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 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 obtained by applying these core techniques with non-optimized version.

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

5.2.1 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.
5.2.2 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, they 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 pre-configured or provisioned dynamically on the servers. The information can then be updated either by human administrators or automatically by the servers themselves exchanging up-
dates 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 5.2.3.

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.

5.2.3 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 con-
figuration 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 us-
ing network assisted discovery scheme is described in Chapter 9. I provide 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.
CHAPTER 5. MOBILITY OPTIMIZATION TECHNIQUES

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 mechanisms 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
Figure 5.1: Inter dependency chart for network elements

for 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 between 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 net-
work 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 up-
date 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 form-
ing 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 cou-
pled 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.

5.2.4 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 In-
formation 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
Figure 5.2: Deployment of application layer information discovery

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 information 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
Figure 5.3: Information population and query process

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 incurred 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 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
Table 5.1: Information service query processing

<table>
<thead>
<tr>
<th>Query Type</th>
<th>Response</th>
<th>Processing delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current PoA: AP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neighbor 0 selected</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Query 2</strong>: Provide list of network elements for Neighbor 0</td>
<td>Target Network Channel: 10 SSID: ITSUMO newpool1 Router address: 10.10.10.52 Router MAC: 00:00:39:ce:63:be6c Subnet: 255.255.255.0 DHCP Server: 10.10.10.52</td>
<td>Total: 1473 API: 991 Network: 451 Server processing: 13 Client processing: 18</td>
</tr>
</tbody>
</table>

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 service. 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.

5.3 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), AAAlh (AAA home)) in each domain. In nutshell, the home AAA domain (where the user belongs to) is equipped with a home AAA (AAAlh) 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
Figure 5.5: Illustration of roaming environment

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 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.

### 5.3.1 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.

### 5.3.2 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 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

\(^1\)STA and mobile node are used interchangeably
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 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).
Ruckfosth et al. [RL04] propose a different approach where a combination of Fast Mobile IPv6 [Koo05] and IEEE 802.11i frames are used to inform users 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.

5.3.3 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 [DOF09], [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 manages 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 schemes reduce the delay invoked by the authentication process during the handover between 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.

5.3.3.1 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 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.

### 5.3.3.2 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 pre-emptive 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.

5.3.4 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.

In this experimental testbed I have used hostapd software [Hos] and madwifi driver
Figure 5.9: Interaction among the functional components

[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 Network 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 Diameters 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 mobiles 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 5.2.

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

\textbf{Taauth} 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.

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

\textbf{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 hand-
shake can be considered as independent events. In fact, \textit{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 \textit{Tconf} remains same.

The first two columns in Table 5.2 show the results for \textit{non-roaming} and \textit{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 $T_{auth}$ (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 inter-
work 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

<table>
<thead>
<tr>
<th>Types of authentication</th>
<th>Post authentication</th>
<th>802.11i pre-authentication</th>
<th>Network layer assisted pre-authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Types of movement</td>
<td>Non roaming</td>
<td>Roaming</td>
<td>Non roaming</td>
</tr>
<tr>
<td>$T_{auth}$</td>
<td>61 ms</td>
<td>599 ms</td>
<td>98 ms</td>
</tr>
<tr>
<td>$T_{config}$ 2 AP</td>
<td>-</td>
<td>-</td>
<td>638 ms</td>
</tr>
<tr>
<td>$T_{assoc}$ 4 way handshake</td>
<td>-</td>
<td>-</td>
<td>177 ms</td>
</tr>
<tr>
<td>Total</td>
<td>79 ms</td>
<td>616 ms</td>
<td>831 ms</td>
</tr>
<tr>
<td>Time affecting handover</td>
<td>79 ms</td>
<td>616 ms</td>
<td>17 ms</td>
</tr>
</tbody>
</table>

5.4 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
(Encrypting Security Payload). Thus, when the IP address of any one of the commu-
nicating 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.

5.4.1 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 these 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].
5.4.2 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.

5.4.3 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 5.4.1. 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 5.11 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 5.11: Anchor agent-assisted security association](image)

Figure 5.11 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 5.12. These tunnels are the additional systems resources expended while handoff delay is reduced by avoiding re-establishment of security association.

\(^2\)DMZ has been defined in Appendix C
I describe the details of the architecture, implementation, and experimental verification in [DZM*04]. I briefly describe the techniques and associated results here.

![Diagram](image)

**Figure 5.12: Mobile IP and VPN tunnels**

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).
CHAPTER 5. MOBILITY OPTIMIZATION TECHNIQUES

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 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.

5.4.4 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 5.13 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.

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 5.14 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 5.14a 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 5.14b 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 5.14c shows the signaling sequence when the mobile goes back from CDMA network to the 802.11 network.

Figure 5.15 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 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 5.17 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 Conferencing tool) [UCB] to generate the video traffic. Both RAT and VIC are open source software.

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 5.18 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 [NDDS03].
Table 5.3: Effect of duplicate address detection (IPv6) on handoff

<table>
<thead>
<tr>
<th>Handoff Case</th>
<th>Signaling Delay (ms)</th>
<th>Media Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SIP w/ DAD and NUD</td>
<td>SIP w/o DAD and NUD</td>
</tr>
<tr>
<td>H12 (Home-Visited 1)</td>
<td>3829</td>
<td>171</td>
</tr>
<tr>
<td>H23 (Visited 1 – Visited2)</td>
<td>3932</td>
<td>161</td>
</tr>
<tr>
<td>H31 (Visited 2 – Home)</td>
<td>1935</td>
<td>161</td>
</tr>
</tbody>
</table>

### 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.

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 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 5.19 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.

Figure 5.19: Router assisted duplicate address detection

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].

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 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. It 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 ac-
quisition 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 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 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 Route optimization

In Chapter 3, I have defined different processes that are part 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.
5.6.1 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.

5.6.2 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.

### 5.6.3 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 5.20 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.
5.6.4 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 5.21 shows how the packet interceptor module and packet modifier are implemented in MIP-LR. Figure 5.22 shows the experimental
testbed where I have carried out this experiment. The mobile host (mh) moves between the two 802.11 access points. Delay1 and delay2 are emulated delays between CH and HA. These delays were introduced using NIST delay simulator. Lr1 and Lr2 are the location registers in each visited network. Table 5.4 shows the round trip time delay comparison between the interceptor assisted MIP-LR and mobile IP. Ha is the home agent.

Figure 5.21: Packet interception technique for MIP-LR

Table 5.4: Experimental validation of route optimization

<table>
<thead>
<tr>
<th>Packet Size Bytes</th>
<th>Emulated delay 0 ms</th>
<th>Emulated delay 10 ms</th>
<th>Emulated delay 20 ms</th>
<th>Emulated delay 40 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIP (RTT) (ms)</td>
<td>MIP-LR (RTT) (ms)</td>
<td>MIP (RTT) (ms)</td>
<td>MIP (RTT) (ms)</td>
</tr>
<tr>
<td>64</td>
<td>5.9</td>
<td>4.1</td>
<td>25.3</td>
<td>5.4</td>
</tr>
<tr>
<td>128</td>
<td>6.6</td>
<td>4.7</td>
<td>27.4</td>
<td>5.6</td>
</tr>
<tr>
<td>256</td>
<td>8</td>
<td>5.9</td>
<td>28.1</td>
<td>6.2</td>
</tr>
<tr>
<td>512</td>
<td>13.9</td>
<td>10.2</td>
<td>32</td>
<td>10.8</td>
</tr>
<tr>
<td>1024</td>
<td>19.5</td>
<td>13</td>
<td>39.5</td>
<td>14.2</td>
</tr>
</tbody>
</table>
Table 5.4 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 delay1 was varied simulating increase in distance between CH and HA (Home 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.

### 5.6.5 Interception 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 mo-
bile 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.

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.

![Route optimization of signaling traffic](image)

**Figure 5.23:** Route optimization of signaling traffic

Figure 5.23 illustrates the experimental testbed where I have applied the route opti-
CHAPTER 5. MOBILITY OPTIMIZATION TECHNIQUES

mization 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. 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 require any changes in MIP functional behavior. Figures 5.24 and 5.25 illustrate the SIP signaling flow without and with route optimization approaches, respectively.
5.6.6 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 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.
5.6.6.1 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 5.24, 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 \(3(t_1 + t_2 + t_3) + 2(t_4 + t_5) + 2t_6 + 3(P_{HA} + P_{FA})\).

5.6.6.2 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 5.25, 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 \(D_2\) 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 \((D_1 - D_2) = \)
3(t_2 + t_3 - t_7) + 3(P_{HA} + P_{FA} - 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 5.26 shows the handoff time for both without and with the proposed techniques, respectively in the IMS testbed environment.

![Handoff Time Graph]

Figure 5.26: Results of route optimization using packet interceptor

In particular, Figure 5.26 shows the breakdown of delay due to several operations such as layer 2 handoff, PPP link configuration, mobile IP registration, DHCP INFORM for 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 modifying 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 [CD07].

5.6.7 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 5.6.1. Many of the Proxy MIPv6 terms are defined in Appendix C. Figure 5.27 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 5.27: Architecture for binding cache-based route optimization

Figure 5.28 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 trajectory 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.

![Diagram](image)

Figure 5.28: Binding cache-based flow

Table 5.5 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 5.6.5. 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 5.5 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 5.5: Results from route optimization using binding cache approach

<table>
<thead>
<tr>
<th>Additional trip delay between MAG and LMA</th>
<th>50 ms</th>
<th>100 ms</th>
<th>150 ms</th>
<th>200 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route optimization</td>
<td>RO</td>
<td>Non RO</td>
<td>RO</td>
<td>Non RO</td>
</tr>
<tr>
<td>End-to-end media delay (ms)</td>
<td>12</td>
<td>71</td>
<td>12</td>
<td>107</td>
</tr>
<tr>
<td>SIP REGISTER delay (s)</td>
<td>2.34</td>
<td>2.50</td>
<td>2.84</td>
<td>2.82</td>
</tr>
</tbody>
</table>

I have also explained details of this route optimization technique in the IETF draft [CYS+08].

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 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.

### 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
CHAPTER 5. MOBILITY OPTIMIZATION TECHNIQUES

[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 5.29 shows how an anchor agent called mobility agent (MA) can be used to provide hierarchical binding update when the mobile moves within a domain.

![Diagram of network layer mobility agent](image)

Figure 5.29: Functional architecture for hierarchical mobility agent

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 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 5.30 shows the call flow when the MN first moves into a new domain managed by a
mobility agent. Figure 5.31 shows the call flow during subsequent intra-domain movement.

According to the flow shown in Figure 5.30, 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.

![Diagram of Initial intra-domain location update](image)

Figure 5.30: Initial intra-domain location update

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. 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 5.31 shows call flow during subsequent intra-domain movement.

![Figure 5.31: Call flow during subsequent intra-domain handoff](image)

### 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 B2BUA can be used as an anchor point to reduce the binding update delay in [DMC+04]. Figure 5.32 illustrates the functional architecture of how B2BUA can be used to reduce the
CHAPTER 5. MOBILITY OPTIMIZATION TECHNIQUES

binding update delay. Figure 5.33 shows the detailed protocol flow for the B2BUA-based binding update.

![Protocol Flow Diagram]

Figure 5.32: B2BUA-based 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
Figure 5.33: B2BUA-based flow for hierarchical binding update

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 5.34 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.

![Diagram](image)

Figure 5.34: 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 MA₁ 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 MA₁ is still its MA, the MN simply performs an intra-domain location update, informing MA₁ 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).

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

(Note that \( L_g \leq L_d \), 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 5.6 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>
<thead>
<tr>
<th>Architecture</th>
<th>Signaling overhead (bytes/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Local per hop</td>
</tr>
<tr>
<td>Mobile IP</td>
<td>0</td>
</tr>
<tr>
<td>Mobility Agent</td>
<td>( 2L_d/T_s )</td>
</tr>
</tbody>
</table>

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 5.35. 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 5.35, 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*09]. 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 to 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.

### 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.

\(^3\)Reactive tunnel has been defined in the definition section
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. [VFBP01] 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 5.36 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.

![Diagram of tunneling techniques](image)

**Figure 5.36:** Forwarding agent for data redirection

### 5.8.3.1 Reactive Handover Tunnel (RHT)

According to Figure 5.36, 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 5.36 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 5.36 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 handoffs 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 5.37 shows the basic framework for mobility proxy assisted media redirection.

![Diagram of media redirection using SIP-based mobility proxy]

Figure 5.37: Media redirection using SIP-based mobility proxy

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 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 5.37 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 5.38, 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].

![Diagram of mobility proxy](image)

**Figure 5.38: Experimental testbed for mobility proxy**

I provide an analysis of how packet loss is minimized by means of forwarding mechanism using RTP translator. This analysis is based on the experimental testbed shown in Figure 5.38. 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 5.39 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 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].

![Diagram](image)

**Figure 5.40: Time-bound multicasting for IDMP**

Figure 5.40 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 briefly describe below how fast handoff is achieved by applying these optimization tech-
niques in the architecture shown in Figure 5.40.

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 5.41, 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 MNs 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 5.41 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 $SA_{new}$.

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 5.40), 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 5.42 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.

![Figure 5.42: Data redirection using multicast agent](image)

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 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 techniques 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 mul-
timedia 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
queueing 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 5.43 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 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.
CHAPTER 5. MOBILITY OPTIMIZATION TECHNIQUES

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.

\[
\text{BReq[initial] = id, bp, tc, bsz, p}
\]

\[
\text{BAns[initial] = id, bp, bsz, rcode}
\]

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.

\[
\text{BReq[ext] = id, seq, bp, bsz, p, coa}
\]
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

id MN id sent in the BReq[initial]
seq Signal sequence number, must match BReq[ext]
bp New buffering period for this service
bsz New buffer size allocated for this service
rcode Result code provided by BN

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

Service Attributes
The BCP also creates service attributes (state information) within the BN. These at-
tributes include the following:

1. MN Id (id)

2. Buffering period (bp)

3. Negotiated buffer size (bsz)

4. Traffic classification (tc) parameter

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
id MN id sent in the BReq\[initial\]
rcode Result code provided by BN

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 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.

![Protocol flow for buffer control protocol](image)

Figure 5.44: Protocol flow for buffer control protocol

The protocol flow shown in Figure 5.44 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 5.45 shows a typical scenario where I have experimented this technique with media independent pre-authentication. Figure 5.46 shows the protocol flow when PANA is used as buffer control protocol.

![Buffering with media independent pre-authentication](image)

**Figure 5.45: Buffering with media independent pre-authentication**

Table 5.7 and Table 5.8 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.

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:

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.

Table 5.7: Time limited buffering

<table>
<thead>
<tr>
<th>Traffic rate (pkts/sec)</th>
<th>X (ms)</th>
<th>Y (ms)</th>
<th>Packet delay (ms)</th>
<th>Packet loss</th>
<th>Average packet buffered</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>0</td>
<td>N/A</td>
<td>37</td>
<td>0</td>
<td>2.5</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>N/A</td>
<td>42</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>90</td>
<td>0</td>
<td>N/A</td>
<td>44</td>
<td>0</td>
<td>3.5</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>N/A</td>
<td>45</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>
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.

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

<table>
<thead>
<tr>
<th>Traffic rate (pkts/sec)</th>
<th>X (ms)</th>
<th>Y (ms)</th>
<th>Packet delay (ms)</th>
<th>Packet loss</th>
<th>Packets buffered</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>0</td>
<td>30</td>
<td>27</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>20</td>
<td>29</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>10</td>
<td>12</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>30</td>
<td>28</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>20</td>
<td>33</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>90</td>
<td>0</td>
<td>30</td>
<td>69</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>20</td>
<td>46</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>10</td>
<td>11</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>30</td>
<td>69</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>20</td>
<td>46</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>10</td>
<td>11</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
explicit flush message instead of fine-tuning y (noted as PUR if y=0 in Figure 5.46). 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 01, 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 5.7 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 5.8 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 playout buffer at the mobile. Figure 5.47 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.
5.10 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.
5.10.1 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.

5.10.2 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. [YIKH02] 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.

5.10.3 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.

5.10.3.1 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
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.
5.10.3.2 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.

5.10.3.3 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](image)

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 Pre-authentication. I have described those experimental results in Chapter 9.
Table 5.9: Sample MIHF primitives

<table>
<thead>
<tr>
<th>MIH.SAP Primitives</th>
<th>Service category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIH.Capability.Discover</td>
<td>Management</td>
<td>Discover list of Events and Commands supported by MIHF</td>
</tr>
<tr>
<td>MIH_Register</td>
<td>Management</td>
<td>Register with a remote MIHF</td>
</tr>
<tr>
<td>MIH.DeRegister</td>
<td>Management</td>
<td>Deregister from a remote MIHF</td>
</tr>
<tr>
<td>MIH.Event.Subscribe</td>
<td>Management</td>
<td>Subscribe for one or more MIH events with a local or remote MIHF</td>
</tr>
<tr>
<td>MIH.Event.Unsubscribe</td>
<td>Management</td>
<td>Unsubscribe for one or more MIH events from a local or remote MIHF</td>
</tr>
<tr>
<td>Link.Detected</td>
<td>Event</td>
<td>Link of a new access network is detected</td>
</tr>
<tr>
<td>MIH_Link_Up</td>
<td>Event</td>
<td>L2 connection is established</td>
</tr>
<tr>
<td>MIH_Link_Down</td>
<td>Event</td>
<td>L2 connection is lost</td>
</tr>
<tr>
<td>MIH_Link.Going-Down</td>
<td>Event</td>
<td>L2 connectivity is predicted to go down</td>
</tr>
<tr>
<td>MIH_Link_Handover.Imminent</td>
<td>Event</td>
<td>L2 handover is imminent</td>
</tr>
<tr>
<td>MIH_Link.Handover.Complete</td>
<td>Event</td>
<td>L2 handover link handover to a new access network is complete</td>
</tr>
<tr>
<td>MIH_Link_Parameters_Report</td>
<td>Event</td>
<td>Link parameters have crossed specified thresholds</td>
</tr>
<tr>
<td>MIH_Link.Get_Parameters</td>
<td>Command</td>
<td>Get the status of the link</td>
</tr>
<tr>
<td>MIH_Link.Configure.Thresholds</td>
<td>Command</td>
<td>Configure Link parameter thresholds</td>
</tr>
<tr>
<td>MIH_Link.Actions</td>
<td>Command</td>
<td>Control the behavior of set of links</td>
</tr>
<tr>
<td>MIH_Net_HO.Candidate.Query</td>
<td>Command</td>
<td>Initiate Handover</td>
</tr>
<tr>
<td>MIH_MN_HO.Candidate.Query</td>
<td>Command</td>
<td>Initiate MN query request for candidate network</td>
</tr>
<tr>
<td>MIH_NN_HO.Query_Resources</td>
<td>Command</td>
<td>Query available network resources</td>
</tr>
<tr>
<td>MIH_MN_HO.Commit</td>
<td>Command</td>
<td>Notify the serving network of the decided target network information</td>
</tr>
<tr>
<td>MIH_Net_HO.Commit</td>
<td>Command</td>
<td>Network has committed to handover</td>
</tr>
<tr>
<td>MIH_NN_HO.Commit</td>
<td>Command</td>
<td>Notify target network that serving network has committed to handover</td>
</tr>
<tr>
<td>MIH_MN_HO.Complete</td>
<td>Command</td>
<td>Initiate MN Handover complete notification</td>
</tr>
<tr>
<td>MIH_NN_HO.Complete</td>
<td>Command</td>
<td>Handover has been completed</td>
</tr>
<tr>
<td>MIH_Get_Information</td>
<td>Information</td>
<td>Requests to get information from repository</td>
</tr>
<tr>
<td>MIH_Push_Information</td>
<td>Information</td>
<td>Notify the mobile node of operator’s policies or other information</td>
</tr>
</tbody>
</table>

5.10.3.4 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
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 have shown the experimental results associated with the 802.21-based cross layer triggers in Chapter 9.

### 5.10.4 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.
CHAPTER 5. MOBILITY OPTIMIZATION TECHNIQUES

In 802.11-based layer 2 operation, a client that is currently associated with an AP (Access 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 dB’s 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
Chapter 5. Mobility Optimization Techniques

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) though 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.

5.10.4.1 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.

5.10.4.2 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.

### 5.10.4.3 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.

### 5.10.4.4 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 traffic 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.