Chapter 9

Systems evaluation

In this Chapter, I evaluate the overall handoff system where many of the optimization techniques that I have described in Chapter 5 function together to build the complete handoff system. I first illustrate the experimental results from the handoff system for both inter-technology and intra-technology handoff and then validate some of the optimization techniques using Petri net modeling including few mobility scenarios such as multi-interface mobility and simultaneous mobility.

9.1 Introduction

Systems evaluation and validation of several optimization techniques associated with a handoff event can be performed through experimental analysis and analytical modeling. While experimental results are constrained by several systems parameters such as memory, CPU power and other network parameters such as bandwidth, Petri net models can be used to validate the experimental results and perform systems evaluation with the ability to vary the systems parameters. Thus, in order to validate various optimization techniques associated with the handoff system, I have applied both the experimental and modeling approaches. Optimization techniques for many of the handoff components have been described in Chapter 4, each with its own experimental results. However, in order to validate
the overall system performance, I have built a complete handoff system and have implemented these optimization techniques that can work together.

9.2 Experimental validation

In this section, I describe the experimental results from a mobility systems that support multiple types of handoff such as *intra – technology* and *inter – technology* that use single interface and multiple interfaces, respectively. These mobility systems use a set of optimization techniques that I have described earlier in Chapter 5.

In particular, I describe the results from three experimental systems, 1) Media independent preauthentication framework 2) Cross layer trigger assisted handoff 3) Optimized handoff in IMS systems

9.2.1 Media Independent Pre-authentication Framework

I have prototyped a mobility system called media independent pre-authentication [DZO*] that utilizes many of the techniques that I have developed to optimize the basic handoff operations, such as network discovery, authentication, configuration, security association, and binding update. This system also optimizes the link layer handoff delay by avoiding scanning and applies the cross layer optimization techniques, such as “link up” “link down” events. In addition, it reduces packet loss by implementing the dynamic buffering and copy-and-forward mechanism [DvF’06] at the edges of the network. I have applied these proactive techniques to optimize both network layer and application layer mobility protocols over both single interface or multiple interfaces [DKZ’05], [DZO*].

Koodli et al. [Koo05] and Gwon et al. [GFJ03] have developed a few mobility systems that utilize the proactive handoff techniques for both Mobile IPv6 and Mobile IPv4, respectively. However, these systems do not address proactive discovery or preauthentication mechanisms. Also, these mechanisms require signaling exchange between the neighbor-
CHAPTER 9. SYSTEMS EVALUATION

ing routers that work only if the routers have established trust relationship among them. I performed a comparative analysis of Media Independent Pre-authentication (MPA) mechanism with the proactive mode of fast-handoff mechanism of FastMIPv6 and have described the details of the results in [DDF⁺07].

Figure 9.1 shows an MPA-based protocol flow. Assume that the mobile node is already connected to a point of attachment referred to as the old point of attachment (oPoA), and assigned an old care-of address (oCoA). Throughout the communication flow, data packet should not be lost except for the period during the layer 2 switching procedure in Step 5, but MPA can help minimize the packet loss during this period with the help of Information Service (IS), Event Service (ES) and Command Service (CS) of IEEE 802.21. I briefly describe different functional phases of pre-authentication framework.

**Pre-authentication phase:** The mobile finds a CTN (Candidate Target Network) through a discovery process, such as IEEE 802.21, and obtains the address and capabilities of the AA (Authentication Agent), CA (Configuration Agent), and AR (Access Router) in the CTN. The mobile pre-authenticates with the authentication agent. If the pre-authentication is successful, an MPA-SA (Security Association) is created between the mobile node and the authentication agent. Two keys are derived from the MPA-SA, namely, an MN-CA key and an MN-AR key, which are used to protect subsequent signaling messages of a configuration protocol and a tunnel management protocol, respectively. The MN-CA key and the MN-AR key are then securely delivered to the configuration agent and the access router, respectively. Layer-2 pre-authentication can be initiated at this stage.

**Pre-configuration phase:** The mobile node realizes that its point of attachment is likely to change from oPoA to a new one, say, nPoA. It then performs pre-configuration, with the configuration agent using the configuration protocol to obtain an IP address, say nCoA (new care-of address) and other configuration parameters from the CTN. The access router uses the tunnel management protocol to establish a proactive handover tunnel. In the tunnel management protocol, the mobile node registers oCoA and nCoA as the tunnel outer address
and the tunnel inner address, respectively. The signaling messages of the pre-configuration protocol are protected using the MN-CA key and the MN-AR key. When the configuration and the access router are co-located in the same device, configuration and tunnel management may be performed by a single protocol such as IKEv2. After completion of the tunnel establishment, the mobile is able to communicate using both oCoA and nCoA by the end of Step 4.

**Secure proactive handover main phase:** Before the mobile switches to the new point of attachment, it starts secure proactive handover process by executing the proactive binding update operation of a mobility management protocol and transmitting subsequent data traffic over the tunnel. In some cases, it may cache multiple nCOA addresses and perform simultaneous binding with the CH (Corresponding Host) or HA (Home Agent).

**Secure proactive handover pre-switching phase:** The mobile completes the binding update and becomes ready to switch to the new point of attachment. The mobile may execute the tunnel management protocol to delete or disable the proactive handover tunnel and cache nCoA after deletion or disabling of the tunnel. A buffering module at the new Access Router (nAR) begins to buffer the packets (start-buffering) when it receives the tunnel-delete signal. The mobile sends explicit signal to stop buffering and flush the packets after the mobile connects to the new point-of-attachment.

The decision as to when the mobile switches to the new point of attachment depends on the handover policy. In general, mobile-controlled or network-controlled policies can be used to trigger the handoff. The mobile’s signal quality, location, communication cost, and QoS on the received traffic are some factors that can determine the handoff policy. Results presented in this paper are based on Signal-to-Noise Ratio (SNR).

**Switching phase:** It is expected that a link-layer handover occurs in this step. During this phase, any of the layer 2 security association including EAP-based authentication and 802.11i related 4-way handshake may take place. Normally, layer 2 pre-authentication is taken care of by inherent layer 2 pre-authentication support. In this scheme, layer 3
pre-authentication can bootstrap layer 2 authentication leaving only the 4-way handshake during this phase.

**Secure proactive handover post-switching phase:** The mobile executes the switching procedure. Upon successful completion of the switching procedure and layer-2 association, the mobile immediately restores the cached nCoA and assigns it to the physical interface attached to the new point of attachment. If the proactive handover tunnel was not deleted or disabled in Step 4, the tunnel can be deleted or disabled in this phase as well. After this, direct transmission of data packets using nCoA is possible without using a proactive handover tunnel.

I have applied the MPA related techniques and have experimented with two mobility protocols - SIP-based mobility and MIPv6 for both intra-technology and inter-technology handovers.

Figure 9.1: Protocol flow for Media Independent Pre-authentication

### 9.2.2 Intra-technology handoff

In this section, I highlight experimental results with intra-technology handoff.
CHAPTER 9. SYSTEMS EVALUATION

An intra-technology handover is defined when a mobile moves between the same type of access technology such as between 802.11[a,b,n] and 802.11[a,b,n] or between CDMA1XRTT and CDMA1EVDO. In this scenario a mobile may be equipped with a single interface (with multiple PHY types of the same technology) or with multiple interfaces. An Intra-technology handover may involve intra-subnet or inter-subnet movement and thus may need to change its L3 identifier depending upon the type of movement.

Figure 9.2 shows the basic topology of the experimental test-bed where I have experimented intra-technology handoff using 802.11 access networks. The test-bed emulates two different visited domains and a home domain. Each visited domain has several sub-networks. Initially, the mobile resides in network 1. The mobile moves from one visited domain to another domain and in the process changes its subnet. Network 1 is oPoA where the mobile node (MN) initially resides prior to handover. Network 2 is the nPoA, Network 3 is where the correspondent node (CN) resides and finally, Network 4 is where the Home Agent (HA) resides. In case of MPA for MIPv6, the CN starts an RTP session with the MN while the MN is in Network 1 via the HA using an MIPv6 tunnel. MPA creates a proactive handover tunnel between the MN and R2 in Network 2. This is an IPSec tunnel in Encapsulating Security Payload (ESP) mode and I use the Protocol for carrying Authentication for Network Access (PANA) for dynamically establishing and terminating the IPSec tunnel. Before the handoff, the MIPv6 tunneled-traffic between the MN and HA goes through the IPSec tunnel created by MPA with IPSec policy settings. When the configuration agent and router co-locate, a single protocol, such as IKEv2 can take care of both the functions, such as configuration and tunnel management.

Table 9.1 shows the experimental results from the mobility system that demonstrates intra technology handover involving IEEE 802.11 access networks. Both application layer mobility protocol such as SIP-based mobility and network layer mobility such as MIPv6 were used for experimental validation of these techniques. These results demonstrate how several of the proactive optimization techniques can work together to minimize the delays,
packet loss, and jitter. Results also demonstrate the effect of buffering at the edge routers in reducing packet loss at the expense of added delay. These results show the average value taken over 5 runs.

I have described implementation and experimental details of the MPA framework supporting intra technology handoff in [DZO*], [DFD*08].

9.2.3 Inter-technology handoff

Supporting terminal handovers across heterogeneous access networks such as CDMA, 802.11 and GPRS is a clear challenge, as each access network has different QoS, security and bandwidth characteristics. A mobile may be equipped with multiple interfaces, where each interface can support different access technology (802.11, CDMA). A mobile may like to communicate with one interface at any time in order to conserve the power. During the handover the mobile may move out of the footprint of one access technology (e.g., 802.11) and move into the footprint of a different access technology (e.g., CDMA). This will warrant switching of the communicating interface on the mobile as well. This type of Inter-technology handover is often called as Vertical Handover since the mobile makes
movement between two different cell sizes. A vertical handover can be termed as upward vertical handover or downward vertical handover based on the direction of movement such as smaller cell to larger cell or vice versa [SK98]. A mobile moving from 802.11 network to cellular network can be viewed as upward vertical handover. An inter-technology handover may affect the quality-of-service of the multimedia communication, since each access network offers different bandwidth and each of the access specific handoff operations may require different amount of resources.

I have applied the optimization techniques to inter-technology handover involving two specific access technologies, IEEE 802.11 and CDMA2000.

Several types of handoff scenarios are possible involving handoff with multiple inter-
faces. I have experimented with two different handoff scenarios: 1) Break-before-make scenario 2) Make-before-break scenario. I describe these scenarios below.

**Scenario 1: Break-before-make**

In normal handoff scenario involving multiple interfaces, the new interface comes up only after the link to the old interface is down. This scenario can be termed as “break-before-make” and usually gives rise to undesirable packet loss and handoff delay. In the current experimental testbed, without any optimization, the handoff delay and associated packet loss resulted because of PPP configuration delay (16 sec) and binding update delay, such as SIP re-INVITE (1.5 s) and MIP registration delay (500 ms). Lower layer triggers, such as “Link Down” can help expedite the handoff process on the second interface and will help reduce the packet loss. In order to optimize scenario 1, I have applied the fast “Link Down” detection technique along with pre-authentication, proactive configuration techniques, buffering, and copy-and-forwarding technique to reduce the packet loss and delay. Using this technique, the mobile was able to resume the communication in the new network, within 50 ms and the packet loss was reduced to 0.

**Scenario 2: Make-before-break (A)**

In the second scenario, the second interface is prepared proactively while the mobile still communicates using the old interface, and at some point the mobile decides to use the second interface as the active interface and completes the authentication and binding update procedures. This results in fewer packet loss as it uses “make-before-break” techniques to set up the layer 2 configuration while the mobile is still connected to the old network. In a typical break-before-make handoff without any optimization, the mobile prepares the CDMA interface only after current interface (802.11) goes down.

As part of scenario 2, I have developed the make-before-break algorithm to support handoff between CDMA and 802.11 networks. Using this technique, the mobile sets up the layer 3 configuration in the CDMA network using the PPP interface while it still keeps on communicating using the current 802.11 interface. This technique reduces layer 2 and layer
3 configuration related delay and packet loss. However, it uses up more resources, since both the interfaces are active at the same time. I have experimented with this mechanism for both network layer protocol, such as MIPv4 and application layer protocol such as SIP-based mobility over 802.11 and CDMA1xRTT access networks. By using make-before-break technique, there was no packet loss for both MIP and SIP. However, initial jitter was observed for the in-flight packets after the handover from 802.11 to CDMA network and out-of-order packets were received after the handover from CDMA to 802.11 networks. I have published a complete experimental handoff analysis of this work in [DKZ'05].

Make-before-break (B):

In the third scenario, some of the required functions, such as nework selection, context transfer of CDMA network parameters such as PPP state and security associations are established ahead of time using the current interface. This scenario is beneficial from a battery management standpoint. By activating the second interface only after an appropriate network has been selected and the mobile has authenticated itself using the old interface it can utilize battery more efficiently.

From the experiments it is verified that make-before-break technique aided by a combination of proactive optimization methodologies and cross layer triggers can reduce the delay during handover.

9.2.4 Cross layer trigger assisted pre-authentication

In this section, I describe the experimental results from a handover system that uses pre-authentication techniques assisted by the media independent handover functions as defined by IEEE 802.21. IEEE 802.21-based cross layer triggers help in handover preparation. Figure 9.3 shows the interaction between the MPA related functions and 802.21-based cross layer triggers denoted as MIHF (Media Independent Handover Functions).

Figure 9.4 depicts the integrated test-bed setup where I have conducted the experiments involving multiple interfaces. This testbed has used MIHF and MPA components.
The experimental testbed includes two types of access networks such as EV-DO and Wi-Fi that are connected via a core network infrastructure. Currently, the EV-DO service is provided by Verizon. The complete testbed consists of the following entities:

A mobile node (MN) is equipped with Wi-Fi and EV-DO interfaces. The Wi-Fi and EV-DO interfaces have the IP address IP0 and IP1, respectively. The MN runs an MPA client supporting IPsec, IKE, MOBIKE and MIH related services that provide cross layer triggers.

MPA server is equipped with several modules including an authentication agent (AA), tunneling agent, configuration agent (CA), and buffering module. The AA pre-authenticates the MN. The tunneling agent manages an IPsec tunnel from the MN as the PHT (Proactive Handover Tunnel) and performs layer 3 handover using MOBIKE. In this specific scenario the testbed is slightly different from the MPA framework described earlier as the tunneling agent is implemented on a node outside of the target network (i.e., the EV-DO network) not on the AR in the target network because we do not have control of the equipment of the operators network. As a result, the MPA server acts as a proxy AR to the EV-DO network.

A MIH Information Server (IS) within the testbed is populated with the information
of the neighboring network elements such as Wi-Fi access points and cellular network elements.

A correspondent node (CN) is connected to the Internet and communicates with the MN via the Skype voice over IP session. In this mobility scenario, the mobile node engages in a VoIP session with the CN over the Wi-Fi network (path A) and then performs a handover to the EV-DO network (path B). While the MN is still connected to the Wi-Fi network, the MPA stack utilizes the MIH services to trigger an authentication and configuration process with the EV-DO network in anticipation of the mobile node’s move. The MPA engine learns of the target network by querying the MIH Information Server for network information. The MPA stack that triggers the authentication can be either on the MN (for mobile initiated) or on the MPA server (for network initiated handover). Although the current implementation uses signal strength thresholds to trigger the IS (Information Server) query, other policies may also be implemented to trigger several steps in the handover process.

Since the tunnel agent does not reside inside the cellular operator network, all communication to and from the MN needs to go through the MPA server over the PHT, even after L2 handover, as shown in path B.
The MPA agents utilize MIH services for the following purposes:

- Identify when to prepare for handover based on signal thresholds of the active interface. This is done by event subscription to Parameter Reports when the active interface’s signal level in the MN crosses different thresholds.

- Identify candidate networks and their related parameters the mobile is likely to handover to by querying the Information Service. Using the MIH_Link_Actions, \textit{PowerUp} MIH command to power up, connect and configure the EV-DO interface and set up a PHT once pre-authentication procedure is over.

- Using MIH command MIH_Link_Actions, \textit{PowerDown} to turn off the old link once handover is complete.

Figure 9.5 (a) shows the sequence diagram for a mobile-initiated handover from the Wi-Fi network to the EV-DO network. Figure 9.5 (b) shows the flow for network initiated handover from Wi-Fi to EV-DO network. I describe the details of the flows for both mobile-initiated and network-initiated handovers.

Figure 9.5: (a)Mobile initiated (b) Network initiated handover
9.2.5 Mobile Initiated Handover with 802.21 triggers

The MN is initially connected to the Wi-Fi network. I describe the following steps in sequence.

1) **Subscribe Request**: The MPA client first subscribes to the MIH_Link_Param_Report event, which provides link parameter reports when the Wi-Fi signal strength crosses certain values.

2) **Configure Threshold Request**: The MPA client uses an MIH_Link_Configure_Threshold command to establish a set of three Wi-Fi signal strength levels that will trigger notifications. Once a threshold level is crossed, the MIHF will propagate the appropriate notification to the MPA client.

3) **Link Parameter Report (Threshold 1)**:

When the MPA client receives the first event notification reporting that the Wi-Fi signal strength has crossed the first threshold, the MPA client prepares for a potential handover, queries the MIH information server (Steps 4 to 5) for available neighboring networks via the MN’s current serving network. The information server then sends a response with the information that the cellular network is available (Steps 6 to 7).

4) **Link Parameter Report (Threshold 2)**:

When the signal strength weakens further and the second threshold is crossed, the MPA client receives an event notification and starts setting up the cellular connection.

5) **Link Up Request**: The MPA client brings the EV-DO interface up and establishes an EV-DO connection using an MIH_Link_Actions command. It is important to note that this step can be performed after Step 10 if the IP address to be assigned to the EV-DO interface can be obtained in Step 10, however, this optimization will require the EV-DO network to support MPA.

6) **MPA Proactive Handover**: The MPA client starts pre-authentication, pre-configuration through the serving Wi-Fi interface.

7) **Link Parameter Report**: When the MPA client receives the third Link Parameter
Report event notification, indicating crossing the third lowest threshold value, the MPA client completes the handover operation via MOBIKE address update.

(13) **Link Power Down Request:** The MPA client then uses an MIH_Link_Actions command to bring down the Wi-Fi interface.

### 9.2.6 Network Initiated Handover with 802.21 triggers

Figure 9.5 (b) shows a sequence diagram for a Network-initiated handover from the Wi-Fi network to the EV-DO network. In addition to the entities depicted in Figure 9.5 (a), a new entity called the serving PoS (Point of Service) in the Wi-Fi network is used to realize a network-initiated handover.

1. **Subscribe Request:** The serving PoS subscribes with the MN to get an MIH_Link_Param_Report event notification, which will provide link parameter reports, when the Wi-Fi signal strength crosses a given value.

2. **Configure Threshold Request:** The serving PoS uses an MIH_Link_Configure_Threshold command to configure the Wi-Fi signal strength level that will trigger event notifications. Once a threshold level is crossed, the MIHF in the mobile node will propagate the appropriate notification to the PoS using the MIH Protocol to provide remote Event Services.

3. **Link Parameter Report:** When the serving PoS receives the event notification reporting that the Wi-Fi signal strength has crossed the specified threshold, the serving PoS queries the MIH information server (Step 4) for available neighboring networks. The information server then reports that the cellular network is available.

4. **Net HO candidate Request:** The serving PoS sends an MIH_Net_HO_Candidate_Query request message to the mobile indicating the candidate networks available for handover. The candidate networks are selected based on the information obtained from the Information Server in Step 4.

5. **MPA Pre-authentication:** Once the target PoS is selected and authentication server is known, the mobile node contacts the MPA server and starts pre-authentication, and sets
up the proactive tunnel through the serving Wi-Fi PoS.

(7) **Link Up Request:** The MPA client verifies the availability of the cellular network as indicated in the MIH.Net.HO.Candidate request message by bringing the EV-DO interface up and establishes an EV-DO connection using an MIH.Link.Actions command.

(8) **Net HO candidate Response:** Once the EV-DO connection is established, the MPA client responds with an MIH.Net.HO.Candidate.Query response message, indicating the EV-DO network as the candidate network.

(9) **N2N HO Query Resource Request/Response:** The serving PoS (Wi-Fi) sends the target PoS (CDMA) a N2N(Network to Network).HO.Query.Resource request message, to verify that the target PoS has resources before committing the handover. Once the serving PoS gets a positive response, it can commit to the handover. While MIH provides a command to indicate handover commitment (i.e., MIH.Net.HO.Commit), I effectively use the MPA proactive handover (Step 9) as the indication of the handover commitment.

(10) **MPA Proactive Handover:** The MPA client completes the handover operation by MOBIKE address update.

(11) **Link Power Down Request:** The MPA client then uses an MIH.Link.Actions command to bring down the Wi-Fi interface.

It is important that MIH handover preparation and MPA pre-authentication procedures complete before the mobile makes a handover to the target network.

### 9.2.7 Handover preparation time

The handover preparation time does not directly affect the handover performance and user experience. However, the amount of time the mobile needs to prepare for handover depends upon the speed of mobile (e.g., pedestrian, vehicular), cell size (e.g., pico cell, macro cell) and type of handover (e.g., single interface, multiple interface). Generally, it is important to reduce the handover preparation time to make the system more resilient to sudden changes in the network characteristics.
This handover preparation time in the experimental scenarios includes the following components:

1. (i) Propagation of the Link events from the link layer to the MIH user (i.e., local MIH user, in case of MN initiated handover and remote MIH user in Network-initiated handover)

2. (ii) Querying the IS database

3. (iii) MIHF internal operations

4. (iv) MPA layer 3 handover

During the experiment, I have measured the time delays for execution of the operations (ii), (iii) and (iv). While the delays I have measured are for different MIH related operations in the network initiated handover scenario described in Figure 9.5, some of these measurements can be applied to mobile initiated handoff scenario as well.

Tables 9.2, 9.3, 9.4 and 9.5 show the values measured for each of the above operations.

**Information Service Transaction Delay:** I measured different operations in the information server that constitute the transactions associated with a request. This sequence starts with receiving a *Get Information* request message containing an IS query and finishes by sending the corresponding response. Table 9.2 shows five values measured for each operation and their average. The average information server transaction execution time is 26.6 ms with lower bound of 13 ms and upper bound of 53 ms.

**MIH message composition and parsing delay:**

Depending on the MIH message type, the time for message composition and parsing might vary. This depends on the number of TLVs included in each message and the TLV type, which dictates the complexity of its composition and parsing. Tables 9.3 and 9.4 show the minimum, maximum and average values for the time taken for for different sub-operations associated with message composition and parsing delay, respectively. These values are taken into account for calculating the handover preparation time.
### Table 9.2: Processing time in the Information Server

<table>
<thead>
<tr>
<th>Measurement #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Info request parsing</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3.8</td>
</tr>
<tr>
<td>Pass indication from MIHF to MIH user</td>
<td>2</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3.8</td>
</tr>
<tr>
<td>Query processing</td>
<td>5</td>
<td>29</td>
<td>5</td>
<td>25</td>
<td>6</td>
<td>14</td>
</tr>
<tr>
<td>Get Info response composition</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2.8</td>
</tr>
<tr>
<td>Get Info response sending</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>2.2</td>
</tr>
<tr>
<td>Total time Processing in the Info server</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>26.6</td>
</tr>
</tbody>
</table>

### Table 9.3: MIH message composition time

<table>
<thead>
<tr>
<th>Measurement Point</th>
<th>Message Type</th>
<th>Execution Time (ms) (average, min, max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MN</td>
<td>Link Parameter Report Indication</td>
<td>1.6, 0, 2</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Register Response</td>
<td>4.4, 3, 8</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Subscribe Request</td>
<td>4.8, 3, 11</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Get Info Request</td>
<td>6.2, 5, 2</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Net HO Candidate Request</td>
<td>25.4, 10, 51</td>
</tr>
<tr>
<td>Info Server</td>
<td>Get Info Response</td>
<td>2.8, 2, 3</td>
</tr>
</tbody>
</table>
Table 9.4: MIH message parsing time

<table>
<thead>
<tr>
<th>Measurement Point</th>
<th>Message Type</th>
<th>Execution Time (ms) (average, min, max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MN</td>
<td>NET HO Candidate Query Request</td>
<td>12.6, 6, 19</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Subscribe Response</td>
<td>12, 7, 17</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Configure Threshold Response</td>
<td>40.2, 10, 54</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Link Parameter report Indication</td>
<td>21.2, 14, 50</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Get Info Response</td>
<td>11.4, 8, 17</td>
</tr>
<tr>
<td>Info Server</td>
<td>Get Info Request</td>
<td>3.8, 3, 5</td>
</tr>
</tbody>
</table>

**MIH performance for MPA triggering**

I measured the time it took to perform all the MIH related operations in our network initiated handover scenario that occurred starting with the initial handover trigger (i.e., crossing signal strength threshold in the MN and creation of the Link Parameter Report Indication) until triggering the MPA handover operation. Table 4 shows the average execution time of five measurements for each of the specified operations with the corresponding lower and upper bounds.

In order to calculate the total MIH MPA triggering operations, the following network propagation delays need to be added:

1. MN Serving PoS round trip propagation delay (MN-PoS-RTT).


In the current experimental testbed, I estimate these delays using round trip ping, which are 1.5 ms for MN-PoS-RTT and 0.3 ms for PoS-IS-RTT, bringing the MIH time to trigger MPA to 148.4 ms in the testbed environment.

These round trip propagation delays can be adjusted for a real network environment.
estimate a realistic network performance. Since the MN and its serving PoS are relatively close to each other, I estimate their round trip propagation delay, MN-PoS-RTT as 5 ms. I estimate the serving PoS-Information Server round trip propagation delay, PoS-IS-RTT as 30 ms. Thus, in a realistic network the time it would take for MIH to trigger the MPA pre-authentication and handover would be approximately = 146.6 ms + 5 ms (MN-PoS-RTT) + 30 ms (PoS-IS-RTT) = 181.6 ms. This time does not include the propagation of the link event from the link layer to the MIHF that I have not measured.

Table 9.5: Delays for MIHF related components

<table>
<thead>
<tr>
<th>Measurement point</th>
<th>Operation description</th>
<th>Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Min (ms)</td>
</tr>
<tr>
<td>MN</td>
<td>Compose/transmit Link Parameter Report Ind.</td>
<td>10</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Recv/parse/process Link Parameter Report Ind.</td>
<td>20</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Compose and transmit Get Info Request</td>
<td>11</td>
</tr>
<tr>
<td>Info Server</td>
<td>Receive/parser/process Get Info Request</td>
<td>10</td>
</tr>
<tr>
<td>Info Server</td>
<td>Compose/Send Get Info Resp.</td>
<td>3</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Receive/parser/process Get Info Response</td>
<td>10</td>
</tr>
<tr>
<td>Serving PoS</td>
<td>Compose/send Net HO candidate Req.</td>
<td>11</td>
</tr>
<tr>
<td>MN</td>
<td>Receive/process Net HO candidate Req.</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
</tr>
</tbody>
</table>

**Delays due to MPA operation**

MPA related delays are attributed to several factors such as delays due to pre-authentication, setting up proactive handover tunnels and sending the binding update for data redirection.
In the current testbed, I have measured delays for these components. As shown in Figure 9.5, pre-authentication and proactive tunnel setup took place before the PPP link was setup. Alternatively, these two operations could take place in parallel with PPP configuration operations that may take up to 2-5 seconds. Measurement shows that complete pre-authentication operation took about 2.175 ms. This time delay consists of several factors, such as four round trip signaling associated with EAP-GPSK (Extensible Authentication Protocol - Generalized Pre Shared Key), generation of keys at the authentication server and message processing delays at the end hosts. Proactive handover tunnel setup time was measured to be 4.730 ms that includes the time for IKE handshake to set up IPSec tunnel in ESP (Encapsulating Security Payload) mode, and initial MOBIKE signaling exchange. These two operations take place over the Wi-Fi interface in the previous network. Final step in the MPA operation is binding update and it is performed using MOBIKE address update mechanism. It took around 400 ms to complete the round trip MOBIKE signaling over a PPP link.

An estimation of the MIH handover preparation time before triggering MPA operation in a realistic network is less than 200 ms, which is less than 10 percent of the time MPA pre-authentication procedures would take. This seems to be a satisfactory time to allow proper timing of the MPA operation and handover procedure.

Information server transaction delay and MIHF performance can be improved by improving query execution time, message composition and message parsing time.

9.3 Handoff Optimization in IMS

In this section, I describe how some of the proactive optimization techniques can improve the handoff performance in IMS (IP Multimedia Subsystem). I have built a complete experimental prototype IMS system that I introduced in Chapter 5 to illustrate the route optimization technique. Here I demonstrate how the handoff delay is reduced when security
association is performed proactively by transferring the security context between points of attachment on two different networks. This specific system does not optimize layer 2 and layer 3 related operations but focus only on optimization due to security association.

### 9.3.1 Non-optimized handoff mode

In a regular non-optimized mode of operation, a new call context is created every time the mobile moves to the new network. During the handoff MN completes all the handoff functions at layer 2 and layer 3 as described earlier in Chapter 5. Specifically, after the MN establishes PPP access to the new network, it performs the MIP binding and it obtains the server configuration information via DHCP. Then the SIP related handoff functions are performed, starting with the SIP re-registration and the security association is re-establishment using AKA procedure. If the MN moves during an active session, session maintenance is carried out with the transmission of an encrypted SIP re-INVITE message that carries the SDP description of the ongoing session. Upon receipt of this message, P-CSCF (Proxy Call Session Control Function) creates a new call context for the same mobile and controls PDSN on the visited network to allow the traffic. This results in the resumption of the media in the new access network. The message flow for the non-optimized operational mode is provided in Figure 9.6.

### 9.3.2 Optimization with reactive context transfer

In reactive mode of operation, all the layer 2 and layer 3 related operations take place like non-optimized mode. The detailed message flow is provided in Figure 9.7. By comparing with Figure 9.6 (non-optimized mode) the difference between the two handoff operational modes are evident. In particular, the session maintenance information message (e.g., re-INVITE) that carries the SDP description of the active session does not play any role in context creation and thus does not affect the media handoff delay. The context created in the new visited networks P-CSCF is transferred from the old visited networks P-CSCF. The
objective of this approach is to reduce the handoff delay by eliminating the dependence on the session maintenance messages (INVITE and 200 OK).

After the radio handoff is over and PPP access is complete in the new network, MN performs the regular MIP binding and obtains the required configuration information using DHCP. MN initiates a SIP REGISTER message via the new P-CSCF. When this message reaches the S-CSCF (Serving Call Session Control Function), the S-CSCF informs the old P-CSCF to transfer the context of the active session to the new P-CSCF. At this point the old P-CSCF transfers the context to the new P-CSCF, and the context is created in the new network. After the completion of the SA setup via AKA (Authentication and Key Management) between MN and new P-CSCF, traffic is allowed at PDSN, and the session resumes.

### 9.3.3 Optimization with proactive security context transfer

Proactive mode of handoff minimizes the delay due to security association and context creation more than the reactive mode and thus, reduces the media interruption more than
the reactive mode. In this mode, the context creation in the new IMS network and security association with the new P-CSCF are completed while the MN is still in the old network. This technique works in conjunction with the proactive discovery of neighboring P-CSCFs ahead of time and accuracy of its movement profile. The carriers could use information service discovery methods like IEEE 802.21 or IEEE 802.11u to obtain the information about the neighboring networks and servers. Various techniques such as signal strength threshold can be used to determine mobiles precise movement pattern.

Figure 9.8 provides the detailed message flows of the proactive handoff. Prior to the MNs radio handoff, some of the handoff functions are done proactively in the old network. Specifically, the MN, utilizing the DHCP INFORM, acquires the addresses of P-CSCFs from the neighboring IMS networks.

In this case, DHCP server is equipped with the information about the servers in the neighboring domains. After the MN has identified the new neighboring network, it is likely to move, it informs its current P-CSCF about the address of its new P-CSCF. The current P-CSCF transfers the context of the active session (e.g., SDP, CDR information) to the new
Figure 9.8: Optimized handoff with proactive context transfer

P-CSCF. Similarly, a new security association is established between new P-CSCF and the mobile after the mobile sends a MoveNotify message to S-CSCF when the movement is imminent. This mechanism performs proactive AKA operation by transferring the security context from the current P-CSCF to the new P-CSCF ahead of time by creating a transient AKA during the handoff. The mobile performs regular AKA after it moves to the new network. Thus, the new PDSN opens its gate for this specific mobiles media even before the mobile has moved. After the mobile re-establishes its connection in the new network, and completes the MIP operation, media starts flowing eliminating the delay due to AKA procedure and context creation. Mobile’s SIP related signaling such as re-REGISTER or re-INVITE do not affect the media handoff delay here. Re-registration in the new network helps to renew the transient AKA.

Based on the message flow description, it is obvious that the proactive handoff operational mode is very promising. We compare the results of each handoff operation in the following section.
9.3.4 Performance results

The focus of this performance analysis is to highlight the relative effectiveness of proactive handoff compared to other two handoff techniques. In Figure 9.9, I plot the delays associated with the different handoff functions that contribute to the overall handoff delay for three different handoff scenarios. On the average, the mobile was subjected to 3,666 ms delay for proactive handoff, 9,685 ms delay for reactive handoff and 12,526 ms delay for non-optimized handoff. Number of packets lost is proportional to the handoff delay and depends on the packet generation rate.

Overall handoff delay consists of delays due to different operations such as layer 2 configuration, layer 3 configuration, binding update, registration, security association and media redirection. As is evident, proactive handoff does not contribute to any delay due to DHCP-based server discovery, context transfer and SIP-based security association compared to reactive or non-optimized case. On the other hand, non-optimized case is subjected to maximum delay due to additional signaling messages during SIP-based security association and context creation phase. Layer 2 handoff delay, PPPoE (PPPoE over Ethernet) access delay and MIP binding delay more or less remain the same for all three handoff scenarios. Similarly, in case of reactive handoff, besides layer 2 delay, major component of the delays came from SIP registration, security association and context transfer. Since I have used Mobile IP, these results are inclusive of inherent trombone routing delays and can further be reduced when the routing mitigation techniques described in Chapter 5 are applied. A different mobility protocol such as MIPv6 or SIP-based mobility may result in smaller binding update delay. SIP related signaling required for context transfer and security association contributes to additional handoff time for non-optimized case compared to reactive case, since it needs to create the context with an additional re-INVITE signaling.

In order to gain an insight into the effect of these optimization techniques in a real deployment scenario, I have taken some additional results. I used NIST delay simulator and varied the emulated distance between the home network and the visited network by intro-
CHAPTER 9. SYSTEMS EVALUATION

Figure 9.9: Comparison of optimized handoff components

Producing delays of 0 ms through 500 ms, with an increment of 50 ms. I provide additional handoff results in Table 9.10.

These results show only the components of the handoff delay that are affected due to additional delay introduced between the home network and the visited network. Since the experimental results indicate a basic trend, we show the second order approximation of the values by taking the values of the trend line. From the analysis it appears that delays related to layer 2 and layer 3 configuration do not get affected because these operations do not involve home network. Delay due to mobile IP binding update increases for all the three cases as the emulated distance between the home network and visited network is increased, but there is an appreciable increase in delay for SIP security association in case of non-optimized and reactive handoff. In proactive case, the additional network transport delay did not have any effect on the handoff delay component related to SIP, AKA and context transfer. The additional handoff delay in proactive case was contributed by the increased MIP update delay only.
Table 9.6: Effect of emulated distance on handoff components

<table>
<thead>
<tr>
<th>Types Of Handoff</th>
<th>SIP, AKA, Context Transfer Delay (ms)</th>
<th>MIP Update Delay (ms)</th>
<th>L2 PPP Delay (ms)</th>
<th>SIP, AKA, Context Transfer Delay (ms)</th>
<th>MIP Update Delay (ms)</th>
<th>L2 PPP Delay (ms)</th>
<th>SIP, AKA, Context Transfer Delay (ms)</th>
<th>MIP Update Delay (ms)</th>
<th>L2 PPP Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>51</td>
<td>2736</td>
<td>1,010</td>
<td>62</td>
<td>1523</td>
<td>3,999</td>
<td>41</td>
<td>2,329</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>152</td>
<td>2693</td>
<td>1,375</td>
<td>161</td>
<td>1744</td>
<td>4,584</td>
<td>145</td>
<td>2,217</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>252</td>
<td>2650</td>
<td>1,741</td>
<td>261</td>
<td>1964</td>
<td>5,170</td>
<td>248</td>
<td>2,194</td>
</tr>
<tr>
<td>150</td>
<td>0</td>
<td>352</td>
<td>2607</td>
<td>2,107</td>
<td>360</td>
<td>2184</td>
<td>5,756</td>
<td>352</td>
<td>2,172</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>453</td>
<td>2563</td>
<td>2,472</td>
<td>459</td>
<td>2405</td>
<td>6,342</td>
<td>455</td>
<td>2,150</td>
</tr>
<tr>
<td>250</td>
<td>0</td>
<td>553</td>
<td>2520</td>
<td>2,838</td>
<td>558</td>
<td>2625</td>
<td>6,927</td>
<td>559</td>
<td>2,128</td>
</tr>
<tr>
<td>300</td>
<td>0</td>
<td>654</td>
<td>2477</td>
<td>3,203</td>
<td>658</td>
<td>2845</td>
<td>7,513</td>
<td>663</td>
<td>2,106</td>
</tr>
<tr>
<td>350</td>
<td>0</td>
<td>755</td>
<td>2434</td>
<td>3,569</td>
<td>757</td>
<td>3066</td>
<td>8,099</td>
<td>766</td>
<td>2,084</td>
</tr>
<tr>
<td>400</td>
<td>0</td>
<td>855</td>
<td>2391</td>
<td>3,935</td>
<td>856</td>
<td>3286</td>
<td>8,685</td>
<td>870</td>
<td>2,061</td>
</tr>
<tr>
<td>450</td>
<td>0</td>
<td>956</td>
<td>2347</td>
<td>4,300</td>
<td>955</td>
<td>3506</td>
<td>9,270</td>
<td>973</td>
<td>2,039</td>
</tr>
<tr>
<td>500</td>
<td>0</td>
<td>1,057</td>
<td>2304</td>
<td>4,666</td>
<td>1,055</td>
<td>3726</td>
<td>9,856</td>
<td>1077</td>
<td>2,017</td>
</tr>
</tbody>
</table>

9.4 Systems validation using Petri net-based models

In this section, I introduce Petri net-based models for some of the optimization techniques that I have described in Section 5. I apply MATLAB-based Petri net tool [MMP] to model some of the handoff functions as described in Chapter 3 and validate the optimization techniques by way of Petri net-based behavioral analysis methods and evaluate the systems performance by using cycle time and Floyd algorithm. I then use three different scheduling techniques for few of the handoff operations and apply Cycle time and Floyd Algorithms approaches to validate the systems performance. I also illustrate how certain sequence of transitions may give rise to deadlocks by doing a reachability and matrix analysis.

9.4.1 MATLAB-based modeling for handoff functions

Here I describe the results from MATLAB-based modeling of many of the handoff functions. This MATLAB-based Petri net tool is used to study the behavioral properties such as reachability analysis, markings, liveness and systems performance of the handoff models.

Figure 9.10 shows the MATLAB-based model to illustrate the sequence of a few of the handoff operations, such as discovery, attachment, configuration, and authentication.
This is MATLAB equivalent of the model shown in Figure 4.14. Places P13, P14 and P15 represent resource places representing battery power, bandwidth and CPU cycles, respectively. It also shows the markings associated with these sets of models as generated from the MATLAB model. These markings represent the step-wise execution of the handoff events.

![MATLAB-based model of four handoff functions](image)

Figure 9.10: MATLAB-based model of four handoff functions

Similarly, Figures 9.11, 9.12, 9.13 illustrate MATLAB-based models for three different handoff sequences as shown in Figures 4.17, 4.18 and 4.19, respectively. These models illustrate coverability trees and markings as obtained from the MATLAB-based Petri net Tool. Matrices $A_i$, $A_o$ and $A$ represent the input, output and incidence matrix for the petri net handoff model shown in Figure 9.11. These matrices are obtained from the MATLAB-based petri net model. As discussed in Section 4, many of the behavioral properties of the handoff functions can be obtained from these matrices using matrix analysis method.
M130 = [0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,1,0,0,1,1]
M131 = [0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,1,1]
M132 = [0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,1,0,1,1]
M133 = [0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,0,1,0]
M134 = [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,0,1,1]
M135 = [0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,0,0,0,1,1]
M136 = [0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,1,0,0,0,1,1]
M137 = [0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,1,1,0,1,1]
M138 = [0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,0,0,1,1,1]

Figure 9.11: Sequential handoff operations

\[
A_i = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &.....
Figure 9.12: Concurrent security and scanning operations
Figure 9.13: Concurrent security, subnet and scanning operations
9.4.2 Petri net-based model of optimized security association

As discussed in Chapter 5.4, addition of an external home agent reduces the number of signaling messages exchanged between the mobile and the VPN gateway when the mobile changes its IP address due to handover. Thus, while the mobile does not need to re-establish a new security association, it still needs to set up an additional Mobile IP tunnel with the external home agent. Thus, there is a tradeoff between additional resources needed due to an additional external home agent and avoidance of extra signaling due to re-establishment of security association.

Figure 9.15 shows the protocol flows associated with both the cases, one when mobile needs to re-establish the security association and one when the mobile does not need to re-establish the security association at the cost of additional external home agent. I derive the Petri net models for both of the systems, one with external home agent and one without the external home agent and evaluate the systems performance of each of the systems based on the experimental results.

Figure 9.16 shows the equivalent Petri net models corresponding to the call flows for
two different scenario as shown in Figure 9.15. Section 5.4 demonstrates the experimental results and highlights how delay and packet loss are reduced when an external home agent is introduced as an anchor agent to help maintain the security association even when the IP address changes. Experimental results comparing the optimized approach with non-optimized version is described in Section 5.4. Table 9.7 shows the results of different operations from the experimental setup. In particular, timings for IKE signaling exchange, security context establishment, tunnel creation, tunneling and detunneling operations for MIP and IPSec, binding update time by the external home agent are applied to the Petri net model in order to determine the systems performance evaluated using cycle time and Floyd algorithm.

In order to evaluate the optimized system, any of the three Petri net-based methods described in Section 4 can be applied. I verify the above optimized system by using a matrix-based solution as described in Section 4.9. It also demonstrates that in order to achieve the desired performance, the system needs to utilize more resources, such as an additional home agent that introduces triple encapsulation.
Figure 9.16: Petri net model for security association

Table 9.7: Timings with security association during handoff

<table>
<thead>
<tr>
<th>Transition</th>
<th>Handoff operation</th>
<th>Time taken for operation (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>IKE exchange</td>
<td>30</td>
</tr>
<tr>
<td>t2</td>
<td>Security context establishment</td>
<td>400</td>
</tr>
<tr>
<td>t3</td>
<td>VPN tunnel creation</td>
<td>6000</td>
</tr>
<tr>
<td>t4</td>
<td>MIP tunnel creation</td>
<td>10</td>
</tr>
<tr>
<td>t5</td>
<td>Tunneling/de-tunneling Mobile IP data</td>
<td>5</td>
</tr>
<tr>
<td>t6</td>
<td>Tunneling/de-tunneling VPN data</td>
<td>60</td>
</tr>
<tr>
<td>t7</td>
<td>External MIP update</td>
<td>300</td>
</tr>
</tbody>
</table>

9.4.3 Petri net-based model for hierarchical binding update

In Section 5.7.3, I have experimentally shown how hierarchical binding update reduces the delay contributed by the global binding update using network layer and application layer mobility protocols. In order to achieve that, I introduce an additional anchor point closer to the mobile that takes care of hierarchical binding update. Here, I introduce an equivalent Petri net model that demonstrates hierarchical binding update mechanism. Matrix analysis method can be used to study the behavior of the optimized model and evaluation of systems performance can be done by cycle time and Floyd algorithm. Figure 9.17 shows the flow
for inter-domain and intra-domain binding update and the respective petri net models.

**Figure 9.17: Petri net model with hierarchical mobility agent**

### 9.4.4 Petri net-based model for redirection of inflight data

In Section 5.8, I have experimentally demonstrated how using different optimization techniques packet loss due to media redirection delay can be reduced. Figure 9.18 shows an equivalent Petri net model for one of these optimization techniques (e.g., mobility proxy-based approach). This model also gives the ability to verify the correctness of this optimization technique and evaluate the systems performance.

### 9.4.5 Petri net-based model of optimized configuration

As described in Section 5, Duplicate Address Detection (DAD) process takes the most amount of time during layer 3 configuration. There are several ways to reduce the time taken by DAD process as described in Section 5. I use Petri net to verify one specific mechanism where duplicate address detection is performed during the identifier acquisition phase. This specific mechanism eliminates the time taken by the neighbor discovery
process that is often performed by the client after the address is obtained. However, this mechanism adds extra load to the server, as an intermediate server needs to collect the information about the addresses that are being used (some of those addresses could be statically configured and some may have been configured using DHCP) and the network also requires additional amount of bandwidth due to the periodic multicast announcement by the server that carries the addresses that are already being used in the network. The specific model has the ability to verify that the specific optimization technique is deadlock free and can reduce the time for DAD process at the expense of additional resources. If there are not enough resources, then there will be deadlock in the system. Figure 9.19(a) shows the protocol flow for optimized DAD mechanism and Figure 9.19 (b) shows the associated Petri net model.

Similarly, equivalent Petri net models can be derived for any of the optimization techniques discussed in Chapter 5. Then, these Petri net models can be used to derive the systems performance and behavioral properties of the optimized systems using any of the methods described in Chapter 4.
9.5 Scheduling handoff operations

In this section, I describe different ways these primitive handoff operations can be scheduled and evaluate the overall systems performance for three different schedules. A timed Petri net can be easily adapted to illustrate the scheduling of handoff operations. One basic approach is through the use of heuristic search for the optimal or near optimal schedule in the reachability tree of the Petri net model. Scheduling techniques can help derive specific schedule to optimize certain performance index such as handoff delay under certain systems resource constraints.

Scheduling of a handoff operation needs to take into account the following guidelines.

1. Required systems performance such as handoff delay and packet loss are usually achieved under certain resource constraints. Thus, minimum cycle time achieved under a specific schedule is considered to be equivalent to maximum systems performance.

2. A specific handoff schedule should not suffer from any deadlock condition where a specific operation cannot proceed because of non-availability of data from the previous operation or because of lack of resources.
3. Both the resources and precedence relationship among the events need to be modeled to allow maximum flexibility. Thus, the Petri net model depends on both the resources required and the precedence graph. Van Bruseel et al. [VBPV93] illustrate how scheduling in FMS (Flexible Manufacturing Systems) can be affected by the resource constraints and precedence relationship among the events. Data dependency among the handoff events and resource availability in the system during handoff operation can determine the extent of parallelism that is possible among the handoff related operations.

In Petri net models, a cycle time represents the delay during the handoff process under resource constraints and thus can be attributed to the overall efficiency of the system. However, scheduling of execution of these processes that are part of this mobility event plays an important role in determining the overall cycle time and systems performance.

A Petri net-based model can be used to analyze various types of mobility events, such as intra-subnet, intra-technology, inter-subnet, and inter-technology handoff. Corresponding Petri net based optimization models can be derived by applying the optimization techniques to the generalized mobility model. These techniques can be applied to the processes that are part of the overall system or in a hierarchical manner to each of these sub-processes. In this section, I primarily categorize handoff optimization techniques based on sequential, concurrent, and proactive modes of scheduling of events and model these in Petri nets. Depending upon the type of scheduling technique, the systems resources expensed during a specific operation will vary over a period of time. While a sequential handoff operation takes more time compared to a proactive or concurrent operation, the optimized models using concurrent or proactive operations will need to exhaust more resources for a given period of time.

In order to conduct a performance analysis, I have initially considered two handoff related operations, namely discovery and authentication. I apply three different scheduling mechanisms to schedule these two handoff related operations and study the overall
CHAPTER 9. SYSTEMS EVALUATION

performance. Figures 9.20, 9.21, and 9.22 illustrate how these two specific handoff operations in IEEE 802.11 environment can be represented in a Petri net model using sequential, concurrent and proactive optimization techniques, respectively. Optimality of the system performance is obtained by comparing the handoff performance (cycle time) and resource utilization (number of tokens) for these handoff methodologies.

9.5.1 Sequential scheduling

Figure 9.20 shows a Petri Net model that represents when discovery and authentication related operations are performed in sequence. Tokens represent the resources of different types. The number of tokens needed for each type of operation varies depending upon the amount of resources needed during each of these operations.

In general, scanning is part of layer 2 discovery process and is followed by layer 2 authentication, 4-way handshake and finally, the association with the layer 2 access point. P0, P1, P2, P3 and P4 are places that represent different states of discovery and authentication. Shared resources are represented by places such as $P_B$, $P_M$ and $P_P$ that represent bandwidth, memory and processing power, respectively. Number of tokens in these shared places represent the amount of resources expended during each of these operations. For example, one token can represent 100 kb of bandwidth for resource place $P_B$ that represents the shared resources of bandwidth.

9.5.2 Concurrent scheduling

Figure 9.21 shows the Petri net model when two of the handoff related operations, scanning and authentication take place concurrently at the expense of additional bandwidth resources due to extra signaling messages. This speeds up the overall handoff operation but consumes more shared resources during a given time.


9.5.3 Proactive scheduling

Figure 9.22 illustrates the Petri net model where some of the handoff related operations are performed proactively. The mobile intends to move from its current network to the target network. Many of the handoff related operations such as discovery of the target network elements and authentication with the target network elements are performed ahead of time while the mobile is in the current network. Thus, the shared resources (e.g., access bandwidth) are utilized in the current network and some additional resources are used due to operations such as tunneling and proactive IP address caching. $P_{B1}$, $P_{M}$ and $P_{D}$ are shared resources that are used in the current network and $P_{B2}$ and $P_{P}$ are shared resources expensed in the target network.

9.6 Verification of systems performance

The systems performance of a Petri net model for a mobility event can be verified in several ways. We illustrate two scenarios here.

In one scenario, the minimum cycle time can be obtained from the Petri net model
Figure 9.21: Concurrent handoff operations

Figure 9.22: Proactive handoff operations
by investigating the number of circuits, the number of transitions, and delay associated with each transition. Thus, the performance requirement expressed in cycle time $C$ can be satisfied if and only if $CN_k - T_k \geq 0$ for all circuits in the net. In the second scenario, the token loading matrix and transition matrix are obtained based on the markings of the Petri net model and the associated values of the transitions. Then, the Floyd algorithm is applied to validate the systems performance by computing the shortest distance between every pair of places.

Similar methods can be applied to compute the cycle time and overall systems performance of the handoff event demonstrating different types of optimization techniques, such as hierarchical binding update, proactive discovery and configuration, and anchor-based security association that are modeled using generalized Petri net. I calculate the cycle time and verify the systems performance based on three scheduling techniques when applied to two basic handoff operations – discovery and authentication. Experimental results of these two operations were used for Petri net modeling.

### 9.6.1 Cycle-time-based approach

Table 9.8 shows the transition times $t1$, $t2$, $t3$, $t4$ and $t5$ for different primitive handoff operations associated with discovery and scanning processes obtained from our experiments [LDOS07].

<table>
<thead>
<tr>
<th>Transition</th>
<th>Handoff operation</th>
<th>Time taken for operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>Disconnection trigger</td>
<td>5 ms</td>
</tr>
<tr>
<td>t2</td>
<td>Scanning</td>
<td>400 ms</td>
</tr>
<tr>
<td>t3</td>
<td>Authentication</td>
<td>50 ms</td>
</tr>
<tr>
<td>t4</td>
<td>4-way handshake</td>
<td>10 ms</td>
</tr>
<tr>
<td>t5</td>
<td>Association</td>
<td>5 ms</td>
</tr>
</tbody>
</table>

It is assumed that there is a handoff delay requirement of 100 ms to support a real-time application. I evaluate the overall cycle time when different schedules are applied to
Table 9.9: Cycle time from Petri net

<table>
<thead>
<tr>
<th>Optimization Schedule</th>
<th>Relevant loop in Petri net</th>
<th>D_i</th>
<th>N_i</th>
<th>Max D/N_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>p0t1p1t2p2t3p3t4p4t5p0</td>
<td>470</td>
<td>1</td>
<td>470</td>
</tr>
<tr>
<td>Concurrent</td>
<td>p0t1p1t3p3t4p0</td>
<td>420</td>
<td>1</td>
<td>420</td>
</tr>
<tr>
<td>Proactive</td>
<td>P1t1p2t4p3t5p1</td>
<td>17</td>
<td>1</td>
<td>17</td>
</tr>
</tbody>
</table>

these handoff operations and verify if the system is conformant to the delay requirement. Table 9.9 shows the cycle time for three different handoff sequences involving discovery and security association of the handoff. It appears from the results that although time for concurrent operation is smaller than the time for sequential operation, proactive operation is the only operation that satisfies the delay bound of 100 ms under these resource constraint.

9.6.2 Using the Floyd algorithm

In order to verify the systems performance of a mobility event and to determine if a specific system satisfies the desired requirement for cycle time, one can generate a Place matrix (P), Transition matrix (Q) from the Petri net model. I describe briefly how the elements of these matrices are formed. Entry (A, B) in the matrix P equals “x” if there are “x” tokens in place A, and place A is connected directly to place B by a transition. Entry (A, B) in the transition matrix “Q” equals t_i if A is an input place of transition t_i, and B is one of its output places. Entry (A, B) contains symbol “w” if A and B are not connected directly. Given a threshold value of cycle time C, one can generate a distance matrix CP-Q. Then, using the Floyd algorithm, one can determine matrix S. By inspecting the values of the diagonal elements of matrix S, it is possible to determine if the system satisfies the desired system performance. There are three cases to figure out the systems performance of the given system: 1) If all diagonal entries of matrix S are positive (i.e., CN_k - T_k > 0 for all circuits), the system performance is higher than the given requirement, 2) if some diagonal entries of matrix S are zeros and rest are positive (i.e., CN_k - T_k = 0 for some circuits and
CN_k - T_k > 0 for the other circuits) the system performance just meets the given requirement,
3) if some diagonal entries of matrix S are negative (i.e., CN_k - T_k < 0 for some circuits),
the system performance is lower than the given requirement.

Equations 9.1 and 9.2 represent the matrices when the Floyd algorithm is applied to verify the systems performance of the mobility event using sequential scheduling and equation 9.3 represents the matrices illustrating the proactive scheduling as shown in Figure 9.18. Values from the mobility event involving discovery and authentication are used to build the token loading matrix P and transition time matrix Q. First element for P matrix is P_{00} and the last element is P_{77}. Distance matrix and S matrix are then derived from these two matrices. By inspecting the values of the S matrix in equation 9.2 that reflect sequential scheduling, it is found that at least one of the diagonal elements is negative. Thus, this specific sequential scheduling cannot meet the desired systems performance of cycle time of 100 ms. Thus, in order to meet the desired performance level, faster facilities could be used to speed up the transition time or more tokens (resources) could be used in the shared places, thereby increasing the level of concurrency.
Equation 9.3 shows the matrices for proactive scheduling based on the transition times obtained from the experiments. By inspecting matrix $S$ in equation 9.3, it appears all the diagonal elements of the matrix are positive. Thus, by applying the Floyd algorithm, it is verified that the proactive scheduling, when applied to discovery and authentication processes, satisfies the systems performance of required minimum cycle time of 100 ms. I have also used several automated tools such as TimeNet [ZGHF99], STPNplay [Ryu04] and Petri net Tool [MMP] to model the behavior of the handoff system, capture the systems performance and evaluate the performance characteristics of the mobility protocols and associated optimizations.
9.7 Deadlocks in handoff scheduling

Scheduling of handoff operations needs to take into account the data dependency and resources. In this section, I describe how a specific handoff schedule might give rise to a deadlock in the system and ways to avoid the deadlocks by changing the schedules or adding resources.

Reachability analysis has been explained in Chapter 4. Reachability analysis is one way of detecting the deadlocks in the handoff system. If a specific schedule generates a coverability tree so that no subsequent transition is allowed at a specific marking (e.g., \( M_i \)), then that specific sequence exhibits a deadlock. Thus, by inspecting the coverability tree of any schedule it is possible to determine if a deadlock exists within a certain specific schedule.

I illustrate a few scenarios that exhibit how deadlocks take place during the handoff process. Then, I propose solutions to take care of this deadlock situation. I use MATLAB-based Petri net tool to construct the coverability trees and verify the deadlock properties.

9.7.1 Handoff schedules with deadlocks

In this section, I illustrate few specific handoff schedules that exhibit a deadlock situation and propose the solutions that will avoid these deadlocks.

If a specific schedule generates a coverability tree so that no subsequent transition is allowed at a specific marking (e.g., \( M_i \)), then that specific sequence exhibits a deadlock. Thus, by inspecting the coverability tree associated with any schedule, it is possible to determine if a deadlock exists in a specific schedule.

I describe few specific scenarios to compare two schedules one with deadlock and one without deadlock.

**Deadlocks due to lack of data:**

A specific handoff schedule could also lead to a deadlock if the sequence of operations
do not follow the data dependency graph. In a normal situation, if the mobile configures its layer 3 identifier and assigns it to the interface as part of layer 3 handover, without finishing L2 handoff (e.g., channel change) operation, the mobile cannot complete the rest of the handoff operations it will lead to a deadlock.

**Deadlocks due to resource sharing:**

A second scenario could be due to the result of concurrent operations resulting in lack of resources. For example, during a concurrent operation a handoff schedule is designed to perform both the operations, layer 2 discovery and authentication in parallel (e.g., layer 2 discovery process starts the authentication process when the former process is still not completed). If there are not enough tokens available in the bandwidth resource place (Pb) that can enable the transition for authentication process after the transition for discovery process is enabled, then the authentication process cannot proceed further.

Figure 9.23 shows a deadlock situation resulting out of lack of resources during a concurrent operation. By investigating the coverability tree, it shows that the system does not get back to its initial state due to deadlock.

![Coverability Tree Diagram](image)

Figure 9.23: Deadlock due to resource constraints

**Deadlocks in simultaneous mobility**
CHAPTER 9. SYSTEMS EVALUATION

Another deadlock scenario would arise due to failure of binding update during simultaneous mobility, where binding update from either client does not complete due to overlapping handoff of the mobiles. The rest of the handoff operations cannot proceed due to non-completion of binding updates for either of the clients. I have described the details of simultaneous problems in Chapter 7. Here, I illustrate Petri net modeling resulting out of simultaneous mobility scenarios. Figure 9.24 shows an equivalent Petri net model of Figure 7.1 where there is no problem due to simultaneous mobility and both the mobiles can communicate with each other. It is apparent that a successful communication between both the mobiles depend on successful completion of configuration and binding updates of both the mobiles.

![Petri net model for simultaneous mobility and Coverability tree](image)

Figure 9.24: (a) Petri net model for simultaneous mobility (b) Coverability tree

However, Figure 9.25 illustrates a Petri net example of a deadlock situation in simultaneous mobility because one of the mobile keeps getting reconfigured.
9.7.2 Deadlock prevention and avoidance in handoff schedule

There are several ways the deadlock situations can be prevented or avoided. Viswanadham et al. [VNJ90] discuss about the deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models. Similar techniques can be applied for prevention or avoidance of deadlocks in the handoff system. Deadlock prevention consists of falsifying one or more of those necessary conditions mentioned in Chapter 4 by using static resource allocation policies so that the deadlocks are completely eliminated. Each of the above deadlock scenarios can be avoided either by adding resources, changing the schedule or by introducing additional components in the network. Reachability graph of a Petri Net model representing a handoff system can be used to arrive at the resource-allocation policies that enforce deadlock prevention.

Since deadlock prevention is accomplished by static policies that is known to result in poor resource utilization and reachability analysis technique to arrive at deadlock prevention policies can become infeasible if the state space is very large, deadlock avoidance techniques are preferred sometimes. Deadlock avoidance techniques attempt to falsify one
or more of the necessary conditions in a dynamic way by keeping track of the current state and possible future conditions. The idea is to let the necessary conditions prevail as long as they do not cause a deadlock but falify them as soon as a deadlock becomes a possibility in the immediate future. Deadlock avoidance leads to better resource utilization.

Deadlocks in scenario 1 can be prevented two ways. In one way deadlock can be prevented by scheduling the handoff operations so as not to enable a transition unless there is data available from the previous operation. Thus, the mobile should not be scheduled to perform layer 3 operation unless layer 2 handoff operation is over. Second case involves introduction of additional operation to avoid the handoff. In the second case, unless a transient tunnel between the mobile and the next hop router is established the rest of the operations such as binding update and media forwarding to the mobile cannot complete leading to mobile’s incomplete handoff. Thus, setting up this additional tunnel is an additional operation that is needed to avoid the possible deadlock in such situation.

One way to prevent the deadlock illustrated in Figure 9.13 is to increase the number of tokens in resource place Pb (e.g., Bandwidth resources) to take care of the concurrency.

Figure 9.26 illustrates how the existing deadlock situation due to lack of resources as shown in Figure 9.23 is taken care of by adding additional resources. It shows how allocation of the additional resources have resulted in elimination of deadlock. I verify that a schedule is deadlock free by doing a reachability analysis. As shown in the coverability tree, in the absence of deadlock, the mobile comes back to its initial state.

Deadlocks in simultaneous mobility scenario can be avoided by installing additional components in the network such as proxies in the network as explained in Chapter 7. Figure 9.27 illustrates the MATLAB-based model and corresponding coverability tree that shows how the deadlock in simultaneous mobility is avoided by introducing retransmission technique or forwarding agent.

Similarly, MATLAB-based models can be used to construct the equivalent coverability tree to determine the presence of deadlocks in the systems where the handoff operations do
not follow proper precedence rules.

## 9.8 Petrinet-based modeling for multi-interface mobility

I illustrate Petri net modeling involving handover between two different types of access networks (e.g., 802.11 and CDMA) covering three scenarios, namely, break-before-make, make-before-break and parallel operations of both the interfaces. As described in Table 3.1 in Chapter 3, each access network has different ways of discovering resource and network parameters; authentication mechanisms and encryption algorithms are also different for each access network. Thus, resource requirement (e.g., battery, bandwidth and CPU) for each of these handoff operations in either access network are different.

When a multi-interface mobile has the coverage for both CDMA and 802.11, both the interfaces start getting configured at the same time and consume battery and CPU resources on the mobile, while bandwidth resources are consumed in each access network separately. Ideally, based on certain policy, the mobile makes a decision regarding which interface should be used to communicate with the other communicating node.
Figure 9.27: (a) Deadlock avoidance simultaneous mobility (b) Coverability tree

During a break-before-make handoff scenario, one interface does not start coming up until the other interface is disconnected. During a break-before-make scenario where one type of interface (e.g., CDMA) comes up after the other interface (e.g., 802.11) goes down, the resource consumptions will be different based on if the mobile is handing over from 802.11 to CDMA or vice-versa.

During a make-before-break operation, when the mobile is still communicating using 802.11 interface, CDMA interface gets into the process of getting activated as the signal-to-noise ratio on 802.11 interface begins to deteriorate. While this helps to reduce the handoff delay it needs more resources in the mobile. Thus, it is an important point to consider when to start the activation of CDMA interface based on the resource availability in the network.

Table 9.10 shows the amounts of resources and time needed to take care of specific operations for two different types of interfaces, namely, CDMA and 802.11. Since these two interfaces have different access characteristics, amounts of resources and timing to complete each of these handoff operations also varies.

Figure 9.28 shows a MATLAB-based model to illustrate a scenario when the mobile is under the coverage of both 802.11 access and CDMA access. Thus, in this scenario, both
of the interfaces will begin its configuration independently. However, since each interface will have its dedicated access network, a single bandwidth (channel resource) place cannot be shared for operations related to both the interfaces, where as battery power and CPU samples would be considered as shared resources for both. Thus, two different bandwidth places can be modeled, one for each type of access network. Each bandwidth resource will be shared among the events within that specific access network. For example, channel resources for identifier configuration for each type of the networks will be shared by channel resources of each type of networks.

Figure 9.28: MATLAB-based model for parallel CDMA and 802.11 operations

Figure 9.29 shows the MATLAB-based model for a multi-interface mobility scenario when the mobile is communicating using its 802.11 interface and as the signal-to-noise ratio
of 802.11 interface starts decreasing, CDMA interface is in the process of being connected.

Figure 9.29: (a) MATLAB-based model for make-before-break (b) Coverability tree

Figure 9.30 shows the MATLAB-based model for a multi-interface mobility scenario where only one interface (e.g., 802.11 or CDMA) is active at any point of time and CDMA interface prepares its interface only after the 802.11 interface is disconnected. Since both the interfaces are not active at the same time, it does not consume that much resources compared to previous two scenarios but this scenario takes the most amount of time for handoff to complete. Unlike the make-before-break case shown in Figure 9.29 here the CDMA interface goes through the process of getting connected only after the 802.11 interface is disconnected at the crossing of a specific signal-to-noise threshold.

### 9.9 Tradeoff analysis: Resources vs. Handoff performance

As discussed in Chapter 4, several types of systems resources are utilized during a handoff operation. Although proactive and handoff operations offer better performance, it utilizes systems resources while the mobile is in the current network and is engaged in these proactive operations. I give some examples illustrating tradeoff between handoff delay and
resource utilization when different levels of proactive operations are put in place.

In case of proactive operations involving multiple candidate target networks, establishing multiple tunnels with the neighboring target networks increases successful handover. However, it also needs more systems resources because of many of the handoff related operations such as tunneling, pre-configuration and pre-authentication processes involved during the proactive process. Pre-authentication process with multiple candidate target networks can happen in several ways.

The very basic level of pre-authentication involves authenticating the mobile with the multiple authentication agents in the neighboring networks, but actual pre-configuration and binding update take place only after layer 2 movement to a specific network is complete.

Similarly, in addition to basic pre-authentication, the mobile can also complete the pre-configuration while in the previous network, but can postpone the binding update until after the mobile has moved to the new network. Like the previous case, in this case the mobile also does not need to set up the pre-configured the tunnels since binding update is actually done after the mobile has moved to the new network.
CHAPTER 9. SYSTEMS EVALUATION

The third type of multiple pre-authentication involves all the three processes to complete while the mobile is in the previous networks, such as authentication, configuration and binding update. But, this specific type of pre-authentication utilizes the most amount of resources.

Some of the resources that are utilized during the preauthentication process are as follows:

1. Additional signaling to complete pre-authentication in the neighboring networks

2. Caching the IP address of the neighboring networks in mobiles for certain amount of time. It needs additional processing in the mobile for storing these IP addresses. In addition, it also uses up the temporary IP addresses from the neighboring networks by keeping these addresses in the cache.

3. There is an additional cost associated with setting up additional transient tunnels between the mobile and with the target routers in the neighboring networks.

4. Binding update with multiple IP addresses obtained from the neighboring networks result in multiple transient data streams between the CN and mobile over these transient tunnels.

When only pre-authentication and pre-configuration are done ahead of time with multiple networks, the mobile sends one binding update to the CN or the home agent.

In case binding update with multiple contact addresses is sent, multiple media streams are forwarded from the CN over the transient tunnels. However, in that case, the mobile needs to send another binding update after the handover with the contact address set to the new care-of-address address where the mobile has moved. This way the CN stops sending media to other neighboring networks where it does not end up moving.

The following is an illustration of this specific case when the mobile sends multiple binding updates while in the previous network but ends up moving to a specific target
network. MN sends a binding update to CH with multiple potential care-of-addresses such as c1, c2, and c3 that were obtained from three neighboring networks. This allows the CN to send transient multiple streams to the mobile over the pre-established tunnels. After the mobile moves to a specific target network, it sends another binding update to the CN with the care-of-address of the mobile in the network where the mobile has moved in. Some of the issues with multiple stream are consumption of extra bandwidth for a small period of time.

Alternatively, one can apply the buffering technique at the target access router or at the home agent. Transient data can be forwarded to the mobile after it has moved in. Forwarding of data can be triggered by the mobile either as part of Mobile IP registration or as a separate buffering protocol.

I discuss some guidelines for the roaming clients that use pre-authentication mechanisms to reduce the handoff delay. These guidelines can help determine the extent of pre-authentication operation that is needed based on a specific type of movement of the client. IEEE 802.11i and 802.11r take advantage of preauthentication mechanism at layer 2. Thus, many of the guidelines observed for 802.11i-based pre-authentication and 802.11r-based fast roaming could also be applicable to the clients that use MPA-based pre-authentication techniques. However, since MPA operations are not limited to a specific subnet and involve inter-subnet and inter-domain handover the guidelines need to take into account other factors such as movement pattern of the mobile, cell size etc.

Time needed to complete pre-authentication mechanism is an important parameter since the mobile node needs to determine how much ahead of time the mobile needs to start the preauthentication process so that it can finish the desired operations before the handover to the target network starts. The time needed to complete pre-authentication operations will vary depending upon the speed of the mobile (e.g., pedestrian, vs. vehicular), cell sizes (e.g., WiFi, Cellular). Cell residence time is defined as the average time the mobile stays in the cell before the next handoff takes place. Cell residence time is dependent upon the
coverage area and velocity of the mobile. Thus, cell residence time is an important factor in determining the desirable pre-authentication time that a mobile should consider.

Since pre-authentication operation involves six sub-operations as described earlier in the chapter and each sub-operation takes some discrete amount of time, only part of these sub-operations may be completed before the handoff depending upon the available delay budget.

For example, a mobile could complete only network discovery and network layer authentication process before the handoff and postpone the rest of the operations to until after the handover is complete. On the other hand, if it is a slow moving vehicle and the adjacent cells are sparsely spaced, a mobile could complete all the desired MPA related operations. Finishing all the MPA related operations ahead of time reduces the handoff delay but adds other constraints such as cell residence time.

I give a numerical example here for a pre-authentication process.

\[ D = \text{Coverage diameter,} \]
\[ v = \text{Mobile's velocity,} \]
\[ RTT = \text{round trip time from AP to AAA server including processing time for authentication Tauth} \]
\[ T_{psk} = \text{Time spent to install keys proactively on the target APs} \]

If for a given value of \( D = 100 \text{ft}, T_{psk} = 10 \text{ ms}, \text{and } RTT = 100 \text{ ms}, \text{if a mobile needs to do only pre-authentication procedure associated with MPA, then the following can be calculated for a successful MPA procedure before the handoff is complete.} \]

\[ 2RTT + T_{psk} < D/v \]
\[ v = 100 \text{ ft}/(200 \text{ ms} + 10 \text{ ms}) = 500 \text{ ft/sec} \]

Similarly, for a similar cell size, if the mobile is involved in both pre-authentication and pre-configuration operations as part of the MPA procedure, and it takes an amount of time \( T_{config}= 190 \text{ ms} \) to complete the layer 3 configuration including IP address configuration, then for a successful MPA operation,
\[2 \text{RTT} + T_{pk} + T_{conf} < \frac{D}{v}\]

\[v = \frac{100 \text{ ft}}{(200 \text{ ms} + 10 \text{ ms} + 190 \text{ ms})} = 250 \text{ ft/sec}\]

Thus, compared to only pre-authentication part of MPA operation, in order to be able to complete both pre-authentication and pre-configuration operations successfully, either the mobile needs to move at a slower pace or it needs to expedite these operations for this given cell size. Thus, the extent of MPA operations will be constrained by the velocity of the mobile.

As an alternative if a mobile does complete all the pre-authentication procedure much ahead of time, it uses up the resources accordingly by way of reserving the IP addresses from the neighboring networks, tunnel setup and additional bandwidth needed to carry these preauthentication related signaling. Thus, during pre-authentication mechanism, there is always a tradeoff between the performance benefit (e.g., low delay, less packet loss) and systems resources. This is also largely governed by network characteristics, cell size and movement speed.