to certain types of IP-layer soft handoff or physical layer soft handoff, as in CDMA systems and bicasting or multicasting schemes.

Figure 7.5 shows a scenario where A and B are two mobile nodes that are in a communication session with each other, during which each node performs zero, one or more handoffs.

Definition The handoff sequence of A is the ordered set

$$H_A = T_A(0), T_A(1), ...T_A(N_A - 1) \tag{7.1}$$

and the handoff sequence of B is the ordered set

$$H_B = T_B(0), T_B(1), ...T_B(N_B - 1) \tag{7.2}$$

where $T_A(i)$ is the handoff time of the ith handoff of A so that $T_A(i) < T_A(j)$ $\forall$ i,j such that 0 < i < j < N_A − 1 and same holds good for B. The function arguments i,j, are the handoff index number. In general, when necessary, we will use subscripts to indicate the mobile node and we will show the handoff index number in function arguments.

Two handoffs are consecutive (with respect to a pair of mobile nodes) if neither of the mobile nodes performs another handoff in between the two handoffs. For example, if the

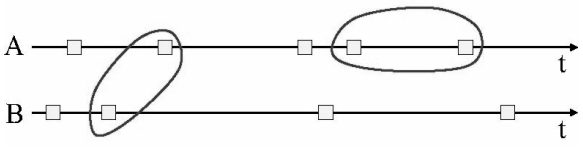


Figure 7.5: Examples of consecutive handoffs

two handoffs are at A and B, at times $T_A(i_0)$ and $T_B(j_0)$, and suppose that A's handoff is earlier, then saying they are consecutive is equivalent to saying $T_A(i) \in H_A : T_A(i_0) < T_A(\text{i}) < T_B(j_0) = \emptyset$ and $T_B(j) \in H_B : T_A(i_0) < T_B(j) < T_B(j_0) = \emptyset$. As defined, then, consecutive handoffs could be at the same mobile node or at two different mobile nodes. Figure 7.5 shows two examples of consecutive handoffs, one in which the two handoffs are at different mobile nodes and one in which they are at the same mobile node.

7.6.3 Binding updates

Definition A binding update is lost if it does not arrive at its intended recipient node.

Definition A binding update makes a belated arrival if it arrives at a network where the destination address used to be valid for the intended recipient node but is no longer valid (for the intended recipient) at the moment of arrival. For example, if A is the sender and B is the intended recipient, and we are considering the binding update for A's ith handoff, then if B's next handoff is its jth, then the binding update makes a belated arrival if and only if $T_A(i) + \gamma_A(i) + \zeta_A(i) + \delta_{A \to B}(i,j) > T_B(j)$

Definition A binding update is lost through belated arrival if it makes a belated arrival and is consequently lost.

A node can be lost not necessarily through belated arrival, but through other possible causes of lost binding updates, such as network congestion, node failure, link failure. Conversely, a node can make a belated arrival and not be lost, e.g., if there is an agent in that network that can forward the binding update to the current location of the intended recipient.

Furthermore, the following assumptions are made about binding updates for simultane-

ous mobility.

1. Binding updates cannot and do not contain information about future moves of the sending node.

2. While two nodes are in a communication session, they get information on the location of the other node only from binding updates, i.e., they do not actively seek the location of the other node, but only passively accept binding updates.

3. Unless otherwise stated, a binding update is sent directly to the most current known address (i.e., known by the sender) of the intended recipient.

4. Regarding the relative timings of binding update latencies and consecutive handoffs of a receiving mobile node, the timescale of the latencies for binding updates is assumed much smaller than the average inter-handoff time. In other words, $\delta << ET(i+1) - T(i)$, where $E(.)$ denotes expectation.

5. It is extremely unlikely that a binding update would be sent and the recipient moves twice before the binding update arrives at the previous network of the recipient.

6. It is also assumed that if there is a forwarding location proxy (defined in Section 7.5.4) in the previous network of the recipient, it will correctly forward the binding update to the recipient, which would only have moved once from the previous network.

7.6.4 Location proxies and binding update proxies

Here I introduce two kinds of stationary proxies for mobility signaling. These proxies, if used carefully, can help prevent the simultaneous mobility problem. These proxies are abstract proxies - the definitions are more about network functionality than specific implementations as network elements.

Thus, it can be seen how familiar network elements like Home Agents can be described as having certain proxy functions, or can be enhanced for such purposes. The abstraction of these proxies will allow general problems and solutions (related to simultaneous mobility) to be discussed without unnecessarily being bogged down by details of specific mobility protocols. It is also assumed that these proxies should be stationary, not mobile.

7.6.4.1 Location proxy

A location proxy (of a mobile node) is a network function that is used to locate the mobile node. There can be three kinds of location proxies. A *forwarding location proxy* will forward messages (including binding updates) to the most recent location that it knows for the mobile node. A *redirecting location proxy* will redirect (e.g., by responding to a query with the latest address) messages to the most recent location that it knows for the mobile node. An *intercepting location proxy* intercepts, and may act on (forwards or redirects), messages in packets not addressed to it. A non-intercepting location proxy only acts on messages in packets addressed to it. The fundamental differences between the types of proxies are shown in Figure 7.6. It is important to note that whereas a forwarding location proxy will pass along messages toward the final destination, a redirecting location proxy will not do this, but just return location information that can be used to send the message toward the final destination. A location proxy is up-to-date with respect to a particular mobile node, if that mobile node continually updates the proxy with its latest address after each move.

7.6.4.2 Proactive location proxy

A *proactive location proxy* keeps a copy of mobility related signaling messages (typically, binding updates, but possibly other messages like Care-of-Test Init, when procedures like the return routability are used before the binding update is sent). It keeps the messages for a short while, X, after receiving and acting on them (i.e., after redirecting and/or for-

warding the message). The messages are kept in the location proxy cache and indexed by the destination node. The messages are discarded after time *X* has elapsed. If during this period of time, the pro-active location proxy receives a binding update from any one of the destination nodes in its location proxy cache, it either (a) redirects to the new address (if it is a redirecting pro-active location proxy); or (b) forwards the corresponding saved message(s) to it, (if it is a forwarding pro-active location proxy).

Here are examples of how some of the mobility components within different mobility protocols behave as different kinds of proxies. The Mobile IP home agent is a forwarding location proxy (of the intercepting kind). DNS servers are non-intercepting redirecting location proxies. MIP-LR Home Location Registers are non-intercepting redirecting location proxies. SIP proxy servers are non-intercepting proxies that can be either forwarding location proxies known as proxy servers in SIP terminology or redirecting location proxies known as redirect servers in SIP terminology. Except for DNS servers, the other examples given here are typically used in mobility schemes as up-to-date location proxies. In TCP migration, though, DNS servers are part of the mobility scheme, and so they are up-to-date location proxies in that scheme. Most existing location proxies are not proactive location proxies. However, proactive location proxies (defined in Section 7.5.4) may be useful to provide solutions to the simultaneous mobility problem. The current solutions take into account signaling only. It is assumed that if the mobility signaling gets to its intended recipient, the mobility schemes should, and must, take care of the data traffic correctly. However, in some cases, e.g., with Mobile IP, the Home Agent forwards both signaling and data, whereas in other cases, e.g., MIP-LR and SIPMM, the location registers or SIP servers are only involved in the signaling.

7.6.4.3 Binding update proxy

A binding update proxy acts on behalf of a mobile node to send its binding updates to its correspondent node's latest addresses. It would typically engage the services of a location

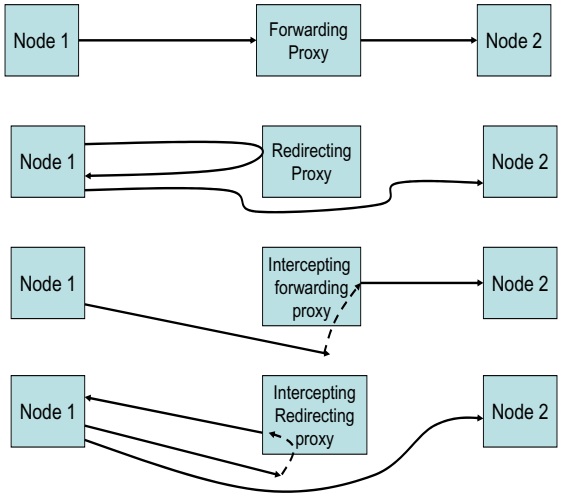


Figure 7.6: Abstract functions of the location proxies

proxy of each correspondent node either for redirection to the correspondent node's latest address, or for forwarding the relevant binding update. At the same time, it also forwards a copy of the message to the latest address it knows for the correspondent node. A mobile node on whose behalf a binding update proxy acts may be referred to as a master of that binding update proxy.

7.6.4.4 Proactive binding update proxy

A proactive binding update proxy not only queries for the latest addresses of the correspondent nodes of its master(s); it also keeps the binding updates for a short while α after receiving and forwarding them. The messages are kept in the binding update proxy cache and indexed by destination node. The messages are discarded after time α has elapsed. If during this period of time, the pro-active binding proxy receives a redirection regarding any one of the destination nodes in its binding update proxy cache, it forwards the corresponding saved binding update(s) to it.

7.7 Analyzing the simultaneous mobility problem

I now prove four lemmas that cover the two cases of what happens when there is a pair of handoffs at two mobile nodes, and the binding update from the earlier one arrives (a) later than the time the other node moves (Lemmas 3.1 and 3.2); or (b) earlier than the time the other node moves (Lemmas 3.4 and 3.5).

Lemma 3.1 Given a pair of consecutive handoffs, one for each of the two mobile nodes in a communication session in normal state, in the absence of location proxies for either mobile node, any binding update sent by the earlier moving mobile node will be lost through belated arrival, if and only if the binding update does not arrive at the other mobile node before it moves.

Proof Suppose without loss of generality that node A moves before node B. Let the handoff times be $T_A(i_0)$ and $T_B(j_0)$, so $T_A(i_0)<T_B(j_0)$. Since node A and node B are in a communication session in normal state up till $T_A(i_0)$, then up till $T_A(i_0)$, anything sent by A arrives at B and vice versa. Since the two handoffs are consecutive, then by definition there is no other handoff in the time interval $[T_A(i_0),T_B(j_0)]$. By our third assumption on binding updates, A's binding update would be addressed to the latest address it has for B. So for time interval $[T_A(i_0),T_B(j_0)]$, anything sent by A will still be addressed to B's pre-handoff address, and still arrives at B, including A's binding update. However, as soon as $t \geq T_B(j_0)$, B would no longer be reachable at its pre-handoff address. In some scenarios, a location proxy for B would be able to prevent the binding update from being lost. However, in the absence of a location proxy, the binding update would just go to B's previous address and disappear there. Hence, it would be lost through belated arrival.

Conversely, suppose A's binding update is lost through belated arrival. As shown, for time interval $[T_A(i_0),T_B(j_0)]$, anything sent by A will still be addressed to B's pre-handoff address, and still arrives at B, including A's binding update. So if it arrives before $T_B(j_0)$, it will not be lost through belated arrival. Thus, A's binding update cannot arrive before $T_B(j_0)$. Therefore it arrives after B has moved. This lemma and the next are making as-

sertions about cases where the binding update from the earlier-moving mobile node arrives after the later-moving mobile node has moved. This is shown in Figure 7.7.

Lemma 3.2 Given a pair of consecutive handoffs, one for each of two mobile nodes in a communication session in normal state (up until the first handoff). In the absence of location proxies for either mobile node (or there might be location proxies but they are not used or involved), the simultaneous mobility problem will occur, if and only if the binding update sent by the earlier moving mobile node does not arrive at the other mobile node before it moves.

Proof I use A and B as the first and second mobile nodes again. If the binding update from A does not arrive at the other node before it moves, then by Lemma 3.1, it is lost through belated arrival. Then, at time $T_B(j)$, B does not have A's new address. Since A's binding update is lost, by the time B is sending its binding update (i.e., $T_B(j)+\lambda_B(j)+\zeta_B(j)$) it will send it to A's previous address. Thus, in the absence of location proxies, B's binding update will also be lost through belated arrival. So both A's and B's binding updates are lost through belated arrival. By definition, the simultaneous mobility problem has occurred.

Conversely, if the simultaneous mobility problem has occurred, then by having both the binding updates having lost through belated arrival, so A's binding update is lost through belated arrival. From the above proof, the following corollary emerges.

Corollary 3.3 Given a pair of consecutive handoffs, one for each of two mobile nodes in a communication session in normal state (up until the first handoff), in the absence of location proxies for either mobile node (or there might be location proxies but they are not used or involved), if the binding update from the node that moved first is lost through belated arrival, the binding update from the node that moved second will also be lost through belated arrival.

Lemma 3.4 Given a pair of consecutive handoffs, one for each of two mobile nodes in a communication session in normal state (up till the first handoff), the simultaneous mobility problem does not occur if the binding update from the node that moved earlier reaches the

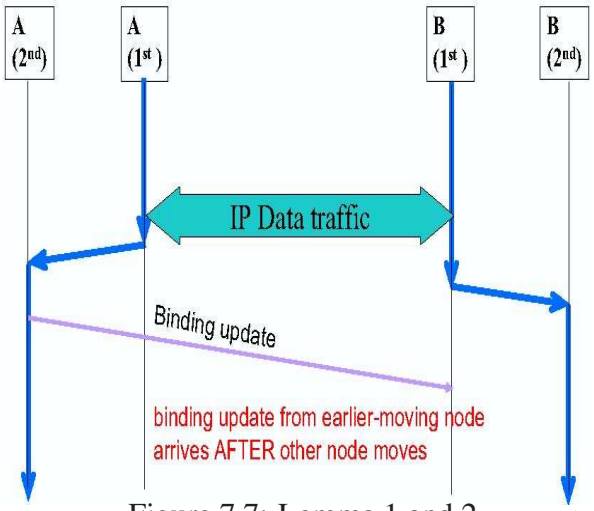


Figure 7.7: Lemma 1 and 2

other node before that node moves.

Proof As in the proof of Lemma 3.1, one can argue that for time interval $[T_A(i_0),T_B(j_0)]$, anything sent by A still arrives at B, including A's binding update. Thus, B can then send its binding update correctly to A's new address after it moves at $T_B(j_0)$. Therefore, the simultaneous mobility problem does not occur.

This lemma and the next are making assertions about cases where the binding update from the earlier-moving mobile node arrives before the later-moving mobile node has moved. This is shown in Figure 7.8. NB: the converse of Lemma 3.1 is not necessarily true, i.e., one cannot say that if the simultaneous mobility problem does not occur, then the binding update from the node that moved earlier reaches the other node before that node moves. The reason this is not necessarily true is that location proxies could be used, as I will demonstrate later. However, I first extend Lemma 3.4 to the case that location proxies are excluded, where I can make a stronger statement.

Lemma 3.5 Given a pair of consecutive handoffs, one for each of two mobile nodes in a communication session in normal state (up till the first handoff), in the absence of location proxies for either mobile node (or there might be location proxies but they are not used/involved), the simultaneous mobility problem does not occur if and only if the binding

update from the node that moved earlier reaches the other node before that node moves.

Proof This has been partially proved in the proof for Lemma 3.4. What remains is to prove that if the simultaneous mobility problem does not occur, the binding update from the node that moved earlier reaches the other node before that node moves. Supposing the simultaneous mobility problem does not occur, that means both binding updates arrive at the other node. B's binding update therefore cannot be lost through belated arrival, so B must have successfully received A's binding update. By the 3rd assumption on binding updates, A's binding update could not have been addressed to B's new location since A moved first. Given that location proxies are not used, there is no way that B could successfully receive A's binding update after $T_B(j_0)$. Therefore, A's binding update must have reached B before $T_B(j_0)$, i.e., before B moved.

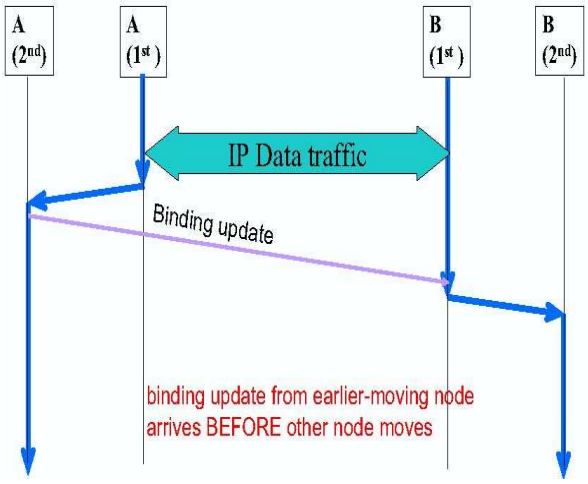


Figure 7.8: Lemma 4 and 5

Remark What about if A's binding update successfully reaches B before it moves, but B's binding up-date does not reach A because A's next handoff happens before the arrival of B's binding update? Does Lemma 3.4 break down? No, in that case the lemma, applied to $H_A(i_0)$and $H_B(j_0)$ would correctly show that the problem is not between those two handoffs, but applied to $H_B(j_0)$ and $H_A(i_0+1)$ would correctly show that the simultaneously mobility problem occurs then with B as the earlier moving node.

7.8 Probability of simultaneous mobility

In this section, I analyze the probability of simultaneous mobility. In [WDSY03] I introduced a simple mathematical model for estimating the probability of occurrence of simultaneous mobility. I briefly describe in this section.

If the probability that any particular handoff (either separately or both at the same time) suffer from simultaneous mobility problem be P_0, and the probability that at least one out of N handoffs in a given session suffers from the simultaneous mobility problem be P_N. Thus P_N = 1 - $(1 - P_0)^N$.

The system gets into an interrupted state if either of the mobiles or both of the mobiles are subjected to simultaneous mobility problem. Inter-handoff times for mobile1 (A) and mobile2 (B) are $\lambda 1$ and $\lambda 2$, respectively.

α is the time taken for A's binding update to reach B. β is the time taken for B's binding update to reach A.

Probability that mobile 1 contributes to the simultaneous mobility problem is $P_1 = \beta/\lambda 1$.

Probability that mobile 2 contributes to the simultaneous mobility problem is $P_2 = \alpha/\lambda 2$.

Thus, probability that there is a simultaneous mobility problem due to handoff by mobile 1, mobile 2 or both is as follows

$P_0 = P_1 + P_2 - (P_1 \times P_2) = \beta/\lambda 1 + \alpha/\lambda 2$ - $[\beta \times \alpha]/[\lambda 1 \times \lambda 2]$

If inter-handoff time for both the mobiles are same, thus $\lambda 1 = \lambda 2 = \lambda$, then

$P_0 = [\alpha+\beta]/\lambda$ - $[\beta \times \alpha]/\lambda^2$

where α and β are the amount of time needed for a binding update to reach from A to B and vice-versa and λ is the average inter-handoff time.

Thus, the simultaneous mobility problem is affected by a combination of end-to-end latency of the packet and inter-handoff time. As part of initial results, I have conducted a preliminary analysis of simultaneous mobility of IP hosts for SIP, MIPv6 and MIP-LR-based mobility protocols.

Consider two consecutive handoffs, one each at mobile nodes A and B. According to

Lemma 3.2, the simultaneous mobility problem occurs if and only if the binding update from the earlier moving node arrives after the other node has moved. Mathematically, this is written as $T_A + \gamma + \zeta + \delta_{A \to B)} > T_B$ (if A is the earlier moving node) or $T_B + \gamma + \zeta + \delta_{B \to A} > T_A$ (if B is the earlier moving node).

Putting the two inequalities together, the following equation is obtained

$$T_A - \alpha < T_B < T_A + \beta \tag{7.3}$$

where $\alpha = \gamma + \zeta + \Delta_B \to_A$ and $\beta = \gamma + \zeta + \Delta_A \to_B$ are convenient short forms.

Then I define the concept of "vulnerability interval" $\beta + \alpha$, which is the time around a handoff during which the two mobile nodes are vulnerable to the simultaneous mobility problem if another handoff occurs at the other mobile node.

It is reasonable to model the handoff times for A and B as independent Poisson processes. In this model, the intervals between consecutive handoffs at A, $\Gamma_A(k-1) = T_A(k)A$ - $T_A(k-1)$, $\Gamma_A(k) = T_A(k+1)A$ - $T_A k$ etc., are independent exponentially distributed random values, and similarly for the corresponding intervals between consecutive handoffs at B. Then it is easy to argue that the probability of the simultaneous mobility problem occurring can be estimated as the following equation.

$$P_0 = \frac{E(\alpha + \beta)}{E(\Gamma)} - \frac{E(\alpha \times \beta)}{E(\Gamma^2)} \tag{7.4}$$

If there are N handoffs occurring at each of the two mobile nodes, then the probability of the simultaneous mobility problem occurring can be estimated by

$$P_N = 1 - (1 - P_0)^N \tag{7.5}$$

Based on experimental measurements, E[α+β] ranges from 50 ms to 500 ms, while λ may range from 5 seconds (movement at vehicular speeds across pico-cells of a few hundred meters in diameter) to 500 seconds or more (larger cells, slower speeds, non-linear

movement pattern). E (Γ) is the average value of inter-handoff time that can be equated to λ. In Figure 7.9, I plot for approximately this range of E[α+β] and λ. Figure 7.9 shows how probability of simultaneous mobility P_0 is affected due to binding update latency and mean handoff time of the mobile based on the Equation 7.4. Figure 7.9(a) shows that for a given inter-handoff time (500 seconds), probability of failure increases as the one-way latency of the binding update increases. While Figure 7.9(b) shows that for a given one-way-delay (50 ms) probability of failure increases as the inter-handoff time decreases. Figure 7.10 shows the probability of simultaneous mobility P_3 when the total number of handoff is 3 based on Equation 7.5. For the same values of inter-handoff time and one-way latency, probability of failure due to simultaneous mobility increases as the number of handoff is increased to 3. As expected, as shown in Figure 7.9(a) the highest probability of simultaneous mobility is when one-way packet latency is the largest and average inter-handoff time is smallest. Thus, the effect of the simultaneous mobility problem could be quite significant. Without fixing the problem, the binding updates of both the Mobile Hosts would never reach the other host, and so the connection would be lost. It is important to note that this analysis is optimistic as it is assumed that the binding update from A to B would not be lost. Since there is a small chance that this binding update might also be lost, the values computed in this analysis could be viewed as merely providing a lower bound on the likelihood of the simultaneous mobility occurring. For the case of simultaneous mobility during session initiation signaling, the probability of failure also depends upon the mobility rate of the mobiles. From the lab measurements, it takes about 200-300 ms to complete the whole session initiation signaling sequence. A complete registration will take about 150 ms. Hence, the probability of simultaneous mobility occurring during session initiation signaling is non-trivial.

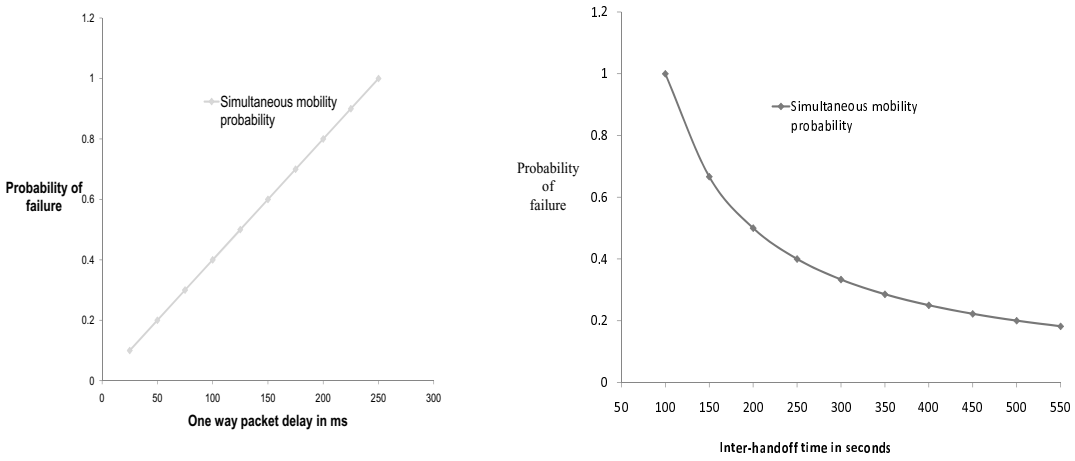


Figure 7.9: Plot of P_0 against latency and mean handoff time

7.9 Solutions

Kravets et al. [KCM01] have hinted at the benefits of some kind of proxies in fixed locations to enable communication to continue even when both end-hosts move simultaneously. However, as far as I know, previous work has not analyzed the problem to the level I have described in Section 7.3 nor has a systematic analysis of solutions applied to a range of mobility protocols been previously provided. I describe some of the solution mechanisms in this section. Solution mechanisms are specific mechanisms and functions that could be used (typically in conjunction with other mechanisms and functions) to provide solution for the simultaneous mobility problem for a given mobility protocol.

The proposed techniques can broadly be classified into three: *soft-handoff*, *sender-based*, *receiver-based* mechanisms.

7.9.1 Soft handoff

Suppose a mobile node can have more than one valid IP address. In such cases, the mobile can have two bindings associated with its home address. This is sometimes referred to as si-

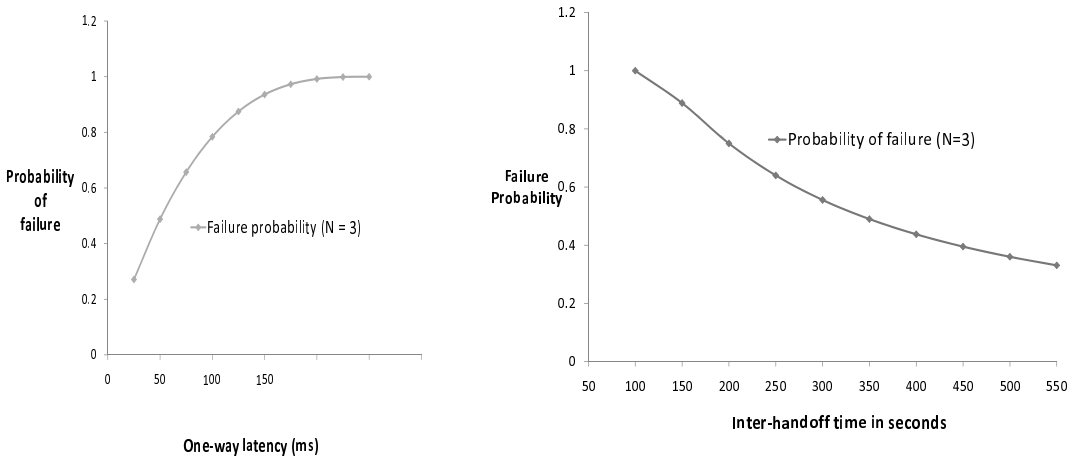


Figure 7.10: Plot of P_N against latency and mean handoff time

multaneous mobility bindings, and should not be confused with the simultaneous mobility problem. I call it as the soft handoff approach, since it is similar to soft handoffs in CDMA mobile systems [WL97]. The idea is that if the previous IP address and new IP address can both be used to reach the mobile node during the handoff process, that can solve the simultaneous mobility problem. Binding updates sent to the previous IP address would arrive correctly. Although, most of the current operating system can support multiple concurrent IP addresses for the wireless interface (s), this is not a universally applicable solution, for the following reasons:

- ◇ The operating system should be able to support multiple concurrent IP addresses for the wireless interface(s).
- ◇ The network interface of the mobile needs to be able to connect simultaneously to multiple base stations that may belong to two different subnets. CDMA technology does provide this ability whereby the network interface can connect to both the network access points simultaneously. However, this is limited to CDMA access technology only.

- ◇ Resource utilization is not efficient because of redundant allocation of bandwidth resources (on both communication paths) during the period of simultaneous mobility bindings.

- ◇ It is also important to make sure that the simultaneous mobility bindings need to be active to ensure that no problems will occur during simultaneous mobility. The longer it waits, the more this specific scheme uses valuable network resources redundantly. Since this solution must work for any radio network technology, use of simultaneous bindings is not a satisfactory solution. Hence, the mechanisms related to the soft handoff approach is not further explained in this paper.

7.9.2 Receiver-side mechanisms

Receiver-side mechanisms typically can be deployed in the previous network or home network of the receiver and act on behalf of a receiver to help it to be located. Retransmission, forwarding, redirecting, proactive-forwarding and proactive-redirecting are some of the mechanisms that have been analyzed for two mobility protocols such as SIP and MIP-LR. Details of the proposed mechanisms and results can be found in references [WDSY03], [DWDSY07]. Following is a list of receiver-side mechanisms.

7.9.2.1 Timer based retransmissions

One could imagine a forwarding location proxy automatically retransmitting a binding update if it has not gotten confirmation that the binding update was successfully received by the intended receiver. This location proxy could be located in the receiver's home network or a visited network (e.g., the previous network or latest network). Location proxies that retransmit based on timeouts are similar to proactive location proxies in that both need to store the message briefly, to retransmit if necessary. The difference is in the conditions for retransmission. A proactive proxy retransmits as soon as a new address is obtained,

whereas a timer-based retransmission may be too slow. In the existing implementations, a stateful SIP proxy could retransmit binding updates (re-INVITE) after the expiration of a timer. This could be located in the home network of the receiver or in the visited network of the receiver. In order to ensure that the re-INVITE message (and other signaling) goes through this server, the Record-Route option could be used in the initial INVITE message to add the server to the signaling path.

7.9.2.2 Regular passive forwarding

Forwarding mechanisms on the receiver side allow binding updates to be forwarded from a location proxy in the previous network to the correct new location of the receiver. Forwarding mechanisms from a previous network may also forward data packets (since the location proxy is forwarding packets, anyway, it might as well forward data packets). One could also imagine such a location proxy in the receiver's home network as well.

Here are some existing implementation where the forwarding agents are in the receiver's previous network. In MIP-RO (Mobile IP with Route Optimization), the previous Foreign Agent serves this role by forwarding the data packets. Unfortunately, this ability is missing from MIPv6, perhaps because no Foreign Agents are used in MIPv6 (and so, there is no natural forwarding agent present in the previous network). Thus, the problem of simultaneous mobility remains in MIPv6. Similarly, SIP-based mobility management and MIP-LR lack such functionality. Similarly, in some examples, the forwarding agents could be in receiver's home network. An example is the Home Agent in MIPv4 and MIPv6. A SIP server in the receiver's home network could also serve in this capacity, e.g., if it places itself in the signaling path using the Record Route field in the initial INVITE message.

7.9.2.3 Pro-active forwarding

Regular passive forwarding may be insufficient to solve the simultaneous mobility problem. A pro-active forwarding location proxy may help where forwarding takes place before the

handoff.

7.9.2.4 Redirecting

Redirecting mechanisms on the receiver side can help to get messages like binding updates to the right place. There are some existing implementations where the redirecting agents are placed in the receiver's previous network and home network. In MIP-RO, the previous Foreign Agent serves this role. In MIP-LR, the HLR does this, but only before a media session begins. Then, it is not involved in control signaling during the communications session. Thus, it does not count as a proper implementation of a solution mechanism for the simultaneous mobility problem.

7.9.2.5 Pro-active redirecting

Regular redirecting may be insufficient to solve the simultaneous mobility problem. A pro-active redirecting location proxy may help in some cases where there is a probability of handover to a number of target networks.

7.9.3 Sender-side mechanisms

Sender-side mechanism typically can be deployed in the home network of the sender, or in the sender itself, and act on behalf of the sender to try to reach the receiver. The receiver may be moving simultaneously with the sender and may not receive the binding update if none of these mechanisms are used. Following is a list of sender-side mechanisms.

7.9.3.1 Timer based retransmissions

A forwarding location proxy automatically retransmits a binding update if it has not gotten confirmation that the binding update was successfully received by the intended receiver. This location proxy could be located in the sender's home network or even in the sender itself (for end-to-end retransmission).

There can be several existing implementations. A stateful SIP server could retransmit binding updates (re-INVITE) after the expiration of a timer.

7.9.3.2 Forwarding (regular, passive type)

Forwarding mechanisms in the sender's home network can help to get messages like binding updates to the right place, but are probably less useful than those on the receiver side because of the time spent by the forwarded signals due to distance between sender's home network and receiver.

7.9.3.3 Pro-active forwarding

Regular, passive forwarding may be insufficient to solve the simultaneous mobility problem. A pro-active binding update proxy may help in some solutions, where it attempts to find the most current location of the receiving node and re-try the forwarding there.

7.9.3.4 Redirecting

Redirecting mechanisms in the sender's home network can help to get messages like binding updates to the right place, but are probably less useful than those on the receiver side.

7.9.3.5 Pro-active redirecting

Regular redirecting may be insufficient to solve the simultaneous mobility problem. A pro-active redirecting location proxy may help in some cases when the target network is not deterministic.

Table 7.1 shows the applicability of these solution mechanisms for different mobility protocols. Table 7.2 shows the strengths and weaknesses of different solutions.

Table 7.1: Comparison of solutions for simultaneous mobility

Solutions			MIP-RO	MIPv6	SIPMM	MIP-LR
Receiver-side	Prior network	Retransmission			Possible	
		Forwarding	Yes			
		Pro-active forwarding				
		Redirecting	Yes			
		Proactive redirecting				
	Home network	Retransmission			possible	
		Forwarding	Yes	Yes	possible	
		Pro-active forwarding				
		Redirecting				Yes
		Pro-active redirecting				
Sender-side	Home network	Retransmission			possible	
		Forwarding			possible	
		Proactive forwarding				
		Redirecting				
		Proactive redirecting				
	At sender	Retransmission			Yes	

Table 7.2: Strength and weakness of different solutions

Solutions	Strengths	Weaknesses
Timer-based retransmission of lost messages	Can be easily implemented with a timer	(a) Difficulty of choosing good time-out values; (b) retransmissions may also be lost
Simultaneous bindings	No significant increase in handoff latency to solve simultaneous mobility	(a) Not supported by all wireless networks; (b) redundant resource utilization; and (c) not clear how long to keep active the simultaneous bindings
Forwarding mechanisms from previous network	Effective if not just data but also signaling is forwarded.	Handoff latency is slightly increased because of the forwarding from the previous network; still vulnerable to simultaneous mobility, but with reduced vulnerability interval
Stationary proxies	Completely eliminates vulnerability interval	Handoff latency is increased

7.10 Application of solution mechanisms

In this section, I describe how the solution mechanisms can be applied to different mobility protocols. I illustrate its applicability to a few mobility protocols, namely Mobile IPv6, SIPMM (SIP-based mobility) and MIP-LR.

7.10.1 Mobile IPv6

Three different solution mechanisms are considered to take care of simultaneous mobility problem in MIPv6. These are described as follows.

7.10.1.1 Forwarding proxy in previous network

As described earlier, for MIP-RO, Foreign Agents in the previous network act as forwarding proxies. However, for MIPv6, Foreign Agents are not used. Thus, ordinary routers in the previous network need to be augmented with forwarding proxy functionality. This would

involve significant challenges and modifications to MIPv6. For example, a mechanism would be needed to securely update the router with the latest IP address of the mobile node. However, getting ordinary IPv6 routers to perform this kind of forwarding might pose deployment bottleneck and thus some kind of agent might need to be introduced.

7.10.1.2 Combination of sender-side and receiver-side mechanisms

I proposed a combination of sender-side proactive binding update proxies and receiver-side pro-active redirecting location proxies as a general solution in [WDY+03]. Here I propose that the same technique can also be applied to MIPv6. The Home Agents of the sender and receiver, respectively, can serve as the pro-active binding update proxy and pro-active redirecting location proxy. Return routability has to be modified so that CTI message goes through the sender's Home Agent. Another modification to MIPv6 is that the binding update must first be reverse-tunneled to a mobile node's own Home Agent before being forwarded to the correspondent node. The revised MIPv6 update procedure will work as follows. Let there be two mobile nodes A and B. A sends its CTI and binding update messages to B through A's Home Agent, rather than directly to B's care-of address. However, A's Home Agent will then forward these messages to B at B's care-of address. A's Home Agent, acting as a pro-active binding update proxy will also keep a copy of any such message for a period τ. It would then query B's Home Agent (a pro-active location proxy for B) to find out if B has a newer address. B's Home Agent responds immediately but keeps a copy of the query for a period ρ. If before this period is over, B's Home Agent receives a registration for B at a new address, it pro-actively corrects its query. A's Home Agent will then forward the message to the new address. This solution is illustrated in Figure 7.11.

The selection of τ and ρ should be carefully chosen based on reasonable estimates of the appropriate signaling and computational delays of the network. It is clear that $\tau > \rho$, so A's Home Agent can respond to any query response correction from B's Home Agent.

7.10.1.3 Receiver-side mechanisms only

Two solutions discussed so far are not good since they require significant changes to MIPv6. A more MIPv6-centric solution is preferable. I therefore consider a solution where just the receiver's Home Agent is involved. A is the sender (of the CTI, HTI or binding update). A sends all these control messages to B using B's *home address*, thus forcing B's Home Agent to be involved. B's Home Agent will act as a pro-active forwarding location proxy (a slight modification from its usual role as a forwarding location proxy), forwarding the control message to B as usual, but keeping a copy of it for time τ . If it gets any binding updates from B during that time, it pro-actively forwards the message to B. This solution is shown in Figure 7.12. However, it requires some modifications to be made at the home agent. Home agents need to behave like proactive forwarding location proxy in addition to behaving like a forwarding location proxy. The main modification to mobile nodes implementing this solution is also small - to send the CTI and binding update to the home address of the correspondent node instead of directly to its care-of address.

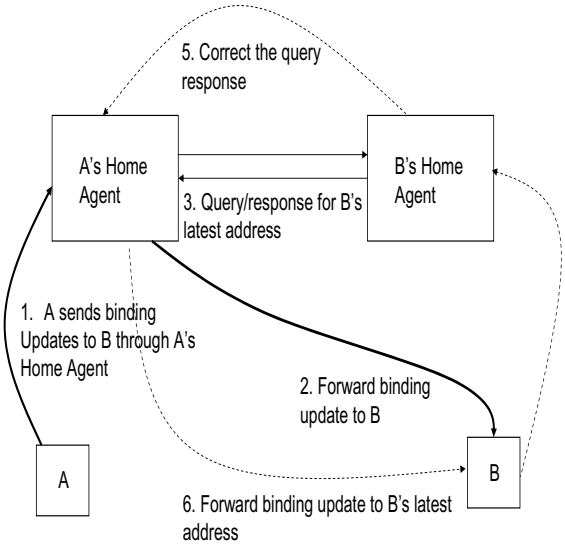


Figure 7.11: Sender and receiver side mechanism

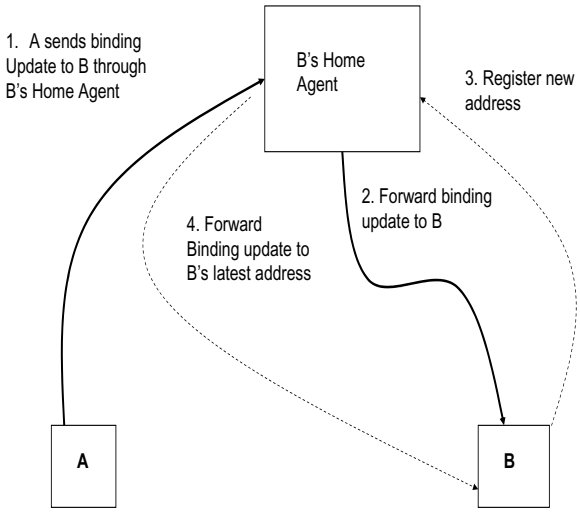


Figure 7.12: Receiver side mechanism

7.10.1.4 Evaluation

It is clear that the third solution, with receiver-side mechanisms only (sending messages to the other node's Home Agent) is the cleanest solution with the least changes to MIPv6. The adding (in the second solution) of a query and response capability to Home Agents is quite a drastic change for MIPv6. After the removal of Foreign Agents in going from MIPv4 to MIPv6, the adding of a forwarding proxy in the previous network with substantial functionality (in the first solution) is not desirable as it increases delay.

7.10.2 MIP-LR

In this section, I describe how these solution mechanisms can be applied to take care of simultaneous mobility problem in MIP-LR.

7.10.2.1 Forwarding proxy in previous network

There are two types of MIP-LR: one with Foreign Agents [JRG+98], and one without [JRY+99]. The version without Foreign Agents uses advertisement agents. The forwarding proxies use interceptor function that intercepts the binding update and sends it to the new

address of the mobile.

For the purposes of placement of the interceptor function, it does not matter whether Foreign Agents or advertisement agents are in use. The point is that there is some kind of agent in each of the foreign networks, and the interceptor function can be placed here. MIP-LR needs modification so the mobile node sends a binding update to the Foreign Agent (or Advertisement Agent) in the previous subnet as soon as it obtains its new IP address.

7.10.2.2 Sender-side and receiver-side mechanisms

In this solution, the binding update sent by a mobile node to its HLR has a list of correspondent nodes and their addresses. The HLR, which already performs the role of a redirecting location proxy, is enhanced to be a pro-active redirecting location proxy. It also acts as a pro-active binding update proxy, since it already obtains the current binding information as part of MIP-LR updating after each handoff. In order to do this, the HLRs must be enhanced to pro-actively retransmit binding updates and to query other HLRs for correspondent nodes' addresses. In order to minimize changes to MIP-LR, the HLR-initiated binding updates are only sent when necessary, i.e., when the queries return a newer address for a correspondent node than the one provided by the mobile node.

7.10.2.3 Evaluation

It is recommended not to consider a solution using only receiver-side mechanisms as has been done for MIPv6. This is because the HLR would then have to become a pro-active forwarding proxy. Such a change is too much for MIP-LR, one of whose points is that no forwarding location proxies are used (but multiple replicated HLRs are used).

7.10.3 SIP-based mobility

SIP allows much flexibility in placement and usage of SIP servers in the signaling path between two mobile nodes. The Record-Route field is an optional field in the SIP header

that allows SIP servers to remain in the signaling path between two SIP end-nodes during a communications session. It is assumed that often there will be a SIP server in each mobile node's home network that serves as an up-to-date location proxy for it. An up-to-date location proxy keeps a record of the most recent location of the mobile. SIP also provides inbuilt retransmission technique.

7.10.3.1 Timer-based retransmission

SIP has an in-built retransmission capability, where messages are retransmitted after a time-out if the acknowledgement is not received. During mid-session mobility, a re-INVITE may get lost even if it goes through the SIP server that keeps the most recent registration status of the destination. However, SIP allows for automatic retransmissions of INVITEs (including re-INVITEs) by SIP UAs if a response (OK message) is not received within a specified time. Stateful SIP servers can also retransmit (re-)INVITEs.

One problem with timer-based retransmissions is that significant latency could be added to the handoff when messages are lost due to simultaneous mobility.

Another problem is that there is no guarantee that the retransmission would not also be lost. For example, the retransmission may be sent directly to the old address of the Correspondent Host, bypassing network elements (e.g., the relevant SIP servers, or the Home Agent of the Correspondent Host) that know the latest address of the Correspondent Host. Figure 7.13 shows how a server assisted retransmission technique can be useful to solve the simultaneous mobility problem.

7.10.3.2 Forwarding proxy in previous network

Like the first proposed solution for MIPv6, I consider adding a forwarding location proxy to the previous network of a mobile node. For SIP-based mobility, the most natural choice of the forwarding proxy could be an entity similar to an RTP (Real-time Transport Protocol) translator [Sch], since these are already used to forward media traffic, among other things.

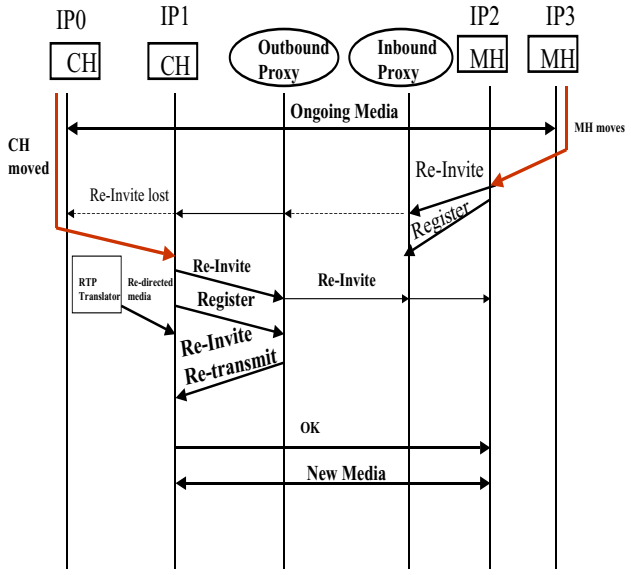


Figure 7.13: Server-assisted retransmission mechanism

Having a SIP server in the previous network that can forward this signaling is one solution. However, if one keeps on adding SIP servers in previous networks to the signaling path (using Record-Route), the signaling path becomes inefficient as the mobile node moves and the signaling path goes through more and more SIP servers in previous networks. Although the existing RTP translator can only forward data traffic from the previous network, similar mechanism can be deployed that can intercept the signaling traffic and forward it to the new location of the mobile. For receiving update signaling, the RTP translator can be enhanced to act as a SIP signaling translator without generally translating the RTP.

7.10.3.3 Receiver-side mechanisms

For SIP-based mobility scheme, the receiver-side home network SIP server already has some location proxy functionality that can be modified to act as pro-active location proxy. I first consider the case where it acts as a forwarding location proxy (it can also be a redirecting location proxy, which I will consider in the next paragraph). The SIP server immediately retransmits the re-INVITE upon receiving a REGISTER message from the destination of a pending re-INVITE. This is basically the same solution as the third one for MIPv6 (the preferred solution). Hence, the Figure 7.11 also applies here, by replacing B's

Home Agent with B's home network SIP server, and binding update with re-INVITE. A difference is that, in order to get the SIP server to be in the signaling path for the re-INVITE request, the Record-Route field can be used. No modifications are needed on the mobile nodes, since SIP conveniently already has the Record-Route feature, unlike with MIPv6, where they have to be slightly modified to send control signaling to the home address of the other node rather than directly to the care-of address. The conversion of the SIP server to a pro-active one is in some ways easier than the conversion of a MIPv6 Home Agent to a pro-active forwarding proxy. This is because there is already the notion of stateful SIP servers that can retransmit messages like the re-INVITE if no acknowledgement has been received by the time a timer expires.

7.10.3.4 Sender-side and receiver-side mechanisms

If the home network SIP server is modified to become a pro-active redirecting location proxy, instead of a pro-active forwarding location proxy, then it needs to interact with a pro-active forwarding proxy closer to the sender in the signaling path. In particular, when there is a SIP server in each mobile node's home network, there needs to be a pro-active forwarding proxy in the sender's home network. This is similar to the chosen solution for MIP-LR, where the two HLRs were involved in this way. One difference is that the Record-Route feature will be needed to keep both SIP servers in the signaling path.

7.10.3.5 Evaluation

It would appear that either the 2nd or 3rd solution is equally simple to implement, given that SIP servers of both types (forwarding and redirecting) are available. With MIPv6, on the other hand, the clear preference was for the receiver-side solution, given that Home Agents are forwarding location proxies only.

7.11 Concluding remarks

Although the original MIP did not suffer from the simultaneous mobility problem, newer mobility management protocols like MIP-LR, SIP-based mobility and MIPv6 do face this problem. In this chapter, I have identified the problem of simultaneous mobility, introduced a new analytical framework, and then used the framework to prove some new theorems, analyze solution mechanisms, and propose and compare solutions for simultaneous mobility for MIPv6, SIP-based mobility and MIP-LR. I also conduct a probability analysis on the likelihood of occurrence of simultaneous mobility.

The problem is further compounded by the expected rise in popularity of at least two of the three protocols we considered, namely MIPv6 and SIP-based mobility. Additionally, with the rise of smaller pico-cells in certain segments of the wireless market and higher mobility rates, there may be more frequent occurrences of simultaneous mobility in the future. I have explored a number of approaches to deal with the simultaneous mobility problem. In some of the protocols, there is existing functionality that partially helps solve the simultaneous mobility problem, or that can be modified to handle simultaneous mobility. For example, with SIP-based mobility, forwarding entity similar to RTP can be used to forward signaling, including binding updates, that might have been sent to the previous network.

Most recently, I have introduced the effect of simultaneous mobility problem for MIPv6 in the MEXT working group within the IETF. Realizing that there is lack of solution to deal with the simultaneous mobility, a new section has been added in RFC 3775 bis [JPA09] to take care of simultaneous mobility problem arising out of return routability procedures. As an alternative to the solutions discussed in Section 7.9, following modification has been added to the draft.

> In some scenarios, such as simultaneous mobility, where both correspondent host and mobile host move at the same time, or in the case where the corre-

spondent node reboots and loses data, route optimization may not complete, or relevant data in the binding cache might be lost.

- ◇ Return routability signaling MUST be sent to the correspondent node's home address if it has one (i.e., not to the correspondent node's care-of address if the correspondent node is also mobile.)
- ◇ If Return routability signaling timed out after $MAX\_RO\_FAILURE$ attempts, the mobile node MUST revert to sending packets to the correspondent node's home address through its home agent.
- ◇ The mobile node may run the bidirectional tunneling in parallel with the return routability procedure until it is successful. Exponential backoff SHOULD be used for retransmission of return routability messages.

The return routability procedure may be triggered by movement of the mobile node or by sustained loss of end-to-end communication with a correspondent node (e.g. based on indications from upper-layers) that has been using a route optimized connection to the mobile node. If such indications are received, the mobile node MAY revert to bi- directional tunneling while re-starting the return routability procedure.

Chapter 8

Handoff optimization for multicast streaming

In this chapter, I propose few optimization techniques that expedite the delivery of multicast stream during handoff in a hierarchically scoped multicast architecture. First, I propose a hierarchically scoped multicast content distribution network, describe the functional components of the architecture and their implementation, introduce the optimization techniques to reduce the join and leave latencies for multicast traffic and finally compare the performance results in the prototype testbed from both optimized and non-optimized handoffs. The previous chapters have focused on fast handoff techniques for unicast traffic. However, in this chapter, I apply some of the optimization techniques that were discussed in Chapter 5 to provide fast delivery for multicast traffic in a hierarchically scoped multicast environment.

8.1 Summary of key contribution and indicative results

Currently, multicast-based content distribution systems lack flexible features such as local and global program management and automatic advertisement insertion. These systems also do not support fast-handoff when the mobile moves between the subnets. Unlike

unicast traffic, multicast traffic is receiver oriented. A mobile receiving multicast traffic is subjected to handoff delay and associated media interruption due to multicast *join* latency during its movement between layer 2 access points or layer 3 subnets. Multicast *join* latency is contributed due to periodic IGMP (Internet Group Management Protocol) router query interval and random amount of time the client waits before it can send IGMP client report. This *join* latency can be as large as 2 minutes in duration and disrupts the streaming media during the mobile's movement.

I proposed and implemented a hierarchical scope-based multicast streaming architecture that enables local and global program management and real-time advertisement insertion using RTCP(Real Time Control Protocol)-based feedback control information.

In order to reduce the handoff latency for multicast join, I proposed both proactive and reactive triggering techniques. As part of the reactive mechanism, I developed an application layer triggering technique that sends unsolicited RTCP *join* to join the multicast tree after the mobile hands off to the new network instead of network layer IGMP. The server that receives the RTCP *join* in turn uses IGMP report to join the upstream router.

As part of the proactive technique, I proposed an application layer proxy and multicast address announcer so that the local server can join the multicast tree on behalf of the mobile as the mobile is impending to handover to the new network. While the multicast proxy and the server join the upstream router using IGMP, the mobile triggers the multicast stream by using RTCP Join to multicast proxy.

A hierarchical scope-based architecture provides the ability to manage local and global program by using local servers in the content distribution network. By using the feedback signal such as RTCP, my proposed mechanism provides the ability to control the advertisement duration without relying on any additional signaling. By using an application layer triggering technique such as RTCP, the mobile does not need to depend upon layer 3 IGMP router query interval nor does it depend upon multicast support in the kernel. Having the ability to trigger multicast streaming during mobile's configuration process, the mobile

optimizes the operations in parallel. Compared to traditional unoptimized multicast handoff approaches, my proposed proactive optimization techniques can reduce the handover latency by a factor of 10 when the probability of presence of a multicast group is low. Proposed proactive and parallel triggering techniques perform better by a factor 4 compared to the proposed reactive techniques when the probability of presence of multicast group is low (e.g., 0.2).

In the rest of the chapter, I describe the details of the hierarchical scope-based multicast architecture, elaborate the proposed mechanisms that allow local, global program management, advertisement insertion. I describe the experimental testbed where I implemented the architecture and the fast-handoff mechanisms that I proposed. I also compare the results of my proposed fast-handoff mechanisms with that of non-optimized systems.

8.2 Introduction

CDN (Content Distribution Network) distributes the contents from the origin server to the replica servers that are situated closer to the end clients. The replica servers in a CDN store a very selective set of content and only the requests for that set of content are served by the CDN. This mechanism provides reduced access delay for any specific content and consumes lower bandwidth in the core of the network. There are a few commercial content distribution networks (CDN), namely Akamai [Aka], Digital Island [Dig] and Edgecast [Edg] that distribute information from many news media, namely CNN and New York Times. Figure 8.1 shows a sample content distribution network and shows how the local affiliates distribute the global program and local advertisement to the end users.

Mobile content distribution network (CDN) can use multicast technology to distribute the content from a single source to multiple replica servers (local affiliates) and end recipients more efficiently. Unlike unicast traffic, the multicast communication is receiver initiated. Thus, triggering techniques play an important role for efficient and timely mul-

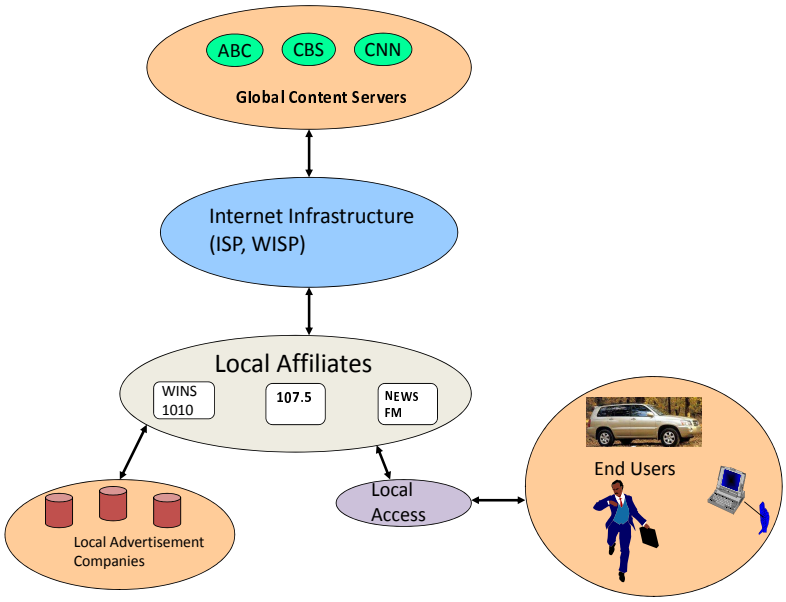


Figure 8.1: Example of content distribution network

ticast multimedia stream delivery. In order to maintain minimum loss and latency during the client's movement, it is desirable to minimize the handoff time and to enable almost instantaneous delivery of multicast streams by using optimized triggering techniques for initiating the stream delivery.

I have introduced multicast mobility in Chapter 2. I re-introduce it in the context of join latency and leave latency. Figure 8.2 shows how a mobile moves from one access point to another access point within the same router (Router R1) and then moves to a new subnet connected to router R2. After the handoff to new subnet, the mobile rejoins the same multicast group and a new multicast tree is constructed. IGMP (Internet Group Management Protocol) is used between the mobile node and router 2 and router 2 uses PIM-SM (Protocol Independent Multicast - Sparse Mode) to join the multicast tree that the mobile is part of prior to handoff.

Handoff delay during multicast stream delivery from a single source while the client moves to the next cell consists of several components, namely detection of a new cell, subnet or domain (Δ_1), address acquisition and network configuration (Δ_2), triggering of

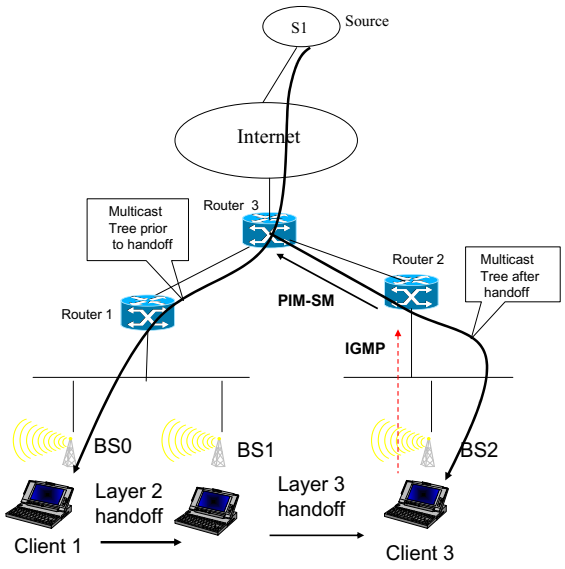


Figure 8.2: Handoff for multicast streams

multimedia stream to be delivered in the new subnet (Δ_3) and delivery of the multimedia stream (Δ_4). While some of these delay factors are common to both unicast and multicast traffic (e.g., cell or subnet detection, IP parameter configuration), in this chapter, I concentrate on the optimization techniques that will allow faster delivery for multicast streaming traffic. Faster delivery of multicast traffic is dependent upon "join" latency. Join latency is defined as the elapsed time between a host joining the multicast group and router sending a multicast packet towards the mobile. Figure 8.3 shows the protocol interaction between the mobile and the first hop router (Router R1) that uses IGMP and interaction between router 2 and router 3 using PIM-SM. This figure shows how router R2 keeps sending IGMP router query message to all host multicast address (e.g., 224.0.0.1) at a periodic interval and the mobile sends a response after it has joined a specific multicast group. On receipt of the IGMP query response message, router 2 joins the new multicast group by sending PIM-SM join message to the upstream router 3.

Fast handoff techniques can be used at several layers to expedite the delivery of multicast streams. Delivery delay for multicast traffic depends on layer-2 handoff delay and

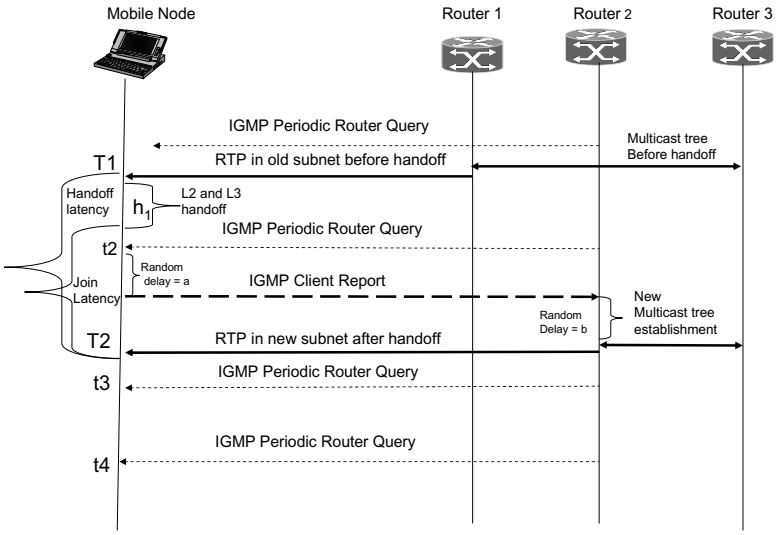


Figure 8.3: IGMP flow during subnet handoff

join latency contributed by Internet group management protocol(IGMP) [Fen97] at layer 3. Leave latency is defined as the time period during which the multicast traffic is still allowed to flow in the previous cell even if the mobile has left the cell. Thus, leave latencies contribute to the waste of bandwidth due to flow of multicast stream in the previous cell even after the mobile has left the cell. Several methods such as IGMP snooping [WSJW02], Cisco group management protocol (CGMP) [FT] take care of handoff for multicast streams at layer 2. As discussed in RFC 3170 [QA01], mobile receivers within a domain can do expedient "joins" and "leaves", whereas a mobile can send to a multicast address without explicitly joining the address.

In layer 3, triggering delay is caused by IGMP query report [Fen97] that helps the node to be part of the new multicast tree after the mobile hands over. However, if there is currently at least another active participant in the subnet, the mobile host can continue to receive the traffic without waiting to hear the membership query from the router. A typical query interval for IGMP is by default 125 seconds [Wil00], although this value is configurable in the multicast routers. In order to avoid flooding, the LAN with IGMP

messages, this value cannot be made very small. Flament et al. [FGL+99] show that by using IGMP, a host will wait on average for 65 seconds in order to continue to receive the multicast traffic after the handoff. This is because IGMP was not designed for roaming clients in a wireless environment.

Typical leave latency is about 2 minutes after the host has moved to a new subnet, i.e., traffic still flows to the previous cell even after the client has moved out thus wasting bandwidth in the previous cell.

In layer 2, when destination cell is part of the same subnet multicast stream continues in both the cells. Although layer 2 triggering delay is avoided, it however contributes to the waste of bandwidth, if there is no active participant in the adjacent cell. However, a multicast switch stops the multicast traffic from flowing to the neighboring cell if there is no mobile in the target cell. In case of a client's movement between access points within a subnet, CGMP (Cisco Group Management Protocol) or IGMP snooping [Wil00] takes care of triggering the multicast stream at layer 2. CGMP works in conjunction with IGMP and controls the multicast traffic flow in layer 2.

Unlike unicast traffic, layer 3 configuration time (e.g., acquisition of a new IP address) on the client does not affect the multicast stream delivery delay. On other hand, several other components, such as detection of new cell, subnet, triggering time and time needed for the router to join the upstream router affect the multimedia delivery. For example, as per Figure 8.3, the media is discontinued for a period of T_1 - T_2, where T_1 denotes the time when media traffic is received by the mobile before handoff, and T_2 denotes the time when media traffic is received by the mobile after the handoff. Router 2 in Figure 8.3 keeps on sending IGMP router query messages at a regular interval (e.g., at time period t_2, t_3, and t_4). After the layer 2 and layer 3 handoff delay that is denoted by h_1, the mobile waits for the IGMP router query. According to the Figure 8.3, the router query is received at time t_2 after the handoff. After receiving the router query, the mobile waits for a random period of time period a before it sends IGMP query response. Once the router 2 receives the client report,

it will join the upstream multicast tree. There is a delay of b to join the upstream multicast tree. Thus, total handoff delay due to multicast join latency is t_2-T_1-h_1+a+b. Since the waiting time for an IGMP query message could range between 0 and 125 seconds, the handoff in IP multicast will result in a large gap in streaming traffic during handoff.

8.3 Key principles

Following are some of the key principles that should be considered to optimize the delivery of multicast traffic and reduce the handoff delay and packet loss.

1. Reduction in join latency reduces the time taken for data delivery after mobile's handoff to the new cell or subnet.

2. Tunnel overhead can be eliminated by avoiding the dependence on the home agent often used in home subscription-based approach.

3. Reduction in leave latency reduces additional bandwidth consumption in the previous network. Leave latency in the previous network can be reduced by the proxies that send unsolicited leave message on behalf of the mobiles.

4. Parallel operation among the handoff functions reduces the overall triggering delay for media delivery.

5. Proactive join to a multicast tree reduces the join latency after the handoff to the new network.

6. Fast-handoff techniques can be applied at multiple layers based on the movement of the mobile.

8.4 Related work

Several papers have discussed the group *join* and *leave* behavior in the Internet, effect of channel surfing, and mobility for multicast stream. The process of joining or leaving a specific multicast group while changing the cell or subnet is similar to surfing a TV or radio by flipping channels studied by Fergusen et al. [Fer94]. Almeroth et al. [AA97] [Alm00] have described the multicast group behavior in the Internet, and have cited results about surfing delay based on the analysis of Mbone (multicast backbone)'s [Eri94] temporal statistics. These results show that within a time interval of 2 minutes, a user leaving one session either joins another session or becomes inactive. Although this is very similar to a mobility event for multicast, where the user leaves one group and rejoins the same group in the next cell, that study has not taken into account the mobility of the users and the associated handoff parameters.

Many of the architectural issues associated with mobile hosts in a multicast environment have been described in [XP97] , [VC99], and [ABN95]. Wu et al. [Wu99], and McAuley et al. [MBM+99] propose ways for taking care of fast delivery of multicast stream when the end-hosts move within a domain. Wu et al. [Wu99] propose a handover solution with pre-registration in order to provide fast-handoff for the multicast streams while moving between subnets. This is accomplished by sending a unicast signal to the neighboring station about the multicast address that it is subscribed to in order for the neighboring station to be able to join the multicast tree even before the client moves in to the neighboring cell. This solution assumes that there is a mobility support agent (MSA) in each subnet that invokes the join message on behalf of the mobile.

McAuley et al. propose mobile multicast proxy [MBM+99] where the proxy's clients do not themselves directly participate in the multicast tree, but the multicast proxy that participates in the multicast tree formation for the groups that its clients are members of. In this case, a multicast proxy performs a function similar to a designated router, however, the multicast proxy can be outside the member's subnet and can forward multicast messages

to its receivers using unicast, multicast or a limited scope broadcast.

There are also proposals to extend Mobile IP to support mobility for multicast users. However, mobile IP-based bi-directional tunneling solution puts the multicasting burden on the Home Agent (HA). In this case, a user desiring to join a certain multicast group joins this group through HA using IGMP. When the user moves to a foreign network, the HA is responsible for tunneling multicast packets to the user. When an HA has a number of users in the same multicast group visiting the same foreign network, tunneling multiple multicast packets to the foreign network is inefficient. If multiple HAs have users in the same visited network that are part of the same multicast group, multiple copies of the same multicast packets are also tunneled.

Mobile multicast (MoM) [WHMB98] uses a Mobile IP-based approach to take care of mobility for multicast traffic. That paper proposes to reduce the problem in bi-directional tunneling when many HAs tunnel the same multicast packet to a foreign network. In this case one HA gets elected to tunnel multicast packets to a foreign network. Range-based MoM [LW00] takes the MoM approach one step further and elects a multicast agent close to FA to tunnel multicast packets to the foreign network.

In order to avoid the duplication of multicast packets being tunneled to foreign networks, one proposed solution is remote subscription. In this case user desiring to join a multicast group will do so in each visited network through the Foreign Agent (FA). However, this requires that after each handoff the user must rejoin a multicast group. In addition, the multicast trees used to route multicast packets will be updated after every handoff to track the multicast group members. Remote subscription mechanism has briefly been introduced in Chapter 2. In order to limit the tree updates or limit duplication of multicast packets, proxy or agent-based solutions have been proposed. For example, in the Mobicast solution [TP00a], users continue to rejoin the multicast group in each visited network. This architecture adopts domain foreign agent (DFA) concept to shield all mobility within the foreign domain from the main multicast delivery tree. In this scenario the DFA will send or

receive the multicast traffic to a multicast group. When the MH is receiving the multicast traffic, DFA will use a translated multicast address within its network to prevent multicast updates due to mobility.

Mysore et al. [MB97] propose a scheme to take care of loss of transient data for the mobile hosts by assigning a location independent unique multicast address to each mobile host. However, this scheme does not discuss the mobility of multicast sessions in a hierarchical environment where the mobiles are assigned locally scoped multicast address.

Multicasting with local scoping becomes more attractive for the mobile users experiencing intra-domain hand-off because of its ease of deployment and its ability to provide more flexible services such as localized advertisement, news broadcast, and location specific information in the wireless environment. Multicasting with local scoping also takes care of global multicast address assignment problem. My proposed approach expedites stream delivery in a hierarchical multicast environment. It takes advantage of localized scope-based IP multicasting techniques in a wireless environment that could be applicable to Internet Radio and TV surfing in a mobile Internet.

8.5 Mobility in hierarchical multicast architecture

Many of the existing solutions are layer 2- and layer 3-based techniques and have not considered application layer techniques. They also do not take into account localized multicasting in a hierarchically scoped environment where the mobile hosts could be operating in a private network with limited scope. Localized multicasting implies that the clients are assigned multicast address with local scoping where time-to-live (TTL) decides the limit to a very few subnets. This approach will avoid multicast address exhaustion and will reduce the overlapping of multicast addresses.

My proposed fast handoff techniques are based on application layer triggers and apply many of the fast-handoff techniques, namely proactive and parallel operations as defined in

Chapter 5.

I have designed and prototyped a hierarchical scope-based multicast content distribution network called MarconiNet [DS01]. This architecture uses the IETF protocols, SDP (Session Description Protocol) [HJ98], SAP [HPW00], SIP [RSC+02] and benefits from RTCP (Real-Time Transport Control Protocol) to provide many flexible features such as localized advertisement, news broadcast, location specific information, QoS guarantee and optimized intra-domain handoff for the mobile users. There are four main functional components in this architecture, namely radio station client (RSC), radio antenna server (RAS) or local station, advertisement or media server and Internet multimedia client (IMC). Figure 8.4 shows the functional diagram for hierarchical streaming architecture that I have implemented and experimented with fast-handoff techniques for multicast mobility. This architecture assumes that there is multicast connectivity throughout the network, but if there is lack of multicast connectivity at certain parts of the network, then there are some possible application layer solutions [Fin03] that can be deployed. I have described different ways of supporting mobility between the multicast enabled networks and non-multicast enabled networks [DCC+03] by using local proxy-based and UDP-based tunnels.

The proposed multicasting architecture consists of two tiered (hierarchically-scoped) IP multicast sessions. At the higher level of the two, global multicast association exists between the broadcasting stations (RSCs) and the local stations (RASs). At the lower level of the hierarchy, a locally scoped multicast session is created for each broadcasting station between the server and the listening clients (IMC) that can be privately scoped. The local server interacts with the advertisement server to provide stream control using protocols such as SIP and RTSP [SRL98]. I have prototyped different functional components of the architecture such as local and global program management, channel monitor, application layer triggering, security and handoff involving multiple servers. However, in this thesis, I focus my experimental analysis on fast-handoff techniques only.

I have described the details of the functional modules associated with this architecture

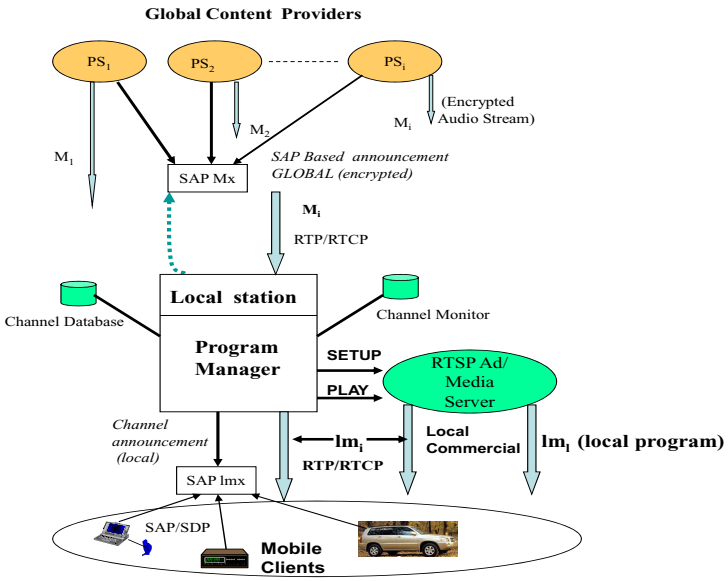


Figure 8.4: Hierarchical scope-based streaming architecture

in [DS04]. I briefly describe below the functions of each of these modules.

8.5.1 Channel announcement

A global streaming server (e.g., Radio/TV station or an individual broadcaster) can broadcast its programs potentially to the global audience. Thus, a global station RSC_i sends its programs live on a unique multicast address M_i globally scoped and encrypted over RTP/UDP (where M_i differs for each station). These global broadcasting stations send their session announcement using subset of SDP parameters to a global multicast address M_x which is also encrypted. This common global multicast M_x address contains a list of programs broadcast by the main radio stations (RSCs). I have designed a java based interface called JSDR that provides a hierarchical searching functionality compared to traditional SDR tool [Han96].

8.5.2 Channel management

The channel management module manages global and local programs in the local server. Each RAS (Radio Antenna Server or local server) gets a global encryption key that it uses to listen to the global common multicast address M_x and get the listing of the channels. The local server broadcasts part of the list to the local domain, and hence creates a local announcement database. The subset of channel descriptions announced by each global station provides sufficient data for building a local channel database. This local announcement database contains the list of the supported channels, each with their appropriate attributes, such as name of the program, duration, type of content, place of origin. Local station sends this program-index to a locally scoped common multicast address lm_x for announcement using SAP (Session Announcement Protocol) [HPW00]. SAP helps to announce the multicast session directory. A SAP announcer periodically multicasts an announcement packet to a well known multicast address and port and a SAP listener on a multicast client learns of the multicast scopes it is within and listens on the well known SAP address and port for those scopes. SAP uses session description protocol (SDP) [HJ98] to describe all the session parameters. SDP is intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. The announcement on lm_x is not encrypted since it gives the ability to the local stations to figure out what is being relayed by the local station. RAS also maintains a pair of multicast addresses for each channel. It keeps the mapping of the globally scoped multicast address M_i on which the radio station sends its program and the locally scoped multicast address lm_i where it gets relayed to. RAS receives the audio stream on the global multicast address M_i and redirects it onto the local multicast address lm_i for the IMCs (Internet Multimedia Clients). Local programs are sent on a specific locally scoped multicast address lm_l.

The client keeps on sending RTCP (Real-time Transport Control Protocol) packets to the management server as long as it receives the audio streams over RTP on a particular multicast address. Information from the RTCP packets can be used for billing, audio quality

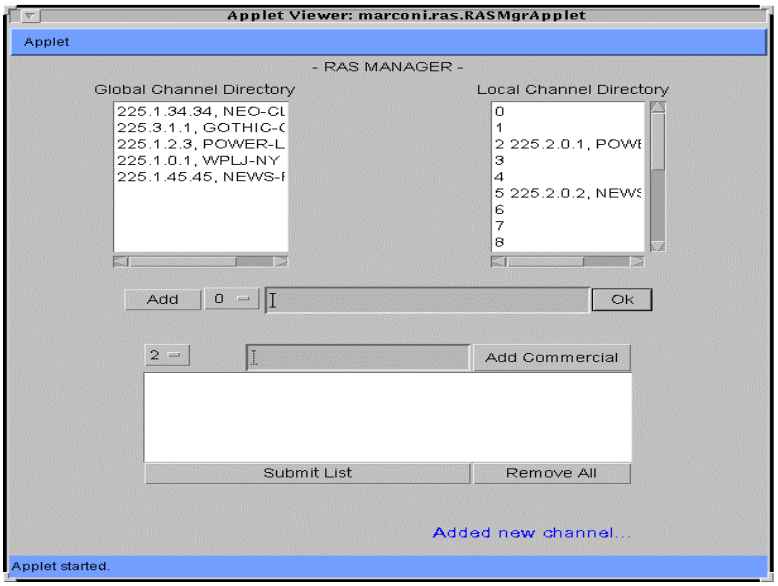


Figure 8.5: Channel manager at the local server

feedback and also membership information for a particular multicast group.

Figure 8.5 shows a screenshot of the channel manager that I have implemented in the testbed. It shows the listing of global program, local program and local advertisement insertion.

8.5.3 Channel tuning

Internet multimedia clients tune to the locally-scoped common multicast address lm_x to determine the currently available program using the JSDR tuner that is based on SAP and SDP. According to the SAP specification, the antenna server will update the announcement information every several minutes or so. The client can tune to a particular channel to get the details of the program that is available.

8.5.4 Local advertisement insertion

Localized information insertion can either be provisioned or it can be event driven based on external event such as an emergency notification. It is assumed that each global broadcast-

ing station is aware of the starting time and duration for commercial break ahead of time or has the control over the time for break.

Through the RTCP report, the global station notifies the local stations for commercial break. On receiving the signal for commercial break, the management server at the local station requests the local RTSP (Real Time Streaming Protocol) server to play the local advertisement on a specific locally scoped multicast address lm_l assigned to that station. It uses a set of RTSP commands like SETUP, PLAY and STOP to control the stream delivery on the locally scoped multicast address. During this time, the local server stops forwarding the RTP stream from M_i to lm_i in the local domain. The local advertisement runs for a specific time based on the information conveyed by RTCP reports. After the advertisement time is over, the local server begins relaying the global program.

8.5.5 Channel monitor

The channel monitor provides statistics of how many clients are tuned to a specific multicast address. For each local channel being diverted, an additional RTCP signaling channel is created with different port. Each listener periodically sends RTCP SDES (Source Description) packets to notify the Local Station (RAS) who is listening to what. SDES is one of the five types of packets that RTCP offers with packet type 2. The SDES packet is a three-level structure composed of a header and zero or more chunks, each of which is composed of items describing the source identified in that chunk. SDES packet can provide participant identification and supplementary details, such as location, e-mail address and telephone number. RAS maps each listener to the desired channel, which in turn increases the number of listeners for that particular channel. The listener-to-channel mapping is destroyed via RTCP BYE packets or via RTCP timeout feature. This also decreases the number of listeners for the associated channel. Figure 8.6 shows the screenshot of the channel monitor that gives the statistics of how many clients are tuned to a particular channel.

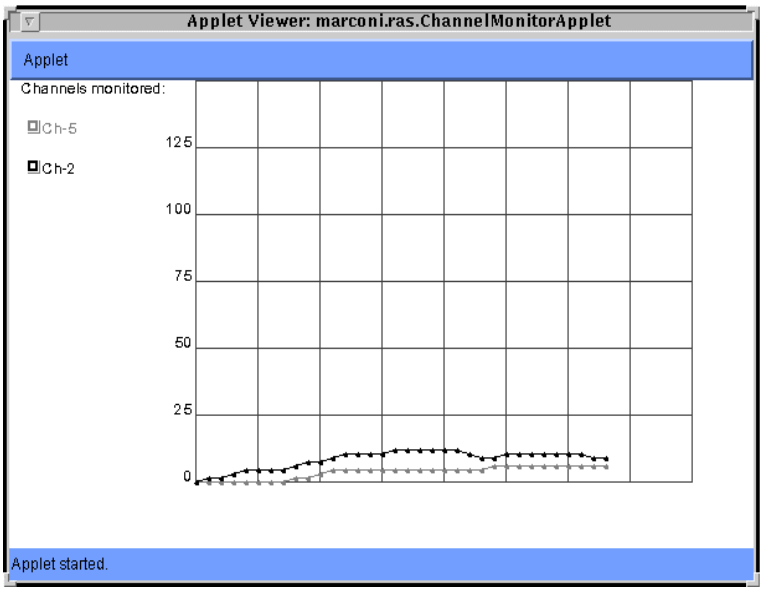


Figure 8.6: Channel monitor

8.5.6 Security

The proposed architecture offers four levels of security overall, namely global announcement encryption, global multicast stream encryption, local audit encryption, and user authentication. By using global announcement encryption, one can separate global announcements from the local announcements. The local IMC (Internet Multimedia Client) do not get access to the global announcements and can only view the local announcements. By using a global encryption key during the announcement by the global radio stations, the scheme does not allow the local Internet multimedia clients to find out about the global channels and thus gives the control over to the local stations to announce only a subset of these channels to the local clients. The security model for global multicast stream should effectively prevent IMCs, as well as the non-paying RASs, from receiving the broadcast content. Thus, each radio station (RSC) must maintain a secret key and encrypt all outgoing content so that only ciphertext stream is transmitted. The basic strategy is to generate a symmetric encryption key at the station and securely distribute this key to a particular RAS upon its payment. Global multicast stream encryption can also be extended to local

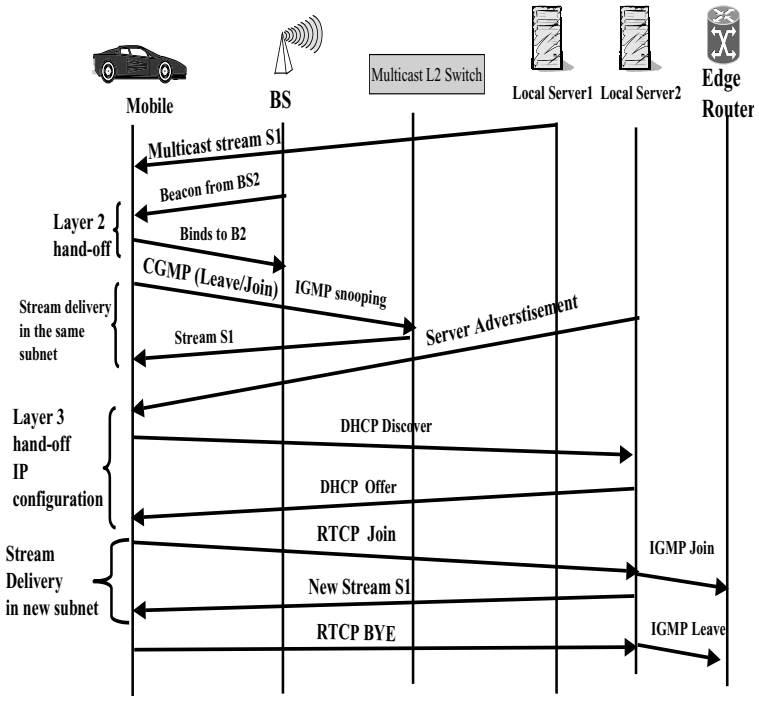


Figure 8.7: Handoff flow for multicast traffic

section as a second level hierarchy. Advertising companies can also be authenticated so that unauthorized companies cannot hijack the local advertisement insertion system.

8.6 Optimization techniques for multicast media delivery

In this section, I propose few mobility optimization techniques for multicast streaming and experiment with these techniques in a hierarchical scope-based multicast system that I discussed in Section 8.4 and have implemented these in a testbed. Figure 8.7 shows the protocol flow for multicast media delivery during handoff that uses a combination of RTCP- and IGMP-based triggering.

In the following section, I describe four different optimization techniques that provide faster multicast stream delivery by reducing the join latency. My proposed techniques are based on remote subscription-based approaches and do not use mobile IP and thus avoid the tunnels between the home network and visited networks.

Figure 8.8 shows an experimental testbed where I have implemented the functional components and have demonstrated several fast-handoff techniques for multicast streaming application. I_a, I_b, I_c I_d are the globally routable subnets connected to the primary interfaces of the local servers S1, S2 and S3, and S4 respectively, whereas I_a, I_b, I_c are the local subnets connected to the secondary interfaces of the respective servers. The access points are connected to the secondary interfaces of the respective servers. In this case, the mobile (represented as an automobile) does handoff between the cells and is thus subjected to layer 2 and layer 3 handoff.

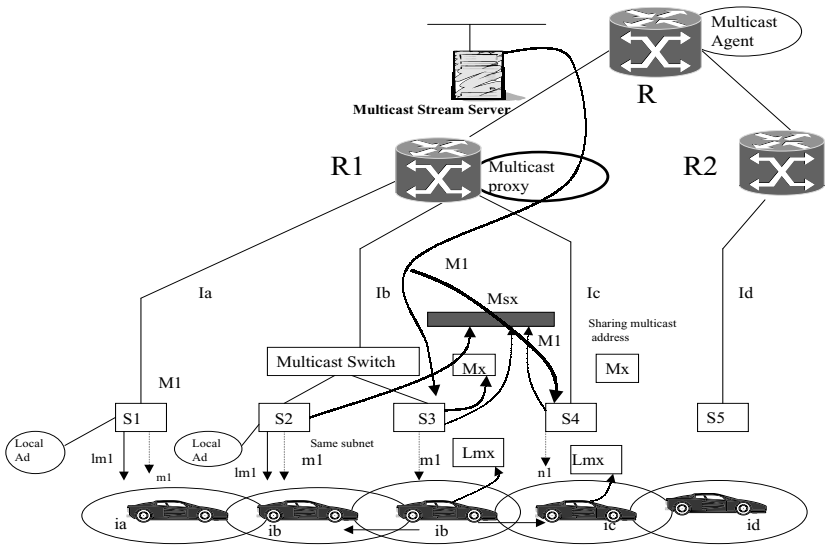


Figure 8.8: Fast-handoff for multicast stream

8.6.1 Reactive triggering

My proposed reactive triggering technique uses a combination of application layer triggering using RTCP and IGMP. Normally, when a client moves to a new subnet, it sends the join request via IGMP. According to [KMG99] IGMP can also be modified to provide aggregate group report to reduce the join latency. As part of the proposed technique, I have implemented an application layer triggering mechanism based on RTCP to facilitate the

join and leave processes. Triggering at the lower level of the hierarchy is accomplished using RTCP, however, the local server triggers the upstream router using IGMP. Using an RTCP-based triggering technique offers a solution at the user space compared to IGMP which works on network layer. Figure 8.7 illustrates the communication flow among mobile, base station, local servers and the upstream router. It illustrates reactive triggering of multicast stream using RTCP-based application layer triggering techniques on lower hierarchy. In this specific figure, it shows how the mobile node moves from one cell to another cell and in the process is subjected to layer 2 and layer 3 handoff. In this approach, the mobile does not need to wait for the IGMP report, nor does it wait for a random time to send IGMP query response. The mobile also uses RTCP BYE packet when it decides to leave a specific multicast group. A BYE packet is RTCP packet of type 203 and it is generated when a participant leaves the session or when it changes its SSRC. This application layer process helps to reduce the leave latency caused by IGMP leave process.

8.6.2 Proactive triggering

Proactive join method reduces the join latency for a client that is about to handoff at the expense of additional bandwidth in the adjacent cell for certain duration. I describe two kinds of proactive join schemes below: pre-registration with a multicast proxy agent and pre-registration with a multicast address announcer.

8.6.2.1 Pre-registration with multicast proxy agent

In the first approach, I propose proxy agents in each subnet. These proxy agents join the upstream multicast tree on behalf of the local downstream servers even before the clients move into the new subnet. Thus, the multicast proxy sends the IGMP query messages to its upstream router beforehand on behalf of the local servers and can help forward the global stream on the respective global multicast addresses to the areas where these clients are about to move in the neighboring subnet for a specific period of time determined by the

client entering to the cell.

As shown in Figure 8.8, router R1 is equipped with a multicast proxy agent. This multicast proxy is on the same subnet as the streaming server S4. Multicast proxy module can also be installed in a dedicated server. The local server S3 proactively notifies the proxy agent about the impending host's subscribed global multicast address (M1). On receiving the notification from the local server S3, the multicast proxy joins the upstream router using IGMP on behalf of server S4. Since the multicast proxy and the server S4 share the same subnet, it helps the server S4 to join the multicast tree ahead of time before the mobile moves into neighboring cell. After the mobile has moved into the neighboring cell, the mobile receives the traffic using RTCP trigger.

8.6.2.2 Pre-registration with a multicast address announcer

The second approach does not include a proxy agent. Rather, for each of neighboring stations sharing an overlapping area with another station there is an associated multicast announcement address. Each local server can subscribe to this address and find out the group address that incoming client is subscribed to. Just before a mobile node leaves the cell the mobile sends the movement imminent signal to the local announcement address about the currently subscribed address. The local server in turn announces this address to the shared multicast addresses that the neighboring local stations subscribe to in the globally scoped address space. In the absence of this association, the neighboring server sends an IGMP message to the upstream router and redirects the stream to the local cells even before the client has moved to the new cell. This helps minimizing the interruption of multicast data.

This mechanism has also been illustrated in Figure 8.8. In this figure, M_{sx} is the announcement address where the neigboring servers (e.g., S2, S3, and S4) about the multicast address of the impending host. Each of the neighboring local servers learns about the subscribed address by the mobile by tuning to the announcement address. This specific

method avoids the additional proxy agent on each subnet.

8.6.3 Triggering during mobile's configuration

Using this approach, mobile's group membership information can also be passed during the client's configuration in the new network. Right after the node has handed over to a new subnet it can send the request for the previously subscribed multicast address as part of DHCP discover message during layer 3 configuration process. During the process of obtaining the IP address from the DHCP server, the client can send the unsolicited "JOIN" to the server for the desired locally scoped multicast address. The server in turn can join the upstream multicast router for the desired multicast group. Thus, the server can join the desired multicast group at the same time the client is in the process of getting its new IP address configured. This process allows the client to join a multicast group during client's configuration itself. This is an example of optimization technique where the client performs two operations in parallel, namely configuration and join during the process of configuring its layer 3 identifier.

There are trade-offs associated with each of these above optimization techniques. Application layer triggering technique can only be applicable to RTP-based traffic, as it heavily depends upon RTCP report. Pre-registration technique helps to reduce the join latency at the cost of bandwidth in the previous cell and subnet. The JOIN operation during mobile's configuration process provides a better solution as it does not need any additional network element but needs more resources due to parallel operation of "Join" operation and layer 3 configuration.

8.7 Experimental results and performance analysis

In this section, I highlight the experimental results where I have optimized the join latency and compare these results with the non-optimized version. I also perform an analytical

comparison of the optimized versions with non-optimized part.

8.7.1 Experimental results

I have conducted a series of handoff experiments for multicast traffic involving cell and subnet mobility to study the effect of handoff on multicast traffic and how the proposed optimization techniques can improve the handoff performance. I used multimedia applications such as RAT (Robust Audio Tool) [SHK+95], VIC (Video Conferencing) and measured the time for movement detection, IP address acquisition, join and leave latency by using network layer IGMP, CGMP (Cisco Group Management Protocol) and application layer RTCP signaling. I mainly focus on improving join and leave latencies in the testbed. Figure 8.9 shows the testbed where I have experimented with fast-handoff using a multicast proxy. Figures 8.10, 8.11 and 8.12 demonstrate the effect of layer 2 and layer 3 handoff on multicast stream for join latency and leave latency, and Figure 8.14 shows how the proactive optimization techniques improve the handoff performance by reducing the packet loss. Figure 8.13 illustrates the results from ping-pong experiment i.e., a mobile moves back and forth between subnets in rapid succession and effect of leave latency. I explain each of these figures in more detail below.

Figure 8.10 shows the effect of layer 2 handoff on multicast stream delivery. As shown in Figure 8.7, the mobile is subjected to a layer 2 handoff when it moves from server S2 to server S3. During layer 2 handoff, there is no join latency, as the subnet does not change during mobile's movement and I did not use a multicast switch that would have contributed to layer 2 join latency, but there is waste of bandwidth due to leave latency in the previous network. Figure 8.10 shows the experimental results for multicast mobility during a layer 2 handoff. Since the mobile does not change any subnet, the mobile does not need to use any configuration protocol such as DRCP to configure itself and thus DRCP packet sequence does not appear in Figure 8.10.

Figure 8.11 shows the effect of layer 3 handoff on join latency of multicast stream deliv-

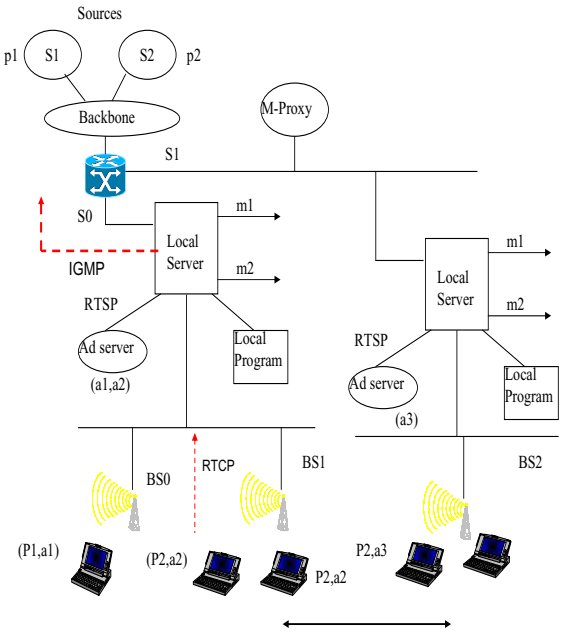


Figure 8.9: Experimental testbed for handoff

ery. As per Figure 8.8, this scenario reflects when the mobile client moves from local server S3 to S4 and in the process changes its subnet and IP address (e.g., IP address changes from I_a to I_b). As shown in Figure 8.11 a join latency of 60 seconds is observed during subnet movement. Figure 8.11 shows the sequence of execution of several protocols, namely RTP, DRCP (Dynamic Rapid Configuration Protocol), IGMP router query and client query response during the handoff. As soon as the layer 2 handoff is over, the mobile uses DRCP to obtain the new IP address. The router sends periodic IGMP query to a well known multicast address. In the absence of unsolicited IGMP join operation, the mobile waits to receive the IGMP query and then sends the IGMP query response. The upstream router then joins the multicast tree before the RTP traffic is received by the mobile. This process contributes to the 60 second multicast join latency.

Figure 8.12 shows the effect of layer 3 handoff on leave latency for the multicast stream during subnet handoff. The mobile is subjected to a maximum leave latency of 3 minutes. This contributes to the waste of bandwidth in the previous cell. However, leave latency can

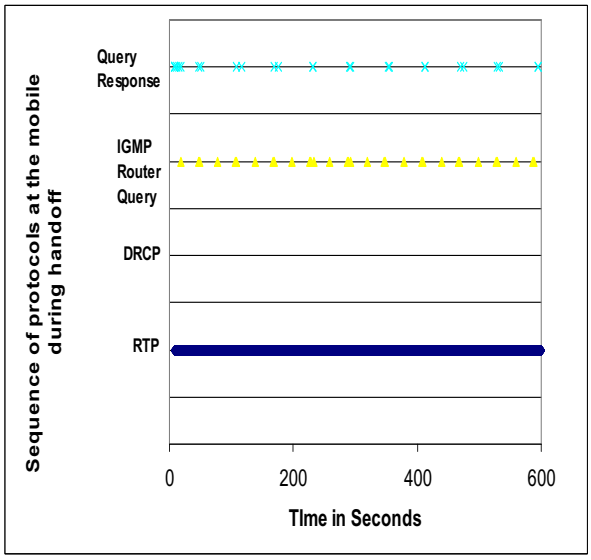


Figure 8.10: Effect of layer 2 handoff on multicast

help to reduce the effect of join latency when a mobile is subjected to handoff back and forth successively in a rapid manner often known as ping-pong.

Figure 8.13 shows the scenario when the mobile is subjected to multiple handoffs. In addition, it shows the ping-pong effect i.e., mobile moves back and forth between two subnets very frequently. Because of the associated leave latency, when the mobile returns to the previous subnet due to ping pong movement, packet loss is reduced.

Figure 8.14 shows the results of how proxy-assisted proactive join techniques can reduce the join latency to almost zero. Although during the experiment, the mobile node moved back and forth between the subnets, I have shown the instance of one subnet move. Since the mobile leaves the previous network, it cannot send the leave report to the access router. Thus, this proactive optimization technique that reduces join latency cannot reduce the leave latency in the previous subnet. Leave latency in the previous network can be reduced if a proxy in the previous network sends the leave report to the access router on behalf of the mobile or the mobile sends a leave report when it is anticipating to move to the new network. Kim and Han [KH04] describe the use of multicast handoff agent (MHA) in the base stations that send *join* and *leave* messages on behalf of the mobile. In the

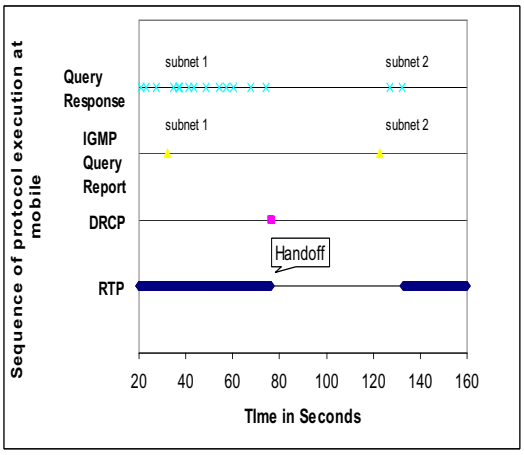


Figure 8.11: Effect of layer 3 handoff on join latency

next section, I describe the performance analysis of optimized version and non-optimized version.

8.7.2 Performance analysis

The multicast handoff latency consists of the IGMP query waiting time, the random backoff time for IGMP report and multicast tree establishment time. Some of the parameters that are used for performance evaluation are as follows:

T_Q: the interval time between the IGMP query messages sent by the multicast router,

T_c = RTCP interval from the client,

T_R: the random backoff time before the response is sent by the client,

T_h: the layer 2 and layer 3 handoff delay,

T_d: the transmission delay of the wired link,

T_w: the transmission delay of the wireless link,

P_m: the probability that there is at least one mobile in target network subscribed to the multicast group as the mobile in the current network.

For performance analysis certain assumptions are made: For the sake of simplicity the processing time for IGMP messages and processing time for proactive operation are ig-

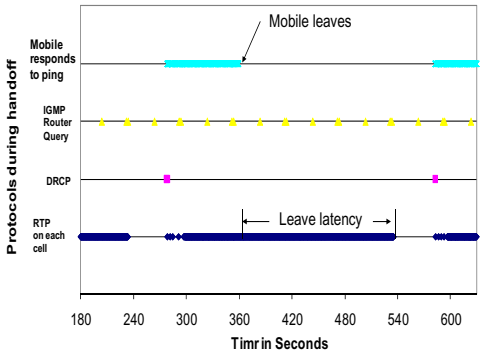


Figure 8.12: Effect of layer 3 handoff on leave latency

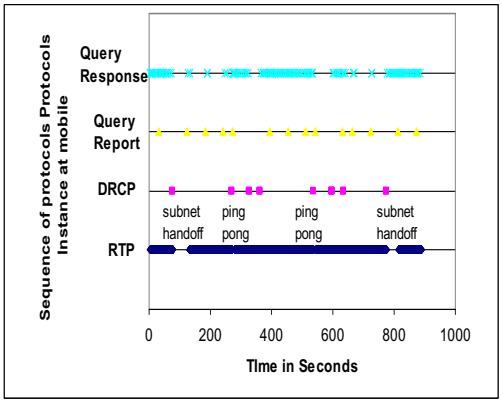


Figure 8.13: Effect of ping-pong on multicast traffic

nored. Packet transmission delay for both signaling (e.g., IGMP query and IGMP response) and media (e.g., RTP packet) are assumed to be same.

8.7.2.1 Non-optimized version

In case of non-optimized version, the total handoff latency for a multicast stream including the media delivery delay can be defined as

$$T_L = T_h + (2(T_d + T_w) + T_R + T_Q) \times (1 - P_m) \tag{8.1}$$

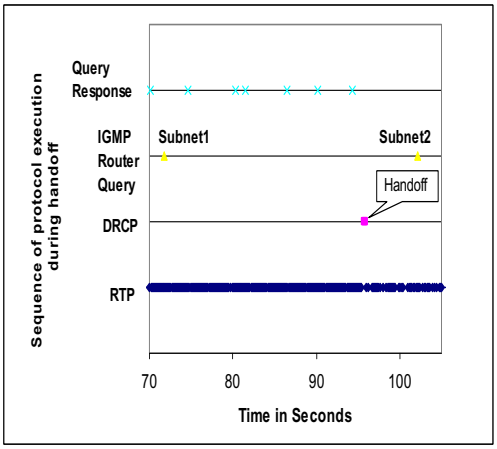


Figure 8.14: Effect of proactive join technique on multicast traffic

Although layer 2 and layer 3 handoff latencies are common to all the four handoff approaches, the join latency will vary for each of these proposed optimized versions. In case of an non-optimized handoff that uses IGMP-based handoff, the *join* latency is denoted as

$$T_{join} = (T_R + T_Q + T_d + T_w) \times (1 - P_m) \tag{8.2}$$

8.7.2.2 Reactive triggering using RTCP

In case of RTCP-based reactive triggering approach, the client does not need to generate an IGMP query response. The client sends an RTCP join as soon as it detects the new network. The local server sends an unsolicited *join* to its upstream router without waiting for the periodic IGMP router query report. This process eliminates the delay contributed by the periodic IGMP query interval. Thus, the *join* latency during RTCP-based triggering approach is denoted as,

$$T_{join} = (T_c + T_d + T_w) \times (1 - P_m) \tag{8.3}$$

and handoff latency including media delivery is denoted as

$$T_L = T_h + (T_c + 2(T_d + T_w)) \times (1 - P_m) \tag{8.4}$$

8.7.2.3 Proxy-based proactive join

In case of proactive join scenario, where the local server joins on behalf of the mobile before the mobile moves into the target network, the join latency is denoted as

$$T_{join} = T_w \times (1 - P_m) \tag{8.5}$$

and the handoff latency including media delivery is denoted as

$$T_L = T_h + 2 \times T_w \times (1 - P_m) \tag{8.6}$$

8.7.2.4 Join during handoff

In case where the mobile joins the multicast tree during the configuration process resulting in parallel operation, there is no additional *join* latency, the join latency is included as part of layer 2 and layer 3 handoff delay. Thus, the total handoff latency including media delivery in case of multicast join during configuration process is:

$$T_L = T_h + (2 \times T_d + T_w) \times (1 - P_m) \tag{8.7}$$

From a sample router configuration, I take the following values to get the results and compare the non-optimized version with all the three optimized versions and also compare the three optimized versions with each other.

T_R = IGMP Query interval = 60 sec

T_c = RTCP join interval = 5 sec

T_Q = Random backoff time = 10 sec

T_h = L2 and L3 handoff = 3 sec

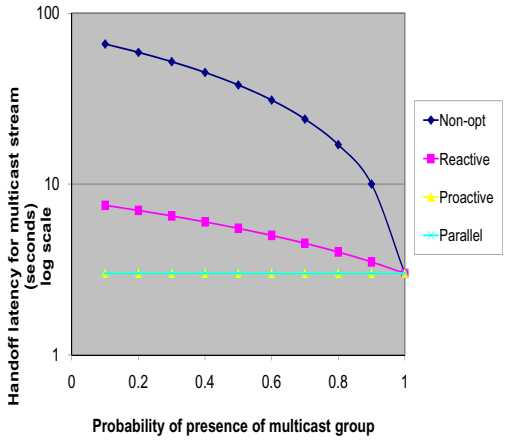


Figure 8.15: Comparison of non-optimized vs. optimized techniques

T_d = packet transmission delay on the wired side 1 ms

T_w = packet transmission delay on the wireless access 5 ms

Figure 8.15 shows the comparison of three optimized versions with the non-optimized version. Figure 8.16 compares the handoff latency among the three optimized versions, reactive, proactive and parallel. As it is evident from both the figures, when there is another client subscribed to the same multicast address in the neighboring subnet (i.e., probability of presence of multicast group set to 1) all the three optimized approaches and the non-optimized do not suffer from any join latency. However, when there is no other client present in the neighboring subnet (i.e., probability of presence of multicast group is 0), proactive and parallel operations produce the same results and work better than reactive and non-optimized versions.

8.8 Concluding remarks

The proposed hierarchical scope-based multicast streaming architecture provides local control while distributing content over the Internet by using local proxies at the edges of the network. These proxies receive the multicast traffic on a global multicast address and transmits it on a locally scoped multicast address. Having the ability to use locally scoped addresses

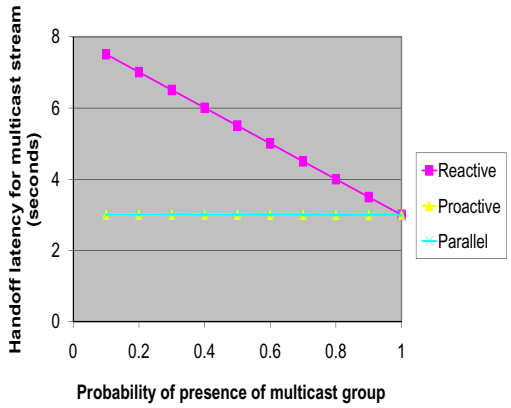


Figure 8.16: Comparison of optimized techniques

will alleviate the global multicast address allocation problem or overlapping problem for the multicast receivers in the neighboring networks. Use of the local proxies and real-time localized advertisement insertion mechanism using RTCP feedback provides a good way to manage local program and global program based on the demography and interest of the local community. Application layer triggering techniques avoid the Join delays contributed by IGMP-based router query reports that may take up to 2 minutes during a mobile's movement across subnets.

Although proxy-assisted proactive technique provides better performance compared to the application layer reactive technique, based on the movement pattern of the mobile, one can either use proxy-assisted proactive multicast or application layer reactive triggering technique or combination of both. For example, a proactive multicasting technique would work the best when the mobile can discover the neighboring networks ahead of time and can predict the target network it may move to, such as a vehicle moving in a high-way. In a city like environment where the movement of the mobile cannot be predicted easily, application layer reactive triggering techniques will work better. On the other hand, even if the multicast join during mobile's configuration provides comparable performance as proactive triggering, one needs to extend the configuration protocol such as DHCP and it is also essential that the local streaming server is equipped with DHCP server functionality. Thus,

based on a specific performance requirement, network topology, and mobile's movement pattern, a mobile can adopt either of the proposed optimization techniques.

Chapter 9

System evaluation

In this chapter, I evaluate the overall handoff system where many of the optimization techniques that I have developed in Chapter 5 function together to build the complete handoff system. I first illustrate the experimental results from few of these systems for both inter-technology and intra-technology handoff and then validate some of the optimization techniques using Petri net modeling. I demonstrate how Petri net can model some of the behavioral properties of the handoff system such as deadlocks. I also compare the performance of a few scheduling techniques that could be applied to handoff.

9.1 Summary of key contribution and indicative results

Currently, optimization techniques for each of the handoff components at different layers are implemented independently. These optimized components have not been used in an integrated fashion to build a complete optimized handoff system. Existing experimental results from the optimization techniques are derived for each of the individual handoff component only (e.g., discovery, authentication, configuration). There is also no work that compares the model based results with experimental results to verify the prediction of systems performance. There is no existing system evaluation technique that can verify the correctness of the handoff system or detect the system anomaly of the handoff system. such

as deadlocks.

I built several indicative handoff systems using the reactive, proactive and cross layer optimization techniques that I have developed for each of the handoff components as described in Chapter 5. I built the equivalent mobility models for each of these handoff indicative systems and compared the experimental results with the results from the equivalent mobility models to determine how changes in certain handoff parameters affect the overall handoff system.

Indicative results from systems evaluation of the experimental systems demonstrate the effectiveness of the optimization techniques that I have developed. These results also demonstrate the feasible scenarios where some of these optimization techniques can work together to produce a partial or complete handoff system. Comparing the results from Petri net-based mobility models with that of experiments for these optimization techniques demonstrate the correctness of prediction. Petri net-based behavioral analysis of these optimization techniques can determine the existence of system anomaly such as deadlocks.

A handoff system that uses many of these optimization techniques for different handoff components indicates possible sequence of these handoff components i.e., if some of these can work in parallel and verify how the cross layer techniques can be used to expedite many of the handoff operations that work in sequence. Verifying the results from the experimental system with that of corresponding mobility models demonstrate the effectiveness of these mobility models. These mobility models can also evaluate the systems performance based on the sequence of handoff operations and resource availability. It can also determine how the level of concurrency and additional resources may affect the systems performance. The behavioral analysis of the mobility model can demonstrate certain operational aspects of the system such as deadlocks that would not be evident otherwise from the experimental results.

In the rest of the Chapter, I discuss the experimental results from a few indicative systems, results from Petri net based models for several optimization techniques that I de-

veloped, verification of systems performance for different handoff sequences, deadlock detection and deadlock avoidance.

9.2 Introduction

Systems evaluation and validation of several optimization techniques associated with a handoff event can be implemented through experimental analysis, simulation and analytical modeling. While experimental results are limited by several constraints, such as systems parameters, namely memory, CPU power and other network parameters such as bandwidth, Petri net models can be used to validate the experimental results and perform systems evaluation with the ability to vary the systems parameters. Thus, in order to validate various optimization techniques associated with the handoff system, I have applied both the experimental and modeling approaches. Optimization techniques for many of the handoff components have been described in Chapter 5, each with its own experimental results. However, in order to validate the system performance, I have built a handoff system by implementing the optimization techniques for these handoff components that work together.

9.3 Experimental validation

In this section, I describe the experimental results from a handoff system that support multiple types of handoffs, namely *intra-technology* and *inter-technology* where the mobile uses single interface and multiple interfaces, respectively. These handoff systems use a set of optimization techniques that I have described earlier in Chapter 5.

In particular, I describe the results from three experimental systems that use optimization techniques to reduce the handoff delay and packet loss. The three experimental systems are, 1) Media independent preauthentication framework, 2) Cross layer trigger assisted preauthentication 3) Optimized handoff in IMS-based network.

9.3.1 Media independent pre-authentication framework

The experimental results show that without any optimization, 4 seconds of handover delay is observed when the mobile moves between two homogeneous access networks (e.g., between 802.11 networks) and 200 packets were lost due to this handover delay. The situation is worse for heterogeneous handover, where it may take up to 15 seconds for the mobile to authenticate and establish connectivity in CDMA network.

I have prototyped a mobility system called media independent pre-authentication (MPA) [DZO+05] that utilizes many of the techniques that I have developed to optimize the basic handoff operations, namely network discovery, authentication, configuration, security association, and binding update. This system also reduces the link layer handoff delay by avoiding scanning and applies the cross layer optimization techniques, such as "link up" and "link down" events. In addition, it reduces packet loss by implementing the dynamic buffering and copy-and-forward mechanism [DvF+06] at the edges of the network.

I have applied these proactive techniques to optimize both network layer and application layer mobility protocols over both single interface or multiple interfaces [DKZ+05], [DZO+05].

Koodli et al. [Koo05] and Gwon et al. [GFJ03] have developed mobility systems that utilize the proactive handoff techniques for both Mobile IPv6 and Mobile IPv4, respectively. However, these systems do not address proactive discovery or pre-authentication mechanisms. Also, these mechanisms require signaling exchange between the neighboring routers that work only if the routers have established trust relationships among them. I performed a comparative analysis of media independent pre-authentication (MPA) mechanism with the proactive mode of fast-handoff mechanism of FastMIPv6 [Koo05] and have described the details of the results in [DDF+07].

Figure 9.1 shows the protocol flow based on an MPA-based framework. Assume that the mobile node is already connected to a point of attachment referred to as the old point of attachment (oPoA), and is assigned an old care-of address (oCoA). Throughout the com-

munication flow, data packets should not be lost except for the period during the layer 2 switching procedure in step 5, but MPA procedures can help minimize the packet loss during this layer 2 handover period with the help of the information service (IS), event service (ES) and command service (CS) of IEEE 802.21. I briefly describe different functional phases of the pre-authentication framework.

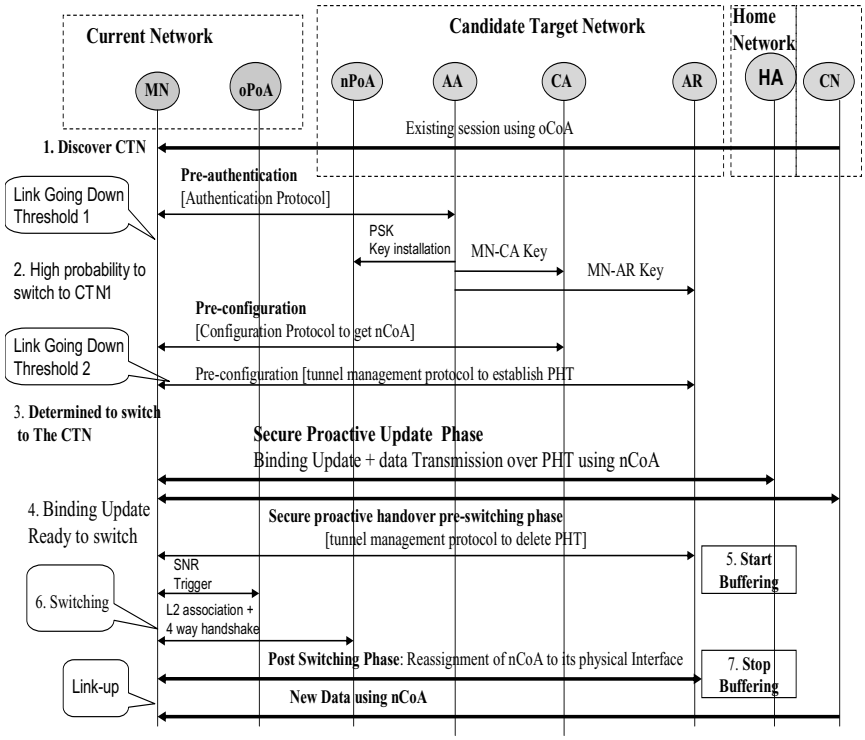


Figure 9.1: Protocol flow for media independent pre-authentication

9.3.1.1 Pre-authentication phase:

The mobile finds a CTN (Candidate Target Network) through a discovery process, such as IEEE 802.21, and obtains the address and capabilities of the AA (Authentication Agent), CA (Configuration Agent), and AR (Access Router) in the CTN. The mobile pre-authenticates with the authentication agent. If the pre-authentication is successful, an MPA-SA (Security Association) is created between the mobile node and the authentication agent. Two keys are derived from the MPA-SA, namely, an MN-CA key and an MN-AR key, which are

used to protect subsequent signaling messages of a configuration protocol and a tunnel management protocol, respectively. The MN-CA key and the MN-AR key are then securely delivered to the configuration agent and the access router, respectively. Layer-2 pre-authentication can be initiated at this stage.

9.3.1.2 Pre-configuration phase:

The mobile node realizes that its point of attachment is likely to change from oPoA (Old point-of-attachment) to a new point-of-attachment (nPoA). It then performs pre-configuration, with the configuration agent using the configuration protocol to obtain an IP address, say nCoA (new care-of address) and other configuration parameters from the CTN. The access router uses the tunnel management protocol to establish a proactive handover tunnel with the mobile. As part of the tunnel management protocol, the mobile node registers oCoA and nCoA as the tunnel outer address and the tunnel inner address, respectively. The signaling messages of the pre-configuration protocol are protected using the MN-CA key and the MN-AR key. When the configuration and the access router are co-located in the same device, configuration and tunnel management may be performed by a single protocol such as IKEv2 [Kau05]. After completion of the tunnel establishment, the mobile is able to communicate using both oCoA and nCoA by the end of step 4 in Figure 9.1.

9.3.1.3 Secure proactive handover main phase

Before the mobile switches to the new point of attachment, it starts the secure proactive handover process by executing the proactive binding update operation of any mobility management protocol such as MIPv6 and SIP-based mobility and transmitting subsequent data traffic over the tunnel. In some cases, it may cache multiple nCoA addresses and establish simultaneous binding with the CH (Corresponding Host) or HA (Home Agent).

9.3.1.4 Secure proactive handover pre-switching phase

The mobile completes the binding update and becomes ready to switch to the new point of attachment. The mobile may execute the tunnel management protocol to delete or disable the proactive handover tunnel and cache nCoA after deletion or disabling of the tunnel. A buffering module at the new Access Router (nAR) begins to buffer the packets (start-buffering) when it receives the signal to delete the tunnel. The mobile sends an explicit signal to stop buffering and flush the packets after the mobile connects to the new point-of-attachment. Details of the buffering modules and buffering protocols are described in Chapter 5.

The decision as to when the mobile switches to the new point of attachment depends on the handover policy. In general, mobile-controlled or network-controlled policies can be used to trigger the handoff. The mobile's signal quality, its location, communication cost, and QoS on the received traffic are some factors that can determine the handoff policy. Results presented in this chapter are based on signal-to-noise ratio (SNR) as the trigger to handoff.

9.3.1.5 Switching phase:

Link-layer handover occurs in this step. During this phase, any of the layer 2 security association including EAP-based authentication and 802.11i related 4-way handshake may take place. Normally, layer 2 pre-authentication is taken care of by inbuilt layer 2 pre-authentication support such as 802.11i. However, using MPA scheme, layer 3 pre-authentication can bootstrap layer 2 authentication leaving only the four-way handshake during this phase.

9.3.1.6 Secure proactive handover post-switching phase:

The mobile executes the switching procedure. Upon successful completion of the switching procedure and layer-2 association, the mobile immediately assigns the cached nCoA to the physical interface attached to the new point of attachment. If the proactive handover tunnel

was not deleted or disabled during step 4, the tunnel can be deleted or disabled in this phase as well. After this, direct transmission of data packets using nCoA is possible without using a proactive handover tunnel.

I have applied the MPA related optimization techniques and have experimented with two mobility protocols, namely SIP-based mobility [WS99] and MIPv6 [JPA04] supporting both intra-technology and inter-technology handovers.

9.3.2 Intra-technology handoff

In this section, I highlight experimental results with intra-technology handoff.

An intra-technology handover is defined when a mobile moves between the same type of access technologies, such as between 802.11[a,b,n] and 802.11 [a,b,n] or between CDMA 1xRTT and CDMA 1x EV-DO. In this scenario, a mobile may be equipped with a single interface (with multiple PHY types of the same technology) or with multiple interfaces. An intra-technology handover may involve intra-subnet or inter-subnet movement and thus the mobile may need to change its L3 identifier depending upon the type of movement.

Figure 9.2 shows the topology of the experimental test-bed that I used to experiment intra-technology handoff using 802.11 access networks. The test-bed emulates two different visited domains and a home domain. Each visited domain has several sub-networks. The mobile moves from one visited domain to another domain and in the process changes its subnet. Network A is oPoA where the mobile node (MN) initially resides prior to handover. Network B is the nPoA, Network C is where the correspondent node (CN) resides and finally, Network D is home network where the Home Agent (HA) and AAA server reside.

The configuration protocol is DHCP, the authentication agent (AA) is a PANA [FOP+08] server with a backend Diameter server to carry out EAP-TLS (Extensible Authentication Protocol) [ABV+04]. The configuration agent (CA) is a DHCP Relay Agent and the next access router (nAR) is an edge router that runs over Linux operating system. For IPv6

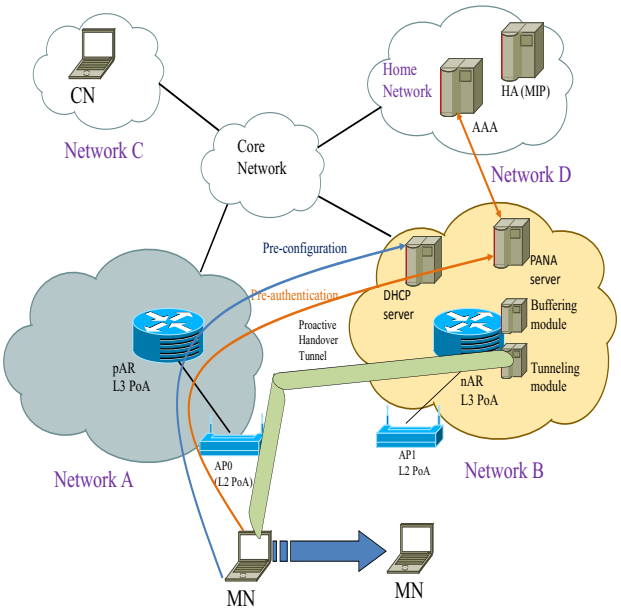


Figure 9.2: Media independent preauthentication testbed

network nAR behaves as the configuration agent in network B.

MPA mechanisms work independent of the underlying mobility management protocols. I have demonstrated the benefits of MPA using both SIP Mobility and MIPv6 as mobility management protocols. In case of MPA for MIPv6, the CN starts an RTP session with the MN while the MN is in Network 1 via the HA using an MIPv6 tunnel. MPA creates a proactive handover tunnel between the MN and nAR in Network B. This is an IPSec tunnel in Encapsulating Security Payload (ESP) mode and I use the Protocol for carrying Authentication for Network Access (PANA) for dynamically establishing and terminating the IPSec tunnel. Before the handoff, the MIPv6 tunneled-traffic between the MN and HA goes through the IPSec tunnel created by MPA with IPSec policy settings. When the configuration agent and router co-locate, a single protocol, such as IKEv2 can take care of both the functions, such as configuration and tunnel management.

In case of SIP-based mobility, an IP-IP-based tunnel is used as proactive handover tunnel between the mobile node (MN) and the next Access Router (nAR). After a successful connection setup using SIP, voice traffic flows between the MN and the CN. This voice traffic is carried over RTP/UDP. I have used RAT (Robust Audio Tool) as the media agent

and the streaming traffic is generated using a CODEC with a spacing of 20 ms between packets. SIP Re-Invite is used as the binding update over the proactive handover tunnel before the mobile moves to the target network. Pre-authentication and buffering procedures are carried out similar to MIPv6.

Table 9.1 shows the experimental results from the mobility system that demonstrate intra-technology handover involving IEEE 802.11 access networks. Both application layer mobility protocol such as SIP-based mobility and network layer mobility such as MIPv6 were used for experimental validation of these techniques. These results demonstrate how several of the proactive optimization techniques can work together to minimize the handover delays, packet loss, and jitter. Results also demonstrate the effect of buffering at the edge routers in reducing packet loss at the expense of added delay. These results shown in Table 9.1 are the average values taken over 5 runs.

I have described implementation and experimental details of the MPA framework for intra technology handoff in [DZO+05], [DFD+08].

9.3.3 Inter-technology handoff

Supporting terminal handovers across heterogeneous access networks such as CDMA, 802.11 and GPRS is a challenge, as each access network has different QoS, security and bandwidth characteristics. A mobile may be equipped with multiple interfaces, where each interface can support a different access technology (802.11, CDMA). A mobile may like to communicate with one interface at any time in order to conserve power. During the handover the mobile may move out of the footprint of one access technology (e.g., 802.11) and move into the footprint of a different access technology (e.g., CDMA). This will require switching of the communicating interface on the mobile as well. This type of inter-technology handover is often called vertical handover since the mobile makes movement between two cells of different sizes. A vertical handover can be termed as upward vertical handover or downward vertical handover based on the direction of movement such as smaller cell to larger

Table 9.1: Delay and packet loss during proactive handoff

Mobility Type	MIPv6				SIP Mobility	
Handoff optimization Parameters	Buffering Disabled + route optimization (RO) Disabled	Buffering Enabled+ RO Disabled	Buffering Disabled +RO Enabled	Buffering Enabled +RO Enabled	Buffering Disabled	Buffering Enabled
L2 handoff (ms)	4.00	4.3	4.00	4.00	4.00	5.00
Avg. packet loss per handoff	1.3	0	0.7	0	1.50	0
Avg. inter-packet interval (ms)	16.00	16.00	16.00	16.00	16.00	16.00
Avg. inter-packet arrival during handover (ms)	21.00	45.00	21.00	67.00	21.00	29.00
Avg. packet jitter (ms)	n/a	29.3	n/a	50.6	n/a	13.00
Buffering period (ms)	n/a	50.00	n/a	50.00	n/a	20.00
Avg. buffered packets	n/a	2.00	n/a	3.00	n/a	3.00

cell or vice versa [SK98]. For example, a mobile moving from 802.11 network to cellular network can be viewed as upward vertical handover. An inter-technology handover may affect the quality-of-service of the multimedia communication, since each access network offers different bandwidth and each of the access specific handoff operations may require different amount of resources.

I have applied the optimization techniques to inter-technology handover involving two specific access technologies, namely IEEE 802.11 and CDMA2000.

Several types of handoff scenarios are possible involving handoff with multiple interfaces. I have experimented with two different handoff scenarios: *Break-before-make sce-*

nario, and *make-before-break scenario*. I describe these scenarios below.

9.3.3.1 Break-before-make scenario

In the normal handoff scenario involving multiple interfaces, the new interface comes up only after the link to the old interface is taken down. This scenario can be termed as "break-before-make" and usually gives rise to undesirable packet loss and handoff delay. In my experimental testbed, without any optimization, the handoff delay and associated packet loss resulted because of PPP configuration delay (16 seconds) and binding update delay e.g., SIP re-INVITE (1.5 seconds) for SIP-based mobility and MIP registration delay (500 ms) for MIPv6. Lower layer triggers, such as "Link Down" can help expedite the handoff process on the second interface and will help reduce the packet loss. In order to optimize scenario 1, I have applied the fast "Link Down" detection technique along with pre-authentication, proactive configuration techniques, buffering, and copy-and-forwarding technique to reduce the packet loss and delay. Using this technique, the mobile was able to resume the communication in the new network within 50 ms and the packet loss was reduced to 0.

I describe below two types of make-before-break scenarios.

9.3.3.2 Make-before-break scenario A

In the first type of make-before-break scenario, the second interface is prepared pro-actively while the mobile still communicates using the old interface, and at some point the mobile decides to use the second interface as the active interface and completes the authentication and binding update procedures. This results in fewer packet loss as it uses "make-before-break" techniques to set up the layer 2 configuration on the new network while the mobile is still connected to the old network. In a typical break-before-make handoff without any optimization, the mobile prepares the CDMA interface only after the current interface (802.11) goes down.

As part of this scenario, I have developed the make-before-break algorithm to support handoff between CDMA and 802.11 networks. Using this technique, the mobile sets up the layer 3 configuration in the CDMA network using the PPP interface while it still keeps on communicating using the current 802.11 interface. This technique reduces layer 2 and layer 3 configuration related delays and packet loss. However, it uses up more power resources, since both the interfaces are active at the same time. I have experimented with this mechanism for both network layer protocol, namely MIPv4 and application layer mobility, namely SIP-based mobility over 802.11 and CDMA 1xRTT access networks. By using the make-before-break technique, there was no packet loss for both MIP and SIP-based mobility. However, initial jitter was observed for the in-flight packets after the handover from 802.11 to CDMA network and out-of-order packets were received after the handover from CDMA to 802.11 networks. I have published a complete experimental handoff analysis involving multiple interfaces in [DKZ+05].

9.3.3.3 Make-before-break scenario B

In the second type of make-before-break scenario, some of the required functions, such as network selection, context transfer of CDMA network parameters such as PPP state and security associations are established ahead of time using the current interface. This scenario helps to conserve energy. By activating the second interface only after an appropriate network has been selected and the mobile has authenticated itself using the old interface, mobile can utilize battery efficiently for authentication.

I have confirmed from the experiments [DKZ+05] that make-before-break technique aided by a combination of proactive optimization methodologies and cross layer triggers can reduce the binding update and configuration delay during handover. Figure 9.3 shows the comparison of audio output at the mobile for both optimized and non-optimized handovers.

Figure 9.3 (A) shows comparison for homogeneous handover, whereas Figure 9.3 (B)

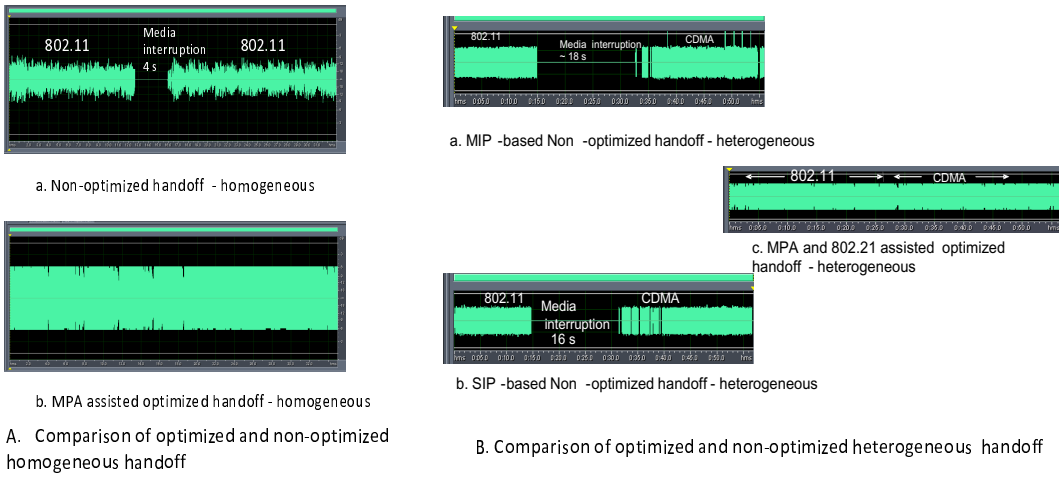


Figure 9.3: Comparison of optimized vs. non-optimized handoff

shows the comparison for handover between 802.11 and CDMA networks. Similar to the results shown in Table 9.1, optimized homogeneous handover does not have any media interruption compared to 4 seconds media interruption for non-optimized handover. Make-before-break optimization technique for heterogeneous handover results in no media interruption compared to 16 seconds media interruption for SIP-based mobility and 18 seconds media interruption for non-optimized handover.

9.3.4 Cross layer trigger assisted pre-authentication

In this section, I describe the experimental results from a handover system that uses pre-authentication techniques assisted by the media independent handover functions as defined by IEEE 802.21. IEEE 802.21-based cross layer triggers help in handover preparation. Figure 9.4 shows the interaction between the MPA related functions and 802.21-based cross layer triggers known as MIHF (Media Independent Handover Functions).

Figure 9.5 depicts the experimental integrated test-bed that uses the MIHF and MPA components shown in Figure 9.4.

The experimental testbed includes two types of access networks, namely EV-DO and Wi-Fi that are connected via a core network infrastructure. The complete testbed consists

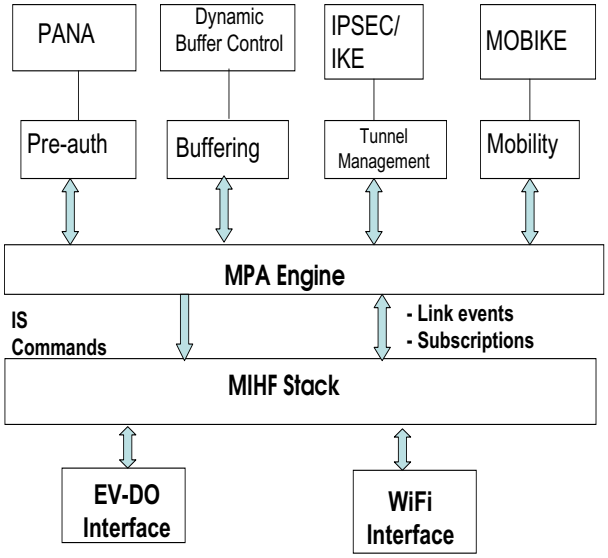


Figure 9.4: Interaction between MIHF and MPA components

of the following entities:

A mobile node (MN) is equipped with IEEE 802.11 (Wi-Fi) and CDMA 1xEVDO interfaces. The IEEE 802.11 (Wi-Fi) and EV-DO interfaces have the IP addresses IP0 and IP1, respectively. The MN runs an MPA client that supports IPSec, IKE, MOBIKE and MIH related services that provide cross layer triggers.

The MPA server is equipped with several modules including an authentication agent (AA), tunneling agent, configuration agent (CA), and buffering module. The AA pre-authenticates the MN. The tunneling agent manages an IPSec tunnel from the MN as the PHT (Proactive Handover Tunnel) and performs layer 3 handover using MOBIKE [SE06]. This testbed differs slightly from the MPA framework described in Section 9.2.2 as the tunneling agent is implemented on a node outside of the target network (i.e., the EV-DO network), not on the AR in the target network because we do not have control of the equipment of the operators network. As a result, the MPA server acts as a proxy AR (Access Router) to the EV-DO network.

A MIH information server (IS) within the testbed is populated with the information of the neighboring network elements such as Wi-Fi access points and cellular network

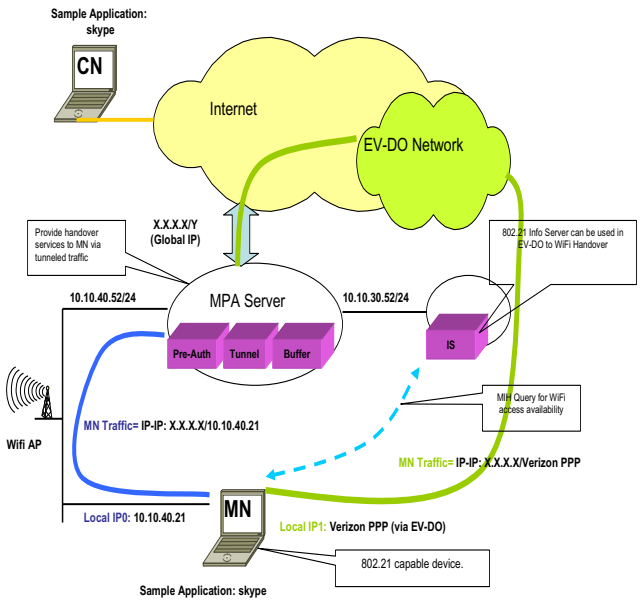


Figure 9.5: Experimental testbed MIHF assisted MPA

elements.

A correspondent node (CN) is connected to the Internet and communicates with the MN via a Skype [BS06] voice-over-IP session. In this mobility scenario, the mobile node engages in a VoIP session with the CN over the Wi-Fi network (path A) and then performs a handover to the EV-DO network (path B). While the MN is still connected to the Wi-Fi network, the MPA stack utilizes the MIH services to trigger an authentication and configuration process with the EV-DO network in anticipation of the mobile node's move. The MPA engine learns of the target network by querying the MIH Information Server for network information. The MPA stack that triggers the authentication can be either on the MN (for mobile initiated) or on the MPA server (for network initiated handover). Although the current implementation uses signal strength thresholds to trigger the IS (Information Server) query, other policies may also be implemented to trigger several steps in the handover process.

Since the tunnel agent does not reside inside the cellular operator network, all communication to and from the MN needs to go through the MPA server over the PHT (Proactive Handover Tunnel), even after L2 handover, as shown in path B.

The MPA agents utilize MIH services for the following purposes:

- Identify when to prepare for handover based on signal thresholds of the active interface. This is done by event subscription to Parameter Reports when the active interface's signal level in the MN crosses different thresholds.

- Identify candidate networks and their related parameters the mobile is likely to handover to and their related parameters by querying the Information Service. Using the MIH_Link_Actions, *PowerUp* MIH command to power up, connect and configure the EV-DO interface and set up a PHT once pre-authentication procedure is over.

- Using MIH command MIH_Link_Actions, *PowerDown* to turn off the old link once handover is complete.

Figure 9.6 (a) shows the sequence of operations for a mobile-initiated handover from the Wi-Fi network to the EV-DO network. Figure 9.6(b) shows the communication flow for network initiated handover from Wi-Fi to EV-DO network. I describe the details of the flows for both mobile-initiated and network-initiated handovers.

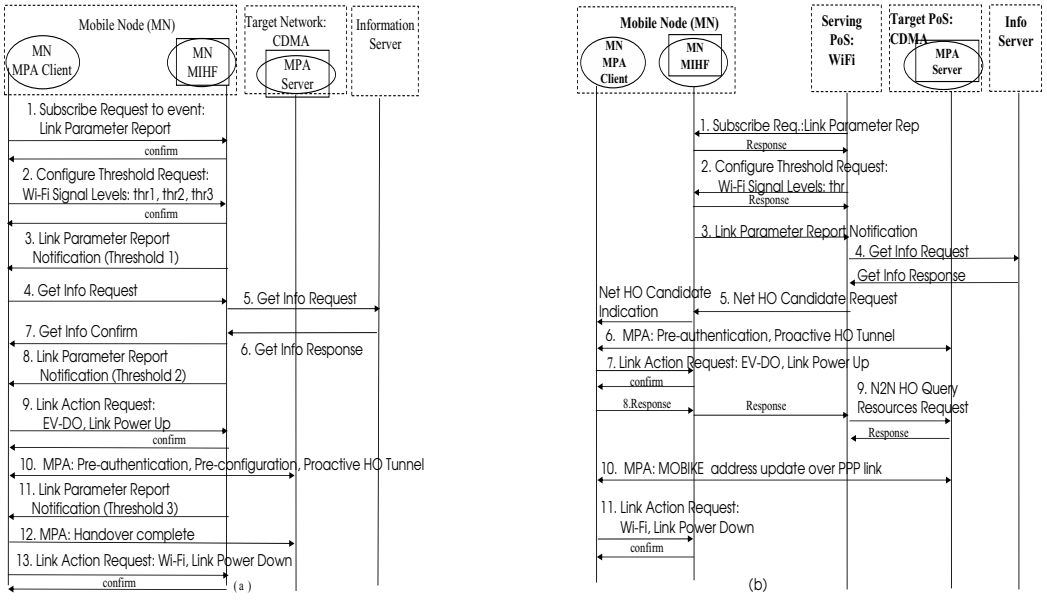


Figure 9.6: (a) Mobile initiated (b) Network initiated handover

9.3.5 Mobile initiated handover with 802.21 triggers

Figure 9.6 (a) shows a sequence diagram for a mobile-initiated handover from Wi-Fi network to the EV-DO network. The MN is initially connected to Wi-Fi network and then hands over to EV-DO network. I describe the following steps in sequence.

(1) Subscribe Request: The MPA client first subscribes to the MIH_Link_Param_Report event, which provides link parameter reports when the Wi-Fi signal strength crosses certain values.

(2) Configure Threshold Request: The MPA client uses an MIH_Link_Configure_Threshold command to establish a set of three Wi-Fi signal strength levels that will trigger notifications. Once a threshold level is crossed, the MIHF will propagate the appropriate notification to the MPA client.

(3) Link Parameter Report (Threshold 1): When the MPA client receives the first event notification reporting that the Wi-Fi signal strength has crossed the first threshold, the MPA client prepares for a potential handover, queries the MIH information server (Steps 4 to 5) for available neighboring networks via the MN's current serving network. The information server then sends a response with the information that the cellular network is available (Steps 6 to 7). Steps 4-7 are not explained for brevity. **(8) Link Parameter Report (Threshold 2):**

When the signal strength weakens further and the mobile crosses the second threshold, the MPA client receives an event notification and starts setting up the cellular connection.

(9) Link Up Request: The MPA client brings the EV-DO interface up and establishes an EV-DO connection using an MIH_Link_Actions command. It is important to note that this step can be performed after Step 10 if the IP address to be assigned to the EV-DO interface can be obtained in Step 10, however, this optimization will require the EV-DO network to support MPA.

(10) MPA Proactive Handover: The MPA client starts pre-authentication and pre-configuration through the serving Wi-Fi interface.

(11) Link Parameter Report: When the MPA client receives the third Link Parameter Report event notification, indicating crossing the third threshold value, the MPA client completes the handover operation via MOBIKE address update.

(13) Link Power Down Request: The MPA client then uses an MIH_Link_ Actions command to bring down the Wi-Fi interface.

9.3.6 Network initiated handover with 802.21 triggers

Figure 9.6 (b) shows a sequence diagram for a network-initiated handover from the Wi-Fi network to the EV-DO network. In addition to the entities depicted in Figure 9.6 (a), a new entity called the serving PoS (Point of Service) in the Wi-Fi network is used to realize a network-initiated handover.

(1) Subscribe Request: The serving PoS subscribes with the MN to get an MIH_Link_Param_Report event notification, which will provide link parameter reports, when the Wi-Fi signal strength crosses a given value.

(2) Configure Threshold Request: The serving PoS uses an MIH_Link_Configure_Threshold command to configure the Wi-Fi signal strength level that will trigger a layer 2 event notifications. Once a threshold level is crossed, the MIHF in the mobile node will propagate the appropriate notification to the PoS using the MIH Protocol to provide remote Event Services.

(3) Link Parameter Report: When the serving PoS receives the event notification reporting that the Wi-Fi signal strength has crossed the specified threshold, the serving PoS queries the MIH information server (Step 4) for available neighboring networks. The information server then reports that the cellular network is available.

(5) Net HO candidate Request: The serving PoS sends an MIH_Net_HO_Candidate_ Query request message to the mobile indicating the candidate networks available for handover. The candidate networks are selected based on the information obtained from the Information Server in Step 4.

(6) MPA Pre-authentication: Once the target PoS is selected and authentication server is known, the mobile node contacts the MPA server and starts pre-authentication, and sets up the proactive tunnel through the serving Wi-Fi PoS.

(7) Link Up Request: The MPA client verifies the availability of the cellular network as indicated in the MIH_Net_HO_Candidate request message by bringing the EV-DO interface up and establishes an EV-DO connection using an MIH_Link_Actions command.

(8) Net HO candidate Response: Once the EV-DO connection is established, the MPA client responds with an MIH_Net_HO_Candidate_Query response message, indicating the EV-DO network as the candidate network.

(9) Network-to-Network HO Query Resource Request/Response: The serving PoS (Wi-Fi) sends the target PoS (CDMA) a N2N(Network to Network)_HO_Query_Resource request message, to verify that the target PoS has resources before committing the handover. Once the serving PoS gets a positive response, it can commit to the handover. While MIH provides a command to indicate handover commitment (i.e., MIH_Net_HO_Commit), I use the MPA proactive handover (Step 9) as the indication of the handover commitment.

(10) MPA Proactive Handover: The MPA client completes the handover operation by MOBIKE address update.

(11) Link Power Down Request: The MPA client then uses an MIH_Link_Actions command to bring down the Wi-Fi interface.

It is important that MIH handover preparation and MPA pre-authentication procedures complete before the mobile makes a handover to the target network. In the next section, I describe the timing for different handover operations.

9.3.7 Handover preparation time

The handover preparation time does not directly affect the handover performance and user experience. However, the amount of time the mobile needs to prepare for handover depends upon the speed of mobile (e.g., pedestrian, vehicular), cell size (e.g., pico cell, macro cell)

and type of handover (e.g., single interface, multiple interface). Generally, it is important to reduce the handover preparation time to make the system more resilient to sudden changes in the network characteristics.

This handover preparation time in the experimental testbed includes the following components:

(i) Propagation of the Link events from the link layer to the MIH user (i.e., local MIH user, in case of MN initiated handover and remote MIH user in Network-initiated handover)

(ii) Querying the IS database

(iii) MIHF internal operations

(iv) MPA layer 3 handover

During the experiment, I have measured the time delays for execution of the operations (ii), (iii) and (iv). While the delays I have measured are for different MIH related operations in the network initiated handover scenario described in Figure 9.6, some of these measurements can be applied to mobile initiated handoff scenario as well.

Tables 9.2, 9.3, 9.4 and 9.5 show the values measured for each of the above operations. I describe the delays associated with each of these operations below.

9.3.7.1 Information service transaction delay:

I measured different operations in the information server that constitute the transactions associated with a request. This sequence starts with receiving a *Get Information* request message containing an IS query and finishes by sending the corresponding response. Table 9.2 shows five values measured for each operation and their average. The average information server transaction execution time is 26.6 ms with lower bound of 13 ms and upper bound of 53 ms.

Table 9.2: Processing time in the Information Server

Measurement #	1	2	3	4	5	Average
Get Info request parsing (ms)	3	3	4	4	5	3.8
Pass indication from MIHF to MIH user (ms)	2	10	2	3	2	3.8
Query processing (ms)	5	29	5	25	6	14
Get Info response composition (ms)	3	2	4	3	2	2.8
Get Info response sending (ms)	2	1	1	5	2	2.2
Total time (ms) processing in the Info server						26.6

9.3.7.2 MIH message composition and parsing delay:

Depending on the MIH message type, the time for message composition and parsing might vary. This depends on the number of TLVs included in each message and the TLV type, which dictates the complexity of its composition and parsing. Tables 9.3 and 9.4 show the minimum, maximum and average values for the time taken for different sub-operations associated with message composition and parsing delay, respectively. These values are taken into account for calculating the handover preparation time.

Table 9.3: MIH message composition time

Measurement Point	Message Type	Execution Time (ms) (average, min, max)
MN	Link Parameter Report Indication	1.6, 0, 2
Serving PoS	Register Response	4.4, 3, 8
Serving PoS	Subscribe Request	4.8, 3, 11
Serving PoS	Get Info Request	6.2, 5, 2
Serving PoS	Net HO Candidate Request	25.4, 10, 51
Info Server	Get Info Response	2.8, 2, 3

Table 9.4: MIH message parsing time

Measurement Point	Message Type	Execution Time (ms) (average, min, max)
MN	NET HO Candidate Query Request	12.6, 6, 19
Serving PoS	Subscribe Response	12, 7, 17
Serving PoS	Configure Threshold Response	40.2, 10, 54
Serving PoS	Link Parameter report Indication	21.2, 14, 50
Serving PoS	Get Info Response	11.4, 8, 17
Info Server	Get Info Request	3.8, 3, 5

9.3.7.3 MIH performance for MPA triggering

I measured the time it took to perform all the MIH related operations in the network initiated handover scenario that occurred starting with the initial handover trigger (i.e., crossing signal strength threshold in the MN and creation of the Link Parameter Report Indication) until triggering the MPA handover operation. Table 9.4 shows the average execution time of five measurements for each of the specified operations with the corresponding lower and upper bounds. Table 9.5 shows the timing associated with each of the MIH related operations.

In order to calculate the total MIH MPA triggering operations, the following network propagation delays need to be added:

1. MN Serving PoS round trip propagation delay (MN-PoS-RTT).

2. Serving PoS Information Server round trip propagation delay (PoS-IS-RTT).

In the current experimental testbed, I estimate these delays using ping messages that give round trip value for ICMP messages which are 1.5 ms for MN-PoS-RTT and 0.3 ms for PoS-IS-RTT, bringing the MIH time to trigger MPA to 148.4 ms in the testbed

environment.

These round trip propagation delays can be adjusted for a real network environment to estimate a realistic network performance. Since the MN and its serving PoS are relatively close to each other, I estimate their round trip propagation delay, MN-PoS-RTT as 5 ms. I estimate the serving PoS-Information Server round trip propagation delay, PoS-IS-RTT as 30 ms as the PoS and IS (Information Server) usually located in the network core would be separated by few hops. Thus, in a realistic network, the time it would take for MIH to trigger the MPA pre-authentication and handover would be approximately = 146.6 ms + 5 ms (MN-PoS-RTT) + 30 ms (PoS-IS-RRT) = 181.6 ms. This time does not include the propagation of the link event from the link layer to the MIHF that I have not measured.

Table 9.5: Delays for MIHF related components

Measurement point	Operation description	Execution time		
		Min (ms)	Average (ms)	Max (ms)
MN	Compose/transmit Link Parameter Report Ind.	10	10.4	11
Serving PoS	Recv/parse/process Link Parameter Report Ind.	20	28.8	53
Serving PoS	Compose and transmit Get Info Request	11	14.4	22
Info Server	Receive/parse/process Get Info Request	10	21.6	44
Info Server	Compose/Send Get Info Resp.	3	5	9
Serving PoS	Receive/parse/process Get Info Response	10	20	28
Serving PoS	Compose/send Net HO candidate Req.	11	31.2	56
MN	Receive/process Net HO candidate Req.	8	15.2	22
	Total		146.6	

9.3.7.4 Delays due to MPA operation

MPA related delays are attributed to several factors such as delays due to pre-authentication, setting up proactive handover tunnels and sending the binding update for data redirection. In the current testbed, I have measured delays for these components. As shown in Figure 9.6, pre-authentication and proactive tunnel setup took place before the PPP link was setup. Alternatively, these two operations could take place in parallel with PPP configuration operations that may take up to 5 seconds. Measurement shows that complete pre-authentication operation took about 2,175 ms. This time delay consists of several factors, such as four round trip signaling associated with EAP-GPSK (Extensible Authentication Protocol - Generalized Pre Shared Key), generation of keys at the authentication server and message processing delays at the end hosts. Proactive handover tunnel setup time was measured to be 4,730 ms that includes the time for IKE handshake to set up IPSec tunnel in ESP (Encapsulating Security Payload) mode, and initial MOBIKE signaling exchange. These two operations take place over the Wi-Fi interface in the previous network. The final step in the MPA operation is the binding update and it is performed using the MOBIKE address update. It took around 400 ms to complete the round trip MOBIKE signaling over a PPP link.

An estimation of the MIH handover preparation time before triggering MPA operation in a realistic network is less than 200 ms, which is less than 10 percent of the time MPA pre-authentication procedures would take. This seems to be a satisfactory time to allow proper timing of the MPA operation and handover procedure.

Information server transaction delay and MIHF performance can be improved by improving query execution time, message composition and message parsing time.

9.4 Handoff optimization in IP multimedia subsystem

In this section, I describe how some of the proactive optimization techniques that I have designed can improve the handoff performance in IMS (IP Multimedia Subsystem) by reducing the packet loss and handover delay. I have built a complete experimental prototype IMS system that I briefly introduced in Chapter 5 to illustrate the route optimization technique. Here I demonstrate how the handoff delay is reduced when security association is performed pro-actively by transferring the security context between points of attachment on two neighboring networks. This specific system does not optimize layer 2 and layer 3 related operations but focuses only on optimization to the security association. Soon after this work was published [DMD+07] 3GPP created a new working group called multimedia session continuity (MMSC) and produced a technical specification [3GP08] to define different scenarios to support fast-handoff for multimedia traffic in an IMS environment. Figure 5.42 shows the experimental testbed where I have experimented with these optimization techniques implemented.

9.4.1 Non-optimized handoff mode

In a non-optimized mode of operation, a new call context is created at the proxy server every time the mobile moves to the new network. A call context consists of the call data records and the media parameters associated with a specific call. During the handoff MN completes all the handoff functions at layer 2 and layer 3 as described earlier in Chapter 3. Specifically, after the MN establishes PPP access to the new network, it performs the MIP binding and then it obtains the server configuration information via DHCP. Then the SIP related handoff functions are performed starting with the SIP re-registration and re-establishment of security association using AKA procedure. If the MN moves during an active session, session maintenance is carried out with the transmission of an encrypted SIP re-INVITE message that carries the SDP description of the ongoing session. Upon

receipt of this message, the P-CSCF (Proxy Call Session Control Function) creates a new call context for the same mobile and interacts with the PDSN (Packet Data Serving Node) on the visited network to allow the traffic. It is important to point out that mobility binding is taken care of by mobile IP here and SIP re-INVITE is used for creating the call context at the P-CSCF. Call context consists of call data record (CDR) and other bandwidth and port related information of the media. Successful creation of call context at the P-CSCF results in the resumption of the media in the new access network. The message flow for the non-optimized operational mode is provided in shown Figure 9.7.

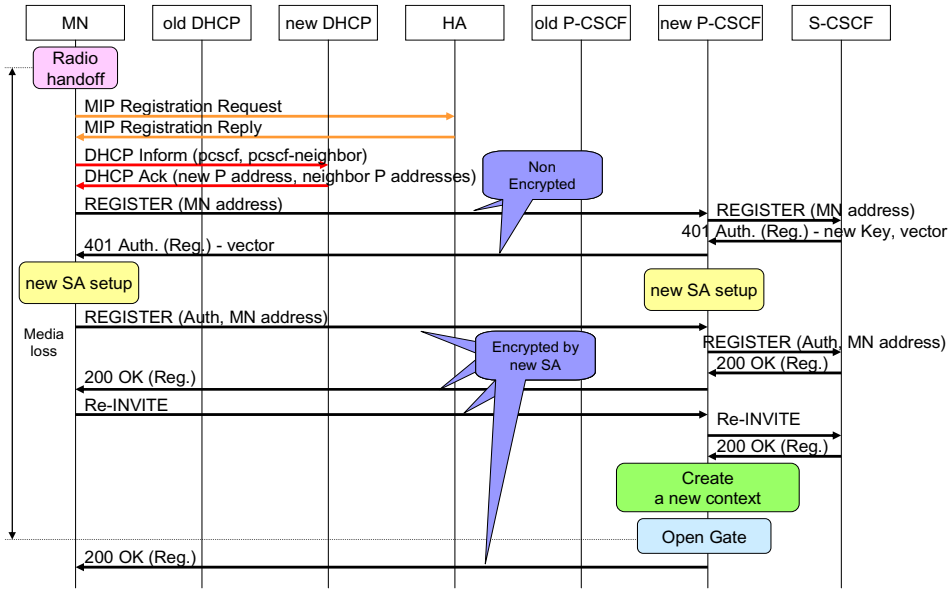


Figure 9.7: Call flow for non-optimized handoff for IMS

9.4.2 Optimization with reactive context transfer

In the reactive mode of operation, all the layer 2 and layer 3 related operations take place like non-optimized mode. The detailed message flow is provided in Figure 9.8. By comparing with Figure 9.7 (non-optimized mode) the difference between the two handoff operational modes are evident.

In particular, the session maintenance information message, such as re-INVITE that carries the SDP description of the active session does not play any role in context creation and thus does not affect the media handoff delay. The context created in the new visited network's P-CSCF is transferred from the old visited network's P-CSCF. The objective of this approach is to reduce the handoff delay by eliminating the dependence on the session maintenance messages (re-INVITE and 200 OK) after the handoff. Session maintenance is carried out to create context at P-CSCF i.e., out-bound SIP proxy.

After the radio handoff is over and PPP (Point-to-Point) connection is complete in the new network, the MN performs the MIP binding and obtains the required configuration information using DHCP. The MN then generates a SIP REGISTER message via the new P-CSCF. When this message reaches the S-CSCF (Serving Call Session Control Function), the S-CSCF informs the old P-CSCF to transfer the context of the active session to the new P-CSCF. At this point the old P-CSCF transfers the context to the new P-CSCF, and the context is created at the new P-CSCF in the new network. After the security association is set up via AKA (Authentication and Key Management) between MN and new P-CSCF, media traffic is allowed at PDSN, and the session resumes.

9.4.3 Optimization with proactive security context transfer

The proactive mode of handoff minimizes the delay due to security association and context creation more than the reactive mode and thus, reduces the media interruption further compared to the reactive mode. In this mode, the context creation in the new IMS network and security association with the new P-CSCF are completed while the MN is still in the old network. This technique works in conjunction with the proactive discovery of neighboring P-CSCFs while the mobile is in the previous network. As discussed in Chapter 5, the carriers could use information service discovery methods like IEEE 802.21 or IEEE 802.11u to obtain the detailed information about the neighboring networks and servers.

Figure 9.9 provides the detailed message flows of the proactive handoff. Prior to the

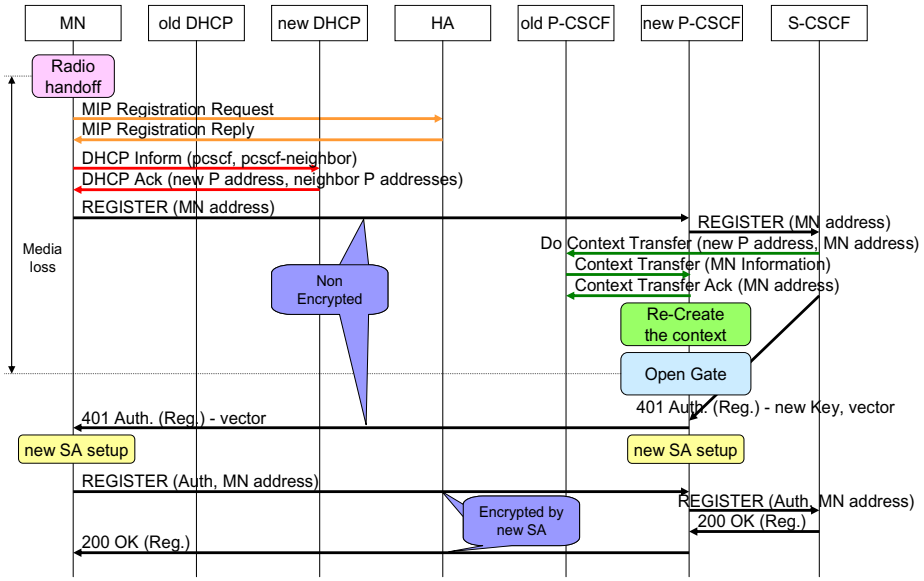


Figure 9.8: Optimized handoff with reactive context transfer

MN's layer 2 handoff, some of the handoff functions (e.g., server discovery, establishment of security association) are done pro-actively while the mobile is still in the old network. Specifically, the MN, utilizing the DHCP INFORM message, acquires the addresses of P-CSCFs from the neighboring IMS networks.

In this case, DHCP server is populated with the information about the SIP servers in the neighboring networks. After the MN has identified the new neighboring network it is likely to move, it informs its current P-CSCF about the address of its new P-CSCF. The current P-CSCF transfers the context of the active session (e.g., SDP parameters and CDR (call data recrod) parameters) to the new P-CSCF. Similarly, a new security association is established between the new P-CSCF and the mobile after the mobile sends a Move_Notify message to S-CSCF when the movement is imminent. This mechanism performs a proactive AKA (Authentication and Key Agreement) operation by transferring the security context from the current P-CSCF to the new P-CSCF ahead of time by creating a transient AKA during the handoff. The mobile performs the normal AKA procedure after it moves to the new network. Thus, the new PDSN allows the specific mobile's media even before the

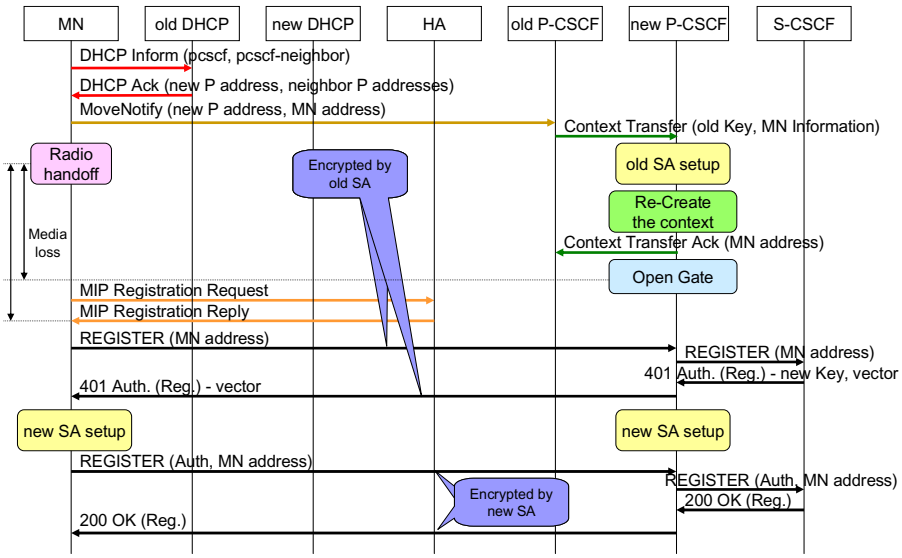


Figure 9.9: Optimized handoff with proactive context transfer

mobile has moved. After the mobile re-establishes its connection in the new network, and completes the MIP operation, media continues, eliminating the delay due to the AKA procedure and context creation. Mobile node's SIP-related signaling such as re-REGISTER or re-INVITE do not affect the media handoff delay here. Re-registration in the new network helps to renew the transient AKA that was established in the previous network.

Based on the message flows in Figure 9.8 and Figure 9.9, it is obvious that the proactive handoff techniques reduce the handoff delay that is typically caused due to re-establishment of security association. I compare the results of each handoff operation in the following section.

9.4.4 Performance results

The focus of this performance analysis is to highlight the relative effectiveness of proactive security association compared to other two handoff techniques, namely non-optimized and reactive mode. In Figure 9.10, I plot the delays associated with the different handoff related functions that contribute to the handoff delay for these three different handoff scenarios.

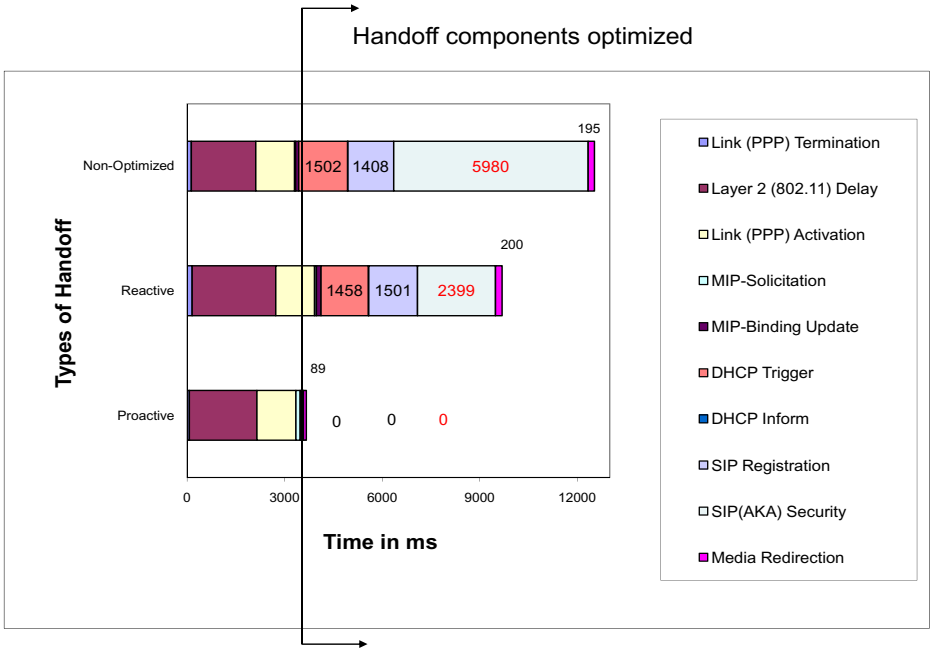


Figure 9.10: Comparison of optimized handoff components

On average, the mobile was subjected to 3,666 ms delay during proactive mechanism, 9,685 ms delay during reactive security association and 12,526 ms delay for non-optimized handoff. The number of packets lost is proportional to the handoff delay and depends on the packet generation rate.

Overall handoff delay consists of delays due to different operations such as layer 2 configuration, layer 3 configuration, binding update, registration, security association and media redirection. As is evident, proactive handoff does not contribute to any delay due to DHCP-based server discovery, context transfer and SIP-based security association compared to reactive or non-optimized case. On the other hand, the non-optimized case suffers from maximum delay due to additional signaling messages during SIP-based security association and context creation phase. Layer 2 handoff delay, PPPoE (PPPover Ethernet) access delay and MIP binding delay remain the same for all three handoff scenarios. In case of reactive handoff, besides layer 2 delay, the major component of the delays came from SIP registration, security association and context transfer process.

Since I have used Mobile IPv4 for mobility, these results are inclusive of inherent tri-

angular routing delays and can further be reduced when the routing mitigation techniques described in Chapter 5 are applied. Mobility protocols such as MIPv6 or SIP-based mobility may result in smaller binding update delay because they avoid triangular routing. SIP related signaling required for context transfer and security association contribute to additional handoff time for non-optimized case compared to reactive case, since it needs to create the context with an additional re-INVITE signaling.

In order to gain an insight into the effect of these optimization techniques in a real deployment scenario, I have used NIST delay simulator [CS03] and varied the emulated distance between the home network and the visited network by varying the delays from 0 ms through 500 ms, with an increment of 50 ms. I provide additional handoff results in Table 9.6.

Table 9.6: Effect of emulated distance on handoff components

One way delay	Proactive Handoff			Reactive Handoff			Non-optimized Handoff		
	AKA, Context Transfer Delay (ms)	MIP Update Delay (ms)	L2 PPP Delay (ms)	AKA, Context Transfer Delay (ms)	MIP Update Delay (ms)	L2 PPP Delay (ms)	AKA, Context Transfer Delay (ms)	MIP Update Delay (ms)	L2 PPP Delay (ms)
0	0	51	2736	1010	62	1523	3999	41	2239
50	0	152	2693	1375	161	1744	4584	145	2217
100	0	252	2650	1741	261	1964	5170	248	2194
150	0	352	2607	2107	360	2184	5756	352	2172
200	0	453	2563	2472	459	2405	6342	455	2150
250	0	553	2520	2838	558	2625	6927	559	2128
300	0	654	2477	3203	658	2845	7513	663	2106
350	0	755	2434	3569	757	3066	8099	766	2084
400	0	855	2391	3935	856	3286	8685	870	2061
450	0	956	2347	4300	955	3506	9270	973	2039
500	0	1057	2304	4666	1055	3726	9856	1077	2017

These results show only the components of the handoff delay that are affected when additional delay is introduced between the home network and the visited network. From the analysis it appears that delays related to layer 2 and layer 3 configuration do not get affected due to additional emulated delay because these operations do not involve the home network. Delays due to the mobile IP binding update increases for all the three cases as the emulated distance between the home network and visited network is increased, but there is an appreciable increase in delay for SIP-based security association in case of non-

optimized and reactive handoff. In proactive handoff case, the additional network transport delay did not have any effect on the handoff delay component related to SIP, AKA and context transfer. The additional handoff delay in proactive case was contributed by the increased MIP update delay only.

9.5 Systems validation using Petri net-based models

In this section, I introduce Petri net-based models for some of the optimization techniques that I have described in Chapter 5. I apply a MATLAB-based Petri net tool [MMP03] to model some of the handoff functions as described in Chapter 3 and validate the optimization techniques by way of Petri net-based behavioral analysis methods and evaluate the systems performance by using cycle time and Floyd algorithm. I then evaluate three different scheduling techniques for few of the handoff operations and apply *cycle time* and *Floyd algorithms* approaches to validate the systems performance. I also illustrate how certain sequence of transitions may give rise to deadlocks by doing a reachability and matrix analysis.

9.5.1 MATLAB-based modeling for handoff functions

In this section, I describe the results from MATLAB-based modeling of many of the handoff functions. This MATLAB-based Petri net tool is used to study the behavioral properties such as reachability analysis, markings, liveness and systems performance of the handoff models.

Figure 9.11 shows the MATLAB-based model to illustrate the sequence of a few of the handoff operations, namely discovery, attachment, configuration, and authentication. This figure is MATLAB equivalent of the model shown in Figure 4.14. Places P13, P14 and P15 in Figure 9.11 represent resource places representing energy, bandwidth and CPU cycles, respectively. This figure also shows the markings associated with these sets of models as

generated from the MATLAB model. These markings represent a step-wise execution of the handoff events.

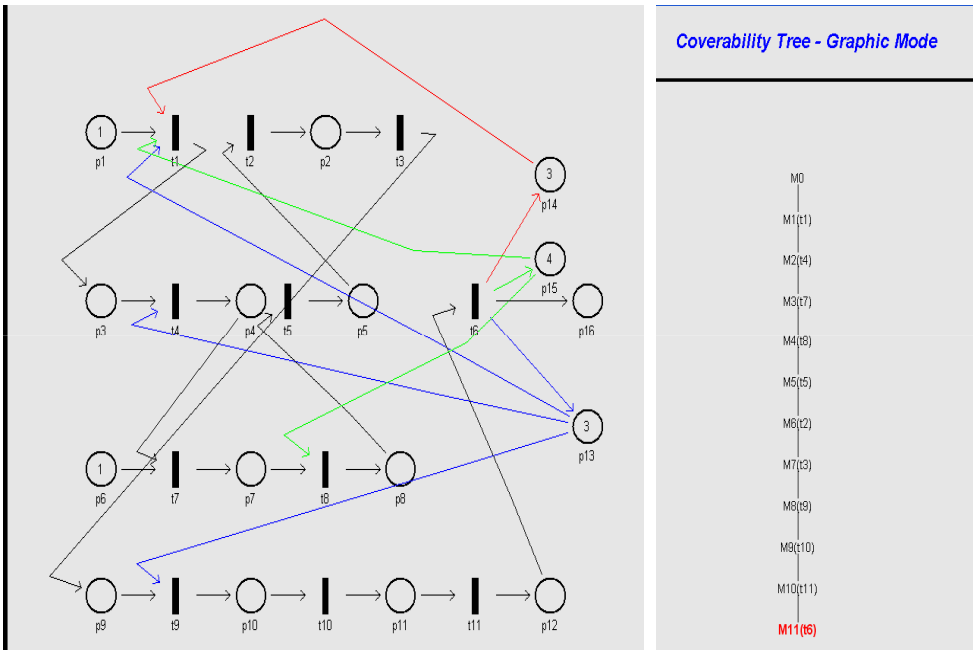
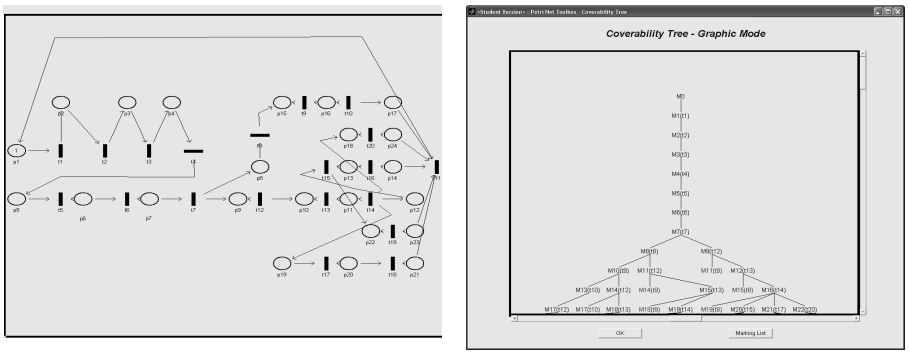


Figure 9.11: MATLAB-based model of four handoff functions

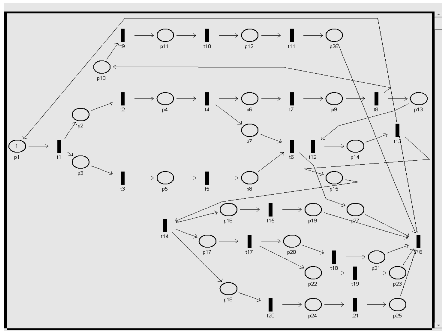
Figures 9.12, 9.13, 9.14 illustrate MATLAB-based models for three different handoff sequences as shown in Figures 4.16, 4.19 and 4.20, respectively. These models illustrate coverability trees and markings as obtained from the MATLAB-based Petri net Tool. Figure 9.12 shows the MATLAB-based modeling, reachability tree and corresponding marking of the sequential handoff operations. Figure 9.13 shows a handoff model and parameters illustrating concurrent security and scanning operations. Figure 9.14 shows the MATLAB-based model for handoff event where security, L2 discovery and L3 discovery operations are performed in parallel. Matrices A_i, A_o and A represent the input, output and incidence matrix, respectively for the petri net handoff model shown in Figure 9.12. These matrices are obtained from the MATLAB-based petri net model. Figure 9.15 shows a MATLAB-based screenshot of these matrices. As discussed in Chapter 4, many of the behavioral properties of the handoff functions can be obtained from these matrices using matrix analysis method. I discuss some of these behavioral properties in Section 9.6.

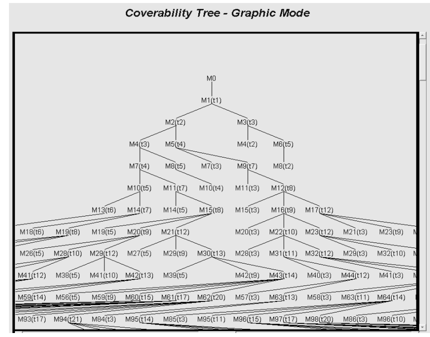


M130 = [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,1,1]
M131 = [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,1,0,1,1]
M132 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,0,1,1]
M133 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,1,0,0,1,0,1,0]
M134 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,1,1,0,1]
M135 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,1,0,0,1,1]
M136 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,1,1]
M137 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,1,0,1,1]
M138 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,0,1,1]

Figure 9.12: Sequential handoff operations

$$A_i = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}$$

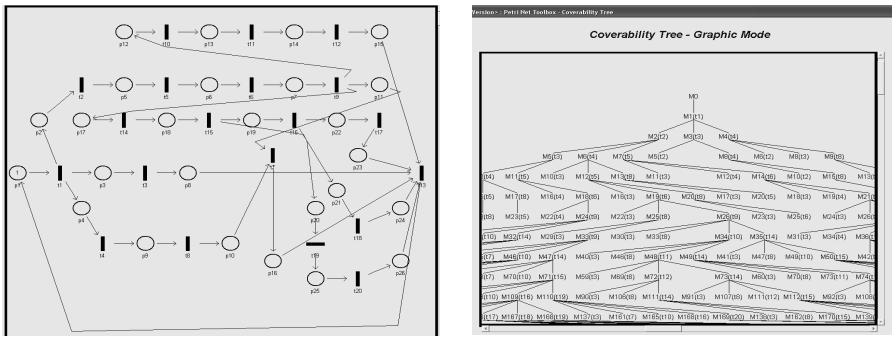




M529 = [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,1,0,1,0,1,0,1]
M530 = [0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,0]
M531 = [0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,1,1,0]
M532 = [0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,1,0,1,1,0]
M533 = [0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,1,0,1,1,0]
M534 = [0,0,0,0,0,0,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,1,0,1,0,1,0,1,0,0]
M535 = [0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,0,1,1,0]
M536 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,1]
M537 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,1,1,1]
M538 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,1,0,1,1,1]
M539 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,1,0,1,1,1]
M540 = [0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,1,0,1,0,1,0,1,0,1]
M541 = [0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,0,1,1,0]
M542 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,0,1,1,1]

Figure 9.13: Concurrent security and scanning operations

$$A_o = \begin{bmatrix}
0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0
\end{bmatrix}$$



M559 = [0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1,0,1,0,1]
M560 = [0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,1,0,1]
M561 = [0,0,1,0,0,0,0,0,0,1,1,0,0,0,1,0,0,0,0,0,0,0,1,1,0,1]
M562 = [0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,0]
M563 = [0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,1,0,1,0,0,1]
M564 = [0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,1,0,1,0,1]
M565 = [0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,1,0,1]
M566 = [0,0,0,0,0,0,0,1,0,1,1,0,0,0,1,0,0,0,0,0,0,0,1,1,0,1]
M567 = [0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,1]
M568 = [0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,1]

Figure 9.14: Concurrent security, L2 discovery and L3 discovery operations

$$A = \begin{bmatrix}
-1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1
\end{bmatrix}$$

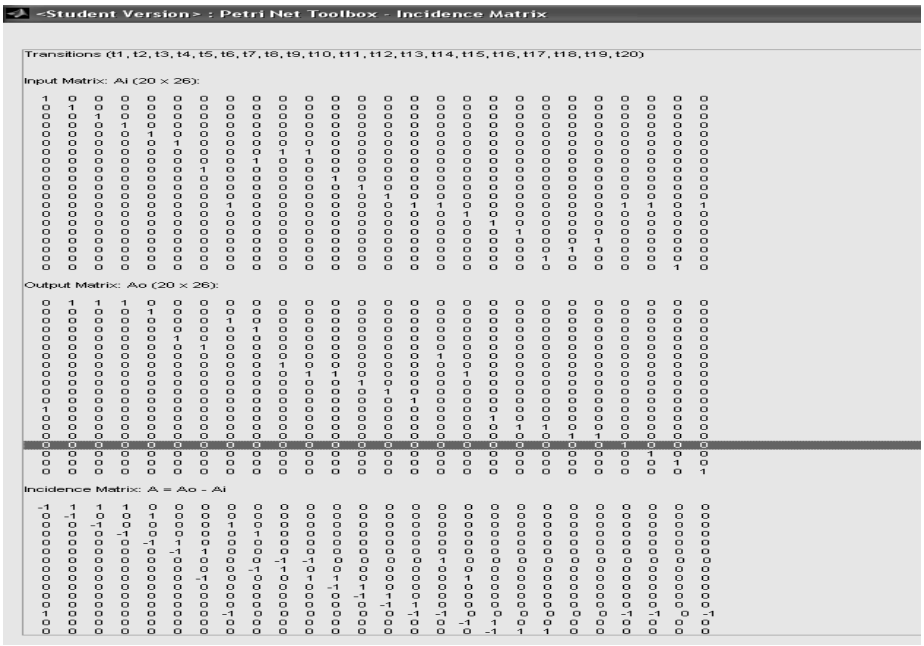


Figure 9.15: Input, output and incidence matrix

9.5.2 Petri net-based model for optimized security association

As discussed in Section 5.4, addition of an external home agent (i.e., external to the user's enterprise network) reduces the number of signaling messages exchanged between the mobile and the VPN gateway when the mobile changes its IP address during handoff. Thus, while the mobile does not need to re-establish a new security association, it still needs to set up an additional mobile IP tunnel with the external home agent. Thus, there is a trade-off between additional resources needed due to additional external home agent, additional tunnels and avoidance of extra signaling due to re-establishment of security association.

Figure 9.16 shows the protocol flows associated with both the cases, one when mobile needs to re-establish the security association and the other when the mobile does not need to re-establish the security association at the cost of additional external home agent. The network element i-HA shown in Figure 9.16(a) is the internal home agent and resides within the mobile's enterprise network and the network element x-HA shown in Figure 9.16(b) is the external home agent. I derive the Petri net models for both of the systems, one with

external home agent and one without the external home agent and evaluate the systems performance of each of the systems based on the experimental results.

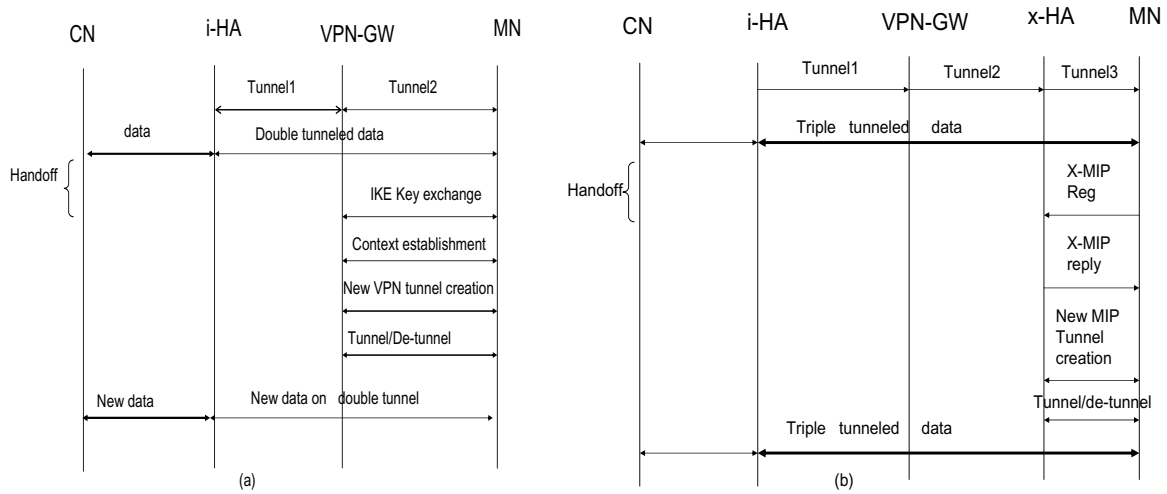


Figure 9.16: Security association with and without external home agent

Figure 9.17 shows the Petri net models corresponding to the call flows for two different scenarios shown in Figure 9.16. In Chapter 5.4, I have demonstrated the experimental results and have highlighted how delay and packet loss are reduced when an external home agent is introduced as an anchor agent to help maintain the security association even when the IP address changes. Experimental results comparing the optimized approach with non-optimized version are described in Section 5.4. Table 9.7 shows the results of the handoff operations from the experimental setup. In particular, timings for IKE signaling exchange, security context establishment, tunnel creation, tunneling and de-tunneling operations for MIP and IPSec, binding update time by the external home agent are applied to the Petri net model to determine the systems performance by way of cycle time and Floyd algorithm.

In order to evaluate the optimization of a Petri net system, any of the three Petri net-based methods described in Chapter 4 can be applied. I verify the above optimized system by using a matrix-based solution as described in Chapter 4.9. These results demonstrate that in order to achieve the desired performance, the system needs to utilize more resources,

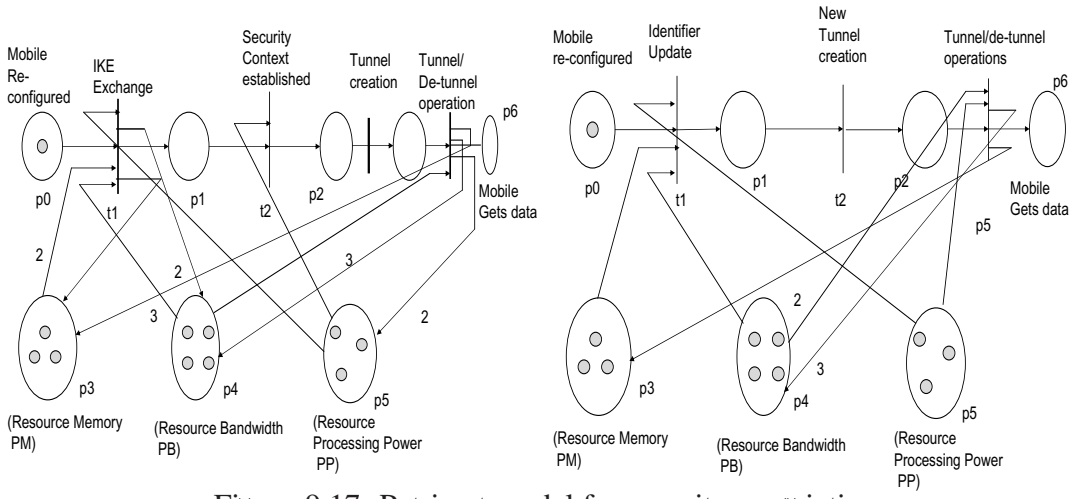


Figure 9.17: Petri net model for security association

Table 9.7: Timings with security association during handoff with xHA

Transition	Handoff operation	Time taken for operation (ms)
t1	IKE exchange	30
t2	Security context establishment	400
t3	VPN tunnel creation	6000
t4	MIP tunnel creation	10
t5	Tunneling/de-tunneling Mobile IP data	5
t6	Tunneling/de-tunneling VPN data	60
t7	External MIP update	300

such as an additional home agent that uses up more resources due to triple encapsulation.

9.5.3 Petri net-based model for hierarchical binding update

In Section 5.7.3, I have experimentally shown how hierarchical binding update reduces the delay contributed by the global binding update using network layer and application layer mobility protocols. In order to achieve that, I introduce an additional anchor point closer to the mobile that takes care of hierarchical binding update. Here, I introduce an equivalent Petri net model that demonstrates hierarchical binding update mechanism. Matrix-based

analysis methods can be developed to evaluate the systems performance by using cycle time-based approach and Floyd algorithm. Figure 9.18 shows the communication flow for inter-domain and intra-domain binding update and Figure 9.19 shows the respective Petri net model.

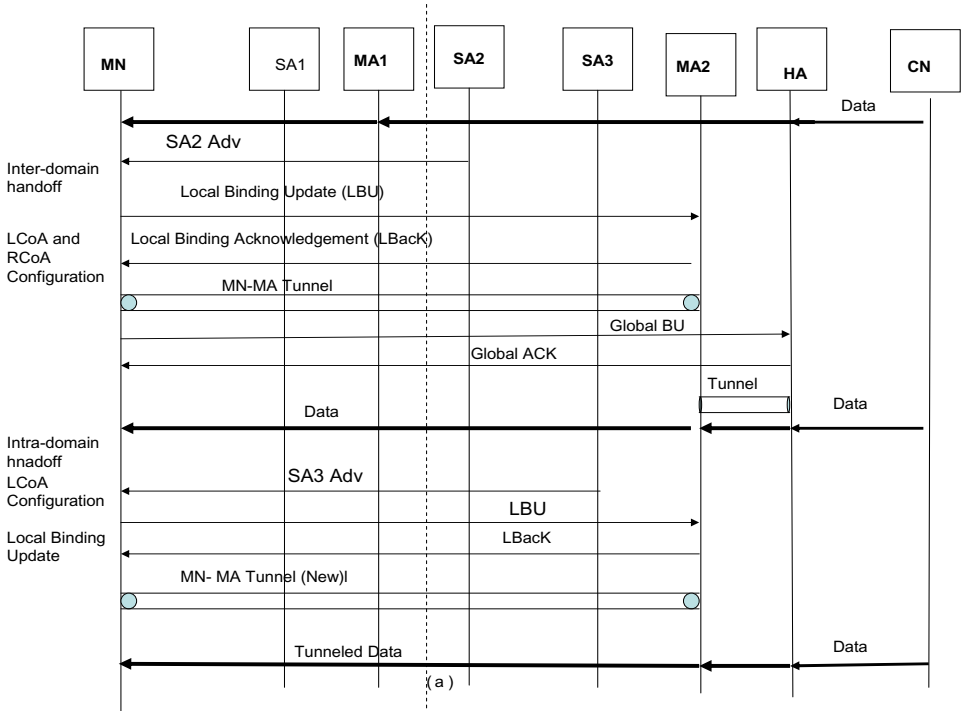


Figure 9.18: Hierarhical mobility management flow

9.5.4 Petri net-based model for redirection of inflight data

In Section 5.8, I have experimentally demonstrated how using different optimization techniques packet loss due to media redirection delay can be reduced. Figure 9.20 shows an equivalent Petri net model for one of these optimization techniques, namely the mobility proxy-based approach [HDS03]. This model also allows to verify the correctness of this optimization technique such as liveness and deadlock properties and evaluate the systems performance.

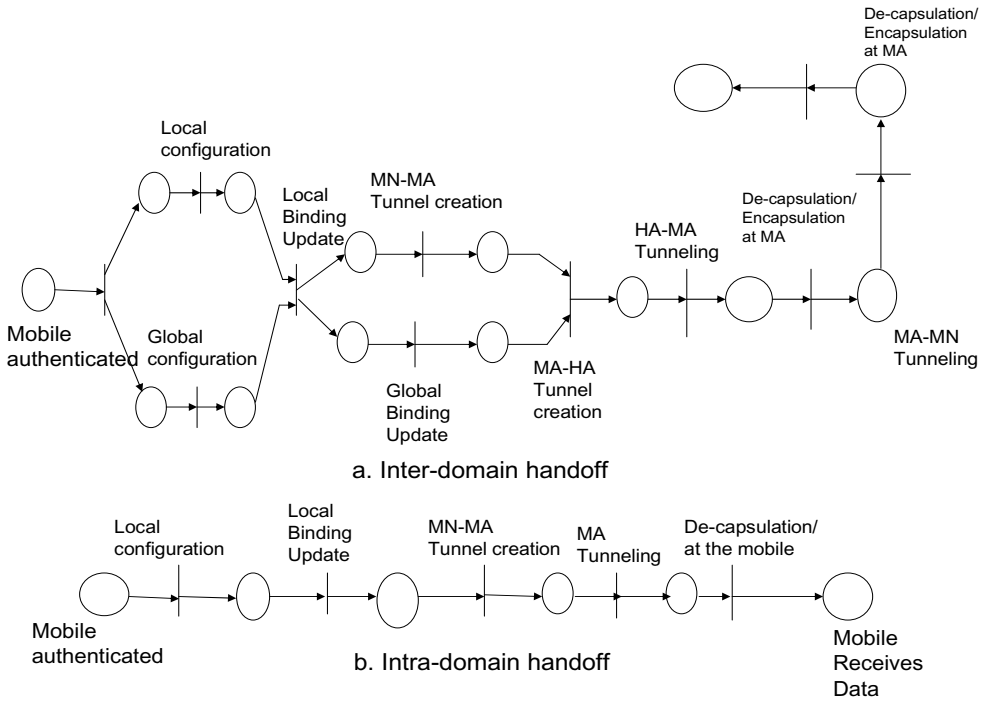


Figure 9.19: Petri net model for inter-domain and intra-domain

9.5.5 Petri net-based model of optimized configuration

As described in Chapter 5, the duplicate address detection (DAD) process takes the most time during layer 3 configuration. Chapter 5 also describes several ways to reduce the time taken by the DAD process. I use Petri net to verify one specific optimization mechanism [DMCS06] where duplicate address detection is performed during the layer 3 identifier acquisition phase. This specific mechanism eliminates the time taken by the neighbor discovery process that is often performed by the client after the address is obtained. However, this mechanism adds extra load to the server or router, as this intermediate network entity (e.g., server or router) needs to collect the information about the addresses that are being used (some of those addresses could be statically configured and some may have been configured using DHCP) and the network also requires additional amount of bandwidth due to the periodic multicast announcement by the server that carries the addresses that are already being used in the network. The specific model has the ability to verify that the specific optimization technique is deadlock free and if it can reduce the time for DAD process at the expense of additional resources. If there are not enough systems resources, namely

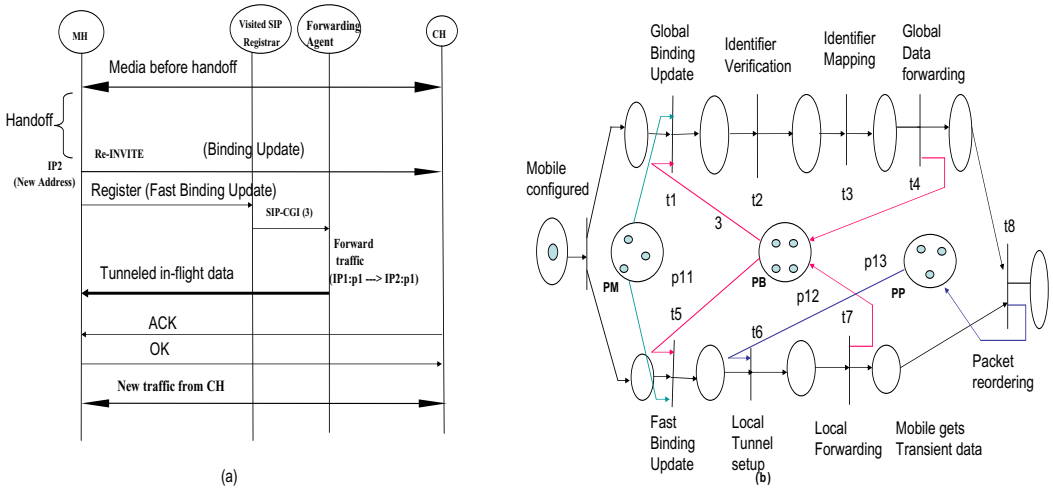


Figure 9.20: Petri net model for media forwarding

bandwidth, memory, CPU cycles, then there will be deadlock during some of these operations. Figure 9.21 (a) shows the protocol flow for optimized DAD mechanism and Figure 9.19 (b) shows the associated Petri net model.

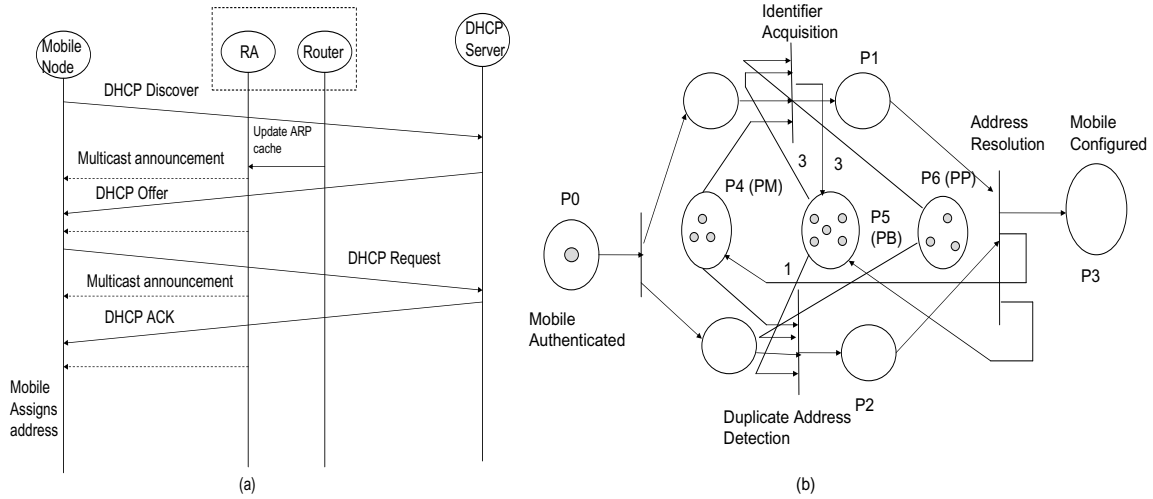


Figure 9.21: Petri net model for optimized DAD

Similarly, equivalent Petri net models can be derived for any of the optimization techniques discussed in Chapter 5. Then, these Petri net models can be used to derive the systems performance and behavioral properties of the optimized systems using any of the

methods described in Chapter 4.

9.5.6 Petri net-based model for multicast mobility

As discussed in Chapter 2 IGMP query interval and query response contribute to the handoff delay for multicast traffic. In Chapter 8, I have described several optimization techniques that help to reduce the handoff delay and packet loss for multicast streaming traffic. Here, I illustrate both non-optimized and optimized versions of multicast mobility using Petri net model. Figure 9.22(a) shows how the mobile's connectivity is delayed due to triggering delay resulting out of IGMP router query interval. On the other hand, 9.22(b) shows how an unsolicited IGMP join helps to reduce the join latency during handoff.

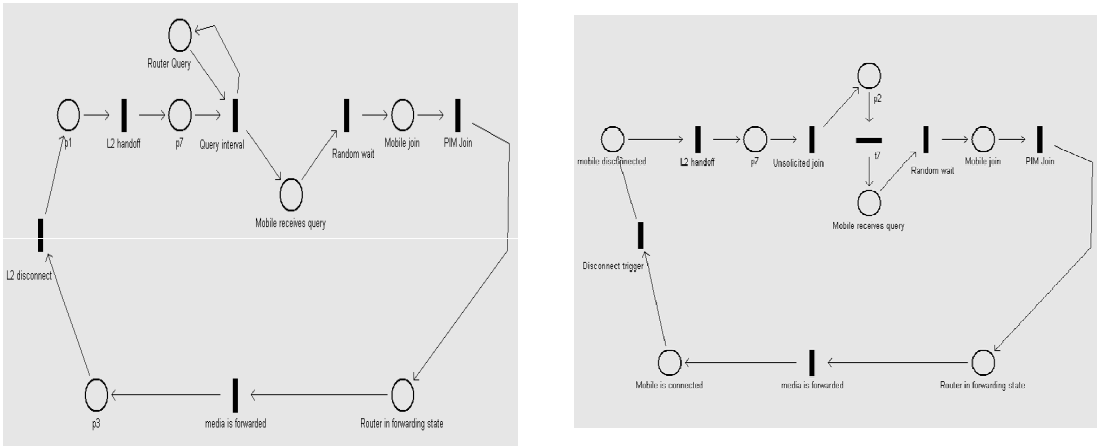


Figure 9.22: Petri net model for multicast mobility

9.6 Scheduling handoff operations

In this section, I illustrate different ways some of the primitive handoff operations described in Chapter 3 can be scheduled and evaluate the overall systems performance for three different schedules, namely sequential, concurrent, and proactive. A timed Petri net can be used to illustrate these scheduling for handoff operations. One basic approach is through

the use of heuristic search for the optimal or near optimal schedule in the reachability tree of the Petri net model. Scheduling techniques can help derive specific schedule to optimize certain performance index such as handoff delay under certain systems resource constraints. Resource constraints can be defined as the limitation on systems resources beyond which a handoff operation will fail due to lack of resources.

Scheduling of a handoff operation needs to take into account the following general guidelines.

1. Required systems performance such as handoff delay and packet loss are usually achieved under certain resource constraints, namely battery power, CPU cycles and available channel resources (effective user bandwidth). Thus, minimum cycle time achieved under a specific schedule is considered to be equivalent to maximum systems performance.

2. A specific handoff schedule should not suffer from any deadlock condition where a specific operation cannot proceed because of non-availability of data from the previous operation or because of lack of resources.

3. Both the resources and precedence relationship among the events need to be modeled to allow maximum flexibility. Thus, the Petri net model depends on both the resources required and the precedence graph. Van Bruseel et al. [VBPV93] illustrate how scheduling in FMS (Flexible Manufacturing Systems) can be affected by the resource constraints and precedence relationship among the events. Data dependency among the handoff events and resource availability in the system during handoff operation can determine the extent of parallelism that is possible among the handoff related operations.

In Petri net models, a cycle time represents systems performance under resource constraints during handoff operation and thus can be attributed to the overall efficiency of the system. However, scheduling of execution of these processes that are part of this mobility

event plays an important role in determining the cycle time and systems performance of any handoff event.

A Petri net-based model can be used to analyze various types of mobility events, such as intra-subnet, intra-technology, inter-subnet, and inter-technology handoff. Corresponding Petri net based optimization models can be derived by applying the optimization techniques to the generalized mobility model. These techniques can be applied to each of the sub-processes of the handoff process in a hierarchical manner. In this section, I primarily categorize handoff optimization techniques based on sequential, concurrent, and proactive modes of scheduling of events and model these in Petri nets. Depending upon the type of scheduling technique, the systems resources expensed during a specific operation will vary over a period of time. While a sequential handoff operation takes more time compared to a proactive or concurrent operation, the optimized models using concurrent or proactive operations will need to spend more instantaneous resources during the handoff period. Instantaneous resources could be defined as the peak bandwidth, battery power or amount of CPU cycles that are used when these operations take place.

In order to conduct a performance analysis for different sequence of operations, I have initially considered two handoff related operations, namely discovery and authentication. I apply three different scheduling mechanisms to schedule these two handoff related operations and study the overall performance. Figures 9.23, 9.24 and 9.25 illustrate how these two specific handoff operations in the IEEE 802.11 environment can be represented in a Petri net model using sequential, concurrent and proactive optimization techniques, respectively. Optimal system performance is obtained by comparing the handoff performance (cycle time) and resource utilization (number of tokens) for these handoff methodologies. I have described petri-net-based handoff analysis using Petri net, and evaluate the effect of different scheduling schemes on handoff [DLS+07] and [DLSW09].

9.6.1 Sequential scheduling

Figure 9.23 shows a Petri net model that represents the state transitions when discovery and authentication related operations are performed in sequence. As described in Chapter 4, the resources used during the handoff operations are represented as tokens. The number of tokens needed for each type of operation varies depending upon the amount of resources needed during each of these operations.

In general, scanning is part of layer 2 discovery process and is followed by layer 2 authentication, 4-way handshake and finally, the association with the layer 2 access point. In Figure 9.23 P_0, P_1, P_2, P_3 and P_4 are places that represent different states of discovery and authentication. Shared resources are represented by places such as P_B, P_M and P_P that represent effective user bandwidth (e.g., number of bits transfered), memory and processing power, respectively. I have explained these resources in Section 4.6. Number of tokens in these shared places represent the amount of resources expended during each of these operations. For example, one token can represent 100 kbits of data for resource place P_B that represents the shared resources of bandwidth.

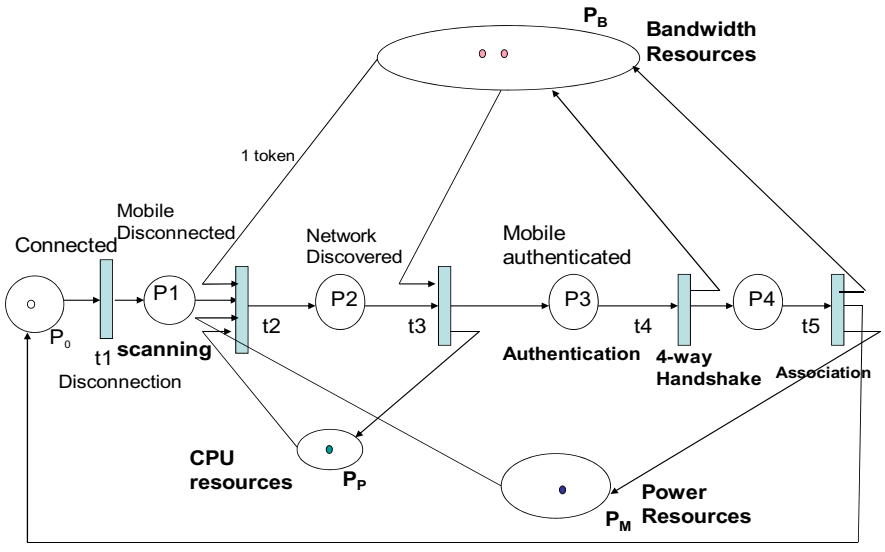


Figure 9.23: Sequential handoff operations

9.6.2 Concurrent scheduling

Figure 9.24 shows the Petri net model when two of the handoff related operations, namely scanning and authentication take place concurrently at the expense of additional bandwidth resources as a result of additional signaling messages. For example, referring to Table 4.4, layer 2 scanning operation results in 298 bytes of signaling messages and four-way handshake operation results in about 504 bytes of signaling messages. These parallel operations speed up the overall handoff operation but the mobile consumes more shared resources during that specific period. As a result, peak resource usage goes up due to parallel operations.

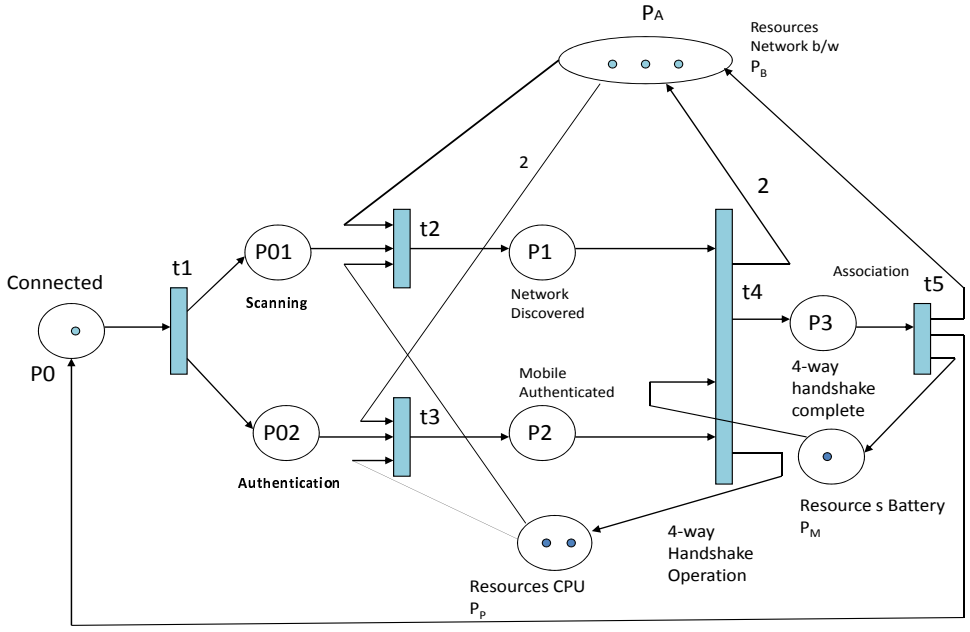


Figure 9.24: Concurrent handoff operations

9.6.3 Proactive scheduling

Figure 9.25 illustrates the Petri net model where some of the handoff related operations are performed pro-actively. During a proactive scenario, the mobile intends to move from its current network to the target network. Many of the handoff related operations such as discovery of the target network elements and authentication with the target network elements are performed ahead of time while the mobile is in the current network. Thus, the shared

resources (e.g., channel resources in the access network), bandwidth in the core network are utilized while the mobile is in the current network and some additional resources such as power and CPU cycles are also used to support operations such as tunneling and proactive IP address caching. For example, as per Table 4.4 in Chapter 4, local caching of IP address will require additional 6 CPU cycles and tunneling operation with the target router will require additional 60 bytes of control message, 3 CPU cycles and will consume 384 nJ (nano Joule) of battery. P_{B1}, P_M and P_D are shared resources that are used in the current network and P_{B2} and P_P are shared resources expensed in the target network.

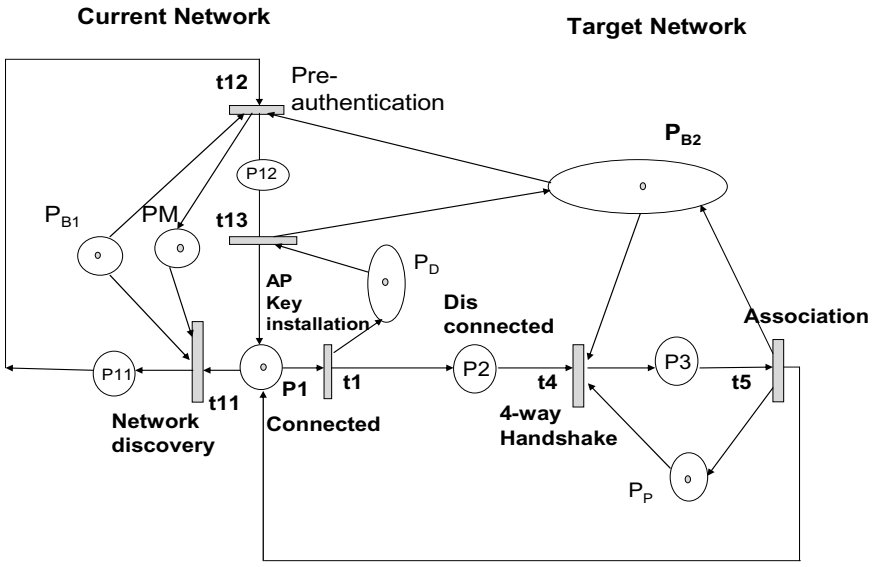


Figure 9.25: Proactive handoff operations

9.7 Verification of systems performance

The systems performance of a Petri net model for a mobility event can be verified in several ways. I illustrate two scenarios here.

In one scenario, the minimum cycle time can be obtained from the Petri net model by investigating the number of circuits[1], the number of transitions, and delay associated

[1]Circuit has been defined in Chapter 3

with each transition. Thus, the performance requirement expressed in cycle time C can be satisfied if and only if CN_k- $T_k \geq 0$ [2] for all circuits in the net. In the second scenario, the token loading matrix and transition matrix are obtained based on the markings of the Petri net model and the associated values of the transitions. Then, the Floyd algorithm is applied to validate the systems performance by computing the shortest distance between every pair of places.

I calculate the cycle time and verify the systems performance based on these three scheduling techniques when applied to two basic handoff operations – discovery and authentication. Experimental results of these two operations were used as the transition timing for Petri net modeling. Similar methods can be applied to compute the cycle time and overall systems performance of the handoff event demonstrating different types of optimization techniques, namely hierarchical binding update, proactive discovery and configuration, and anchor-based security association that are modeled using Petri net.

9.7.1 Cycle-time-based approach

I have introduced the details of cycle time-based approach in Chapter 4. I apply this approach to these three optimization techniques. Table 9.8 shows the transition times *t1, t2, t3, t4 and t5* for different primitive handoff operations associated with discovery and scanning processes obtained from our experiments [LDOS07].

Table 9.8: Experimental results - Layer 2 operations

Transition	Handoff operation	Time taken for operation
t1	Disconnection trigger	5 ms
t2	Scanning	400 ms
t3	Authentication	50 ms
t4	4-way handshake	10 ms
t5	Association	5 ms

[2] Associated terms for Cycle time and Floyd algorithm are defined in Chapter 4.10

Table 9.9: Cycle time from Petri net

Optimization Schedule	Relevant loop in Petri net	D_i	N_i	Max D/N_i
Sequential	p0t1p1t2p2t3p3t4p4t5p0	470	1	470
Concurrent	p0t1p1t3p3t4p0	420	1	420
Proactive	P1t1p2t4p3t5p1	17	1	17

This analysis works under an assumption that there is a handoff delay requirement of 100 ms to support a real-time application under a specific set of resources. I evaluate the overall cycle time when different schedules are applied to these handoff operations and verify if the system is conformant with the delay requirement. Table 9.9 shows the cycle time for three different handoff sequences involving discovery and security association of the handoff. It appears from the results that although time for concurrent operation is smaller than the time for sequential operation, proactive operation is the only operation that satisfies the delay bound of 100 ms under these resource constraint.

9.7.2 Using the Floyd algorithm

I have described the details of how Floyd algorithm can be used to study the systems performance in Chapter 4. In this section, I apply Floyd algorithm to study performance of these three handoff sequences.

Equations 9.1 and 9.2 represent the matrices when the Floyd algorithm is applied to verify the systems performance of the mobility event using sequential scheduling and equation 9.3 represents the matrices illustrating the proactive scheduling as shown in Figure 9.25. Values from the mobility event involving discovery and authentication are used to build the token loading matrix P and transition time matrix Q. First element for P matrix is P_{00} and the last element is P_{77}. Distance matrix and S matrix are then derived from these two matrices. By inspecting the values of the S matrix in equation 9.2 that reflect sequential scheduling, it is found that at least one of the diagonal elements is negative. Thus, this

specific sequential scheduling cannot meet the desired systems performance of cycle time of 100 ms. In order to meet the desired performance level, faster facilities could be used to speed up the transition time or more tokens (resources) could be used in the shared places, thereby increasing the level of concurrency.

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 2 & 0 & 2 & 2 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} w & 5 & w & w & w & w & w & w \\ w & w & 400 & w & w & w & w & w \\ w & w & w & 50 & w & 50 & 50 & 50 \\ w & w & w & w & 10 & 10 & w & w \\ 5 & w & w & w & w & 5 & 5 & w \\ w & w & 400 & 50 & 10 & w & 50 & 50 \\ w & w & 400 & w & w & w & w & w \end{bmatrix}, \tag{9.1}$$

$$\mathbf{CP} - \mathbf{Q} = \begin{bmatrix} \infty & 95 & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & -300 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & -50 & \infty & -50 & -50 & -50 \\ \infty & \infty & \infty & \infty & -10 & -10 & \infty & \infty \\ -5 & \infty & \infty & \infty & -10 & -10 & \infty & \infty \\ \infty & \infty & -200 & 150 & 190 & \infty & 150 & 150 \\ \infty & \infty & -300 & \infty & 90 & 90 & \infty & \infty \end{bmatrix}, \mathbf{S} = \begin{bmatrix} -270 & 195 & -105 & -155 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \end{bmatrix} \tag{9.2}$$

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} w & 5 & w & w & w \\ w & w & 10 & w & w \\ 5 & w & w & 5 & 5 \\ w & w & 10 & w & w \\ w & w & 10 & w & w \end{bmatrix} \tag{9.3}$$

$$\mathbf{CP} - \mathbf{Q} = \begin{bmatrix} \infty & 95 & \infty & \infty & \infty \\ \infty & \infty & -10 & \infty & \infty \\ -5\infty & \infty & -5 & -5 & \\ \infty & \infty & 90 & \infty & \infty \\ \infty & \infty & 90 & \infty & \infty \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 90 & 95 & . & . & . \\ 105 & 100 & . & . & . \\ . & . & 85 & . & . \\ . & . & . & . & 85 \\ . & . & . & . & 85 \end{bmatrix} \tag{9.4}$$

Equation 9.3 shows the matrices for proactive scheduling based on the transition times obtained from the experiments. By inspecting matrix S in equation 9.4, it appears all the diagonal elements of the matrix are positive. Thus, by applying the Floyd algorithm, it is verified that the proactive scheduling, when applied to discovery and authentication processes, satisfies the systems performance of the required minimum cycle time of 100 ms. I have also used several automated tools such as TimeNet [ZGFH99], STPNplay [Ryu04] and Petri net Tool [MMP03] to model the behavior of the handoff system, capture the systems performance and evaluate the performance characteristics of the mobility protocols and associated optimizations. By analyzing these Petri net-based models, one can easily predict the performance of a handoff system given a set of resources when various scheduling mechanisms are applied.

9.8 Petrinet-based modeling for multi-interface mobility

In this section, I illustrate Petri net modeling involving handover between two different types of access networks (e.g., 802.11 and CDMA) covering three scenarios, namely, *par-*

allel operations, *break-before-make*, and *make-before-break* and of both the interfaces. As described in Table 3.1 in Chapter 3, each access network has different ways of discovering resources and network parameters; authentication mechanisms and encryption algorithms are also different for each access network. Thus, resource requirement (e.g., battery, bandwidth and CPU) for each of these handoff operations in either access network are different.

9.8.1 Multi-homing scenario

When a multi-interface mobile has the coverage for both CDMA and 802.11, both the interfaces start getting configured at the same time and consume battery and CPU resources on the mobile, while bandwidth resources are consumed in each access network independently. Ideally, based on certain policy such as the type of application or tariff, the mobile makes a decision regarding which interface should be used for communication.

9.8.2 Break-before-make scenario

During a break-before-make handoff scenario, one interface does not start getting configured until the second interface is disconnected. During a break-before-make scenario where one type of interface (e.g., CDMA) comes up after the other interface (e.g., 802.11) goes down, the resource consumption during handoff will be different based on whether the mobile is handing over from 802.11 to CDMA or vice-versa.

9.8.3 Make-before-break scenario

During a make-before-break operation, when the mobile is still communicating using 802.11 interface, CDMA interface starts getting activated as the signal-to-noise ratio on 802.11 interface begins to deteriorate. While this helps to reduce the handoff delay compared to break-before-make case, it requires more battery power in the mobile since both the interfaces stay powered on for sometime. In certain types of make-before-break scenario, many

of the handoff related operations such as resource discovery, authentication, security context transfer for CDMA interface are taken care of by 802.11 interface thus could delay the activation of CDMA interface. Delaying the activation of CDMA interface helps in reducing the battery usage for the mobile. Thus, it is an important point to consider when to start the activation of CDMA interface based on the resource availability in the network.

9.8.4 MATLAB-based Petri net modeling for multi-interface

Resource modeling of multiple interfaces as part of performance modeling of multi-interface mobility allows a more principled decision of which interfaces to use for handoff operations. In case of multi-interface mobility, currently, it is not obvious to figure out which interface needs to be turned on when and for which handoff operation (e.g., discovery, configuration). Interaction among multiple interfaces i.e., when to switch up or switch down which interface can be determined by Petri net modeling. Since each interface has different resource requirement for the same kind of handoff operation, Petri net model allows one to find out which operation can be carried out by which interface given resource constraints such as battery, CPU and bandwidth.

I model above three scenarios using the MATLAB-based Petri net tool. Table 9.10 shows the amounts of resources and time needed to take care of specific handoff related operations for two different types of interfaces, namely, CDMA1X-EvDO and IEEE 802.11. These values are based on the experimental results from the testbed as described in Chapter 5. Since these two interfaces have different access characteristics, amounts of resources and timing to complete each of these handoff operations also vary. In Chapter 4, I have discussed the energy required to transmit and receive a bit for by CDMA and 802.11. As discussed in [SE09], it takes more amount of energy to transmit a bit using CDMA interface compared to 802.11 interface. Layer 2 authentication process for 802.11 and CDMA are different and will need different signaling messages and amounts of bytes exchanged. Also, PPPoE gives rise to additional 8 bytes of header for the same handoff related operations.

Table 9.10: Resources and timing for 802.11 and CDMA

Operations	Resources in 802.11			Resources in CDMA			Timing (ms)	
	Battery (nJ)	Bytes	CPU (cycles)	Battery (nJ)	Bytes	CPU (cycles)	802.11	CDMA
Discovery	414000	345	12	1968000	328	9	745	422
Layer 2 Authentication	4126800	3439	29	1392000	232	14	106	200
Configuration	2257200	1881	22	5454000	909	12	510	850
Security Association	940800	784	10	4752000	792	10	640	4500
Binding Update	422400	352	18	2160000	360	18	168	599

The MATLAB-based model in Figure 9.26 illustrates a scenario when the mobile is under the coverage of both 802.11 access and CDMA access. Thus, in this scenario, both of the interfaces will begin its configuration independently. However, since each interface will have its dedicated access network, a single bandwidth (channel resource) place cannot be shared by the operations related to both the interfaces, whereas battery power and CPU samples can be considered as shared resources for both. Thus, two different bandwidth places can be modeled, one for each type of access network. Each type of bandwidth resource will be shared among the handoff events within that specific access network. For example, channel resources for CDMA network will be shared by identifier configuration and discovery operation within CDMA network.

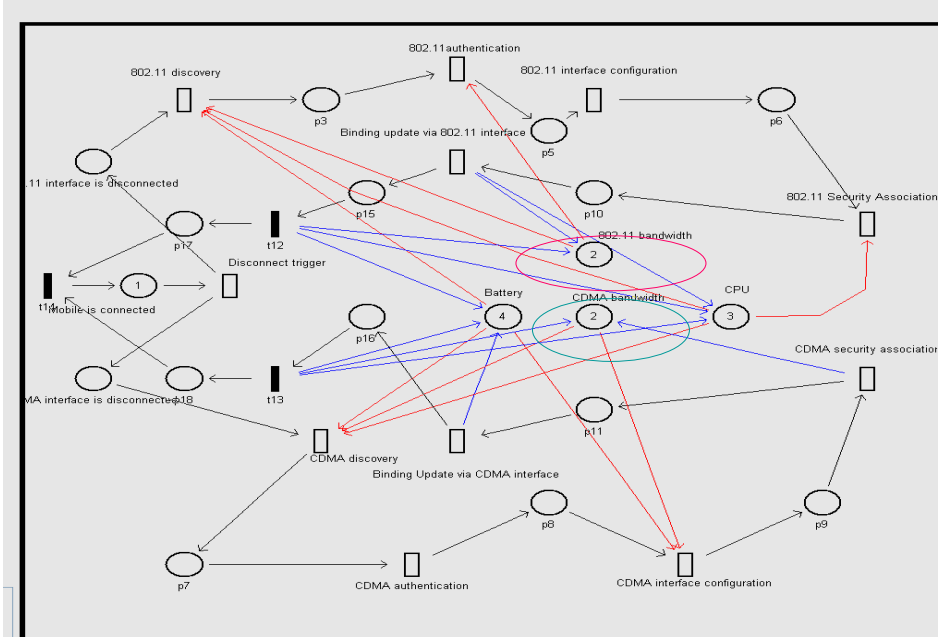


Figure 9.26: MATLAB-based model for parallel CDMA and 802.11 operations

Figure 9.27 shows the MATLAB-based model for a multi-interface mobility scenario when the mobile is communicating using its 802.11 interface. As the signal-to-noise ratio of 802.11 interface decreases, CDMA interface is in the process of being connected.

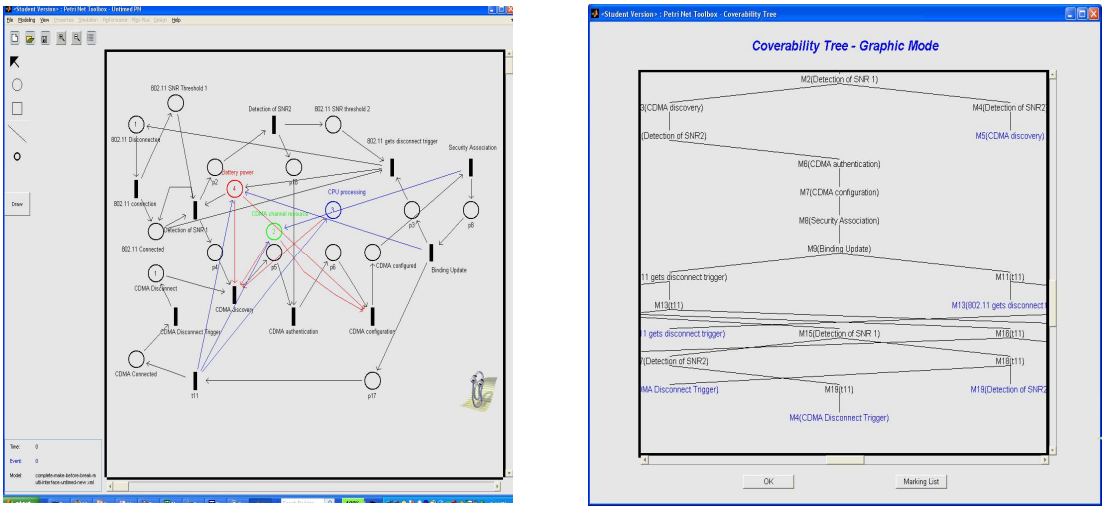


Figure 9.27: (a) MATLAB-based model for make-before-break (b) Coverability tree

In the second type of make-before-break scenario, many of the handoff related operations for the second interface (e.g., CDMA) are performed by the currently serving interface(e.g., 802.11). For example, operations like discovering the next point of attachment for CDMA interface, authenticating the CDMA interface, setting up the PPP context for the CDMA interface can still be performed by the currently serving interface without bringing up the CDMA interface until after most of the handoff related operations for the CDMA interface are completed by the 802.11 interface. This will delay powering on the CDMA interface and will not use as much resources as the previous make-before-break case where both the interfaces are up when CDMA interface is in the process of coming up. However, the 802.11 interface will end up exchanging additional handoff related signaling on behalf of the CDMA interface.

Figure 9.28 shows the MATLAB-based model for a multi-interface mobility scenario where only one interface (e.g., 802.11 or CDMA) is active at any point of time and CDMA

interface prepares itself only after the 802.11 interface is disconnected. Since both the interfaces are not active at the same time, it does not consume that much resources compared to previous two scenarios but this scenario takes the most amount of time for handoff to complete. Unlike the make-before-break case shown in Figure 9.27, here the CDMA interface goes through the process of getting connected only after the 802.11 interface is disconnected at a specific signal-to-noise threshold.

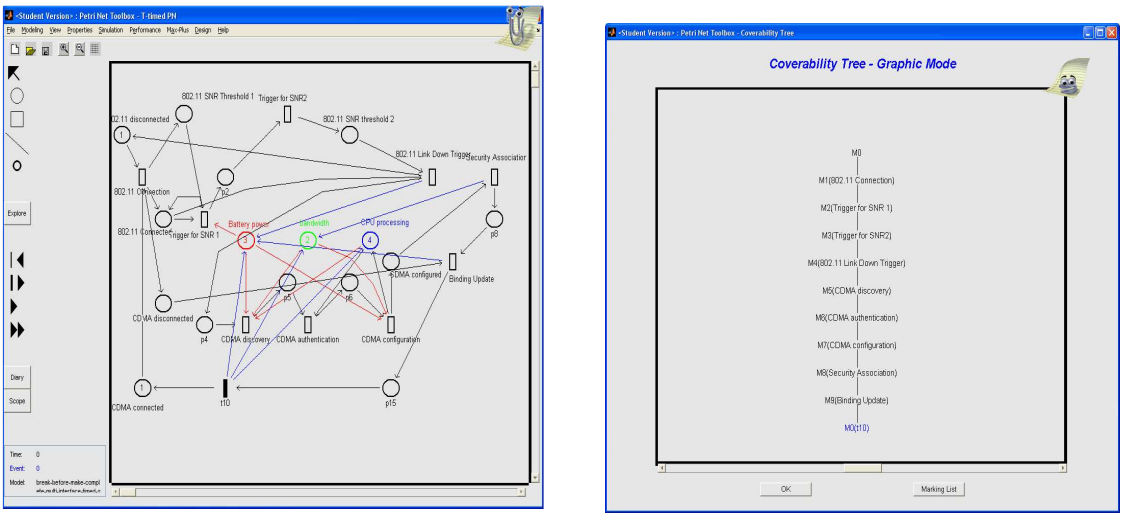


Figure 9.28: (a) MATLAB-based model for break-before-make (b) Coverability tree

9.9 Deadlocks in handoff scheduling

Optimization intrinsically requires cross-layer designs and speculative execution in preparation for possible handover. Thus, sequence of handoff operations (e.g., parallel, proactive, sequential) plays a role in determining the extent of optimization that can be achieved. The use of Petri nets has allowed resource dependencies and data dependencies to be explicitly represented and thus allows for checking on blocking and deadlocks. Scheduling of handoff operations needs to take into account the data dependency and resource availability. In this section, I describe a few scenarios that might give rise to a deadlock in the system and

ways to avoid the deadlocks by changing the schedules or by adding resources.

Reachability analysis has been explained in Chapter 4. Reachability analysis is one way of detecting the deadlocks in the handoff system. If a specific schedule generates a coverability tree so that no subsequent transition is allowed at a specific marking (e.g., M_i), then that specific sequence exhibits a deadlock. Thus, by inspecting the coverability tree of any handoff schedule it is possible to determine if a deadlock exists within a certain specific schedule.

I illustrate a few scenarios that exhibit how deadlocks take place during the handoff process. Then, I propose solutions to take care of this deadlock situation. I use MATLAB-based Petri net tool to construct the coverability trees and verify the deadlock properties and then demonstrate how deadlocks are eliminated.

9.9.1 Handoff schedules with deadlocks

In this section, I illustrate few specific handoff schedules that exhibit a deadlock situation and propose the solutions that will avoid these deadlocks.

If a specific schedule generates a coverability tree so that no subsequent transition is allowed at a specific marking (e.g., M_i), then that specific sequence exhibits a deadlock. Thus, by inspecting the coverability tree associated with any schedule, it is possible to determine if a deadlock exists in a specific schedule.

I describe few specific scenarios to compare two schedules one with deadlock and one without deadlock.

9.9.1.1 Deadlocks due to lack of data

A specific handoff schedule could also lead to a deadlock if the sequence of operations do not follow the data dependency graph. In this case, a specific handoff operation cannot proceed due to lack of data that is expected from another handoff operation. For example, in a normal situation, if the mobile configures its layer 3 identifier and assigns it to the

interface as part of layer 3 handover, without finishing L2 handoff (e.g., channel change) operation, the mobile cannot complete the rest of the handoff operations and it will lead to a deadlock.

9.9.1.2 Deadlocks due to resource sharing

A second scenario could be due to the result of concurrent operations resulting in lack of resources. For example, if during a concurrent operation a handoff schedule is designed to perform both the operations, layer 2 discovery and authentication in parallel (e.g., layer 2 discovery process starts the authentication process when the former process is still not completed). If there are not enough tokens available in the bandwidth resource place (Pb) that can enable the transition for authentication process after the transition for discovery process is enabled, then the authentication process cannot proceed further.

Figure 9.29 shows a deadlock situation resulting out of lack of resources during a concurrent operation. By investigating the coverability tree, it shows that the system does not get back to its initial state due to deadlock.

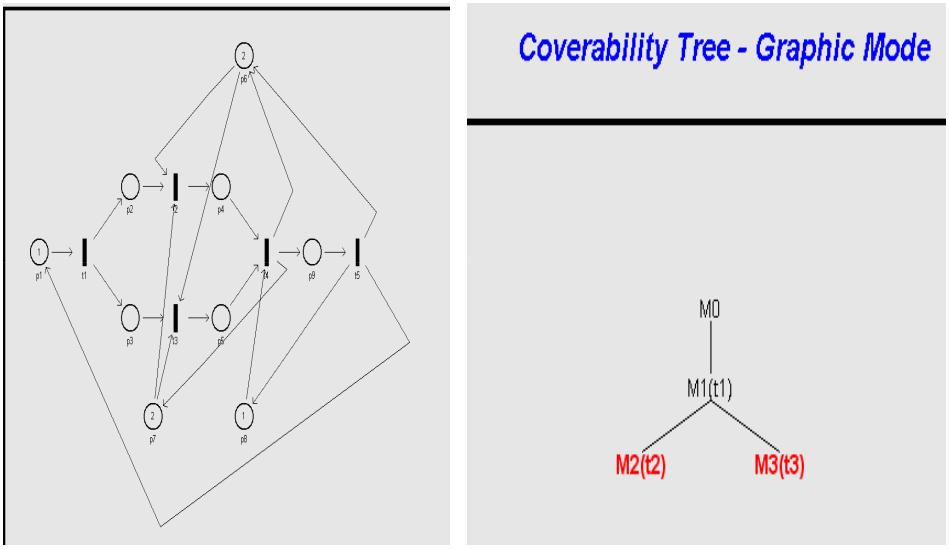


Figure 9.29: Deadlock due to resource constraints

9.9.1.3 Deadlocks in simultaneous mobility

Another deadlock scenario can arise due to failure of binding update during simultaneous mobility, where binding update from either client does not complete due to overlapping handoff of the mobiles. The rest of the handoff operations cannot proceed due to non-completion of binding updates of either of the clients. I have described the details of simultaneous problems in Chapter 7. Here, I illustrate Petri net modeling resulting out of simultaneous mobility scenarios. Figure 9.30 (a) shows an equivalent Petri net model of Figure 7.1 where there is no problem due to simultaneous mobility and both the mobiles can communicate with each other. Figure 9.30 (b) shows the corresponding coverability tree. It is apparent that a successful communication between both the mobiles depend on successful completion of configuration and binding updates of both the mobiles.

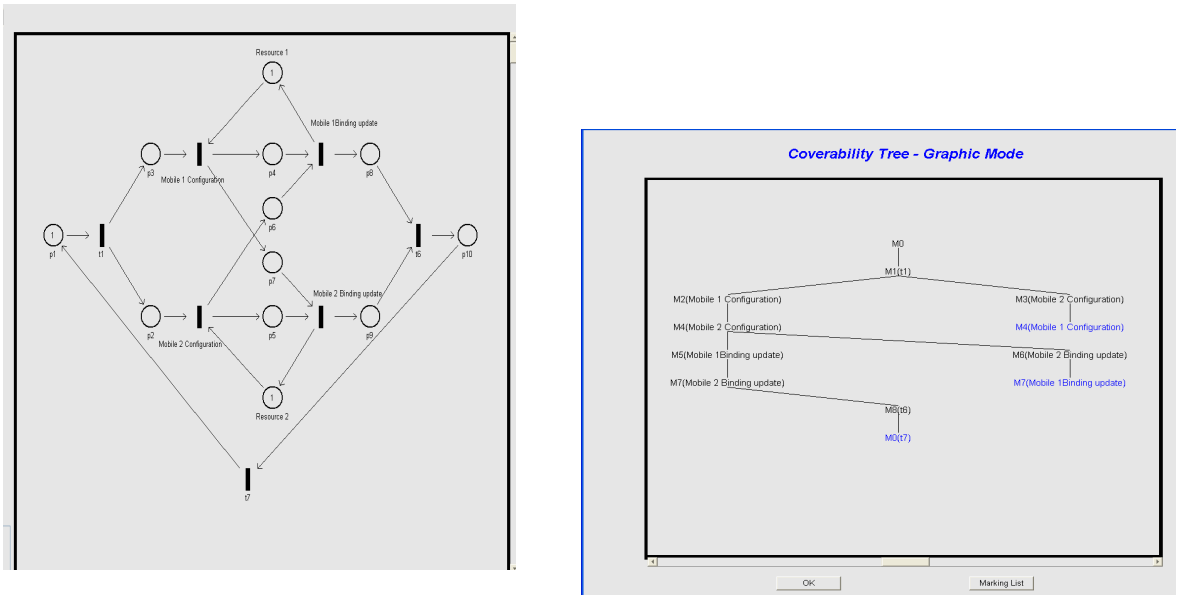


Figure 9.30: (a) Petri net model for simultaneous mobility (b) Coverability tree

Figure 9.31 illustrates a Petri net example of a deadlock situation in simultaneous mobility because one of the mobile keeps getting reconfigured. Figure 9.32 shows the equivalent coverability tree, that shows the markings.

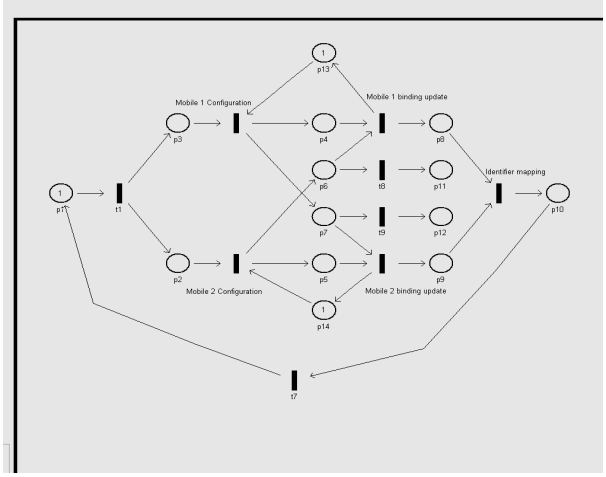


Figure 9.31: Deadlock in simultaneous mobility

9.9.2 Deadlock prevention and avoidance in handoff schedule

There are several ways the deadlock situations can be prevented or avoided. Deadlock prevention consists of falsifying one or more of those necessary conditions mentioned in Chapter 4 by using static resource allocation policies so that the deadlocks are completely eliminated. Since deadlock prevention is accomplished by static policies that is known to result in poor resource utilization and reachability analysis technique to arrive at deadlock prevention policies can become infeasible if the state space is very large, deadlock avoidance techniques are preferred sometimes. Deadlock avoidance techniques attempt to falsify one or more of the necessary conditions in a dynamic way by keeping track of the current state and possible future conditions. The idea is to let the necessary conditions prevail as long as they do not cause a deadlock but falsify them as soon as a deadlock becomes a possibility in the immediate future. Deadlock avoidance leads to better resource utilization.

Viswanadham et al. [VNJ90] discuss about the deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models. Similar techniques can be applied for prevention or avoidance of deadlocks in the handoff system. Each of the above deadlock scenarios can be avoided either by adding resources, changing the schedule or by

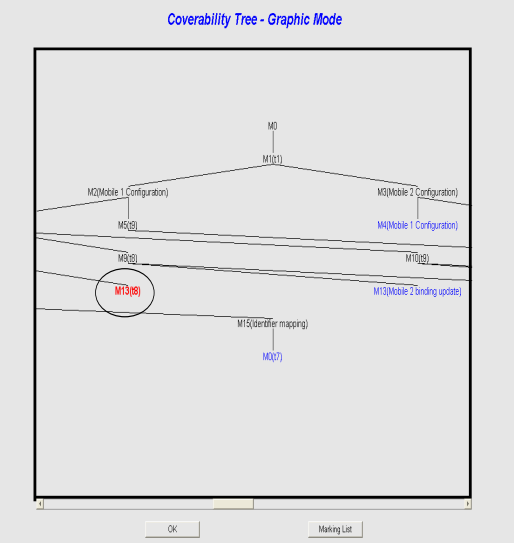


Figure 9.32: Coverability tree for deadlock in simultaneous mobility

introducing additional components in the network. Reachability graph of a Petri net model representing a handoff system can be used to arrive at the resource-allocation policies that enforce deadlock prevention.

Deadlocks in scenario 1 can be prevented in two ways. In one way deadlock can be prevented by scheduling the handoff operations so as not to enable a transition unless there is data available from the previous operation. Thus, the mobile should not be scheduled to perform layer 3 operation unless layer 2 handoff operation is over. Second case involves additional operations to avoid the handoff. For example, in the second case, unless a transient tunnel between the mobile and the next hop router is established the rest of the operations such as binding update and media forwarding to the mobile cannot complete and will lead to mobile's incomplete handoff. Thus, setting up this additional tunnel is an additional operation that is needed to avoid the possible deadlock in such situation.

One way to prevent the deadlock illustrated in Figure 9.33 is to increase the number of tokens in resource place Pb (e.g., Bandwidth resources) to take care of the concurrent operations. Figure 9.33 illustrates how the existing deadlock situation due to lack of resources

during a concurrent operation as shown in Figure 9.29 is taken care of by adding additional resources. It shows how allocation of the additional resources have resulted in elimination of deadlock. I verify that a schedule is deadlock free by doing a reachability analysis. As shown in the coverability tree, in the absence of deadlock, the mobile comes back to its initial state.

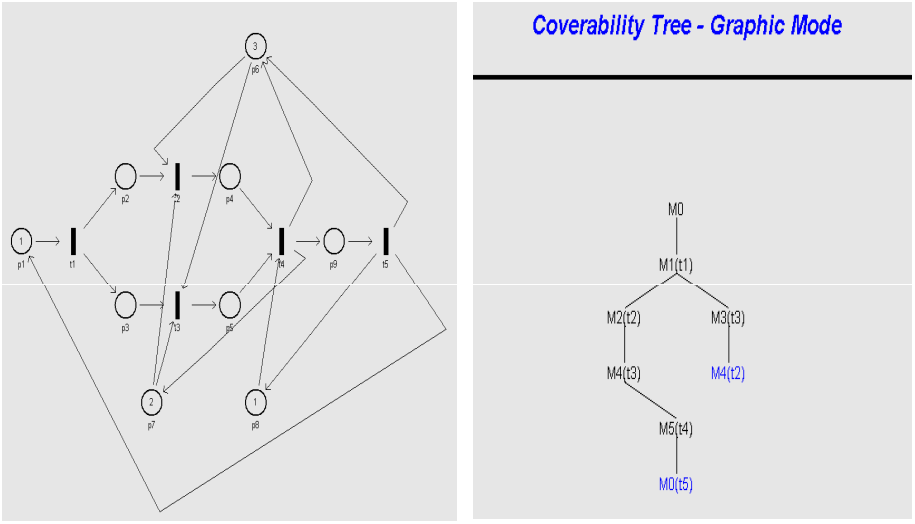


Figure 9.33: Avoidance of deadlock in concurrent operation

Deadlocks in simultaneous mobility scenario can be avoided by installing additional components in the network such as proxies in the network as explained in Chapter 7. Figure 9.34 illustrates the MATLAB-based model and corresponding coverability tree that shows how the deadlock in simultaneous mobility is avoided by introducing retransmission technique or forwarding agent.

Similarly, MATLAB-based models can be used to construct the equivalent coverability tree that can determine the presence of deadlocks in the systems where the handoff operations do not follow proper precedence rules.

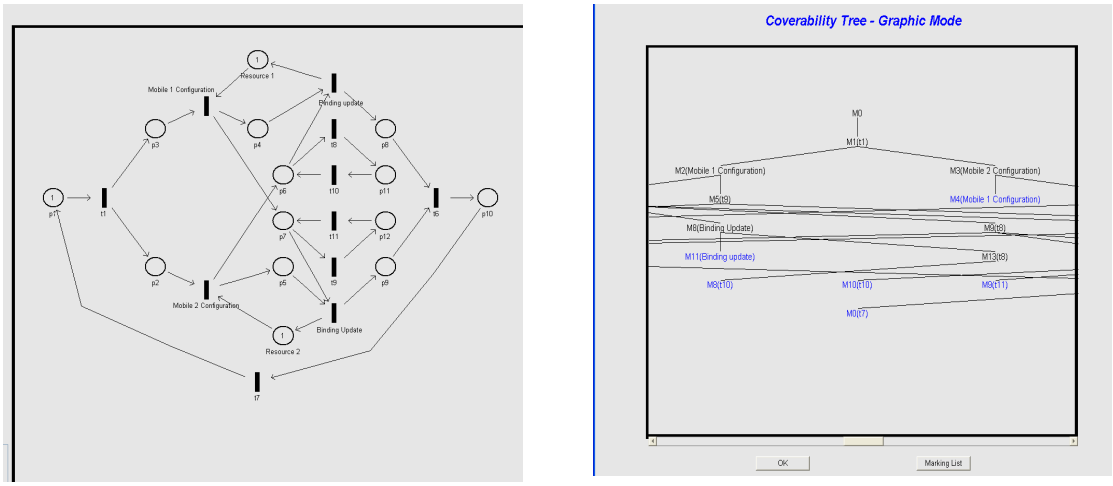


Figure 9.34: (a) Deadlock avoidance simultaneous mobility (b) Coverability tree

9.10 Analysis of level of concurrency and resources

In this section, I describe how the amount of concurrency and resources affect the systems performance for handoff operations by using Petri net model. In a Petri net based model, systems performance is determined by cycle time. Procedure for cycle time computation is based on formulating the cycle time as a minimal cost-to-time ratio cycle problem [Law01] where resources needed can be treated as the cost and time taken is the amount of time needed to complete a set of handoff related operations for a specific circuit within a Petri net.

A lower cycle time implies higher systems performance. Given two sequences of handoff systems, a lower cycle time implies less amount of resources needed to complete the task within a certain amount of time. Thus, a cycle time provides an indication of tradeoff between systems resources and the task completion time, and is thus a measure of handoff time under certain available resources.

Figure 9.35 shows different handoff related operations for a layer 2 handoff as the mobile moves from old access point to one of the target APs. These handoff related operations demonstrate the delays due layer 2 discovery, authentication, 4-way handshake and asso-

ciation with a new access point. I have explained these experimental results for layer-2 handoff operations in [LDOS07] for both roaming and non-roaming scenarios.

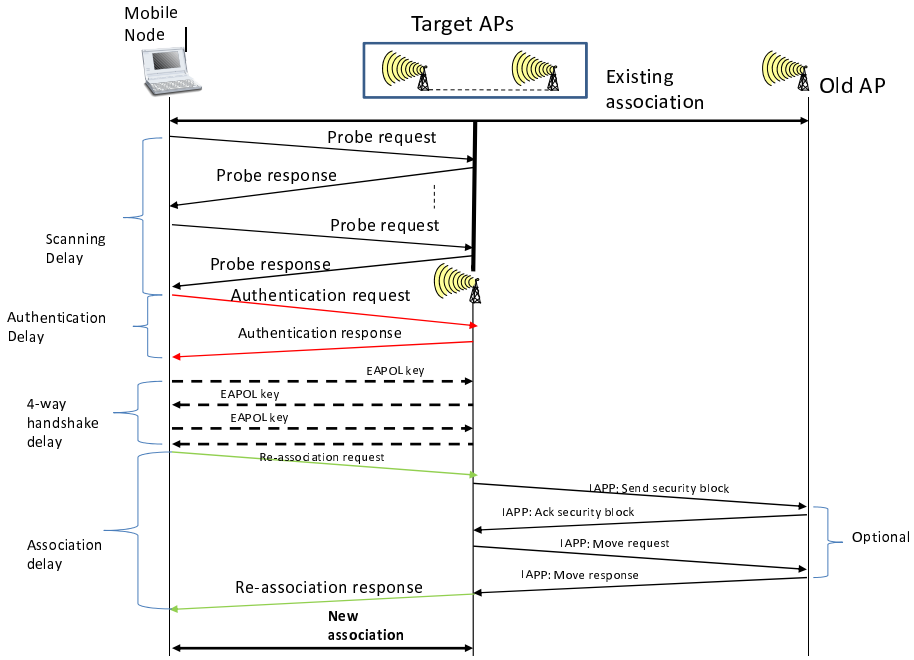


Figure 9.35: Illustration of layer 2 handoff

Next I describe several Petri net models based on the handoff operation shown in Figure 9.35 using four possible sequence of handoff operations and resources. I then apply Floyd algorithm to verify if a specific sequence of handoff operations meets a certain systems performance in terms of cycle time. This analysis shows how changing the concurrency level (e.g., no parallel operations, two operations in parallel or three operations in parallel) or changing the resource parameters does affect the required systems performance.

Figure 9.36 shows the Petri net model illustrating sequential layer 2 handoff operations, namely layer 2 discovery, authentication, 4-way handshake and association. The transition timings $t1$, $t2$, $t3$, $t4$ and $t5$ are the values obtained from experimental results [LDOS07]. Given a required systems performance value of C one can use Floyd algorithm to determine whether this value of systems performance is achieved under a certain level of concurrency of operations and resource availability. Since Floyd algorithm can only be applied to deci-

sion free Petri net [3] these Petri net models need to be converted to corresponding decision free Petri nets where applicable.

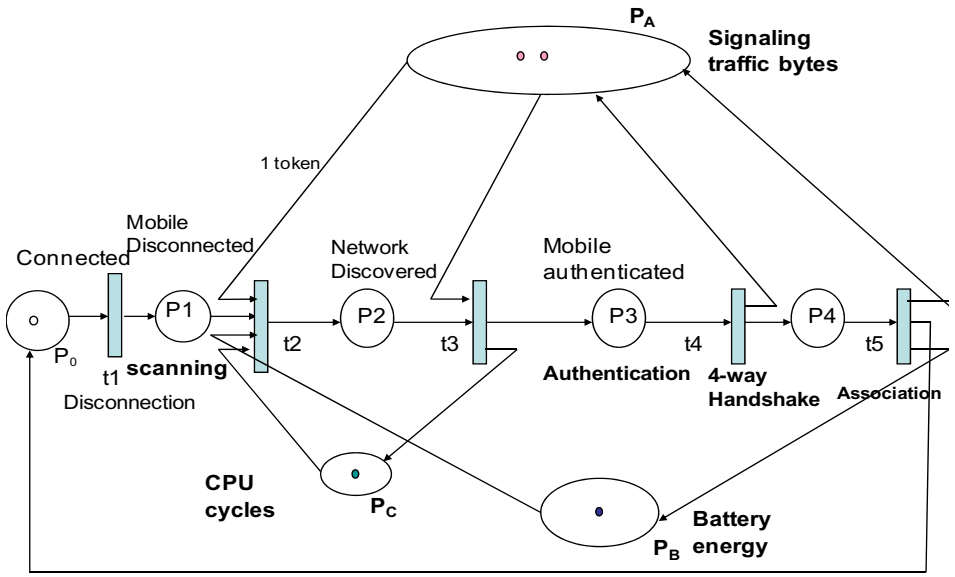


Figure 9.36: Sequential layer 2 operations

Figure 9.37 shows a model illustrating two layer 2 handoff operations in parallel, namely scanning and authentication. By applying the Floyd algorithm, I derive the S matrix to determine the shortest distance between every pair of places for a given value of C. For example, for the sequential model shown in Figure 9.36 Equation 9.5 shows the S matrix for a C value of 100 whereas Equation 9.6 shows the S matrix for a C value of 500. Although a lower value of C (e.g., 100) compared to a value of C = 500, offers a better systems performance resulting in lower handoff delay, it is evident that this specific sequence cannot meet the required systems performance at C=100. In order to obtain a cycle time lower than (C=500), it is required to increase the level of concurrency among the handoff operations.

[3] Decision-free system is defined in Chapter 4

$$\mathbf{S}(\mathbf{100}) = \begin{bmatrix} -370 & 35 & -305 & -355 & -365 & -370 & -315 & -370 \\ -465 & -370 & . & . & . & . & . & . \\ -65 & . & -265 & . & . & . & . & . \\ -15 & . & . & -370 & . & . & . & . \\ -5 & . & . & . & -370 & . & . & . \\ 135 & . & . & . & . & -260 & . & . \\ -365 & . & . & . & . & . & -365 & . \\ -365 & . & . & . & . & . & . & -350 \end{bmatrix} \tag{9.5}$$

$$\mathbf{S}(\mathbf{500}) = \begin{bmatrix} 30 & . & . & . & . & . & . & . \\ . & 30 & . & . & . & . & . & . \\ . & . & 540 & . & . & . & . & . \\ . & . & . & 540 & . & . & . & . \\ . & . & . & . & 940 & . & . & . \\ . & . & . & . & . & 935 & . & . \\ . & . & . & . & . & . & 940 & . \\ . & . & . & . & . & . & . & 135 \end{bmatrix} \tag{9.6}$$

Analyzing Equation 9.6 it shows that all the diagonal entries are positive. Thus, the system meets the performance for a value of C=500.

Figure 9.37 shows two handoff operations, namely discovery and authentication in parallel.

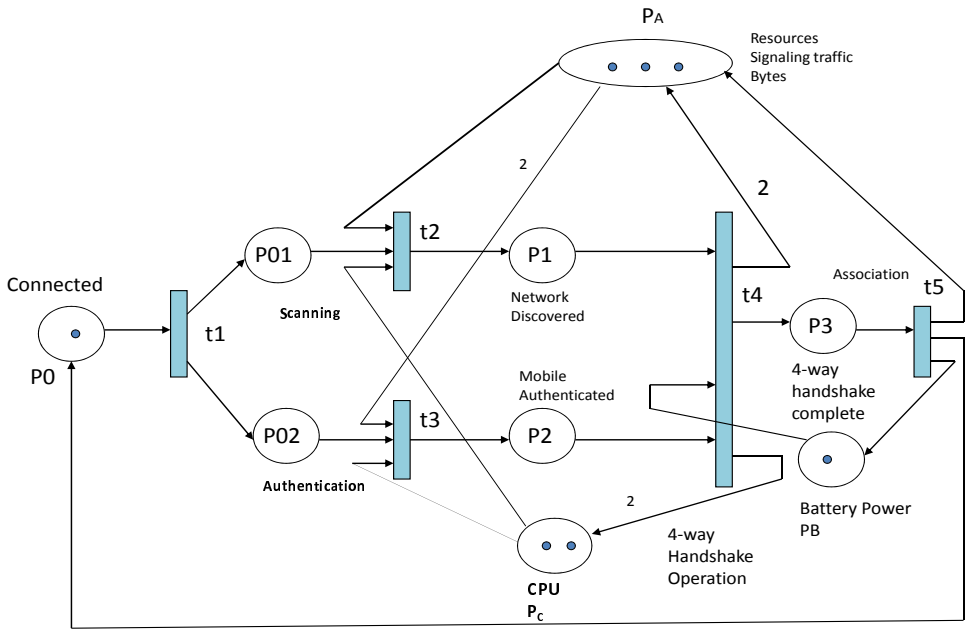


Figure 9.37: Two concurrent L2 handoff operations

$$\mathbf{S(450)} = \begin{bmatrix} 30 & . & . & . & . & . & . & . & . & \\ . & 30 & . & . & . & . & . & . & . & \\ . & . & 380 & . & . & . & . & . & . & \\ . & . & . & 940 & . & . & . & . & . & \\ . & . & . & . & . & 840 & . & . & . & \\ . & . & . & . & . & . & 435 & . & . & \\ . & . & . & . & . & . & . & 940 & . & \\ . & . & . & . & . & . & . & . & 435 & . \\ . & . & . & . & . & . & . & . & . & 840 \end{bmatrix} \tag{9.7}$$

By applying Floyd algorithm and inspecting the corresponding S matrix in Equation 9.7 it is evident that when the level of concurrency is 2, the system can operate under a lower value of C i.e., 450 (better systems performance) compared to a C value of 500 (lower performance) for sequential operations as shown in Equation 9.6. This indicates that handoff delay will be reduced when the system meets the performance with a lower

value of C.

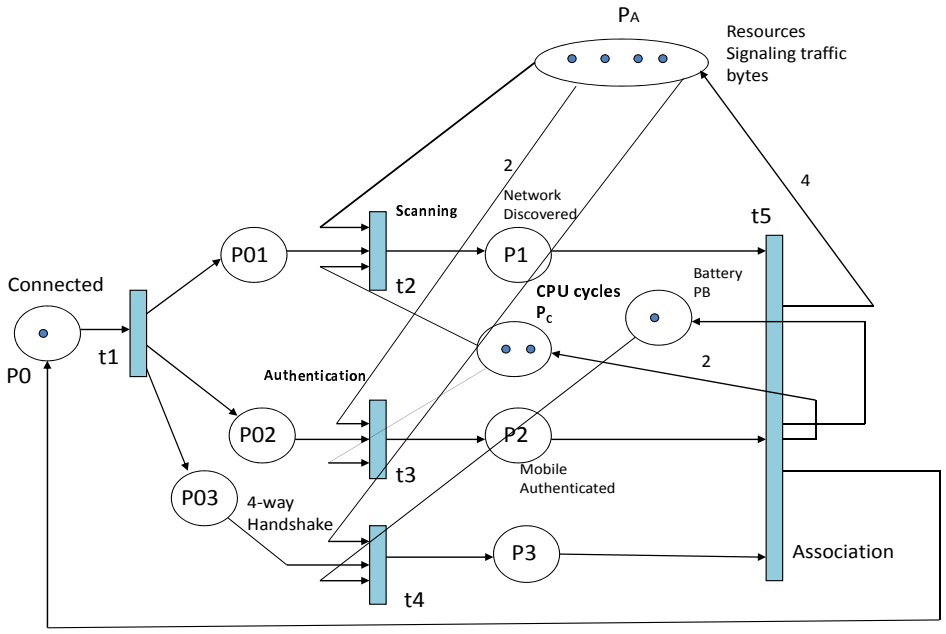


Figure 9.38: Three concurrent L3 handoff operations

I further add concurrency as shown in Figure 9.38 where discovery, authentication and 4-way handshake operations take place in parallel. Compared to Figure 9.37, additional level of concurrency will require more resources in place P_A in order to avoid the deadlocks.

$$\mathbf{S(410)} = \begin{bmatrix} 0 & . & . & . & . & . & . & . & . & . \\ . & 0 & . & . & . & . & . & . & . & . \\ . & . & 395 & . & . & . & . & . & . & . \\ . & . & . & 390 & . & . & . & . & . & . \\ . & . & . & . & 1235 & . & . & . & . & . \\ . & . & . & . & . & 765 & . & . & . & . \\ . & . & . & . & . & . & 395 & . & . & . \\ . & . & . & . & . & . & . & 1235 & . & . \\ . & . & . & . & . & . & . & . & 395 & . \\ . & . & . & . & . & . & . & . & . & 765 \end{bmatrix} \tag{9.8}$$

By applying Floyd algorithm and inspecting the corresponding S matrix in Equation 9.8 it is also evident that by adding concurrency the system meets the systems performance at cycle time C=410.

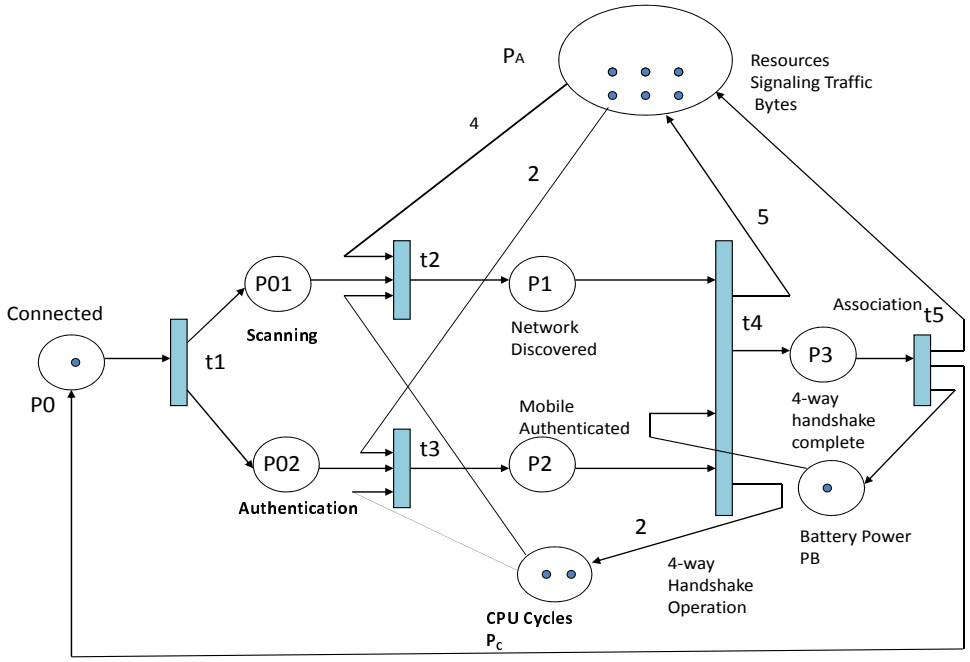


Figure 9.39: Two concurrent handoff operations with additional resources

Petri net models from previous examples show how adding concurrency results in reduction in cycle time (increased systems performance). While keeping the level of concurrency same, the cycle time can also be reduced if there are additional resources dedicated to reduce the transition timing. For example, the Petri net model shown in Figure 9.37 can be converted to a Petri net model as shown in Figure 9.39 with additional resources in P_A. These additional resources will help expediting the scanning and authentication operations. Thus, values of $t2$ and $t3$ will be reduced.

$$\mathbf{S}(\mathbf{200}) = \begin{bmatrix} 80 & 195 & 195 & . & . & . & . & . & . & \\ 195 & 180 & . & . & . & . & . & . & . & \\ . & . & 130 & . & . & . & . & . & . & \\ . & . & . & 1090 & . & . & . & . & . & \\ . & . & . & . & . & 340 & . & . & . & \\ . & . & . & . & . & . & 1855 & . & . & \\ . & . & . & . & . & . & . & 1090 & . & \\ . & . & . & . & . & . & . & . & 185 & . \\ . & . & . & . & . & . & . & . & . & 340 \end{bmatrix} \quad (9.9)$$

By applying Floyd algorithm and inspecting the S matrix in Equation 9.9 it verifies that the model shown in Figure 9.39 does meet the systems performance at a value of $C = 200$.

From the analysis of above Petri net models it is verified that a handoff system can meet a desired lower value of cycle time C either by increasing the number of concurrent operations or by increasing the amount of resources for the same level of concurrency.

9.11 Tradeoff analysis for proactive handoff

As discussed in Chapter 4, several types of systems resources are utilized during a handoff operation. Proactive and concurrent operations reduce the handoff delay and packet loss at the expense of additional resources. For example, although proactive handoff operations offer better performance, it utilizes systems resources while the mobile is in the current network and is engaged in these proactive operations. I discuss different levels of proactive operations that are needed when the mobile is about to move to a target network and investigate the tradeoff between handoff delay and resource utilization during this operation.

When the mobile is about to move to a new network it does perform a set of proactive handoff operations with the target network in order to reduce the handoff delay and

packet loss. When there are multiple neighboring networks, doing proactive operations with additional networks increases the probability of successful handover. However, it also needs more systems resources because of many of the handoff related operations, such as tunneling, pre-configuration and pre-authentication processes involved during the proactive operation.

Figure 9.40 shows an example of how a mobile can perform proactive handoff operation with multiple target networks. It shows the current network and three neighboring networks, namely, network A, network B and network C. It shows how multiple tunnels can be set up between the mobile and target routers and proactive operations can take place with multiple target networks. Establishing proactive operations with multiple networks increases the probability of successful handover but ends up using more resources.

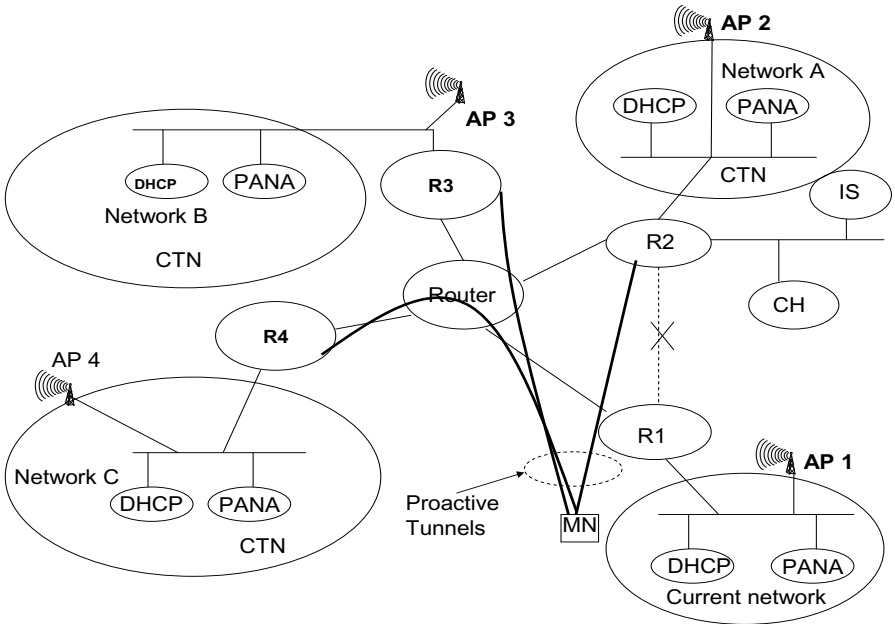


Figure 9.40: Proactive handoff operation with multiple target networks

Some of the resources that are utilized during the proactive operations are defined as follows:

1. Additional signaling to complete pre-authentication with the neighboring networks results in additional bandwidth usage in the current network.

2. Caching the IP address of the neighboring networks in mobiles for certain amount of time. It needs additional processing in the mobile for storing these IP addresses. In addition, it also uses up the temporary IP addresses from the neighboring networks by keeping these addresses in the cache.

3. There is cost (e.g., CPU power) associated with setting up additional transient tunnels (proactive tunnels) between the mobile and the target routers in the neighboring networks and tearing them down once the handover is over. There is also a cost (e.g., bandwidth) associated with maintaining these tunnels during the handover.

4. Binding update with multiple IP addresses obtained from the neighboring networks results in multiple transient data streams between the CN and mobile over these transient tunnels. This will result in additional bandwidth usage because of duplicate data streams.

I describe below several phases of proactive operations. Resource requirement will also vary based on the levels of proactive operation used.

The very basic level of proactive operation involves authenticating the mobile with the multiple authentication agents in the neighboring networks, but pre-configuration and binding update take place only after layer 2 movement to a specific network is complete.

During second level of pro-active operation, the mobile can also complete the pre-configuration while in the previous network, but can perform the binding update until after the mobile has moved to the new network. Like the previous case, here the mobile also does not need to set up the proactive tunnels since binding update is done after the mobile has moved to the new network.

The third type of proactive operation involves all the three processes to complete while the mobile is in the previous networks, such as authentication, configuration and binding update. However, this specific type of pre-authentication utilizes the most amount of resources.

When only pre-authentication and pre-configuration are done ahead of time with multiple networks, the mobile sends only one binding update to the CN or the home agent after it has moved to the new network.

In case of proactive binding update, binding update with multiple contact addresses is sent. These contact addresses are the care-of-addresses from the neighboring networks obtained ahead of time during pre-configuration phase. The following is an illustration of this specific case when the mobile sends multiple binding updates while in the previous network but ends up moving to a specific target network. MN sends a binding update to CH with multiple potential care-of-addresses such as $c1$,$c2$, and $c3$ that were obtained from three neighboring networks. This allows the CN to send transient multiple streams to the mobile over the pre-established tunnels. After the mobile moves to a specific target network, it sends another binding update to the CN with the care-of-address of the mobile in the network where the mobile has moved in. This way the CN stops sending media to other neighboring networks where it does not end up moving. Some of the issues with multiple stream are consumption of extra bandwidth for a small period of time.

Alternatively, one can apply the buffering technique at the neighboring target access routers or at the home agent. Transient data can be forwarded to the mobile after it has moved in. Forwarding of data can be triggered by the mobile either as part of mobile IP registration or as a separate buffering protocol.

I have experimentally verified multiple pre-authentication techniques using location assisted handoff [DCT+07]. In this specific experiment, I created multiple possible target networks. Based on the relative location of the mobile with respect to several access points in the target networks, the mobile completes proactive authentication with one or multiple target networks. I have experimented with two different levels of pre-authentication processes as described above. In one case, the mobile performs multiple authentication, but sends binding update to one access network after setting up the proactive tunnel with the target network. In the second case, the mobile completes pre-authentication with all the

three neighboring networks and sends binding updates over the proactive tunnels.

I discuss some guidelines for the roaming clients that use pre-authentication mechanisms to reduce the handoff delay. These guidelines can help determine the extent of pre-authentication operation that is needed based on a specific type of movement of the client. As discussed in Chapter 5, IEEE 802.11i and 802.11r take advantage of pre-authentication mechanism at layer 2. Thus, many of the guidelines observed for 802.11i-based pre-authentication and 802.11r-based fast roaming could also be applicable to the clients that use MPA-based pre- authentication techniques. However, since MPA operations are not limited to movement within a specific subnet and involve inter-subnet and inter-domain handover, these guidelines need to take into account other factors such as movement pattern of the mobile, cell size.

Time needed to complete pre-authentication mechanism is an important parameter since the mobile node needs to determine how much ahead of time the mobile needs to start the pre-authentication process so that it can finish the desired operations before the handover to the target network starts. The time needed to complete pre-authentication operations will vary depending upon the speed of the mobile (e.g., whether the mobile is moving at vehicular and pedestrian speeds), cell sizes (e.g., WiFi, Cellular). Cell residence time is defined as the average time the mobile stays in the cell before the next handoff takes place. Cell residence time is dependent upon the coverage area and velocity of the mobile. Thus, cell residence time is an important factor in determining the desirable pre-authentication time that a mobile should consider.

Since pre-authentication operation involves six sub-operations as described earlier in the chapter and each sub-operation takes some discrete amount of time, only part of these sub-operations may be completed before the handoff depending upon the available delay budget.

For example, a mobile could complete only network discovery and network layer authentication process before the handoff and postpone the rest of the operations to until after

the handover is complete. On the other hand, if it is a slow moving vehicle and the adjacent cells are sparsely spaced, a mobile could complete all the desired MPA related operations. Finishing all the MPA related operations ahead of time reduces the handoff delay but adds other constraints such as cell residence time.

I give a numerical example here for a pre-authentication process.

D= Coverage diameter,

v= Mobile's velocity,

RTT = round trip time from AP to AAA server including processing time for authentication Tauth

T_{psk} = Time spent to install keys pro-actively on the target APs

If for a given value of D = 100ft, Tpsk = 10 ms, and RTT = 100 ms, if a mobile needs to do only pre-authentication procedure associated with MPA, then the following can be calculated for a successful MPA procedure before the handoff is complete.

$2RTT+T_{psk}$ < D/v

v = 100 ft/(200 ms +10 ms) = 500 ft/sec

Similarly, for a similar cell size, if the mobile is involved in both pre-authentication and pre-configuration operations as part of the MPA procedure, and it takes an amount of time T_{config}= 190 ms to complete the layer 3 configuration including IP address configuration, then for a successful MPA operation,

$2RTT+T_{psk}+T_{config}$ < D/v

v = 100 ft /(200 ms + 10 ms + 190 ms) = 250 ft/sec

Thus, compared to only pre-authentication part of MPA operation, in order to be able to complete both pre-authentication and pre-configuration operations successfully, either the mobile needs to move at a slower pace or it needs to expedite these operations for this given cell size. Thus, the extent of MPA operations will be constrained by the velocity of the mobile.

As an alternative if a mobile does complete all the pre- authentication procedure much

ahead of time, it uses up the resources accordingly by way of reserving the IP addresses from the neighboring networks, resources for tunnel setup and additional bandwidth needed to carry these pre-authentication related signaling. Thus, during pre-authentication mechanism, there is always a trade-off between the performance benefit (e.g., low delay, less packet loss) and systems resources. This is also largely governed by network characteristics, cell size and movement speed of the mobile.

9.12 Concluding remarks

Systems evaluation of some of the indicative handoff systems described in this chapter demonstrates how many of the handoff components can work together using the proposed proactive, reactive and cross layer optimization techniques. For example, I have built the media independent pre-authentication handoff system that use several of my proposed proactive optimization techniques, namely pre-handoff triggers, proactive network discovery, network layer assisted layer 2 pre-authentication, proactive layer 3 configuration, proactive binding update and uses dynamic buffering mechanism to reduce the handoff delay and minimize the packet loss. Based on the available systems resources and the required handoff performance, an enterprise network designer or wireless service provider can pick and choose a set of optimization techniques (e.g., proactive, reactive or cross layer) for different handoff components and build their customized target mobility system. In some cases, they may have the freedom of changing the system parameters and in some cases they cannot change the system parameters or may not be able to install new network element. Thus, based on the performance requirement and specific systems limitation, a service provider may choose a specific optimization technique that would be optimal for their desired operation. The proposed mobility models can be used to build the optimized handoff systems that can predict the systems performance ahead of time before the service providers go ahead, build the actual handoff system and deploy it.

Chapter 10

Conclusions and future work

In this chapter, I discuss some of the general principles of mobility optimization that I inferred during the course of my thesis, highlight the main contribution of my thesis and potential future work that can be carried out.

10.1 General principles of mobility optimization

Currently, there are many mobility protocols available, each with its own strength and weaknesses. Each of these mobility protocols has historically evolved its own optimization techniques without regards for any generic framework. Thus, it is desirable to have a set of guidelines for the protocol designers, mobile users and architects who plan to use these mobility protocols and associated optimization techniques based on their usage requirement and applicability. In this section, I summarize the fundamental parts and factors that drive the systems optimization and describe the fundamental principles of systems optimization for mobility management. Some of these are protocol design methodologies and some are guidelines for any service provider or enterprise that may like to deploy these mobility protocols and relevant optimization techniques.

- Since current mobility protocols and the associated optimization techniques are ad hoc in nature, it is useful to have a systematic analysis of the mobility event while designing the appropriate optimization techniques.

- Since the mobility involves various layers of the protocol stack, it is important to discover the type of mobility the mobile will be subjected to, such as layer 2, layer 3 or application layer. Type of mobility will be determined based on mobile node's mobility pattern, such as cell handoff, subnet handoff or domain handoff, type of application supported on the mobile node, type of access network.

- Since layer 2 handoff optimization techniques are access dependent, it is important to consider the access characteristics of each network, such as channel access algorithm (e.g., CSMA/CA, OFDM, TDMA). For example, CDMA network will have different access characteristics than 802.11 networks. Amounts of resources used (e.g., channel bandwidth) will vary based on the types of access networks.

- Each mobility event (e.g., handoff) can be considered to consist of a set of abstract functions, such as discovery, configuration, authentication, security association, registration, binding update and media delivery. Optimization of these abstract functions can take place independent of each other but often benefit from cross layer triggers.

- A mobility event can be considered as a discrete event dynamic system (DEDS), where each of these abstract functions can be considered as a specific discrete event. Optimizing each of the discrete events can contribute to the overall systems optimization.

- Scheduling of the primitive functions that are part of these handoff events plays an important role in the overall systems behavior including systems performance and resource usage.

- Scheduling of the handoff primitives needs to take into account the data dependency among the abstract operations. Data dependency will determine the extent of parallelism that is possible during the handoff operations.

- Deadlocks need to be avoided during any mobility operation. Deadlocks are typically caused by lack of data from previous primitive operations or lack of available resources needed for an operation. Thus, the scheduling of the primitive events should ensure that there are enough resources available for any kind of parallel or speculative operations and there is availability of data.

- It is important to consider the transport type (e.g., RTP, TCP) supported by an application running on the mobile when it is subjected to handoff as each of these applications has a different performance requirement in terms of packet loss, delay and jitter.

- Since there are several mobility protocols available and each of the mobility protocols is suitable for a specific type of application (e.g, RTP- and TCP-based transport) and specific mobility pattern (e.g., layer 2 handoff, layer 3 handoff, inter-domain handoff) a policy-based mobility management scheme can be appropriate in many cases.

- Since the primitive handoff operations in each layer take place independent of the operations in other layers, cross-layer triggers from lower layers help to expedite the handoff operations in upper layers. Thus, any optimization framework needs to apply any of the available cross layer optimization techniques. IEEE 802.21 has defined one such media independent handover function that provides cross layer triggers to expedite the handover.

- It is always useful to have a handoff model that can predict the systems performance based on the schedules and available systems resources. By varying different systems parameters and resource availability, performance of the system will also vary.

Service providers can use the handoff model to determine the type of protocol and optimization technique needed for a specific scenario.

- Scheduling of handoff primitives will be largely determined by the systems resources and data dependency among the events. Since scheduling of handoff primitives affects the systems performance and it can be changed to meet the performance requirements at the cost of added systems resources.

- Scheduling of the handoff operations can also affect the trade-off between the resources expensed (e.g., battery, CPU, bandwidth) and systems performance (e.g., delay, packet loss). Thus, the types of optimization are largely determined by the extent of trade-off that can be allowed.

- In case of multi-interface mobility, a make-before-break mechanism helps to reduce the delay and packet loss at the expense of additional resources[1], since both the interfaces remain active during handoff. The extent of overlapping operations will be determined by the amount of resources that could be expensed during handoff.

- Proactive operations appear to be most attractive to provide the desired handoff performance (e.g., delay and packet loss) compared to sequential and parallel operations. However, there is a trade-off between amount of resources and performance in case of multiple target networks, since the mobile needs to complete proactive handoff related operations with multiple target networks to increase the probability of successful handover.

10.2 Summary of contribution

This thesis contributes to the general theory of optimized handover. Some of the key contributions include identification of basic properties of a mobility event, formulation of a

[1]Resources are defined in Appendix C

mobility systems model, design of optimization techniques based on some fundamental design principles of optimization and evaluation of the associated optimization techniques by means of analysis, model-based simulation, and experiments. These contribution can be summarized in following three main areas.

First, this thesis has addressed the need for a formal systems model that can characterize a mobility event and the associated mobility optimization methodologies. It provides a systematic and formal approach to a mobility event that works independent of the type of mobility protocol. After a thorough analysis of the abstract operations associated with several mobility protocols, I determined that these basic operations form a set of discrete events that can be modeled as discrete event dynamic system (DEDS). I used deterministic timed transition petri net (DTTPN) to model the mobility event and analyzed its behavioral properties and systems performance. This model has the ability to predict the systems performance based on the availability of the systems resources and the mobility pattern. Analysis of this model and the optimization methodologies will help to define a set of principles and guidelines for designing any new mobility protocol as well as evaluate the effectiveness of a specific mobility protocol in a deployment scenario.

I have developed the Petri net model that can analyze the behavioral properties such as deadlocks and validate systems performance of any type of handoff optimization supporting intra-technology, inter-technology, simultaneous mobility and multi-layer mobility. The model can also perform tradeoff analysis between the handoff performance of these optimization techniques and systems resources. The model-based approach provides the ability to define various handoff schedules under resource constraints and can determine the extent of parallelism and proactive operations that are possible among the handoff components.

Second, I have developed a series of optimization techniques (e.g., reactive, proactive and parallel) for different handoff components and have carried out extensive experiments to validate these optimization techniques. I have applied these optimization techniques to

different mobility scenarios, such as simultaneous mobility, multi layer mobility and multicast mobility, multi-interface mobility and compared the results with the non-optimized version. These series of experiments provide a systematic methodology that can be carried out in a repetitive manner and can be applied to optimize different handoff components.

Third, I have developed a hierarchical scope-based multicast architecture to support multicast streaming using proxies in the access network. This proposed architecture introduces a novel local advertisement insertion technique and program management between local program and global program. I have developed a few optimization techniques to support fast-handoff for multicast traffic in a hierarchically scope-based environment. These techniques are based on proxy-based proactive triggering and application layer triggering to expedite multicast stream delivery by reducing the "Join" latency.

10.3 Future work

My thesis laid the foundation for the systematic approach to mobility event that can be analyzed by a formal model using Petri net. However, this model-based analysis of mobility event can further be enhanced to make it more useful to the wireless community and mobility deployment. The following is a list of future work items, that I believe could be pursued beyond my thesis work.

1. Although I have used this model to validate few of the mobility optimization techniques, this model can be enhanced to study the behavioral properties and systems performance of any type of mobility protocol, such as transport layer and mobility in other types of networks such as ad hoc networks.

2. Using the current model, I was able to study and detect the behavioral properties such as systems deadlocks, investigate the anomaly of a specific schedule and then compare various schedules, such as proactive, reactive and concurrent. This model can be enhanced so that one can use it in an automated fashion to generate a specific

schedule of the handoff operations given a set of resource constraints, performance objectives and dependence graph. Automatic generation of schedules for handoff operations to meet the desired quality of services under available resources will help to use the right set of protocols.

3. Using the systematic analysis of the mobility functions, one can design a customized mobility protocol suitable to one's own set of requirements. Currently, any mobility event depends upon a set of protocols each doing its own desired functions (e.g., DHCP for IP address acquisition, server discovery). However, each of these protocols adds additional overhead when used individually. Using this model, one can design a comprehensive mobility protocol that will define its own set of protocols for each of the desired functions instead of using the existing ones.

4. Current Petri net model has looked at the resource parameters of the mobile only. This model can be enhanced to predict the performance based on the resource parameters of all the network elements that are involved in the mobility event. Other system elements may include layer 2 point of attachment (e.g., Access Point), layer 3 point of attachment (e.g., router), and servers in the network. The distributed resource metrics would be useful for any wireless service providers that may like to have an optimized service deployment.

5. The formalization of key techniques, models of systems dependencies and the ability to calculate or predict optimization metrics provide a foundation for the automated discovery and implementation of mobility optimization. I envision specification of the functional components of mobility protocols as defined in Chapter 3 and then having tools that search for application or context specific optimizations, such as caching, proactive, or cross layer techniques.

I plan to pursue some of the future work listed above beyond my thesis defense and apply the results to real life deployment.

Bibliography

[3GP08] 3GPP. 3GPP TR23893. *Feasibility Study on Multimedia Session Continuity*, June 2008.

[802a] 8021x. URL: http://www.ieee802.org/1/pages/802.1x.html.

[802b] 80221. URL: http://www.ieee802.org/21.

[A07] Third Generation Partnership Project 2 and Telecommunications Industry Association. Voice Call Continuity (VCC). *3GPP2-TIA PN-3-0231 (TIA-1093)/X. P0042*, 2007.

[AA97] Kevin Almeroth and Mahmoud Ammar. Multicast group behavior in the internet's multicast backbone (mbone). *IEEE Communications Magazine*, 35(6), June 1997.

[AAKB08] J. Arkko, B. Aboba, J. Kohonen, and F. Bari. Network discovery and selection problem. RFC 5113, Internet Engineering Task Force, January 2008.

[ABN95] Arup Acharya, Ajay Bakre, and Badri Nath. IP multicast extensions for mobile internetworking. Technical Report LCSR-TR-243, Department of Computer Science, Rutgers University, New Brunswick, New Jersey, April 1995.

[ABO97] W. Almesberger, Jean-Yves Boudec, and P. Oechslin. Application REQuested IP over ATM (AREQUIPA). RFC 2170, Internet Engineering Task Force, July 1997.

[ABV+04] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and Ed. Levkowetz. Extensible authentication protocol (EAP). RFC 3748, Internet Engineering Task Force, June 2004.

[Aka] Akamai. URL: http://www.akamai.com.

[Alm00] Kevin Almeroth. A long-term analysis of growth and usage patterns in the multicast backbone (mbone). In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Tel Aviv, Israel, March 2000.

[AP98] R.M. Amadio and S. Prasad. Modelling IP Mobility. In *CONCUR'98: Concurrency Theory: 9th International Conference*, Nice, September 1998. Springer.

[AS99] B. Aboba and D. Simon. PPP EAP TLS authentication protocol. RFC 2716, Internet Engineering Task Force, October 1999.

[AS04] K. Arabshian and H. Schulzrinne. Gloserv: Global service discovery architecture. In *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous)*, 2004.

[BdSEC91] F. Boussinot, R. de Simone, and V. ENSMP-CMA. The ESTEREL language. *Proceedings of the IEEE*, 79(9):1293–1304, 1991.

[BFC93] Tony Ballardie, P. Francis, and Jon Crowcroft. Core based trees (CBT). In Deepinder Sidhu, editor, *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 85–95, San Francisco, California,

September 1993. ACM. also in *Computer Communication Review* 23 (4), Oct. 1992.

[BK93] U. Belhe and A. Kusiak. Performance analysis of design process using timed petri nets. *Concurrent Engineering*, 1(3):147, 1993.

[BLFM98] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform resource identifiers (URI): generic syntax. RFC 2396, Internet Engineering Task Force, August 1998.

[BMB05] V. Brik, A. Mishra, and S. Banerjee. Eliminating handoff latencies in 802.11 WLANs using multiple radios: Applications, experience, and evaluation. In *ACM SIGCOMM IMC*, Berkeley, CA, October 2005.

[BMN+04] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The secure real-time transport protocol (SRTP). RFC 3711, Internet Engineering Task Force, March 2004.

[BS06] S.A. Baset and H. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. In *IEEE infocom*, volume 6. Citeseer, 2006.

[BSSS03] M. Bauer, P. Schefczik, M. Soellner, and W. Speltacker. Evolution of the UTRAN Architecture. In *3G Mobile Communication Technologies, 2003. 3G 2003. 4th International Conference on (Conf. Publ. No. 494)*, pages 244–248, 2003.

[BZCS96] M.G. Baker, X. Zhao, S. Cheshire, and J. Stone. Supporting mobility in MosquitoNet. In *Proceedings of the 1996 USENIX Technical Conference*, pages 127–140. Citeseer, 1996.

[CAG05] S. Cheshire, B. Aboba, and E. Guttman. Dynamic configuration of IPv4 Link-Local addresses. RFC 3927, Internet Engineering Task Force, March 2005.

[Car00] B. Carpenter. Internet transparency. RFC 2775, Internet Engineering Task Force, February 2000.

[CD98] A. Conta and S. Deering. Generic packet tunneling in IPv6 specification. RFC 2473, Internet Engineering Task Force, December 1998.

[CDK+02] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan. Internet group management protocol, version 3. RFC 3376, Internet Engineering Task Force, October 2002.

[CDS07] T. Chiba, A. Dutta, and H Schulzrinne. Trombone Routing Mitigation Techniques for IMS/MMD Networks. In *Proceedings of IEEE WCNC*, Hong Kong, March 2007.

[CES71] E.G. Coffman, MJ Elphick, and A. Shoshani. System deadlocks. *Computing Surveys*, 3(2):67–78, 1971.

[CFMBdI00] F. Cruz Filho, P. Maciel, E. Barros, and C. de Informatica. Using Petri Nets for Data Dependency Analysis. In *Systems, Man, and Cybernetics, IEEE International Conferences on*, volume 4, pages 2998–3003, Nashiville,TN,USA, 2000.

[CG91] RA Cuninghame-Green. Minimax algebra and applications. *Fuzzy Sets and Systems*, 41(3):251–267, 1991.

[CGK+00] Andrew T Campbell, Javier Gomez, Seung-Hoon Kim, Valko A. G., and Chieh-Yih Wan. Design, implementation, and evaluation of cellular IP. *IEEE Personal Communications*, 7(4):42–49, August 2000.

[CGR+00] F. Cuervo, N. Greene, A. Rayhan, C. Huitema, B. Rosen, and J. Segers. Megaco protocol version 1.0. RFC 3015, Internet Engineering Task Force, November 2000.

[CH90] X.R. Cao and Y.C. Ho. Models of discrete event dynamic systems. *IEEE Control Systems Magazine*, 10(4):69–76, 1990.

[CHK+00] P. Calhoun, T. Hiller, J. Kempf, P. McCann, C. Pairla, A. Singh, and S. Thalanany. Foreign agent assisted hand-off. Internet Draft, Internet Engineering Task Force, November 2000. Work in progress.

[Cho05] JH. Choi. Goals of detecting network attachment in IPv6. RFC 4135, Internet Engineering Task Force, August 2005.

[C.K05] Ed. C.Kaufman. Internet key exchange (ikev2) protocol. RFC 4036, Internet Engineering Task Force, December 2005.

[CLG+03] P. Calhoun, J. Loughney, Erik Guttman, G. Zorn, and J. Arkko. Diameter base protocol. RFC 3588, Internet Engineering Task Force, September 2003.

[CM03] Y. Chen and Q. Mary. Soft handover issues in radio resource management for 3g wcdma networks. *Department of Electronic Engineering Queen Mary, University of London*, September 2003.

[CMP03] P. Calhoun, G. Montenegro, and C. Perkins. Mobile IPv4 regional registration. Internet Draft draft-ietf-mobileip-reg-tunnel-08, Internet Engineering Task Force, November 2003. Work in progress.

[CNP01] Marco Carli, Alessandro Neri, and A. R. Picci. Mobile IP and cellular IP integration for inter access networks handoff. In *Conference Record*

of the International Conference on Communications (ICC), pages 2467–2471, Helsinki, Finland, June 2001.

[CPRW07] R. Chandra, J. Padhye, L. Ravindranath, and A. Wolman. Beacon-Stuffing: Wi-Fi Without Associations. In *Mobile Computing Systems and Applications, 2007. Hot Mobile 2007. Eighth IEEE Workshop on*, pages 53–57, 2007.

[CS03] M. Carson and D. Santay. NIST Net: a Linux-based network emulation tool. *ACM SIGCOMM Computer Communication Review*, 33(3):111–126, 2003.

[CT09] T. Clancy and H. Tschofenig. Extensible Authentication Protocol - Generalized Pre-Shared Key (EAP-GPSK) Method. RFC 5433, Internet Engineering Task Force, February 2009.

[CYD+08] T. Chiba, H. Yokota, A. Dutta, D. Chee, and H. Schulzrinne. Route Optimization for Proxy Mobile IPv6 in IMS Network. In *Proceedings of the 2008 International Conference on Signal Processing and Communication Systems*, 2008.

[DA99] T. Dierks and C. Allen. The TLS protocol version 1.0. RFC 2246, Internet Engineering Task Force, January 1999.

[DAC+03] Ashutosh Dutta, Prathima Agrawal, Jyh-Cheng Chen, Subir Das, David Famolari, Yoshihiro Ohba, Sinichi Baba, Anthony McAuley, and Henning Schulzrinne. Realizing mobile wireless internet telephony and streaming multimedia testbed. *Elsevier, Computer and Communication*, 27:725, October 2003.

[DACS02] Ashutosh Dutta, Onur Altintas, Wai Chen, and Henning Schulzrinne. Mobility approaches for all IP wireless networks. In *SCI*, Orlando, Florida, July 2002.

[DAD+04] A. Dutta, P. Agrawal, S. Das, M. Elaoud, D. Famolari, S. Madhani, A. McAuley, P. Li, M. Tauil, and H. Schulzrinne. Realizing mobile wireless Internet telephony and streaming multimedia testbed. *Computer Communications*, 27(8):725–738, 2004.

[DBJ+05] A. Dutta, J. Burns, R. Jain, D. Wong, K. Young, and H. Schulzrinne. Implementation and Performance Evaluation of Application layer MIP-LR. In *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, volume 2, Maui, HI, June 2005.

[DBW+02] A. Dutta, J. Burns, KD Wong, R. Jain, K. Young, H. Schulzrinne, and A. McAuley. Realization of Integrated Mobility Management Protocol for Ad-Hoc Networks. In *MILCOM*, volume 1, pages 448–454, 2002.

[DCC+03] A. Dutta, J. Chennikara, W. Chen, O. Altintas, and H. Schulzrinne. Multicasting streaming media to mobile users. *IEEE Communications magazine*, 41(10):81–89, 2003.

[DCT+07] A. Dutta, S. Chakravarty, K. Taniuchi, V. Fajardo, Y. Ohba, D. Famolari, and H. Schulzrinne. An experimental study of location assisted proactive handover. In *IEEE GLOBECOM*, Washington DC, 2007.

[DDF+05] A. Dutta, S. Das, D. Famolari, Y. Ohba, K. Taniuchi, T. Kodama, and H. Shulzrinne. Seamless handover across heterogeneous networks-an IEEE 802.21 centric approach. *Proceedings of IWS-WPMC*, 2005.

[DDF+06] Ashutosh Dutta, Subir Das, David Famolari, Yoshihiro Ohba, Kenichi Taniuchi, Victor Fajardo, Toshikazu Kodama, and Henning Schulzrinne.

Secured seamless convergence across heterogeneous access networks. In *World Telecommunication Congress*, Budapest, May 2006. IEEE.

[DDF+07] A. Dutta, S. Das, D. Famolari, Y. Ohba, and H. Schulzrinne. Seamless Proactive Handover across Heterogeneous Access Networks. *Wireless Personal Communication*, 43(3):837–855, August 2007.

[DDG04] H.H. Duong, A. Dadej, and S. Gordon. Proactive context transfer in WLAN-based access networks. In *Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, pages 61–70. ACM New York, NY, USA, 2004.

[DDL+04] Ashutosh Dutta, Subir Das, Peter Li, Anthony McAuley, Yoshihiro Ohba, Shinich Baba, and Henning Schulzrinne. Secured mobile multimedia communication for wireless internet. In *International Conference on Network Sensing and Control*, Taipei, Taiwan, March 2004. IEEE.

[DDM+02] Subir Das, Ashutosh Dutta, Anthony McAuley, Archan Misra, and Sajal Das. IDMP: an intra-domain mobility management protocol for next generation, wireless networks. *IEEE Personal Communications Magazine*, June 2002.

[DEB+03] R. Droms, I. Ed., Jim Bound, B. Volz, Todd Lemon, Charles Perkins, and Megan Carney. Dynamic host configuration protocol for IPv6 (DHCPv6). RFC 3315, Internet Engineering Task Force, July 2003.

[DEF+94] S. E. Deering, D. Estrin, Dino Farinacci, V. Jacobson, Chao-Ming Liu, and Li Wei. An architecture for wide-area multicast routing. In *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 126–135, London, UK, September 1994.

[DFD+08] A. Dutta, D. Famolari, S. Das, Y. Ohba, V. Fajardo, K. Taniuchi, R. Lopez, and H. Schulzrinne. Media-independent pre-authentication supporting secure interdomain handover optimization. *IEEE Wireless Communications*, 15(2):55–64, 2008.

[dGG+00] Cees de Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence. Generic AAA architecture. RFC 2903, Internet Engineering Task Force, August 2000.

[DH98] S. Deering and R. Hinden. Internet protocol, version 6 (IPv6) specification. RFC 2460, Internet Engineering Task Force, December 1998.

[Dig] DigiIsland. URL: http://www.digitalisland. com.

[DJT03] T. Dreibholz, A. Jungmaier, and M. Tuxen. A new scheme for IP-based Internet-mobility. In *CONFERENCE ON LOCAL COMPUTER NETWORKS*, volume 28, pages 99–108. IEEE, 2003.

[DKZ+05] A. Dutta, B. Kim, T. Zhang, T. Technologies, S. Baba, K. Taniuchi, Y. Ohba, and H. Schulzrinne. Experimental analysis of multi interface mobility management with sip and mip. In *IEEE Wireless Conference*, Maui, HI, June 2005.

[DLS+07] A. Dutta, B. Lyles, H. Schulzrinne, T. Chiba, H. Yokota, and A. Idoue. Generalized Modeling Framework for Handoff Analysis. In *Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC07)*, Athens, Greece, 2007.

[DLSW09] A. Dutta, B. Lyles, H. Schulzrinne, and J. Wang. Systems Modeling for IP-based Handoff Using Timed Petri Nets. In *IEEE International conference on Systems Sciences, HICSS 2009*, Big Island, HI, 2009.

[DMAD00] S. Das, A. Misra, P. Agrawal, and S.K. Das. TeleMIP: Telecommunications-enhanced mobile IP architecture for fast intradomain mobility. *IEEE Personal Communications*, 7(4):50–58, 2000.

[DMC+03] A. Dutta, S. Madhani, W. Chen, O. Altintas, and H. Schulzrinne. MobiCom poster: optimized fast-handoff schemes for application layer mobility management. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(1):17–19, 2003.

[DMC+04] A. Dutta, S. Madhani, W. Chen, O. Altintas, and H. Schulzrinne. Fast-handoff schemes for application layer mobility management. In *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 3. IEEE, September 2004.

[DMCS06] Ashutosh Dutta, Sunil Madhani, Wai Chen, and Henning Schulzrinne. GPS assisted fast-handoff mechanism for Real-Time communication. IEEE Sarnoff Symposium, Princeton, NJ, 2006.

[DMD+07] A. Dutta, K. Manousakis, S. Das, T. Chiba, and H. Schulzrinne. Mobilty Testbed for 3GPP2-based Multimedia Domain Networks. *IEEE Communications Magazine*, 45(7):118, July 2007.

[DMDL07] A. Dutta, K. Manousakis, S. Das, and FJ Lin. Mobility Testbed for 3GPP2-Based Multimedia Domain Networks. *IEEE Communications Magazine*, 45(7):118–126, 2007.

[DMZ+06] Ashutosh Dutta, Sunil Madhani, Tao Zhang, Yoshihiro Ohba, Kenichi Taniuchi, and Henning Schulzrinne. Network discovery mechanisms for fast-handoff. In *Third International Conference on BROADNETS*, San Jose, 2006.

[DOF+10] A. Dutta, Y. Ohba, V. Fajardo, K. Taniuchi, and H. Schulzrinne. A Framework of Media-Independent Pre-Authentication (MPA). Internet Draft draft-irtf-mobopts-mpa-framework-07, Internet Engineering Task Force, April 2010. Work in progress.

[dPPC03] J. del Prado Pavon and S. Choi. Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement. In *IEEE International Conference on Communications*, pages 1108–1113. IEEE, 2003.

[Dro97] R. Droms. Dynamic host configuration protocol. RFC 2131, Internet Engineering Task Force, March 1997.

[Dro99] R. Droms. Procedure for defining new DHCP options. RFC 2489, Internet Engineering Task Force, January 1999.

[DS01] Ashutosh Dutta and Henning Schulzrinne. A streaming architecture for next generation Internet. In *Conference Record of the International Conference on Communications (ICC)*, page 7, Helsinki, June 2001.

[DS04] A. Dutta and H. Schulzrinne. MarconiNet: overlay mobile content distribution network. *IEEE Communications Magazine*, 42(2):64–75, 2004.

[DSC+06] A. Dutta, H. Schulzrinne, T. Chiba, H. Yokota, and S. Das. Comparative Analysis of Network Layer and Application Layer IP Mobility Protocols for IPv6 Networks. In *Proceedings of WPMC 2006*, San Diego, September 2006.

[DTC+09] S. Das, M. Tauil, Y.H. Cheng, A. Dutta, D. Baker, M. Yajnik, D. Famolari, et al. IEEE 802.21: Media independent handover: Features, applicability, and realization. *IEEE Communications Magazine*, page 113, 2009.

[DVC+01] Ashutosh Dutta, Faramak Vakil, Jyh-Cheng Chen, Miriam Tauil, Shinichi Baba, and Henning Schulzrinne. Application layer mobility management scheme for wireless Internet. In *3G Wireless*, page 7, San Francisco, May 2001.

[DvF+06] Ashutosh Dutta, Eric van den Berg, David Famolari, Victor Fajardo, Yohihiro Ohba, Kenichi Taniuchi, and Henning Schulzrinne. Dynamic buffering scheme for mobile handoff. In *IEEE PIMRC*, Helsinki, 2006.

[DWB+02] Ashutosh Dutta, Daniel Wong, James Burns, Ravi Jain, Ken Young, Anthony McAuley, and Henning Schulzrinne. Realization of integrated mobility management for ad-hoc networks. In *IEEE Milcom*, Annaheim, California, October 2002.

[DWDSY07] K. Daniel Wong, A. Dutta, H. Schulzrinne, and K. Young. Simultaneous mobility: analytical framework, theorems and solutions. *Wireless Communications and Mobile Computing*, 7(5), 2007.

[DZM+04] Ashutosh Dutta, Tao Zhang, Sunil Madhani, Kenichi Taniuchi, Kensaku Fujimoto, Henning Schulzrinne, Yoshihiro Ohba, and Yasuhiro Katsube. Secure universal mobility for wireless internet. In *The Second ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, Philadelphia, PA, October 2004.

[DZM+05] A. Dutta, T. Zhang, S. Madhani, K. Taniuchi, K. Fujimoto, Y. Katsube, Y. Ohba, and H. Schulzrinne. Secure universal mobility for wireless internet. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(3):45–57, 2005.

[DZO+05] Ashutosh Dutta, Tao Zhang, Yoshihiro Ohba, Kenichi Taniuchi, and Henning Schulzrinne. MPA assisted proactive handoff scheme. In *ACM Mobiquitous*, pages 155–165. SIGMOBILE, ACM, July 2005.

[(Ed01] G. Montenegro (Ed.). Reverse tunneling for mobile IP, revised. RFC 3024, Internet Engineering Task Force, January 2001.

[Edg] Edgecast. URL: http://www.edgecast.com.

[EE04] J. Manner (Ed.) and M. Kojo (Ed.). Mobility related terminology. RFC 3753, Internet Engineering Task Force, June 2004.

[EF94] K. Egevang and P. Francis. The IP network address translator (NAT). RFC 1631, Internet Engineering Task Force, May 1994.

[Eri94] H. Eriksson. Mbone: The multicast backbone. *Communications of the ACM*, pages 54–60, 1994.

[Fen97] W. Fenner. Internet group management protocol, version 2. RFC 2236, Internet Engineering Task Force, November 1997.

[Fer94] D. A Ferguson. Measurement of mundane TV behaviors: Remote control device flipping frequency. *Journal of Broadcasting and Electronic Media*, 38:35–47, January 1994.

[FGL+99] M. Flament, F. Gessler, F Lagergren, O. Queseth, R. Stridh, M. Unbehaun, Jean-Lien C. Wu, and J. Zander. An approach to 4th generation wireless infrastructures, scenarios and research issues. In *VTC*, Houston, Texas, May 1999.

[Fin03] R. Finlayson. The UDP multicast tunneling protocol. Internet Draft draft-finlayson-umtp-09, Internet Engineering Task Force, November 2003. Work in progress.

[FJP07] E. Fogelstroem, A. Jonsson, and C. Perkins. Mobile IPv4 regional registration. RFC 4857, Internet Engineering Task Force, 2007.

[FKC+05] H. Fathi, K. Kobara, S. Chakraborty, H. Imai, and R Prasad. On the Impact of Security on the Latency in WLAN 802.11. In *IEEE Global Telecommunications Conference, 2005. GLOBECOM'05*, St Louis, MO, Nov 2005.

[Flo62] R.W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.

[FN01] L.M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *IEEE INFOCOM*, volume 3, pages 1548–1557. Citeseer, 2001.

[FOP+08] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, and A. Yegin. Protocol for carrying authentication for network access (pana). RFC 5191, Internet Engineering Task Force, May 2008.

[Fre] Free Radius. http://www.freeradius.org/.

[FS07] A.G. Forte and H. Schulzrinne. Cooperation Between Stations in Wireless Networks. In *Network Protocols, 2007. ICNP 2007. IEEE International Conference on*, pages 31–40, Beijing, China, 2007.

[FSS06] Andrea Forte, Sangho Shin, and Henning Schulzrinne. Passive duplicate address detection for dynamic host configuration protocol (DHCP). Technical Report cucs-011-06, Columbia University, Computer Science Department, March 2006.

[FT] D. Farinacci and A. Tweedly. T. Speakman," Cisco Group Management Protocol (CGMP).

[Gas05] M. Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly Media, Inc., 2005.

[Geo04] M. Georgides. Context transfer support for IP-based mobility management. In *CCSR Awards for Research Excellence*, 2004.

[GFJ03] Youngjune Gwon, Guangrui Fu, and Ravi Jain. Fast handoffs in wireless LAN networks using mobile initiated tunneling handoff protocol for ipv4 MITHv4. In *IEEE Wireless Communications and Networking*, pages 1248–1253, March 2003.

[GLD+08] S. Gundavelli, K. Leung, V. Devarapali, K. Chowdhury, and B. Patil. Proxy mobile ipv6. RFC 5213, Internet Engineering Task Force, August 2008.

[GPVD99] E. Guttman, C. Perkins, J. Veizades, and M. Day. Service location protocol, version 2. RFC 2608, Internet Engineering Task Force, June 1999.

[Gre01] M. Greis. 3GPP TS23. 107 v5. 0.0. *Quality of Service, Concept and Architecture*, 2001.

[H+03] Youngnam Han et al. Advance duplicate address detection. Internet draft, Internet Engineering Task Force, July 2003. Work in progress.

[HA07] R. Housely and B. Aboba. Guidance for authentication, authorization, and accounting (aaa) key management. RFC 4962, Internet Engineering Task Force, July 2007.

[Han96] M. Handley. The sdr session directory: An mbone conference scheduling and booking system. *URL http://ugwww. ed. ac. uk/mice/archive/sdr. html*, 1996.

[HC98] D. Harkins and D. Carrel. The internet key exchange (IKE). RFC 2409, Internet Engineering Task Force, November 1998.

[HC06] H Holbrook and B Cain. Source-Specific Multicast for IP. RFC 4607, Internet Engineering Task Force, August 2006.

[HD99] A.M. Hormozi and L.F. Dube. Establishing Project Control: Schedule, Cost, and Quality. *SAM Advanced Management Journal*, 64(4), 1999.

[HDS03] P.Y. Hsieh, A. Dutta, and H. Schulzrinne. Application Layer Mobility Proxy for real-time communication. *3G Wireless*, 2003.

[Her00] G. Herrin. Linux IP Networking: A Guide to the Implementation and Modification of the Linux Protocol Stack. *Connections*, 3:2, 2000.

[HHM06] T. Hansen, T. Hardie, and L. Masinter. Guidelines and registration procedures for new uri schemes. RFC 4395, Internet Engineering Task Force, February 2006.

[HJ98] M. Handley and V. Jacobson. SDP: session description protocol. RFC 2327, Internet Engineering Task Force, April 1998.

[HK89] S. Hares and D. Katz. Administrative domains and routing domains: A model for routing in the internet. RFC 1136, Internet Engineering Task Force, December 1989.

[Hos] Host AP software. http://hostap.epitest.fi/.

[HPW00] M. Handley, C. Perkins, and E. Whelan. Session announcement protocol. RFC 2974, Internet Engineering Task Force, October 2000.

[HTM+05] M. Hempstead, N. Tripathi, P. Mauro, G.Y. Wei, and D. Brooks. An ultra low power system architecture for sensor network applications. *ACM SIGARCH Computer Architecture News*, 33(2):208–219, 2005.

[HTT00] C. Hirel, B. Tun, and KS Trivedi. SPNP Version 6.0. *Computer performance evaluation: Modelling tools and techniques; 11th International Conference; TOOLS*, pages 354–357, 2000.

[HV87] M.A. Holliday and M.K. Vernon. A Generalized Timed Petri Net Model for Performance Analysis. *IEEE Transactions on Software Engineering*, 13:12, 1987.

[IT04] ITU-T. ITU-T X.200. *Open Systems Interconnection - Model and Notation*, July 2004.

[JC98] H. Johner and International Business Machines Corporation. *Understanding LDAP*. IBM Corporation, 1998.

[JLO08] P. Jayaraman, R. Lopez, and Y. Ohba. Protocol for carrying authentication for network access (pana) framework. RFC 5193, Internet Engineering Task Force, May 2008.

[JPA04] D. Johnson, C. Perkins, and J. Arkko. Mobility support in IPv6. RFC 3775, Internet Engineering Task Force, June 2004.

[JPA09] D. Johnson, C. Perkins, and J. Arkko. Mobility support in ipv6. Internet Draft draft-ietf-mext-rfc3775bis-05.txt, Internet Engineering Task Force, October 2009. work in progress.

[JRG+98] R. Jain, T. Raleigh, C. Graff, M. Bereschinsky, and M. Patel. Mobile Internet access and QoS guarantees using mobile IP and RSVP with location registers. In *IEEE International Conference on Communications*, volume 3, pages 1690–1695, Atlanta,GA,USA, June 1998. (IEEE).

[JRMRMT97] FJ Jaimes-Romero, D. Munoz-Rodriguez, C. Molina, and H. Tawfik. Modeling resource management in cellular systems using Petri nets. *Vehicular Technology, IEEE Transactions on*, 46(2):298–312, 1997.

[JRY+99] Ravi Jain, Thomas Raleigh, Danny Yang, L. F. Chang, C. J. Graff, M. Bereschinsky, and Minesh Patel. Enhancing survivability of mobile Internet access using mobile IP with location registers. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, New York, March 1999.

[JS00] W. Jiang and H. Schulzrinne. Modeling of packet loss and delay and their effect on real-time multimedia service quality. In *Proceedings of NOSSDAV*. Citeseer, 2000.

[JSAC01] C.E. Jones, K.M. Sivalingam, P. Agrawal, and J.C. Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7(4):343–358, 2001.

[K+03] Soo Koh et al. Use of SCTP for seamless handover. Internet draft, Internet Engineering Task Force, February 2003. Work in progress.

[KA98a] S. Kent and R. Atkinson. IP encapsulating security payload (ESP). RFC 2406, Internet Engineering Task Force, November 1998.

[KA98b] S. Kent and R. Atkinson. Security architecture for the internet protocol. RFC 2401, Internet Engineering Task Force, November 1998.

[KAQ+99] M. Khalil, H. Akhtar, E. Qaddoura, C. Perkins, and A. Cerpa. Buffer Management for Mobile IP. Technical report, IETF Internet Draftdraft-mkhalil-mobileip-buffer-00. txt, 1999.

[Kau05] C. (Ed.) Kaufman. Internet key exchange (IKEv2) protocol. RFC 4306, Internet Engineering Task Force, December 2005.

[KBR09] R. Khalaf-Bitar and I. Rubin. Throughput-Capacity and Bit-per-Joule Performance of IEEE 802.11 Based Wireless Mesh Networks. In *IEEE, Ad Hoc Networking Workshop*, pages 34–41, 2009.

[KCM01] Robin Kravets, Casey Carter, and Luiz Magalhaes. A cooperative approach to user mobility. *ACM Computer Communications Review*, 31, 2001.

[KCP01] G. Krishnamurthi, R. Chalmers, and C. Perkins. Buffer management for smooth HandOvers in mobile IPv6. Internet draft, Internet Engineering Task Force, March 2001. Work in progress.

[KD73] K. Kosugi and J. Davies. Basic Joseki, 1973.

[Kem07] J. Kempf. Goals for network based localized mobility management. RFC 4830, Internet Engineering Task Force, April 2007.

[KH04] B.S. Kim and K.J. Han. Multicast handoff agent mechanism for all-IP mobile network. *Mobile Networks and Applications*, 9(3):185–191, 2004.

[KHHK01] K.I. Kim, J.L. Ha, E.H. Hyun, and S.H. Kim. New approach for mobile multicast based on ssm. In *In the Proceedings of Ninth IEEE International Conference on Networks, 2001*, pages 405–408, October 2001.

[KK00] R. Kravets and P. Krishnan. Application-driven power management for mobile communication. *Wireless Networks*, 6(4):263–277, 2000.

[KL92] G. Klas and R. Lepold. TOMSPIN-a tool for modeling with stochastic Petri nets. *CompEuro'92.'Computer Systems and Software Engineering', Proceedings.*, pages 618–623, 1992.

[KLPK04] P.S. Kim, M.E. Lee, S. Park, and Y.K. Kim. A New Mechanism for SIP over Mobile IPv6. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 975–984, 2004.

[KMG99] Satwant Kaur, B. Madan, and S. Ganeshan. Multicast support for mobile IP using modified IGMP. In *IEEE WCNC*, pages 948–952, March 1999.

[Koo05] R. Koodli. Fast handovers for mobile IPv6. RFC 4068, Internet Engineering Task Force, July 2005.

[KS05] S. Kent and K. Seo. Security architecture for the internet protocol. RFC 4301, Internet Engineering Task Force, December 2005.

[Law01] E.L. Lawler. *Combinatorial optimization: networks and matroids*. Dover Pubns, 2001.

[LDOS07] R. Lopez, A. Dutta, Y. Ohba, and H. Schulzrinne. Network-layer assisted mechanism to optimize authentication delay during handoff in 802.11 networks. In *ACM Mobiquitous*, Philadelphia,PA, June 2007.

[LE05] J. Levon and P. Elie. Oprofile: A system profiler for linux. *Web site: http://oprofile. sourceforge. net*, 2005.

[LGL+96] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones. SOCKS protocol version 5. RFC 1928, Internet Engineering Task Force, March 1996.

[LLC03] H. Lee, S.W. Lee, and D.H. Cho. Mobility management based on the integration of mobile IP and session initiation protocol in next generation mobile data networks. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 3, 2003.

[LNPK05] J. Loughney, M. Nakhjiri, C. Perkins, and R. Koodli. Context transfer protocol. RFC 4067, Internet Engineering Task Force, July 2005.

[LS+99a] O. Lassila, R.R. Swick, et al. *Resource Description Framework (RDF) Model and Syntax Specification*. World Wide Web Consortium W3C, 1999.

[LS99b] J. Lennox and Henning Schulzrinne. Transporting user control information in SIP REGISTER payloads. Internet draft, Internet Engineering Task Force, March 1999. Work in progress.

[LSCF05] M. Liebsch, A. Singh, H Chaskar, and D. Funato. Candidate Access Router Discovery (CARD). RFC 4066, Internet Engineering Task Force, July 2005.

[LW00] Chunhung Lin and Kai-Min Wang. Mobile multicast support in IP networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Tel Aviv, Israel, March 2000.

[MAD] MADWiFi Driver. http://sourceforge.net/project/madwifi/.

[Mal04] K. Malki. Low latency handoffs in mobile IPv4. Internet Draft draft-ietf-mobileip-lowlatency-handoffs-v4-08, Internet Engineering Task Force, January 2004. Work in progress.

[Mal07] K. El Malki. Low-latency handoffs in mobile IPv4. RFC 4881, Internet Engineering Task Force, June 2007.

[MB97] Jayanth Mysore and Vaduvur Bhargavan. A new multicasting-based architecture for Internet host mobility. In *Third Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 161–172, Budapest, Hungary, September 1997.

[MB98] David A. Maltz and P. Bhagwat. MSOCKS: an architecture for transport layer mobility. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, page 1037, San Francisco, California, March/April 1998.

[MB01] A. Miu and P. Bahl. Dynamic host configuration for managing mobility between public and private networks. In *The 3rd Usenix Internet Technical Symposium*, San Francisco, CA, March 2001.

[MBM+99] Anthony McAuley, Ethan Bommaiah, A. Misra, Rajesh Talpade, Sue Thomson, and K. C. Young. Mobile multicast proxy. In *IEEE Milcom*, Atlantic City, New Jersey, November 1999.

[MC06] H. Mostafa and P. Cicak. Hands on Roaming Duration: Petri-Nets Modeling of a Wireless Mobile-IP Procedure in Cisco Platform. *Networking and Services, 2006. ICNS'06. International conference on*, pages 28–28, 2006.

[McB02] B. McBride. Jena: a semantic Web toolkit. *Internet Computing, IEEE*, 6(6):55–59, 2002.

[McB04] B. McBride. The resource description framework (rdf) and its vocabulary description language rdfs. *Handbook on Ontologies*, pages 51–66, 2004.

[MCFB01] P. Maciel, F. Cruz Filho, and E. Barros. A Petri net based method for resource estimation: an approach considering data-dependency, causal and temporal precedences. In *Integrated Circuits and Systems Design, 2001, 14th Symposium on.*, pages 78–84, Mirenopolis,Brazil, 2001.

[McG92] G. McGregor. The PPP internet protocol control protocol (IPCP). RFC 1332, Internet Engineering Task Force, May 1992.

[MDD+02] Archan Misra, Subir Das, Ashutosh Dutta, Anthony McAuley, and Sajal Das. IDMP based fast-handoff and paging in IP based 4G mobile networks. *IEEE Communications Magazine*, 40(3):138–145, March 2002.

[MMGS01] M. Marsan, M. Meo, M. Gribaudo, and M. Sereno. On petri net-based modeling paradigms for the performance analysis of wireless internet access. In *International Workshop on Petrinet and Performance Models*, Aachen,Germany, September 11-14 2001. IEEE.

[MMN05] Julien Montavont, Nicolas Montavont, and Thomas Noel. Enhanced schemes for L2 handover in IEEE 802.11 networks and their evaluations. In *16th International Symposium on Personal Indoor and Mobile Radio Communications*, Berlin, Germany, September 2005.

[MMP03] M.H. Matcovschi, C. Mahulea, and O. Pastravanu. Petri net toolbox for MATLAB. In *11th IEEE Mediterranean Conference on Control and Automation MED'03*, Rhodes,Greece, 2003.

[MMWM01] A. Mcauley, A. Misra, L. Wong, and K. Manousakis. Experience with Autoconfiguring a Network with IP addresses. In *MILCOM*, volume 1, pages 272–276, 2001.

[MN06] R. Moskowitz and P. Nikander. Host identity protocol (hip) architecture. RFC 4423, Internet Engineering Task Force, May 2006.

[Moo04] N. Moore. Edge handovers for mobile IPv6. Internet Draft draft-moore-mobopts-edge-handovers-00, Internet Engineering Task Force, February 2004. Work in progress.

[Moo06] N. Moore. Optimistic duplicate address detection DAD for IPv6. RFC 4429, Internet Engineering Task Force, April 2006.

[Moy93] John Moy. Multicast routing extensions for OSPF. In *International Networking Conference (INET)*, pages BCC–1–BCC–7, San Francisco, California, August 1993. Internet Society.

[MP00] B.A. Miller and R.A. Pascoe. Salutation Service Discovery in Pervasive Computing Environments. *IBM Pervasive Computing white paper, http://www-3. ibm. com/pvc/tech/salutation. shtml, February*, 2000.

[MRMRLM94] C. Molina-Ramirez, D. Munoz-Rodriguez, and E. Lopez-Mellado. Modelling and analysis of telecommunication cellular systems using Petri nets. Technical report, IEEE, 1994.

[MSA03] Arunesh Mishra, Minho Shin, and William Arbaugh. An empirical analysis of the IEEE 802.11 MAC layer handoff process. *ACM Computer Communication Review*, 33(2):93–102, 2003.

[MSPJ+04] A. Mishra, M.H. Shin, NL Petroni Jr, TC Clancy, and WA Arbaugh. Proactive key distribution using neighbor graphs. *IEEE Wireless communications*, 11(1):26–36, 2004.

[MSST98] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet security association and key management protocol (ISAKMP). RFC 2408, Internet Engineering Task Force, November 1998.

[Mur85] T. Murata. Use of resource-time product concept to derive a performance measure of timed Petri nets. In *Proceedings 1985 Midwest Symposium Circuits and Systems, Louisville, USA*, 1985.

[Mur89] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of IEEE*, page 541, April 1989.

[MVH+04] D.L. McGuinness, F. Van Harmelen, et al. OWL web ontology language overview. *W3C recommendation*, 10:2004–03, 2004.

[NAT02] A. Niemi, J. Arkko, and V. Torvinen. Hypertext transfer protocol (HTTP) digest authentication using authentication and key agreement (AKA). RFC 3310, Internet Engineering Task Force, September 2002.

[NDDS03] Nobuyasu Nakajima, Ashutosh Dutta, Subir Das, and Henning Schulzrinne. Handoff delay analysis and measurement for SIP based mobility in IPv6. In *ICC 2003 - Personal Communication Systems and Wireless LANs*, Anchorage,Alaska,USA, May 2003.

[NNS98] T. Narten, E. Nordmark, and W. Simpson. Neighbor discovery for IP version 6 (IPv6). RFC 2461, Internet Engineering Task Force, December 1998.

[O'H04] B. O'Hara. *IEEE 802.11 handbook: a designer's companion*. Institute of Electrical & Electronics Engineers (IEEE), 2004.

[Ope] Open Diameter. http://sourceforge.net/projects/diameter/.

[OQG09] Y. Ohba, Q.Wu, and G.Zorn. Extensible authentication protocol (eap) early authentication problem statement. Internet Draft draft-ietf-hokey-preauth-ps, Internet Engineering Task Force, July 2009. Work in progress.

[OWS+06] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey. Extended rtp profile for real-time transport control protocol (rtcp)-based feedback (rtp/avpf). RFC 4585, Internet Engineering Task Force, July 2006.

[PAGW06] T. Pering, Y. Agarwal, R. Gupta, and R. Want. Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio inter-

faces. In *Proceedings of the 4th International Conference on Mobile systems, Applications and Services*, page 232. ACM, 2006.

[Par02] J. Park. Mobile Multicast Routing Protocol: TBMOM (Timer-Based Mobile Multicast). *RMT Wksp. 2002*, may 2002.

[PC02] S. Pack and Y. Choi. Fast inter-AP handoff using predictive authentication scheme in a public wireless LAN. In *Networks: The Proceedings of the Joint International Conference on Wireless LANs and Home Networks (ICWLHN 2002) and Networking (ICN 2002): Atlanta, USA, 26-29 August 2002*, page 15. World Scientific Pub Co Inc, 2002.

[PCA+04] C. Politis, KA Chew, N. Akhtar, M. Georgiades, R. Tafazolli, and T. Dagiuklas. Hybrid multilayer mobility management with AAA context transfer capabilities for all-IP networks. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 11(4):76–88, 2004.

[Per96a] C. Perkins. IP encapsulation within IP. RFC 2003, Internet Engineering Task Force, October 1996.

[Per96b] C. Perkins. Minimal encapsulation within IP. RFC 2004, Internet Engineering Task Force, October 1996.

[Per02a] C. Perkins. IP mobility support for IPv4. RFC 3220, Internet Engineering Task Force, January 2002.

[Per02b] CE Perkins. Mobile IP. *Communications Magazine, IEEE*, 40(5):66–82, 2002.

[Per02c] Charles Perkins. IP mobility support for IPv4. RFC 3344, Internet Engineering Task Force, August 2002.

[Pet81] JL Peterson. Petri Net Theory and the Modeling of Systems. *PRENTICE-HALL, INC., ENGLEWOOD CLIFFS, NJ 07632, 1981, 290*, 1981.

[PH98] C. Perkins and O. Hodson. Options for repair of streaming media. RFC 2354, Internet Engineering Task Force, June 1998.

[PJ06] J. Peterson and C. Jennigs. Enhancements for authenticated identity management in the session initiation protocol (sip). RFC 4474, Internet Engineering Task Force, August 2006.

[PK91] EP Patsidou and JC Kantor. Application of minimax algebra to the study of multipurpose batch plants. *Computers & chemical engineering*, 15(1):35–46, 1991.

[PKB05] S. Park, P Kim, and B.Volz. Rapid commit option for the dynamic host configuration protocol version 4. RFC 4039, Internet Engineering Task Force, March 2005.

[Plu] U. Plug. Play (UPnP) Forum.

[Plu82] D. Plummer. Ethernet address resolution protocol: Or converting network protocol addresses to 48.bit ethernet address for transmission on ethernet hardware. RFC 826, Internet Engineering Task Force, November 1982.

[Pol96] GP Pollini. Trends in handover design. *Communications Magazine, IEEE*, 34(3):82–90, 1996.

[PRRJ03] N.R. Potlapally, S. Ravi, A. Raghunathan, and N.K. Jha. Analyzing the energy consumption of security protocols. In *Proceedings of the 2003 international symposium on Low power electronics and design*, pages 30–35. ACM New York, NY, USA, 2003.

[PSL04] J. Polk, J. Schnizlein, and M. Linsner. Dynamic host configuration protocol option for coordinate-based location configuration information. RFC 3825, Internet Engineering Task Force, July 2004.

[PW99] Charles E. Perkins and Kuang-Yeh Wang. Optimized smooth handoffs in mobile IP. In *Fourth IEEE Symposium on Computers and Communications (ISCC '99)*, pages 340–346, Red Sea, Egypt, July 1999.

[QA01] B. Quinn and Kevin Almeroth. IP multicast applications: Challenges and solutions. RFC 3170, Internet Engineering Task Force, September 2001.

[Rah93] M. Rahnema. Overview of the GSM system and protocol architecture. *Communications Magazine, IEEE*, 31(4):92–100, 1993.

[Ram74] C. Ramchandani. Analysis of asynchronous concurrent systems by timed Petri nets. *Ph.D dissertation*, 1974.

[RCC+04] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, and S. Banerjee. MAR: a commuter router infrastructure for the mobile internet. In *Proceedings of the 2nd international ACM conference on Mobile systems, applications, and services*, pages 217–230, Boston, MA, June 2004.

[RH80] CV Ramamoorthy and G.S. Ho. Performance evaluation of asynchronous concurrent systems using Petri nets. *IEEE Transactions on Software Engineering*, 6(5):440–449, 1980.

[RH06] Y. Rekhter and S. Hares. A border gateway protocol 4 (bgp-4). RFC 4271, Internet Engineering Task Force, January 2006.

[RKL+04] I. Romdhani, M. Kellil, H.Y. Lach, A. Bouabdallah, and H. Bettahar. IP mobile multicast: Challenges and solutions. *IEEE Communications Surveys & Tutorials*, 6(1):18–41, 2004.

[RL03] P. Roshan and J. Leary. *802.11 wireless LAN fundamentals*. Cisco press, 2003.

[RL04] T. Ruckforth and J. Linder. AAA context transfer for fast authenticated inter-domain handover. *Swisscom SA, Mar*, 2004.

[RLS+00] Ramachandran Ramjee, Thomas F. LaPorta, Luca Salgarelli, Sandra Thuel, Kannan Varadhan, and Li (Erran) Li. IP-based access network infrastructure for next-generation wireless networks. *IEEE Personal Communications Magazine*, 7(4):34–41, August 2000.

[RPSC04] J. Rosenberg, J. Peterson, Henning Schulzrinne, and G. Camarillo. Best current practices for third party call control (3pcc) in the session initiation protocol (SIP). RFC 3725, Internet Engineering Task Force, April 2004.

[RS99] J. Rosenberg and Henning Schulzrinne. An RTP payload format for generic forward error correction. RFC 2733, Internet Engineering Task Force, December 1999.

[RSC+02] J. Rosenberg, Henning Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. RFC 3261, Internet Engineering Task Force, June 2002.

[Ryu04] N. Ryuther. A tool for PetriNet Simulation, 2004. http://www.informatik.uni-hamburg.de/TGI/PetriNets.

[S+96] H. Schulzrinne et al. Personal Mobility for Multimedia Services in the Internet. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 143–162, 1996.

[San95] WH Sanders. UltraSAN Users Manual Version 3.0. *Center for Reliable and High-Performance Computing, University of Illinois*, 1995.

[SB00] Alex C. Snoeren and Hari Balakrishnan. An end-to-end approach to host mobility. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 155–166, Boston, Massachusetts, USA, August 2000.

[SC] Henning Schulzrinne and Columbia University IRT Lab. rtptools: Tools for RTP. http://www.cs.columbia.edu/IRT/software/rtptools/.

[SCeMB06] H. Soliman, C. Castelluccia, K. el Malki, and L. Bellier. Hierarchical mobile IPv6 mobility management HMIPv6. RFC 4140, Internet Engineering Task Force, August 2006.

[SCFJ03] Henning Schulzrinne, Stephen Casner, Ron Frederick, and Van Jacobson. RTP: a transport protocol for Real-Time applications. RFC 3550, Internet Engineering Task Force, July 2003.

[Sch] H. Schulzrinne. Irt/software/rtptools.

[Sch01] Henning Schulzrinne. SIP registration. Internet draft, Internet Engineering Task Force, April 2001. Work in progress.

[SE06] Ed. S. Eronen. IKEv2 mobility and multihoming protocol. RFC 4555, Internet Engineering Task Force, June 2006.

[SE09] N. SALAWU and N.O. Elizabeth. Energy Optimization Mechanism for Mobile Terminals using Vertical Handoff between WLAN and CDMA2000 Networks. volume 15, pages 51–58, 2009.

[Sea04] A. Seaborne. Rdql-a query language for rdf. *W3C Member submission*, 9:29–1, 2004.

[SH08] Henning Schulzrinne and Robert Hancock. GIMPS: general internet messaging protocol for signaling. Internet Draft draft-ietf-nsis-ntlp-15, Internet Engineering Task Force, February 2008. Work in progress.

[SHC+04] V. Shnayder, M. Hempstead, B. Chen, G.W. Allen, and M. Welsh. Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 188–200. ACM New York, NY, USA, 2004.

[SHE+04] Mortaza S. Bargh, Bob Hulsebosch, Henk Eertink, Anand Prasad, Hu Wang, and Peter Schoo. Fast authentication methods for handovers between IEEE 802.11 wireless LANs. In *The Second ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, Philadelphia, PA,, October 2004.

[SHK+95] M. Angela Sasse, V. J. Hardman, I. Kouvelas, C. E. Perkins, O. Hodson, A. I. Watson, Mark Handley, and Jon Crowcroft. RAT (robust-audio tool), 1995.

[Sim94] W. Simpson. The Point-to-Point protocol (PPP). RFC 1661, Internet Engineering Task Force, July 1994.

[Sim05] N. Simulator. 2 (NS2). *URL: http://www. isi. edu/nsnam*, 2005.

[SK98] M. Stemm and R.H. Katz. Vertical handoffs in wireless overlay networks. *Mobile Networks and applications*, 3(4):335–350, 1998.

[SLGW01] R. Steele, J. Li, P. Gould, and J. Wiley. *GSM, cdmaOne, and 3G systems*. John Wiley New York, 2001.

[Spa03] R. Sparks. The session initiation protocol (SIP) refer method. RFC 3515, Internet Engineering Task Force, April 2003.

[SRL98] Henning Schulzrinne, Asha Rao, and R. Lanphier. Real time streaming protocol (RTSP). RFC 2326, Internet Engineering Task Force, April 1998.

[SSFR04] Sangho Shin, Henning Schulzrinne, Andrea Forte, and Anshuman Rawat. Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs. In *ACM MobiWac (ACM International Workshop on Mobility Management and Wireless Access)*, Philadelphia, Pennsylvania, September 2004. ACM.

[SSK00] H.S. Shin, Y.J. Suh, and D.H. Kwon. Multicast routing protocol by multicast agent in mobile networks. In *Proceedings of the Proceedings of the 2000 International Conference on Parallel Processing*, page 271. IEEE Computer Society Washington, DC, USA, 2000.

[SSTK07] R. Shacham, H. Schulzrinne, S. Thakolsri, and W. Kellerer. Ubiquitous device personalization and use: The next generation of IP multimedia communications. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 3(2):12, 2007.

[SSTK08] R. Shacham, H. Schulzrinne, S. Thakolsri, and W. Kellerer. Session initiation protocol (sip) session mobility. Internet Draft draft-shacham-sipping-session-mobility-05, Internet Engineering Task Force, May 2008. Work in progress.

[Sta04] W. Stallings. *IEEE 802.11: Wireless LANs from a to n*. IEEE Computer Society, 2004.

[SW00] Henning Schulzrinne and Elin Wedlund. Application-layer mobility using SIP. *Mobile Computing and Communications Review (MC2R)*, 4(3):47–57, July 2000.

[SXM+00] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, and M. Kalla. Stream control transmission protocol. RFC 2960, Internet Engineering Task Force, October 2000.

[Tac02] K. Tachikawa. *W-CDMA Mobile Communications System*. Wiley, 2002.

[TAG01] S. Tilak and N.B. Abu-Ghazaleh. A concurrent migration extension to an end-to-end host mobility architecture. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(3):26–31, 2001.

[TGM+08] F. Teraoka, K. Gogo, K. Mitsuya, R. Shibui, and K. Mitani. " Unified Layer 2 (L2) Abstractions for Layer 3 (L3)-Driven Fast Handover. Request for Comments 5184, Internet Engineering Task Force, May 2008.

[Tha04] D Thaler. Border Gateway Multicast Protocol. RFC 3913, Internet Engineering Task Force, September 2004.

[TIAa] TIA. Telecommunications Industry Association.

[TIAb] TIA. Tr-45 mobile and personal communications systems standards.

[Tim00] O.W.T. Time. ITU-T Recommendation G. 114. *ITU-T May*, 2000.

[TLP99] C.L. Tan, K.M. Lye, and S. Pink. A fast handoff scheme for wireless networks. In *Proceedings of the 2nd ACM international workshop on Wireless mobile multimedia*, page 90. ACM, 1999.

[TN98] S. Thomson and T. Narten. IPv6 stateless address autoconfiguration. RFC 2462, Internet Engineering Task Force, December 1998.

[TP00a] Cheng Lin Tan and Stephen Pink. Mobicast: A multicast scheme for wireless networks. *Mobile Networks and Applications*, 5(4):259–271, January 2000.

[TP00b] C.L. Tan and S. Pink. MobiCast: A multicast scheme for wireless networks. *Mobile Networks and Applications*, 5(4):259–271, 2000.

[TP01] Antti J. Tuominen and Henrik L. Petander. MIPL mobile IPv6 for linux in HUT campus network mediapoli. In *Proceedings of Ottawa Linux Symposium, 2001*, Ottawa, Canda, June 2001.

[TP06] S. Thajchayapong and J.M. Peha. Mobility patterns in microcellular wireless networks. *IEEE Transactions on Mobile Computing*, pages 52–63, 2006.

[TRV98] Nishit Tripathi, Jeffrey Reed, and Hugh F. VanLandingham. Handoff in cellular systems. *IEEE Personal Communications Magazine*, 5(6), December 1998.

[TS01] D. Tutsch and J. Sokol. Petri Net based Performance Evaluation of USAIAs Bandwidth Partitioning for the Wireless Cell Level. *Proceedings of the 9th international Workshop on Petri Nets and Performance Models PNPM'01*, September 2001.

[TTAV07] D. Thaler, M. Talwar, A. Aggarwal, and L. Vicisano. T. Pusateri," Automatic IP Multicast Without Explicit Tunnels (AMT)", draft-ietf-mboned-auto-multicast-08. txt (work in progress), 2007.

[TV+02] D. Thaler, L. Vicisano, et al. IPv4 automatic multicast without explicit tunnels (AMT). Internet draft, Internet Engineering Task Force, April 2002. Work in progress.

[TYCH05] C.C. Tseng, L.H. Yen, H.H. Chang, and K.C. Hsu. Topology-aided cross-layer fast handoff designs for IEEE 802.11/mobile IP environments. *Communications Magazine, IEEE*, 43(12):156–163, 2005.

[UCB] UCB/LBNL. vic – video conferencing tool. http://www-nrg.ee.lbl.gov/vic/.

[UCL] UCL Multimedia. Robust audio tool (RAT). http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/.

[Val99] A.G. Valkó. Cellular IP: a new approach to Internet host mobility. *ACM SIGCOMM Computer Communication Review*, 29(1):50–65, 1999.

[VBPV93] H. Van Brussel, Y. Peng, and P. Valckenaers. Modelling flexible manufacturing systems based on Petri nets. *CIRP annals*, 42(1):479–484, 1993.

[VC99] Upkar Varshney and Samir Chatterji. Architectural issues in IP multicasting over wireless networks. In *IEEE conference on Wireless Communication and Networking Conference*, New Orleans, June 1999. IEEE.

[VFBF01] Faramak Vakil, David Famolari, Shinichi Baba, and David Famolari. Virtual soft hand-off in IP-Centric wireless CDMA networks. In *3G Wireless 2001*, San Francisco, May 2001. Delson.

[VK04] Hector Velayos and Gunnar Karlsson. Techniques to reduce IEEE 802.11b handoff time. In *Wireless Networking Symposium*, Paris, France, June 2004.

[VM98] Jon-Olov Vatn and Gerald (Chip) Maguire. The effect of using co-located care-of addresses on macro handover latency. In *14th Nordic Tele-traffic Seminar*, Technical University of Denmark, Lyngby, Denmark, August 1998.

[VNJ90] N. Viswanadham, Y. Narahari, and TL Johnson. Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net

models. *IEEE Transactions on Robotics and Automation*, 6(6):713–723, 1990.

[Vog06] C. Vogt. A Comprehensive Delay Analysis for Reactive and Proactive Handoffs with Mobile IPv6 Route Optimization. *Institute of Telematics, Universitaet Karlsruhe (TH), Karlsruhe, Germany, TM-2006-1*, 2006.

[Wal99] J. Waldo. The Jini architecture for network-centric computing. *Communications of the ACM*, 42(7):76–82, 1999.

[Wan98] J. Wang. *Timed Petri nets: Theory and application*. Kluwer Academic Pub, 1998.

[WCL+02] C.H. Wu, A.T. Cheng, S.T. Lee, J.M. Ho, and D.T. Lee. Bi-directional route optimization in mobile IP over wireless LAN. In *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*, volume 2, 2002.

[WD05] KD. Wong and A. Dutta. Simultaneous mobility in MIPv6. In *2005 IEEE International Conference on Electro Information Technology*, page 5, May 2005.

[WDB+03] Daniel Wong, Ashutosh Dutta, Jim Burns, Ken Young, and Henning Schulzrinne. A multilayered mobility management scheme for auto-configured wireless IP networks. *IEEE Wireless Communication Magazine*, 10(5), October 2003.

[WDSY03] Daniel Wong, Ashutosh Dutta, Henning Schulzrinne, and Kenneth Young. Managing simultaneous mobility of IP hosts. In *IEEE International Military Communications Conference (MILCOM)*, Boston, MA, USA, October 2003. IEEE, IEEE.

[WDY+03] KD. Wong, A. Dutta, K. Young, H. Schulzrinne, and T. Technol. Managing simultaneous mobility of IP hosts. In *IEEE Military Communications Conference, 2003. MILCOM 2003*, volume 2, 2003.

[WHK97] M. Wahl, T. Howes, and S. Kille. Lightweight directory access protocol (v3). RFC 2251, Internet Engineering Task Force, December 1997.

[WHMB98] Carey Williamson, T. Harrison, Wayne L. Mackrell, and R. B. Bunt. Performance evaluation of the MoM mobile multicast protocol. *ACM Mobile Networks and Applications (MONET) Journal*, 3(2):189–201, August 1998.

[Wil00] Beau Williamson. *Developing IP Multicast Networks: The definitive Guide to Designing and Deploying CISCO IP Multicast Networks*, volume Volume 1. Cisco Press, San Francisco, January 2000.

[WL97] D. Wong and T.J. Lim. Soft handoffs in CDMA mobile systems. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 4(6):6–17, 1997.

[Won02] KD Wong. Architecture alternatives for integrating Cellular IP and Mobile IP. In *Performance, Computing, and Communications Conference, 2002. 21st IEEE International*, pages 197–204, Phoenix,AZ,USA, April 2002.

[WPD88] D. Waitzman, C. Partridge, and S. Deering. Distance vector multicast routing protocol. RFC 1075, Internet Engineering Task Force, November 1988.

[WS99] Elin Wedlund and Henning Schulzrinne. Mobility support using SIP. In *2nd ACM/IEEE International Conference on Wireless and Mobile Multimedia (WoWMoM)*, Seattle, Washington, August 1999.

[WSJW02] J. Wang, L. Sun, X. Jiang, and Z. Wu. IGMP snooping: a VLAN-based multicast protocol. In *High Speed Networks and Multimedia Communications 5th IEEE International Conference on*, pages 335–340, 2002.

[Wu99] Jean-Lien C. Wu. An IP mobiity support architecture for 4GW wireless infrastructure. In *PCC*, November 1999.

[WWD+02] K. Wong, Hung-Yu Wei, Ashutosh Dutta, Kenneth Young, and Henning Schulzrinne. IP micro-mobility management using host-based routing. In Sudhir Dixit and Ramjee Prasad, editors, *Wireless IP and building the Mobile Internet*. Artech House, 2002.

[XP97] George Xylomenos and George C. Polyzos. IP multicast for mobile hosts. *IEEE Communications Magazine*, 35(1), January 1997.

[Yem83] Y. Yemini. A bang-bang principle for real-time transport protocols. *ACM SIGCOMM Computer Communication Review*, 13(2):262–268, 1983.

[YIHK02] H. Yokota, A. Idoue, T. Hasegawa, and T. Kato. Link layer assisted mobile IP fast handoff method over wireless LAN networks. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 131–139. ACM New York, NY, USA, 2002.

[YP01] SJ Yang and SH Park. An Efficient Multicast Routing Scheme for Mobile Hosts in IPv6-Based Networks. *J. Institute of Elec. Eng. Korea*, 38(8):11–18, 2001.

[Zai94] M. Zaid. Personal mobility in PCS. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 1(4), 1994.

[ZD91] M. Zhou and F. DiCesare. Parallel and sequential mutual exclusions for petri net modeling ofmanufacturing systems with shared resources. *IEEE Transactions on Robotics and Automation*, 7(4):515–527, 1991.

[ZGFH99] A. Zimmermann, R. German, J. Freiheit, and G. Hommel. Timenet 3.0 tool description. In *Int. Conf. on Petri Nets and Performance Models (PNPM'99), Zaragoza, Spain*, 1999.

[ZK99] W. M. Zuberek and W. Kubiak. Timed petri nets in modeling and analysis of simple schedules for manufacturing cells. *Elsevier Science, Computers and Mathematics with Applications*, 37, 1999.

[ZLS+05] S. Zhuang, K. Lai, I. Stoica, R. Katz, and S. Shenker. Host Mobility Using an Internet Indirection Infrastructure. *Wireless Networks*, 11(6):741–756, 2005.

[ZMD+05] T. Zhang, S. Madhani, A. Dutta, Eric VanDenberg, Y. Ohba, K. Taniuchi, and S. Mohanty. Implementation and evaluation of autonomous collaborative discovery of neighboring networks. In *Information Technology: Research and Education, 2005. ITRE 2005. 3rd International Conference on*, pages 12–17, 2005.

[ZR94] M.C. Zhou and A.D. Robbi. Applications of Petri net methodology to manufacturing systems. *Computer Control of Flexible Manufacturing Systems, Chapman and Hall*, pages 207–230, 1994.

[ZRS99] A. Zimmermann, D. Rodriguez, and M. Silva. Modelling and Optimisation of Manufacturing Systems: Petri Nets and Simulated Annealing. In *Proceedings of the 1999 European Control Conference ECC99*, Karlsruhe, 1999.

[ZSDR04] S. Zeadally, F. Siddiqui, N. DeepakMavatoor, and P. Randhavva. SIP and mobile IP integration to support seamless mobility. In *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, volume 3, 2004.

[Zub80] WM Zuberek. Timed Petri nets and preliminary performance evaluation. In *Proceedings of the 7th annual symposium on Computer Architecture*, pages 88–96. ACM New York, NY, USA, 1980.

[Zub91] WM Zuberek. Timed Petri nets definitions, properties, and applications. *Microelectronics and Reliability*, 31(4):627–644, 1991.

[Zub00] WM Zuberek. Hierarchical analysis of manufacturing systems using Petri nets. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 4, Nashville, TN, USA, October 2000.

[ZV99] M.C. Zhou and K. Venkatesh. *Modeling, simulation, and control of flexible manufacturing systems: a Petri net approach*. World Scientific Pub Co Inc, 1999.

Appendix A

RDF schema for application layer discovery

I now illustrate sample RDF schema that can be used for information discovery. For the sake of brevity I provide only a subset of the schema. I show examples that include combination of basic and extended sets of classes and their associated properties. For example a network class will have properties of type L2 and L3. An L2 class will have properties such as network-id, operator, location and neighbor information.

Schema primitives

I present sample primitives in ASN.1 format that can be transported as part of RDF schema.

```
<?xml version=''1.0''?>
<!DOCTYPE rdf:RDF [
 <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
 <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
 <!ENTITY mihbase 'URL_TO_BE_ASSIGNED'>
 ]>

<rdf:RDF xmlns:rdf=''&rdf;'' xmlns:rdfs=''&rdfs;''
xmlns:mihbase=''&mihbase;''
 xml:base=''&mihbase;''>

<rdfs:Class rdf:ID=''Network''>
 <rdfs:subClassOf rdf:resource=''&rdfs;Resource''/>
<rdfs:comment>

Network class has two properties, namely l2 for layer-2 information
and l3 for higher-layer information. Any property can be added to this
class in an extended schema.
  <rdfs:comment/>
</rdfs:Class>
<rdf:Property rdf:ID=''l2''>
 <rdfs:domain rdf:resource=''#Network''/>
 <rdfs:range rdf:resource=''#L2''/>
 <rdfs:comment>
This property is of type L2 class.
 <rdfs:comment/>
</rdf:Property>

<rdf:Property rdf:ID=''l3''>
 <rdfs:domain rdf:resource=''#Network''/>
 <rdfs:range rdf:resource=''#L3''/>
 <rdfs:comment>
This property is of type L3 class.
 <rdfs:comment/>
</rdf:Property>

<rdfs:Class rdf:ID=''L2''>
 <rdfs:subClassOf rdf:resource=''&rdfs;Resource''/>
 <rdfs:comment>
L2 class has properties that are specific to link-layer. The
properties include network-id, operator, location and
neighbor-information properties. Any property can be added to this
class in an extended schema.
 <rdfs:comment/>
</rdfs:Class>
<rdf:Property rdf:ID=''operator''>
 <rdfs:domain rdf:resource=''#L2''/>
 <rdfs:range rdf:resource=''&rdfs;Literal''/>
 <rdfs:comment>
This property contains a name of the operator. It could be the same as
network-id property.
 <rdfs:comment/>
</rdf:Property>
<rdf:Property rdf:ID=''network-id''>
 <rdfs:domain rdf:resource=''#L2''/>
 <rdfs:range rdf:resource=''&rdfs;Literal''/>
 <rdfs:comment>
This property contains an identifier of the network. It may contain an
SSID.
<rdfs:comment/>
</rdf:Property>
</rdf:RDF>
```

Figure A.1: Sample RDF Schema for Information Service

```
Network ::= ENUMERATED{L2info, L3info, Location}
L2info ::= ENUMERATED {802.11, 802.16, GSM, GPRS, W-CDMA, cdma2000}
L3info ::= ENUMERATED {IPv4, IPv6}
Location ::= SEQUENCE {
                        Geo-location ::= String
                        Civic-addr ::= String
                       }
802.11 ::= SEQUENCE {
                     Standards ::= BITMAP{802.11a, 802.11b, 802.11g}
                     SSID_Network_Name ::= String(SIZE(1..32))
                     BSSID ::= NumericString(SIZE(6))
                     Channel ::= INTEGER
                     Phy ::= ENUMERATED{CCK, DSSS, OFDM}
                     Data_Rates ::= INTEGER
                     Network_Service_Provider_Code ::= String
                     Network_Service_Provider_Name ::= String
                     Network_Service_Provider_Tariff ::= String
                     Cipher Suites ::= BITMAP {WEP, TKIP, AES-CCMP}
     Authenticated_Key_Management_Suites::=BITMAP{WEP,Psk,802.1x}
           KeyManagementProtocol::=ENUMERATED{11i4WayHandshake}
         Quality_of_Service ::= ENUMERATED {802.11e}
         Cost ::= INTEGER
         Roaming_List ::= String
         Mobility ::= ENUMERATED {802.11r, 802.11u, 802.21, PreAuth}
                    }
IPv4 ::= SEQUENCE{
               Router_Address ::= String
                     DHCP_Server_Address ::= String
                     DomainName ::= String
                     Subnet ::= String
                     SIP_Server_Address ::= String
    KeyManagementProtocol ::= ENUMERATED{IKEv1, IKEv2}
                     Authentication ::= ENUMERATEED{PANA, UAM}
                     PacketCiphering ::= ENUMERATED{IPsec}
                     Internet_Service_Provider_Code ::= String
                     Internet_Service_Provider_Name ::= String
                     Internet_Service_Provider_Tariff ::= ???
                     Mobility ::= ENUMERATED {MIPv4, CT, CARD,
       Preauth}
Quality_of_Service ::= ENUMERATED{...}
                     VPN_Gateway_Address ::= String
                     NAT_Address ::= String
                    }
MIPv4 ::= SEQUENCE{
                    HomeAgent_Address ::= String
                    ForeignAgent_Address ::= String }
PANA ::= SEQUENCE{
                    PAA_Address ::= String
                    EP_Address ::= String  }
```

Figure A.2: RDF Schema of ASN.1 primitives

Appendix B

Glossary

Glossary

1G First generation cellular network. 1G networks are based on analog systems meant to carry voice only. These were developed around 1980. NMT, AMPS, and TACS are some example 1G systems.

2G Second generation cellular network. 2G networks are an evolution of 1G networks that were introduced during 1990s. 2G networks are digital in nature and provide per-user bandwidth up to 144 kb/s. GSM, IS-54/136 and IS-95 are some example 2G systems.

3G Third generation cellular network. 3G networks can provide per-user bandwidth up to 2 Mb/s and can carry multimedia traffic. WCDMA and CDMA2000 are some example 3G systems.

3GPP Third Generation Partnership Project. Collaborative effort by a group of telecommunications associations to define the standards for 3G networks and development of W-CDMA/UMTS.

3GPP2 Third Generation Partnership Project 2. The standards body and organization used to coordinate the development of 3G networks based on CDMA2000.

4G Fourth generation cellular network. 4G networks are an evolution of 2G and 3G cellular networks and are being defined as part of IMT-2000 and can provide per-user bandwidth up to 100 Mb/s.

AAA Authentication, Authorization and Accounting. AAA is a generic model for IP network access control, initiated and developed by the IETF [dGG+00].

AH Authentication Header. AH is a component of the IPSec protocol suite [KS05] that guarantees connectionless integrity and data origin authentication of IP datagrams.

AKA Authentication and Key Agreement. The AKA process is a challenge-response based mechanism that aims at mutual network/terminal authentication and security key distribution [NAT02].

AMT Automatic Multicast Tunneling. Automatic Multicast Tunneling (AMT) allows multicast communication amongst isolated multicast-enabled sites or hosts, attached to a network which has no native multicast support [TTAV07].

ANSI American National Standard Institute. ANSI is responsible to oversee the development of voluntary consensus standards for products and services in the United States.

ARP Address Resolution Protocol. It is the process of finding out a host's link layer address when only a network layer address is given [Plu82].

AuC Authentication Center. The authentication center is a database used to control the authentication process and compare the users' identification with those recognized as valid by the network in GSM and UMTS network.

AVP Attribute Value Pair. AVP is a fundamental data representation in computing systems and applications. It is a data structure that allows for future extension without modifying existing code or data.

B2BUA Back-to-Back User Agent. B2BUA consists of two SIP user agents where one can initiate the call and the other user agent modifies and terminates the call.

B2BUA can act as a third party call controller and can establish call between two user agents.

BCCH Broadcast Control Channel. The BCCH is a point to multi-point, unidirectional downlink channel used in the GSM cellular standard.

BCE Binding Cache Entry. The relationship between a Mobile Node's home address and its current care-of address is known as a binding. All Nodes participating in MIPv6 are required to maintain a table of these bindings in a binding cache.

BCP Buffer Control Protocol. Using the buffer control protocol [DvF+06], the mobile node communicates with the buffer nodes in the network to reduce the packet loss during handoff by adjusting the buffer value dynamically.

BGR Border Gateway Router. BGR acts as an entry point to the autonomous system (AS) and uses border gateway protocol (BGP) [RH06].

BN Buffering Node. The buffering node is a logical entity in the network that allows the buffering of packets during a handoff.

BSC Base Station Controller. Part of the network that controls one or more base stations and interfaces with the switching center (e.g., MSC in GSM network).

BSS Base Station Sub-System. The overall system that encompasses the BTS and BSC and takes care of handling traffic and signaling between mobile phone and network switching subsystem. The BSS is typically used in 2G and 3G networks.

BSSID Basic Service Set IDentifier. Uniquely identifies each basic service set. The BSSID is the MAC address of the 802.11 wireless access point.

BTS Base Transceiver Station. The base station equipment used to transmit and receive signals to and from the mobile handsets.

CARD Candidate Access Router Discovery. A protocol [LSCF05] that provides network discovery mechanism at layer 3 by way of signaling exchanges between the routers at previous network and target network.

CDMA Code Division Multiple Access. Wireless access mechanism defined for 2G and 3G networks.

CDN Content Distribution Network. A content distribution network (CDN) is a system of computers containing copies of data, placed at various points in a network so as to maximize bandwidth for access to the data from clients throughout the network.

CGMP Cisco Group Management Protocol. CGMP [FT] is a Cisco proprietary group management protocol that manages the multicast groups at layer 2.

CoTI Care-of-Test Init. In MIPv6, a mobile node uses the Care-of Test Init (CoTI) message to initiate the return routability procedure and request a care-of keygen token from a correspondent node.

CS Circuit Switch domain. The CS domain is the subset of the 2G/GSM and 3G/UMTS Core network domain dedicated to the support of circuit-based services such as voice calls.

CSMA/CA Carrier Sense Multiple Access/Collision Avoidance. A mobile uses this mechanism to get access to the IEEE 802.11 type networks.

CTN Candidate Target Network. CTN is considered as one of the possible network attachment points where the mobile might move to.

DAD Duplicate Address Detection. This is a process of verifying the uniqueness of layer 3 identifier, which is the IP address in a subnet. This is often carried out during layer 3 configuration process [NNS98].

DCDP Dynamic Configuration Distribution Protocol. DCDP is a protocol that works in conjunction with DRCP to configure the servers with a block of addresses that can be distributed to the end clients [MMWM01].

DEDS Discrete Event Dynamic System. A DEDS [CH90] may be viewed as a sequence of events. The completion of an activity may initiate one or more new activities. Moreover, the order of sequence of events is not necessarily unique.

DFA Domain Foreign Agent. DFA [TP00a] is a hierarchical mobility agent that sits above the subnet level joins the multicast tree on behalf of the mobile host. Mobile host communicates with DFA using unicast.

DMSP Designated Multicast Service Provider. To avoid tunnel convergence problem a selection is performed by the FA to appoint one HA as the DMSP for a specific multicast group.

DMZ Demilitarized Zone. In computer security, DMZ is a physical or logical subnetwork that contains and exposes an organization's external services to larger, untrusted network, usually the Internet.

DRCP Dynamic Rapid Configuration Protocol. It is a light weight version of DHCP that reduces the number of messages over the air and message size, thereby reducing the configuration time [MMWM01].

DTTPN Deterministic Timed Transition Petri net. Defines a type of timed transition petri net where each of the transitions is associated with deterministic firing time [RH80].

EAP Extensible Authentication Protocol. An authentication framework [ABV+04] that supports multiple authentication methods and can run directly over data link layers such as PPP or IEEE 802 without requiring IP.

EPC Evolved Packet Core. See the definition of SAE.

EPS Evolved Packet System. EPS gives a system view of evolution of UMTS.

ESN Electronic Serial Number. A 32 bit identifier mainly used with AMPS, TDMA and CDMA phones in the United States, compared to IMEI numbers used by all GSM phones.

ESP Encapsulating Security Payload. ESP [KA98a] is a member of the IPSec protocol suite that provides origin authenticity, integrity and confidentiality protection of packets.

FA Foreign Agent. Acts as a decapsulation agent in Mobile IPv4 network.

FACH Forward Access CHannel. It is a downlink access channel that carries control information to terminals known to be located in the given cell in a GSM network.

FEC Forward Error Correction. Forward error correction (FEC) is a system of error control for data transmission, whereby the sender adds redundant data to its messages. This allows the receiver to detect and correct errors without the need for retransmission.

FMIPv6 Fast Mobile IPv6. Optimized version of mobile IPv6 protocol that reduces the handoff delay and packet loss by using layer 3 optimization techniques [Koo05].

FMS Flexible Manufacturing Systems. A flexible manufacturing system (FMS) is a manufacturing system in which there is some amount of flexibility that allows the system to react in case of changes, whether predicted or unpredicted.

FOCC FOrward Control Channel. Used to send signaling messages from a base station to one or multiple mobiles. This is used for 1G networks.

GCoA Global Care of Address. This is the care of address assigned to the mobile when it first enters into a specific domain. It is used for intra-domain mobility protocols such as IDMP [DDM+02].

GFA Gateway Foreign Agent. A hierarchical mobility agent that helps to reduce binding update signaling delay.

GGSN Gateway GPRS Support Node. It acts as a gateway between external packet data networks and a GSM network that supports GPRS.

GIST General Internet Signaling Protocol. It is an IETF protocol [SH08] being developed under NSIS (Next Steps in Signaling) working group defines a common messaging layer that provides a common service for diverse signaling applications.

GPRS General Packet Radio Service. An evolution of GSM whereby packet switching rather than circuit switching is used to provide increased bandwidth.

GPSK Generalized Pre Shared Key. EAP-GPSK is lightweight and seeks to minimize round trips [CT09]. Hence, it is well suited for any type of device, especially those with processing power, memory, and battery constraints.

GRE Generic Routing Extension. A tunneling protocol that can encapsulate variety of network layer protocol packet types inside IP tunnels.

GSM Global System for Mobile communication. It is a 2G cellular standard based on digital TDMA system operating on 200 kHz RF channel characteristics.

GTP GPRS Tunneling Protocol. Is a group of IP-based communication protocols used to carry GPRS within GSM and UMTS networks. GTP is used to carry user data between SGSN, GGSN and between RAN (Radio Access Network) and Core network.

HA Home Agent. An encapsulating agent in the home network [Per02a].

HAWAII Handoff-Aware Wireless Access Internet Infrastructure. Is a micro-mobility protocol designed to take care of mobility when a mobile's movement is confined within a domain.

HIP Host Identity Protocol. A mobility protocol that acts as shim layer between network layer and transport layer to provide the desired mobility functions [MN06].

HLR Home Location Register. A central database within the GSM or CDMA network that contains the information about the mobiles that subscribe to that particular network.

HoTI Home Test Init. As part of return routability procedure in mobile IP, a mobile node sends a Home Test Init message to the correspondent node (via the home agent) to acquire the home keygen token.

HSDPA High-Speed Downlink Packet Access. Is a high-speed enhancement of 3G/UMTS networks for network-to-terminal transmission.

IDEN Integrated Dispatch Enhanced Network, is a mobile telecommunication technologies developed by Motorola which provides its users the benefits of a trunked radio and cellular telephone. IDEN places more users in a spectral space compared to cellular systems and uses time division multiple access (TDMA).

IDMP Intra Domain Mobility Protocol. IDMP [DDM+02] is mobility protocol that optimizes mobility within an administrative domain.

IGMP Internet Group Multicast Protocol. IGMP is used between the host and the router and is used to manage the multicast groups [CDK+02].

IKE Internet Key Exchange. It is the protocol [HC98] used to set up a security association (SA) in the IPSec protocol suite. IKE uses a Diffie-Hellman key exchange to set up a shared session secret, from which cryptographic keys are derived.

IMC Internet Multimedia Client. These are end clients capable of receiving multicast traffic in a hierarchical scoped multicast environment [DS01].

IMEI International Mobile Equipment Identity. A 56 bit number that is unique to every GSM phone. Used by GSM network to identify the valid mobile devices.

IMS IP Multimedia Subsystems. IMS is a 3GPP framework, designed for delivering IP multimedia services to end-users over 3G and 4G network.

IMSI International Mobile Subscriber Identity. A serial number (normally contained within a SIM) that identifies the subscriber in a GSM and UMTS network. An IMSI is usually fifteen digits long. The first three digits are the mobile country code, and the next digits are the mobile network code (MNC).

IPCP Internet Protocol Control Protocol. The IP Control Protocol (IPCP) [McG92] is responsible for configuring, enabling, and disabling the IP protocol modules on both ends of the point-to-point (PPP) link.

IS-95 Interim Standard Number 95. IS-95 is the first CDMA-based digital cellular standard pioneered by Qualcomm. It is also known as CDMAOne or TIA (Telecommunication Industry Associations)-EIA (Electronic Industries Alliance)-95.

ISAKMP Internet Security Association and Key Management Protocol. The Internet Security Association and Key Management Protocol (ISAKMP) [MSST98] defines the procedures for authenticating a communicating peer, creation and

management of Security Associations, key generation techniques, and threat mitigation (e.g., denial of service and replay attacks).

LAI Location Area Identity. Internationally unique identifier comprised of a three decimal digit country code, two to three digit mobile network code, and a location area code. LAI is broadcast regularly by broadcast control channel (BCCH). Mobile station stores LAI in the subscriber identity module (SIM).

LCoA Local Care-of-Address. A new local care of address is assigned to the mobile when it moves between subnets within a mobility domain for both MIPv4 and MIPv6 networks.

LDAP Lightweight Directory Access Protocol. Developed within IETF [WHK97], it is an application protocol for querying and modifying directory services running over TCP/IP.

LMA Local Mobility Agent. Local Mobility Anchor is the home agent for the mobile node in a Proxy Mobile IPv6 domain [GLD+08].

LTE Long Term Evolution. LTE is the fourth generation radio technologies that has resulted out of enhancements to UMTS. It is being defined as part of 3GPP standards process. LTE uses OFDM for the downlink that is, from the base station to the terminal. In the uplink, LTE uses a pre-coded version of OFDM called Single Carrier Frequency Division Multiple Access (SC-FDMA).

M-TMSI M-Temporary Mobile Subscriber Identity. This is used to identify the UE within the MME (Mobility Management Entity).

MA Mobility Agent. MA is an anchor agent closer to the mobile and provides similar functionality as the home agent in MIP.

MAG Mobility Access Gateway. Mobile Access Gateway is a functional component on an access router that manages the mobility-related signaling for a mobile node that is attached to its access link.

MAHT Maximum Acceptable Handoff Time. Defines the amount of media interruption that a mobile can withstand during handoff. Depends upon the type of application.

MAP Mobile Application Part. MAP is the application-layer of SS7 (Signaling System Number 7) protocol that is used for communication among several components of GSM network such as HLR, VLR, MSC, AuC.

MICS Media Independent Command Service. As one of the key functional components of MIHF [DTC+09], MICS commands are used to gather information about the status of the connected links as well as to execute higher layer mobility and connectivity decisions to the lower layers. MICS is useful to optimize the handover among heterogeneous access networks.

MIES Media Independent Event Service. As one of the key functional components of MIHF, MIES provides services to the upper layers by reporting both local and remote events such as Link_Down or Link_Going_Down.

MIHF Media Independent Handover Function. Defines a set of abstract functions that help the mobility protocols to achieve seamless handover between heterogeneous access networks. MIHF was defined by the IEEE 802.21 WG.

MIIS Media Independent Information Service. Media Independent Information Service is one key component of MIHF. It provides a framework and corresponding mechanisms by which an MIHF entity can discover and obtain network information within a geographic area. MIIS has been developed by the 802.21 WG as part of MIHF.

MIN Mobile Identification Number. 10 digit identifier for a mobile subscription. MIN is stored in a database managed by the cellular provider.

MIP Mobile IP. It is a standard developed within IETF that takes care of session continuity of IP-based sessions for the mobile users (see [Per02c], Section 2.6.1.).

MIP-LR Mobile IP with Location Register. A modified version of Mobile IP [JRY+99] that provides survivability and eliminates triangle routing of the data path.

MIP-RO Mobile IP with Route Optimization. A modified version of Mobile IP that reduces the data transfer delay by way of direct path (see Section 2.6.1).

MIP-RR Mobile IP with Regional Registration. An optimized version of Mobile IP that reduces the binding update delay by confining the binding update [Per02b].

MME Mobility Management Entity. MME is an entity part of the EPC network that is in charge of session and user-mobility management.

MMP Micro-mobility Management Protocol. MMP includes a suite of protocols that take care of mobility within an administrative domain [WWD+02].

MOM Mobile Multicast [WHMB98]. MOM is a home subscription-based multicast mobility protocol to reduce the tunnel convergence.

MPA Media Independent Pre-authentication. Defines a framework that allows a mobile to pre-authenticate itself with the target network when the mobile is still in the serving network.

MSC Mobile Switching Center. The switching center where the mobile network interfaces to the public telephone system.

NAMONC Network Assisted MObile and Network Controlled. A method [Mal04] where a mobile host is assisted by the network about the impending layer 2 handoff.

NAT Network Address Translation. Network address translation (NAT) is the process of modifying network address information in datagram packet headers while in transit across a traffic routing device for the purpose of remapping a given address space into another [EF94].

NIMOT Network Initiated, Mobile Terminated. A network node initiates the handoff by sending the handoff triggers to the mobile node.

NMT Nordic Mobile Telephony. NMT is based on analog technology (first generation standard) specified by Nordic countries and was commercially deployed in 1979.

NUD Neighbor Unreachability Detection. NUD is a process of figuring out the existence of the neighbors on a link in IPv6 networks. Usually a mobile uses Neighbor Solicitation [NNS98] to find out the existence of neighbors. In case a neighbor is not found, it is deleted from the neighbor cache entry.

OWL Web Ontology Language. The OWL Web Ontology Language [MVH+04] is designed for use by applications that need to process the content of information instead of just presenting information to humans.

PANA Protocol for carrying Authentication to Network Access. An application layer protocol [JLO08] defined within the IETF that is used to authenticate a user independent of the access network.

PBU Proxy Binding Update. A request message sent by a mobile access gateway to a mobile node's local mobility anchor for establishing a binding between the mobile node's home network prefix(es) assigned to a given interface of a mobile node and its current care-of address.

PC Pilot Channel. It acts as a timing beacon for the system and is used by MSs

in the neighboring cells to assess the suitability of the cell for handover. Pilot channels are used for CDMA networks.

PCF Packet Control Function. This is an entity in a radio access network that controls the transmission of packets between the base station and packet data serving node in a CDMA 2000 network.

PDIF Packet Data Inter-working Function. Acts like an access router in an IP-based non-cellular network.

PDN-GW Packet Data Network. PDN-GW is part of EPC (Evolved Packet Core) and terminates the SGi interface towards the packet network.

PDSN Packet Data Serving Node. Layer 3 point of attachment in CDMA Network.

PHT Proactive Handover Tunnel. This is a tunnel between the mobile node and the router in the target network to complete the proactive handover. Traffic flows through the proactive handover tunnel even before the handover is complete.

PS Packet Switch. The PS domain is the subset of the 2G/GSM and 3G/UMTS core network domain dedicated to the support of packet-based services.

PTK Pairwise Transient Key. PTK is a collection of four keys referred to as temporal keys that are derived from PMK every time a mobile device associates to the access point. These keys are used to encrypt data and protect it from modification.

RA Router Advertisement. The ICMP router discovery messages are called Router Advertisements. Each router periodically multicasts a Router Advertisement from each of its multicast interfaces, announcing the IP address(es) of that interface.

RAN Radio Access Network. It controls and terminates radio signals in 3G and 4G networks.

RAS Radio Antenna Server. Radio Antenna Servers [DS01] are the content servers towards the edge of the networks that are capable of translating streaming content from a globally scoped multicast address to a locally scoped multicast address.

RAT Radio Access Technology. Refers to the type of cellular access that is in use. GSM, W-CDMA and CDMA2000 are all different radio access technologies.

RBMOM Range based Mobile Multicast [LW00]. RBMOM improves upon the performance of MOM by dynamically assigning Designated Multicast Service Provider (DMSP).

RDF Resource Description Framework. A general method for conceptual description or modeling of information that is implemented in web resources; using a variety of syntax formats. RDF is a family of W3C specifications [LS+99a].

RDFS RDF Schema [McB04].

RDQL RDF Data Query Language. RDQL is an SQL-like RDF query language derived from Squish.

RHT Reactive Handover Tunnel. This is a tunnel set up between the router in the previous network and the mobile's new care-of-address so that the transient in-flight traffic can be forwarded to the mobile after the handover. Reactive handover tunnel is used to reduce the packet loss due to handoff.

RNC Radio Network Controller. The RNC is the network element responsible for the control of the radio resources of UTRAN [BSSS03].

RSC Radio Station Client. These are the multicast traffic sources [DS01] in the hierarchical scoped multicast environment.

S-GW Serving Gateway. The serving GW is part of the EPC. It is the functional network entity that forwards and routes packets between eNodeB and Packet Data Network (PDN).

SA Security Association. A security association establishes the shared security information between two network entities to support secure communication.

SAE System Architecture Evolution. SAE defines the packet core of 4G network where LTE is defined as the RAN. SAE is also known as EPC.

SAT Supervisory Audio Tone. One of three tones, at 5970 Hz, 6000 Hz, or 6030 Hz, is transmitted by the base station and repeated back by the mobile.

SC Sync Channel. It allows the MS to achieve time synchronization with the BS and the network.

SCTP Stream Control Transport Protocol. SCTP [SXM+00] is a transport layer protocol serving in a similar role as TCP and UDP. SCTP places messages and control information into separate chunks (data chunks and control chunks), each identified by a chunk header.

SGSN Serving GPRS Support Node. Keeps track of the location information and security information associated with the MSs that are within its service area.

SID System Identifier. System identifier is used by the mobile in a cellular network to ensure that it is on the correct network.

SMR Specialized Mobile Radio is a two-way radio system in which two or mobile transceivers are linked by a single repeater.

SMS-SC Short Message Service-Service Center.

SNR Signal-to-Noise Ratio. SNR is the ratio of the signal to the background noise and is used as an indication of signal quality. It can be used as a measure of receiver performance.

SPARQL SPARQL Protocol and RDF Query Language. It is an RDF Query language and is considered a key semantic web technology.

SPI Security Parameter Index. The SPI is the 32-bit value used to distinguish among different SAs terminating at the same destination and using the same IPSec protocol [KA98b].

SRTP Secure RTP. A profile of the Real-time Transport Protocol (RTP), which can provide confidentiality, message authentication, and replay protection to the RTP traffic and to the control traffic for RTP, the Real-time Transport Control Protocol (RTCP) [BMN+04].

SSID Service Set IDentifier. It identifies a particular 802.11 wireless LAN. The SSID can be up to 32 characters long. The SSID is defined as a sequence of 1-32 octets each of which may take any value.

SSM Source Specific Multicast. Multicast protocol mostly used for multicasting content from one source to many users.

TACS Total Access Communication System. TACS is 1G cellular standard based on analog system used in European countries. TACS was first used in the United Kingdom in 1985 and then later used in other countries including Hong Kong and Japan. Although in Japan it is called JTAC.

TAI Tracking Area Identity. This identifies a tracking area in a PLMN (Public land mobile network).

TDMA Time Division Multiple Access. It allows several users to share the same frequency channel by dividing the signal into different time slots.

TELEMIP Telecommunications-Enhanced Mobile IP. It is a mobility management protocol to take care of mobility within a domain [DMAD00].

TLS Transport Layer Security. The TLS protocol [DA99] allows client/server applications to communicate across a network in a way designed to prevent eavesdropping, tampering, and message forgery. TLS is usually implemented on top of any of the Transport Layer protocols, encapsulating the application specific protocols such as HTTP, FTP, SMTP, NNTP, SIP, and XMPP.

TLV Type-Length-Value. Any information within any communication protocol can be encoded as a type-length-value where the type and length are of fixed size, whereas the value is of variable size.

TMSI A temporary Mobile Subscriber Identity. A temporary identifier assigned to a subscriber by a network.

UMTP UDP Tunneling Multicast Tunneling Protocol. The 'UDP Multicast Tunneling Protocol' (UMTP) enables such a host without multicast connectivity an ad hoc connection to the MBone by tunneling multicast UDP datagrams inside unicast UDP datagrams.

URI Uniform Resource Identifier. URI is an addressing technology that consists of a string of characters used to identify or name a resource on the Internet. URIs were originally defined as two types: Uniform Resource Locators (URLs) which are addresses with network locations, and Uniform Resource Names (URNs), which are persistent names that are address independent [HHM06]).

UTRAN UMTS Terrestrial Radio Access Network. UTRAN represents the access network of a 3G UMTS network.

VCC Voice Call Continuity. 3GPP standards to maintain the call continuity between cellular and IP-based networks.

VLR Visitor Location Register. VLR is a database that holds the information of all the mobiles that are visitors and are under one visited MSC (V-MSC) in a GSM network.

VPN Virtual Private Network. The concept of VPN is to superimpose a private network on top of a public network so that one can get the advantages of a dedicated network. VPNs are often installed by organizations to provide remote access to a secure organizational network.

WCDMA Wideband Code Division Multiple Access. A system based around direct sequence spread spectrum that enables multiple users to access a cellular channel simultaneously.

WPA Wi-Fi Protected Access. It is built around RSN (Robust Security Network). This is an interim standard before IEEE 802.11i was standardized.

XML Extensible Markup Language. XML is a textual data format with a strong support via Unicode for the languages of the world. The Extensible Markup Language (XML) is a subset of SGML (Standard Generalized Markup Language)that can be served, received and processed on the web as easily as HTML.

Appendix C

Definition of mobility related terms

It is useful to define the mobility related terms that are often used in the context of handover and optimization. In this section, I introduce a few of the relevant mobility related terms including some of the relevant ones from RFC 3753 [EE04] and ITU-T X.200 [IT04] that are useful for handover optimization and have been used in different chapters of my thesis.

Definition 5 *A Mobile Node (MN)* is a node that is capable of changing its point of attachment to the network across layers, namely layer 2 or layer 3.

A mobile node may either be a mobile host (no forwarding functionality) or a mobile router (with forwarding functionality). A mobile node can have multiple interfaces.

Definition 6 *An interface identifier* is a unique identifier that is assigned to a specific interface of a node by which it can be addressed for data transfer.

Interface identifier can be a layer 2 identifier or layer 3 identifier depending upon the specific configuration. For example, a MAC address is defined to be a layer 2 identifier, whereas an IP address is a layer 3 identifier.

Definition 7 *Device identifier* is a unique identifier that is fixed and permanent and is assigned to a device during its manufacturing.

A typical device identifier could be the MAC address of the IP-based device or electronic serial number (ESN) in case of a cellular device. In case of multiple interfaces assigned to a device, each interface will have a unique MAC address.

Definition 8 *Network Point of Attachment (PoA)* is defined as the remote endpoint of the link that connects a mobile node to the network associated with a unique link layer or layer 3 identifier. These can be defined as L2 PoA or L3 PoA, respectively.

A point-of-attachment can be a layer 2 point of attachment or layer 3 point of attachment. For example, an 802.11 access point or CDMA base station can be defined as a layer 2 point of attachment, whereas a router is considered to be a layer 3 point of attachment. Old point of attachment (oPoA) is defined as the point of attachment of the mobile node prior to the link-switch. New point of attachment (nPoA) is the new point of attachment of the mobile node that results from the link switch event.

Definition 9 *Path identification* defines the unique connection path between the mobile and the point of attachment on the network.

A path identifier is defined as the pair of interface identifiers, where one of them is associated with the mobile and the other one is associated with the network point-of-attachment in the network.

Definition 10 *Radio cell* defines a geographical area within which an access point provides radio coverage, i.e., where radio communication between a mobile node and the specific access point is possible.

A mobile node uses the radio cell to communicate with the point-of-attachment in the network.

Definition 11 *A subnet* is a logical group of connected network nodes, where two hosts can communicate through a layer-2 connection and do not require a layer-3 entity.

In IP networks, nodes in a subnet share a common network mask (in IPv4) or a network prefix (in IPv6). All hosts within a subnet can be reached in one hop, implying that all hosts in a subnet are connected to the same logical layer 3 link.

Definition 12 *An access router* resides on the edge of an access network and connected to one or more access points.

An access router offers IP connectivity to mobile nodes, acting as a default router to the mobile nodes it is currently serving. The access router may include functionality beyond a simple forwarding service offered by ordinary IP routers, such as buffering, tunneling and caching.

Definition 13 *An Access Network (AN)* is an IP network which includes one or more Access Network Routers.

An access network consists of multiple subnets that could be connected to the same access router or multiple routers.

Definition 14 *An administrative domain* is a collection of networks under the same administrative control and grouped together for administrative purposes.

Definition 15 *Serving Access Router (SAR)* is the access router that currently offers the connectivity to the mobile node.

This is usually the point of departure for the mobile node as it makes its way towards a new Access Router (at which time the Serving Access Router takes the role of the previous access router). There may be several serving access routers serving the mobile node at the same time.

Definition 16 *The Next Access Router (NAR)* is the the access router that offers connectivity to the Mobile Node after a handover.

Definition 17 *Previous Access Router (PAR)* is an access router that offered connectivity to the Mobile Node prior to a handover. This is the serving access router that will cease or has ceased to offer connectivity to the Mobile Node.

Definition 18 *Candidate Access Router (CAR)* is an Access Router to which the Mobile Node may do a handoff.

Definition 19 *Anchor MSC* i*s the MSC from which a handover has been initiated.*

During a handover in GSM network, the anchor MSC stays in the datapath to help reduce the packet loss in case of handoff forward and handoff backward.

Definition 20 *Target MSC* i*s the MSC toward which a handover should take place.*

Target MSC directs base station to assign channel for the mobile that is intended to move to that BS's radio coverage.

Definition 21 *Handover* is the process by which an active mobile node changes its point of attachment to the network, or when such a change is attempted.

The access network may provide features to minimize the interruptions in progress which is termed as optimized handover. Handover term is often interchanged with handoff.

Definition 22 *Layer 2 handover* is a type of handover where the MN changes access points (or some other aspect of radio channel) connected to the same access router's interface.

This type of handover is transparent to the routing at the IP layer.

Definition 23 *Layer 3 handover* is a type of handover where the mobile node changes access points (or some other aspect of radio channel) connected to different subnetwork interface.

The subnetwork interface may be connected to two different interfaces on the same router or two different routers altogether. During this type of handover mobile node moves to a new subnet.

Definition 24 *Roaming* involves formal agreements between operators that allows a mobile to get connectivity from a foreign network.

Roaming includes the functionality by which the users can send their identity to the local Access Network so that inter-AN agreements can be activated and service and applications in the MN's home network can be made available to the user in the visited network.

Definition 25 *Systems resources* are resources on the mobile host that are shared by different elementary operations during a handoff event.

These resources could include mobile node resources such as CPU cycles, battery power and network resources such as amount of bytes that can be transmitted per unit time on a specific access channel (e.g., per user bandwidth).

Definition 26 *Network resource parameters* are the parameters that are needed to perform different handoff related operations.

These resource parameters could include different wireless parameters, such as channel number, frequency, authentication algorithm, and authentication server. A mobile node uses these parameters to carry out the handoff related operations.

Definition 27 *Care-of-Address (CoA)* is an IP address associated with a mobile node while visiting a foreign link; the subnet prefix of this IP address is a foreign subnet prefix.

A packet addressed to the mobile node which arrives at the mobile node's home network when the mobile node is away from home and has registered a care-of Address will be forwarded to that address by the home agent in the home network.

Definition 28 *Home Address (HoA)* is an IP address assigned to a mobile node, used as the permanent address of the mobile node.

This address is within the mobile node's home link. Standard IP routing mechanisms will deliver packets destined for a mobile node's home address to its home link.

Definition 29 *Binding* can be defined as an association when (N)-entities support connectionless-mode by maintaining a binding with the appropriate (N)-SAPs for delivering the connectionless data to the (N+1)-entities (ITU-T X.200).

For example, in case of mobility, a binding can be an association between the temporary Care-of-Address obtained in the visited node and the permanent home address of the mobile node so that the data from the correspondent node can be routed to the mobile node.

Definition 30 *Encapsulation* is the process of adding control information as the header to any protocol data frame often for routing purposes. All the control overhead and data of that protocol is considered as data after the encapsulation.

IP-IP encapsulation often used in Mobile IP or ESP encapsulation used in IPSec are examples of encapsulation.

Definition 31 *Home Agent* is a router on a mobile node's home link with which the mobile node has registered its current care-of-address.

When the mobile node is away from home, the home agent intercepts packets on the home link destined to the mobile node's home address, encapsulates them and tunnels them to the mobile node's registered care-of-address.

Definition 32 *Foreign Agent* is a node on a mobile node's visited network that intercepts the traffic from the home agent and delivers it to the mobile node connected to the same visited node.

When the mobile node is connected to a visited network, it can use foreign agent as the care-of-address so that packets destined to the mobile node can be captured by the foreign agent and delivered to the mobile node in the visited network.

Definition 33 *Encapsulation agent* encapsulates the data and sends it to the decapsulation agent that removes the header and delivers the data to the target node.

Home Agent is an encapsulation agent that encapsulates the data traffic using source and destination IP headers and sends it to the Foreign Agent (FA) that delivers the packet to the target node.

Definition 34 *Decapsulation agent* communicates with the encapsulation agent and decapsulates the data by stripping the header.

A Foreign Agent is a type decapsulation agent that decapsulates the IP-IP tunnel and delivers to the mobile host.

Definition 35 *A Binding Update (BU)* message indicates a mobile node's current mobility binding, and in particular its care-of-address.

The mobile can send its binding update either directly to the other communicating node or to the home agent to associate its home address with the care-of-address.

Definition 36 *Tunneling* is a process of setting up a point-to-point virtual link between two end points by adding encapsulation header on the data so that it can be routed properly over a tunnel.

A Home agent uses tunneling mechanism to send the data from the home network to the foreign agent or to the mobile node in the visited network.

Definition 37 *A Bidirectional tunnel* is a tunnel between two communicating nodes where the data is sent over the tunnel in either direction.

In case of bi-directional tunnel mode operation for mobile IP, both the home agent and foreign agent act like encapsulation and decapsulation agent.

Definition 38 *Horizontal handover* involves mobile nodes moving between access points of the same type (in terms of coverage, data rate and mobility), such as UMTS to UMTS or 802.11 to 802.11.

A horizontal handover is also called as intra-technology handover.

Definition 39 *Vertical handover* involves mobile nodes moving between the access points of different type, such as UMTS to 802.11 or vice-versa.

A vertical handover can also be defined as Inter-technology handover.

Definition 40 *Mobile-initiated handover* requires that the mobile node is the one that makes the initial decision to initiate the handover.

Definition 41 *Network-initiated handover* requires that the network makes the initial decision to initiate the handover.

Definition 42 *Handover latency* is the difference between the time a mobile node is last able to send and/or receive an IP packet by way of the PAR (Previous Access Router), and the time the mobile node is able to send and/or receive an IP packet through the NAR (Next Access Router).

Definition 43 *Smooth handover* is a type of handover when there is no session interruptions when the mobile is in transition from one base station to another.

Smooth handover aims primarily to minimize packet loss, with no explicit concern for additional delays in packet forwarding.

Definition 44 *Fast handover* aims primarily to minimize handover latency, with no explicit interest in packet loss.

However, minimizing handover latency typically results in reduction of packet loss.

Definition 45 *Macro mobility* is defined as when a mobile moves between two networks that belong to two different subnets. These subnets may belong to the same administrative domain or different ones.

Layer 3 mobility support and associated address registration procedures are needed when a mobile node is subjected to macro mobility. Inter-Access Network (AN) handovers typically involve macro-mobility protocols. Mobile-IP can be seen as a means to provide macro mobility solution.

Definition 46 *Micro mobility* usually means movement of a terminal within a mobility domain, where a mobility domain maybe confined to a subnet or collection of subnets within an administrative domain.

Micro-mobility protocols exploit the locality of movement by confining movement related changes and signaling to the access network without having the need to interact with Mobile IP. Examples of some well known IP micro mobility architecture include HAWAII (Handoff-aware Wireless access Internet infrastructure), Cellular IP and HMIP (Hierarchical Mobile IP).

Definition 47 *Fast handover* aims primarily to minimize packet loss, with no explicit concern for additional delays in packet loss.

Definition 48 *During a Make-before-break handover* the mobile node prepares the new connection path before the old one is broken.

Thus, if the mobile has multiple interfaces, it can communicate with both the old AR and new AR at the same time using either of the interfaces. However, only one interface is used for transmitting data while the other interface is engaged in handover preparation.

Alternatively, many of the handover related operations for the second interface can be performed by the first interface. When the mobile has a single interface, a virtual interface is used to enable make-before-break handoff.

Definition 49 *During a Break-before-make handover* The mobile node breaks the existing connection before the new connection is made.

In break-before-make handover case, there is an appreciable amount of handover delay as the second interface comes up only after the first interface is disconnected. In case of handover involving single interface, the same interface gets configured again in the new network.

Definition 50 *Intra-domain handover* refers to a handover scenario where a mobile node's movement is confined to the domain.

A domain can be defined as an administrative domain, DNS-based domain or mobility domain anchored by a mobility agent. There are several types of domain defined by Hares and Katz [HK89].

Definition 51 *Inter-domain handover* refers to a handover scenario where a mobile node moves between two domains.

These two domains can be two different administrative domains, each with its own anchor mobility agent.

Definition 52 *Route optimization* is a process of minimizing the data path between the communicating nodes by having a direct path for data forwarding.

When the mobile node changes its network identifier, data gets rerouted to the new point of attachment and may take a longer route. Route optimization helps to maintain the direct path between the communicating nodes.

Definition 53 *Network Assisted, Mobile and Network Controlled (NAMONC)* handoff allows the MN to be involved in an anticipated IP-layer handoff procedure.

The Mobile Node is therefore assisted by the network in performing an anticipated L3 handoff before it completes the L2 handoff.

Definition 54 *Network Initiated Mobile Terminated (NIMOT)* handoff allows the network to initiate the handoff and register proactively on behalf of the mobile.

This handoff method enables a rapid establishment of service at the new point of attachment so that the effect of the handoff on real-time applications is minimized.

Definition 55 *Network Controlled Handoff (NCHO)* involves the network to handle the necessary RSS measurements and handoff decision.

NCHO is used in first generation cellular systems such as Advanced Mobile Phone System (AMPS) where the mobile telephone switching office (MTSO) is responsible for overall handoff decision.

Definition 56 *Mobile Assisted Handoff (MAHO)* involves the mobile to do RSS measurements and send them periodically to BS.

MAHO is used in Global System for Mobile Communications (GSM), where based on the received measurements, the BS or the mobile switching center (MSC) describes when to handoff.

Definition 57 *Mobile Controlled Handoff (MCHO)* MCHO extends the roles of the MS by giving overall control to it.

The MS and BS both, make the necessary measurements and BS sends them to the MS. Then MS decides when to handoff based on the information gained from the BS and itself.

Definition 58 *During a connected state* a mobile is in the process of actively receiving data from the correspondent node.

After all the handoff operations are complete, the mobile node goes into a connected mode.

Definition 59 *During a disconnected state* a mobile does not receive any data from the correspondent node.

A mobile node is said to be in disconnected mode when it is in the process of handing off to a new point-of-attachment in the network.

Definition 60 *Join latency* is defined as the elapsed time between a host joining the group and the router sending a multicast packet towards the mobile.

A mobile node can send unsolicited join request to the router to trigger the multicast flow failing which it can wait to respond to the router's query.

Definition 61 *Leave latency* is defined as the time between the moment the last host leaves a group and when the routing protocol is notified that there are no more multicast members.

To reduce the leave latency, the last mobile node to leave the group sends group leave message to all router multicast address so that the router prunes the multicast tree. In the absence of specific group leave report from the mobile node e.g., in case of IGMPv1 (Internet Group Management Protocol), the router needs to wait until it does not get any response from any of the clients.

Appendix D

List of publication

Mobility Fast-Handoff - Techniques, Systems prototype and Experiments

1. **Ashutosh Dutta**, Henning Schulzrinne, Sunil Madhani, Onur Altintas, and Wai Chen. Optimized fast-handoff schemes for application layer mobility management. *Mobile Computing and Communications Review (MC2R)*, ACM MC2R, Vol 7, Issue 1, January 2003.

2. Nobuyasu Nakajima, **Ashutosh Dutta**, Subir Das, and Henning Schulzrinne. Handoff delay analysis and measurement for SIP based mobility in IPv6. In *ICC 2003 - Personal Communication Systems and Wireless LANs*, Anchorage,Alaska,USA, May.

3. P-yu Hsieh, **A. Dutta**, H. Schulzrinne, "Application Layer Mobility Proxy for Real-time communication," 3G Wireless, May 2003, San Francisco.

4. Archan Misra, Subir Das, **Ashutosh Dutta**, Anthony McAuley, and Sajal Das. IDMP based fast-handoff and paging in IP based 4G mobile networks. *IEEE Communications Magazine*, 40(3):138–145, March 2002.

5. **A. Dutta**, J. Chennikara, W. Chen, O. Altintas, H. Schulzrinne, "Multicasting streaming media to mobile users," IEEE Communication Magazine, October 2003 Issue

6. **Daniel Wong**, Ashutosh Dutta, Jim Burns, Ken Young, and Henning Schulzrinne. A multilayered mobility management scheme for auto-configured wireless IP networks. *IEEE Wireless Communication Magazine*, 10(5), October 2003.

7. **A. Dutta**, S. Madhani, W. Chen, O. Altintas, H. Schulzrinne. *GPS assisted Fast-handoff Mechanism for Real-Time Communication*, IEEE Sarnoff Symposium, April 2006, Princeton.

8. **A. Dutta**, S. Das, P. Li, A. McAuley, Y. Ohba, S. Baba, H. Schulzrinne, "Secured Mobile Multimedia Communication for Wireless Internet," IEEE ICNSC 2004, Taipei, Taiwan

9. **A. Dutta** and H. Schulzrinne. MarconiNet: overlay mobile content distribution network. *IEEE Communications Magazine*, 42(2):64–75, 2004.

10. **A. Dutta**, S. Madhani, W. Chen, O. Altintas, and H. Schulzrinne. Fast-handoff schemes for application layer mobility management. In *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 3. IEEE, September 2004.

11. **A. Dutta**, T. Zhang, S. Madhani, K. Taniuchi, K. Fujimoto, Y. Katsube, Y. Ohba, H. Schulzrinne, "Secure Universal Mobility for Wireless Internet," ACM WMASH 2004, Philadelphia.

12. K.D. Wong and **A. Dutta**. Simultaneous mobility in MIPv6. In *2005 IEEE International Conference on Electro Information Technology*, page 5, May 2005. **(Best Paper Award)**.

13. **A. Dutta**, T. Zhang, S. Madhani, K. Taniuchi, K. Fujimoto, Y. Katsube, Y. Ohba, H. Schulzrinne Secure Universal Mobility for Wireless Internet, (Extended version) ACM MC2R, July 2005.

14. **Ashutosh Dutta**, Tao Zhang, Yoshihiro Ohba, Kenichi Taniuchi, and Henning Schulzrinne. MPA assisted proactive handoff scheme. In *ACM Mobiquitous, 2005*, page 155. San Diego, CA.

15. **A. Dutta**, S. Das, D. Famolari, Y. Ohba, K. Taniuchi, T. Kodama, and H. Shulzrinne. Seamless handover across heterogeneous networks-an IEEE 802.21 centric approach. *Proceedings of IWS-WPMC*, Aalborg, Denmark, 2005.

16. **Ashutosh Dutta**, Subir Das, David Famolari, Yoshihiro Ohba, Kenichi Taniuchi, Victor Fajardo, Toshikazu Kodama, and Henning Schulzrinne. Secured seamless convergence across heterogeneous access networks. In *World Telecommunication Congress*, Budapest, May 2006. IEEE.

17. K. Daniel Wong, **A. Dutta**, H. Schulzrinne, and K. Young. Simultaneous mobility: analytical framework, theorems and solutions. *Wireless Communications and Mobile Computing*, 7(5), 2007.

18. **Ashutosh Dutta**, Subir Das, David Famolari, Yoshihiro Ohba, Kenichi Taniuchi, Victor Fajardo, Toshikazu Kodama, and Henning Schulzrinne. Secured seamless convergence across heterogeneous access networks. In *World Telecommunication Congress*, Budapest, May 2006. IEEE.

19. **A. Dutta**, H. Schulzrinne, K.D Wong, "Supporting Continuous Services to Roaming Clients, The Handbook of Mobile Middleware, CRC Press, Edited by Bellavista and Corradi.

20. **Ashutosh Dutta**, Sunil Madhani, Tao Zhang, Yoshihiro Ohba, Kenichi Taniuchi, and Henning Schulzrinne. Network discovery mechanisms for fast-handoff. In *Broadnets*, San Jose, 2006. IEEE.

21. **Ashutosh Dutta**, Eric van den Berg, David Famolari, Victor Fajardo, Yoshihiro

Ohba, Kenichi Taniuchi, and Henning Schulzrinne. Dynamic buffering scheme for mobile handoff. In *IEEE PIMRC*, 2006, Helsinki.

22. T. Chiba, **A. Dutta**, and H Schulzrinne. Trombone Routing Mitigation Techniques for IMS/MMD Networks. In *Proceedings of IEEE WCNC*, Hong Kong, March 2007.

23. T. Chiba, H. Yokota, **A. Dutta**, D. Chee, and H. Schulzrinne. Route Optimization for Proxy Mobile IPv6 in IMS Network. In *Proceedings of the 2008 International Conference on Signal Processing and Communication Systems*.

24. R. Lopez, **A. Dutta**, Y. Ohba, and H. Schulzrinne. Network-layer assisted mechanism to optimize authentication delay during handoff in 802.11 networks. In *ACM Mobiquitous*, Philadelphia,PA, June 2007.

25. **A. Dutta**, S. Das, D. Famolari, Y. Ohba, and H. Schulzrinne. Seamless Proactive Handover across Heterogeneous Access Networks. *Wireless Personal Communication*, 43(3):837–855, August 2007.

26. **A. Dutta**, S. Chakravarty, K. Taniuchi, V. Fajardo, Y. Ohba, D. Famolari, H. Schulzrinne, "An Experimental Study of Location Assisted Proactive Handover," IEEE Globecom 2007, Internet Protocol Symposium, Washington D.C.

Mobility Fast-Handoff - Modeling

27. **A. Dutta**, B. Lyles, H. Schulzrinne, T. Chiba, H. Yokota, A. Idoue, "Generalized Modeling Framework for Handoff Analysis," IEEE PIMRC, September 2007, Athens.

28. **A. Dutta**, B. Lyles, H. Schulzrinne, J.Wang, "Systems Modeling for IP-based Handoff using Timed Petri nets," IEEE HICSS, January 2009, HAWAII

Performance - Mobility

29. K. D Wong, H-Yu Wei, **A. Dutta**, K. Young, H. Schulzrinne, "Performance of IP Micro-Mobility Management Scheme using Host Based Routing", IEEE WPMC

2001, Aalborg.

30. F. Anjum, M. Elaoud, D. Famolari, A. Ghose, R. Vaidyanathan, **A. Dutta**, P. Agrawal, "Voice Performance in WLAN Networks - An Experimental Study," IEEE Globecom 2003, San Francisco.

31. **A. Dutta**, J. Burns, R. Jain, D. Wong, K. Young, and H. Schulzrinne. Implementation and Performance Evaluation of Application layer MIP-LR. In *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, volume 2, Maui, HI, June 2005.

32. **A. Dutta**, B. Kim, T. Zhang, S. Baba, K. Taniuchi, Y.Ohba, H.Schulzrinne, "Experimental Analysis of Multi Interface Mobility Management with SIP and MIP," IEEE Wirelesscom 2005, Maui, HI

33. **A. Dutta**, S. Chakrabarty, K. Taniuchi, V. Fajardo, Y.Ohba, D. Famolari, H. Schulzrinne, "An Experimental Study of Location Assisted Proactive Handover," IEEE Globecom 2007, Washington DC.

Mobility and Streaming Architecture

34. **A. Dutta**, H. Schulzrinne, Y. Yemini, "MarconiNet: An Architecture for Internet Radio and TV. 9th International Workshop on Network Support for Digital Audio Video Systems (NOSSDAV 99), New Jersey, 23-25th June.

35. **A. Dutta**, H. Schulzrinne, "A Streaming Architecture for Next Generation Internet," IEEE ICC 2001, June 11-14, 2001, Helsinki, Finland.

36. **A. Dutta**, F. Vakil, J.C Chen, M. Tauil, S. Baba, H. Schulzrinne, "Application Layer Mobility Management Scheme for Wireless Internet," in 3G Wireless May 2001,(San Francisco).

37. K. Chakrabarty, A. Misra, S. Das, A. McAuley, **A. Dutta**, S. Das, ”Implementation and Performance Evaluation of TeleMIP,” IEEE ICC, May 2001, Helsinki.

38. A. Misra, S. Das, **A. Dutta**, A. McAuley, S. Das, ”IDMP-based Fast handoffs and paging in IP-based cellular Networks,” in 3G Wireless 2001, May 2001, San Francisco.

39. S. Das, A. Misra, A. McAuley, **A. Dutta**, S. Das, ”A generalized mobility solution using a dynamic tunneling,” in ICCCD2000, Dec. 2000, Kharagpur, India.

40. A. Misra, S. Das, A. McAuley, **A. Dutta**, S. Das, “Integrating QoS Support in TeleMIP’s Mobility Architecture,” in ICPWC, (Hyderabad, India), pp. 8, Dec. 2000.

41. **A. Dutta**, O. Altintas, H. Schulzrinne, W. Chen, “Multimedia SIP sessions in a Mobile Heterogeneous Access Environment,” 3G Wireless 2002, San Francisco.

42. J. Chennikara, W. Chen, **A. Dutta**, O. Altintas, “Application Layer Multicast for Mobile Users in Diverse Networks,” IEEE Globecom 2002, Taiwan.

43. S. Das, **A. Dutta**, A. McAuley, A. Misra, S. Das, “IDMP: An Intra-Domain Mobility Management Protocol for Next Generation,” IEEE Wireless Magazine, October 2002 - **SAIC Best Paper**

44. T. Chiba, H. Yokota, A. Idoue, **A. Dutta**, S. Das, Fuchun J. Lin, H.Schulzrinne, “Mobility Management Schemes for Heterogeneity Support in Next Generation Wireless Networks,” NGI 2007, May 2007, Norway

45. **A. Dutta**, H. Schulzrinne, W. Chen, O. Altintas, ”Mobility support for wireless streaming multimedia in MarconiNet,” in IEEE Broadband Wireless Summit, Interop 2001, (Las Vegas), pp. 7, May 2001.

46. **A. Dutta**, H. Schulzrinne, S. Das, A. McAuley, W. Chen, O. Altintas, “ MarconiNet supporting Streaming Media over Localized Wireless Multicast,” ACM M-Commerce

2002 Workshop, September, 2002, Atlanta.

47. **A. Dutta**, R. Jain, K. D. Wong, J. Burns, K. Young, Henning Schulzrinne, "Multilayerd Mobility Management for Survivable Network," IEEE MILCOM Proceedings, October 2001, Boston.

48. Y. Ohba, S. Das, **A. Dutta**, "Kerberized Handover Keying: A Media-Independent Handover Key Management Architecture," ACM Mobiarch 2007, Kyoto, Japan.

49. **A. Dutta**, F. Joe Lin, D. Chee, S. Das, B. Lyles, T. Chiba, H. Yokota, H. Schulzrinne "Architecture Analysis and Experimental IPv6 tstbed for Advances IMS," IMSAA 2007, Bangalore, India.

Testbed - Mobility

50. **A. Dutta**, J.C Chen, S. Das, S. Madhani, A. McAuley, S. Baba, N. Nakajima, Y. Ohba, H. Schulzrinne, "Implementing a Testbed for Mobile Multimedia," IEEE Globecom 2001, San Antonio.

51. **A. Dutta**, J. Burns, KD Wong, R. Jain, K. Young, H. Schulzrinne, and A. McAuley. Realization of Integrated Mobility Management Protocol for Ad-Hoc Networks. In *MILCOM*, volume 1, pages 448–454, 2002.

52. **A. Dutta**, P. Agrawal, S. Das, M. Elaoud, D. Famolari, S. Madhani, A. McAuley, Tauil M. Li, P., and H. Schulzrinne. Realizing mobile wireless Internet telephony and streaming multimedia testbed. *Computer Communications*, 27(8):725–738, 2004.

53. **A. Dutta**, K. Manousakis, S. Das, F.J Lin, T. Chiba, H. Yokota, A. Idoue, H.Schulzrinne, "Mobility Testbed for 3GPP2-based MMD Networks," IEEE Communication Magazine, July 2007.

54. S. Das, M. Tauil, Y.H. Cheng, **A. Dutta**, D. Baker, M. Yajnik, D. Famolari, "Media independent handover: Features, applicability, and realization,"IEEE Communication

Magazine, January 2009.

55. M. Tauil, **A. Dutta**, Y.H. Cheng, S. Das, D. Baker, M. Yajnik, D. Famolari, Y. Ohba, V. Fajardo, K. Taniuchi, H. Schulzrinne " Realization of IEEE 802.21 services and pre-authentication framework"IEEE conference Tridentcom 2009, April 2009, Washington DC.

56. M. Tauil, **A. Dutta**, Y.H. Cheng, S. Das, D. Baker, M. Yajnik, D. Famolari, Y. Ohba, V. Fajardo, K. Taniuchi, H. Schulzrinne " Integration of IEEE 802.21 services and pre-authentication framework," to appear in International Journal of Communication Networks and Distributed Systems.

Survey Paper - Mobility

57. **Ashutosh Dutta**, Onur Altintas, Wai Chen, and Henning Schulzrinne. Mobility approaches for all IP wireless networks. In *SCI*, Orlando, Florida, July 2002.

58. T. Chiba, H. Yokota, A. Idoue, **A. Dutta**, S. Das, Fuchun J. Lin, "Gap Analysis and Deployment Architectures for 3GPP2 MMD Networks," IEEE VT Magazine, March 2007.

59. **A. Dutta**, S. Das, T. Chiba, H. Yokota, A. Idoue, H. Schulzrinne, "Comparative Analysis of Network Layer and Application Layer IP Mobility Protocols for IPv6 Networks," WPMC 2006, San Diego, CA.

Potpourri

60. **A. Dutta**, Y. Yemini, "Power Management of LEOs under bursty broadband traffic. AIAA's 17th International Conference on Satellite Systems and Communication. AIAA, 1998 February," Yokohama, Japan, March 1998.

61. S. Khurana, **A. Dutta**, P. Gurung, H. Schulzrinne, "XML based Wide Area Communication with Networked Appliances," IEEE Sarnoff 2004, Princeton, NJ.

62. J. Chennikara, **A. Dutta**, A. McAuley, D. Wong, M. Elaoud, A. Cheng, M. Yajnik, I. Sebuktekin, K. Young, H. Schulzrinne, " Integrated Networking Technologies for Survivable Network," IEEE WCNC, March 2005, New Orleans.

63. T. Zhang, S. Madhani, **A. Dutta**, E. Van den Berg, Y.Ohba, K. Taniuchi, S. Mohanty, " Implementation and Evaluation of Autonomous Collaborative Discovery of Neighboring Networks," IEEE ITRE 2005.

64. **A. Dutta**, H. Cheng, S. Madahani, K.D. Wong, J. Chennikara, K. Young, H. Schulzrinne, A. Patel, "Flexible Call Control Framework for Supporting Multi-party Service," IEEE MILCOM 2005.

65. **A. Dutta**, J. Alberi, A. Cheng, B. Horgan , A. McAuley, D. Chee, B. Lyles, "IPv6 Transition Techniques for Legacy Application," IEEE MILCOM 2006, Washington DC.

66. **A. Dutta**, C. Makaya, S. Das, D. Chee, F. J. Lin, S. Komorita, T. Chiba, H. Yokota, "Self Organizing IP Multimedia Subsystem," IEEE IMSAA 2009, Bangalore. ***Best Paper Award***