Chapter 3

MOBILITY MANAGEMENT ARCHITECTURE FOR 4G NETWORKS

This chapter discusses the design of proposed mobility management (MM) architecture for future 4G networks. As indicated earlier in previous chapters the focus of this thesis for mobility management is connected to exploration of handoff management in heterogeneous convergent network at architectural, protocol, algorithm levels.

This chapter presents handoff Management unit that is capable of achieving a swift handoff between various interfaces offering varied services integrated over an IPv6 core for efficient mobility management. The HMU is capable of managing both vertical as well as horizontal handoff operation utilizing the proposed Fast Binding Algorithm (FBA). The operation of HMU is tested on Vmware simulation environment for horizontal, vertical handoffs and across multiple interfaces. The results are analysed and discussed at the end of this chapter.

3.1 Use case Scenario for Mobility management in 4G Networks

To provide better understanding of the framework, in this section an example is given to demonstrate a scenario for mobility in 4G converged networks from user perspective. The Fig 3.1 and Fig 3.2 show this scenario and can be described as below.
Currently users are all Service Provider bound. They enjoy services provided by their home network operator. 4G networks need to eliminate for the user to be location bound or service provider bound. The future 4G mobile terminal phone is envisioned and conceptualized as in the Fig 3.1. The figure shows that user mobile device is enabled to have the provision of N1, N2, N3, N4, N5, N6, --Nn network operators operating in that geographic location. The consumers would have the freedom to choose any service provider for any specific services; for example User UN can choose N1 for Voice service, N2 for Data service, N3 for SMS service etc... The IMIE (International Mobile Identification Equipment) number and IMSI (International mobile subscriber Identity) Number could be considered for user mobile identity.

This Future scenario demands for a mobility management Architecture wherein the Mobile User can connect to his preferred network provider for specific service and seamlessly move between networks on active sessions.

The major issue in the above scenario occurs when the user is on active session and wants to move to other location. Then the user needs to get connected to other service provider offering the same service at that location. The User should seamlessly connect to the operator NWO on the same active session with negligible handoff latency, so that the active session is continued.
Fig 3.1 Conceptualized future 4G mobile device

In this situation, there could be multiple operators offering same service at that location; which network operator to get connected? (If the user has not specified his preference) To address this issue we require a Handoff management architecture, wherein the user will accept the handover for the Network operator who responds FIRST. This scenario is conceptualized and shown in the Fig 3.2 below.

Fig 3.2 Handoff Use case Scenario
We have introduced a Fast Binding Acknowledgement algorithm to decrease handoff latency. The proposed HMU manages the handoff operations and is designed to support proactive handoff operations. In proactive Handoff operations, the MN decides on the interface over which they would like to access the service offered by the network operators.

3.2 Design considerations for proposed MM Architecture

The design of MM is based on the user scenario presented in the above section. This section discusses the design considerations with assumptions used for our proposed architecture.

3.2.1 IP based Networks

4G Networks considered as heterogeneous networks would integrate over an IPv6 core providing convergence among all the service providers offering different type of service. This thesis assumes such an IP based convergent architecture for the design of mobility management. The IP based convergent framework for Mobility is shown in the Fig 3.3 below. We also assume that the number of services provided by the varied network providers on multiple interfaces with different access technologies converge on IPv6 Core. By envisioning the convergence of all access technologies on IPv6 core, we have proposed a Handoff Management Unit (H MU) capable of achieving a swift handoff between various interfaces offering varied services integrated over an IPv6 core.
Fig 3.3 Convergence of all access technologies on IPv6 core

**3.2.2 ITU-T recommendations** [4]

The realization of Mobility management would be possible only when there are global standards & all network providers and equipment manufacturers agree & adhere to these standards for NGNs. ITU-T (International Telecommunication Union) started this work in 2003 with prominent standardization bodies such as ETSI, IETF, CJK, ATIS and TMF published its NGN Release-1 specifications for the NGNs [1]. These specifications include many features that guide the development of emerging telecom networks. These features presented in the document viewed as the first world-wide standard for further development of NGNs. The release 1 is not in fully implementable stage. However, we can consider these recommendations and proceed for further research.
This thesis assumes the specifications and guidelines specified in the report. One thing that is clear from these recommendations is that with the convergence the user is independent to choose any type of service from different network operators and have seamless mobility among network operators.

### 3.2.3 MIPv6

The Mobile IPv6 (MIPv6) is the protocol used to solve the mobility problem and an IETF standard (RFC3775) [19] (shown in Fig 3.4). Many evolutions and improvements are seen in MIPv6 as the year’s progressed. The MIPv6 implementations have been around from 1998; and they are proposed for different scenarios of mobility management. Each implementation has its own features & differ one to another & none is completely interoperable.

The available different implementations of MIPv6 protocol are Cisco, MIPL (Mobile IPv6 for Linux), KAME (FreeBSD), Microsoft and Lancaster University (Linux) [88].

For Handoff support on Mobility, we have considered MIPv6 implementation on windows platform with incorporation of varied interfaces through virtual routers to support IPv6 switching.
3.2.4 MN (Mobile Node)

The Mobile Node (MN) or user mobile terminal is assumed to have the following:

a) Equipped with multiple access modes (e.g., Multi RF interfaces, wired access, wireless access etc...) that are compatible with IP stack.

b) A high-end multi-functional device with embedded advanced features such as High-speed processor, Advanced mobile OS, Web browser, GPS, GPRS support for all types of technologies and high-end applications etc...
c) To have multiple interfaces to accommodate wide area network connectivity.

d) Only one interface or multiple interfaces can be used at a time depending upon the user or network environment.

The provision of multi interfaces to MN plays a vital role in our scenario of MM. Make-before-break handoff can be achieved easily by initializing a new/existing interface before breaking from the current active interface. This handoff process will reduce handover latency as adopted in our proposed architecture.

3.2.5 Types of Handoff schemes

The 4G is envisioned to provide efficient seamless handoff across multiple heterogeneous networks. From the classification of types of handovers reflected in the Fig 2.5 of chapter 2, we are using seamless handover scheme of rerouting handover schemes. The seamless handover shown in Fig 2.7 based on the concept of changeover between the old and the new link concurrently with only one of them being active. Here the data broadcast done via both links (assuming the Mobile device terminal having multiple interfaces as discussed in the previous section). The old link remains active until the new path is established and activated through a switching action in the network, based on the quality sensitivity of the mobile device. The above seamless handoff scheme is used for both horizontal and vertical handover in our design.
process. In horizontal handover, the process of handover happens within the homogeneous access networks. In vertical handover, the handover happens across heterogeneous access networks. This can be broadly classified as shown in Table 3.1

<table>
<thead>
<tr>
<th>Type of Handover</th>
<th>Horizontal</th>
<th>Vertical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Technology</td>
<td>Homogeneous</td>
<td>Single</td>
</tr>
<tr>
<td>Network Interface</td>
<td>Heterogeneous</td>
<td>Multiple</td>
</tr>
</tbody>
</table>

Table 3.1 Type of handoff scheme used

### 3.2.6 Handover process/operations

Handover is a process in which an on-going transmission or on-going call to a mobile node is transferred from one interface to another interface. The time taken to transfer the context across an interface is known as handoff latency $T_{hl}$ and could be represented as

$$T_{hl} = T_{at} + T_{cr}$$

Where

$T_{at}$ = Interface Association Time

$T_{cr}$ = Interface Connection Re-Establishment Time

### 3.3 HMU Architecture for Mobility Management

MIPv6 used as the main solution for MM in many implementations. Provision of any service anywhere for user could also be addressed with additions to MIPv6 core implementation. We have used MIPv6 core with few additions to facilitate MM in 4G networks. We have
proposed a Handoff management Unit (HMU) for MM in 4G networks as shown in the Fig 3.5

Fig 3.5 HMU Architecture for Mobility Management

The coordination and strong SLAs between service providers and convergence of IP core over MIPv6 is the key for realization of our proposed architecture [6]. The proposed HMU integrates the different network service providers (providing varied services across multiple interfaces) with the multiple mobile (on different interfaces) users. This integration is over MIPv6 core environment. All the services offered by network operators across different interfaces facilitated and managed by HMU. The HMU is designed to handle both vertical and horizontal handoffs as shown in our experimental study discussed later in this chapter. We have used proactive
handoff mechanism in our proposed HMU. In this mechanism, the mobile user decides on the network operator for the service before handoff. From the Fig 3.5 we have considered service providers \( M = 1, 2, 3 \ldots m \) and user nodes \( N = 1, 2, 3 \ldots n \). The HMU integrates the service providers to the users over MIpv6 core network. The \( M \) service providers are connected to HMU through several interfaces \( J = 1, 2, 3 \ldots m \) represented by \( M_{J_m} \). All the nodes \( N \) communicate to HMU through interfaces \( I = 1, 2, 3 \ldots n \) represented by \( N_{i_n} \). The HMU is equipped with the Seamless Connectivity Manager (SCM), the Interface Management System (IMS) and the Policy Manager (PM). All these synchronize with each other to deliver efficient MM Architecture.

As specified in our design considerations, the mobile node (terminal) consists of multiple interfaces in 4G networks. The IMS interacts with SCM. The SCM is the monitoring unit of the HMU, which communicates with the IMS and PM for effective handoff management. The SCM monitors the ongoing communications and facilitates new service requests from the nodes. Since HMU has multiple interfaces for interaction with the service providers and with the nodes, IMS of HMU will manage these interfaces. 4G networks envisioned to provide services to users without prior subscription. Users as the option to choose specific service with any service provider. The policy manager (PM) of HMU uses priority resolution to address this issue. The PM will also facilitate the
implementation of all the policies with respect to handoff management. During handover of active session, there will be many contenders (service providers) offering to provide same service. The HMU needs a mechanism to choose the service provider where there is low latency during handoff. This resolution effectively managed by PM of HMU. The PM uses a Fast Binding Acknowledgement (FBA) algorithm for priority resolution in our approach.

3.3.1 **Functional description of the HMU**

The HMU could be considered as a monitored bridge between the service providers and the user nodes integrated over the MIPv6 core. The HMU integrates the varied services offered by different service providers to the user nodes. The HMU is unconditionally the most important integrating unit of the proposed architecture.

Consider a node \( N = n \) \( (n = 1, 2, 3, \ldots n) \) which requests for a service \( NSn \) over the interface \( Nin \) to the HMU. The user node could specify a preferred service provider \( M= m(pref) \) providing service \( MSm(pref) \) or else the HMU decides on the service provider. The HMU receives the request through the IMS and sends the request to the SCM. If the SCM receives a \( MSm(pref) \) service request, then it would ask the PM to send a service request to service provider \( m(pref) \) over the interface \( MJm(pref) \) through the IMS. The IMS interacts with service provider \( m(pref) \) over interface \( MJm(pref) \). The SCM on getting a confirmation from the service provider \( M= m(pref) \) initiates a
connection to provide a service $NS_n = MS_m(pref)$ over interface $NI_n$. This kind of a handoff is also referred to as a proactive handoff.

If the user node provides no service provider preference and requests for a service $NS_n$ over interface $NI_n$, the HMU is responsible for the service provider selection. The request received by the SCM is sent to the PM for service provider decision making. The PM uses the FBA algorithm for provider decision making. Based on the FBA the PM responds that service provider $M=m$, offering $MS_m$ over interface $MJ_m$ would be best suited. The SCM sends a service request to service provider $m$ and initiates the service delivery to node $N=n$ over interface $NI_n$. The SCM monitors the service connections provided and is also responsible for handoff management in case of a link failure or service failure.

The HMU is capable of handling both horizontal and vertical handoff operations. Network connectivity between the HMU, the nodes and service providers established over the MIPv6 (known for its mobility management and multi addressing capabilities).

3.3.2 FBA-Fast Binding Acknowledgement Algorithm

With multiple service providers, integrating and offering services through the HMU, priority resolution is a very important factor taken for the design of the HMU. The priority resolution is carried out by the PM using the FBA algorithm when the user node has no specific handoff decision taken. When the user node requests for a
service NS\(n\) over interface Nin to the HMU without any user priority, the HMU relies on the FBA for decision making. The FBA algorithm is based on the quickness in response from service providers M. The response time from the service providers M could be represented as \(Trp_m\) where \(r = 1, 2, 3, \ldots m\). The internal timer of the PM could be represented as \(Tp_m\).

The working of the FBA algorithm is given below. With the algorithm given below.

FBA Algorithm:

1. Receive request to provide service NS\(n\) to node N
2. Start PM Timer \(Tp_m\)
3. Send service provision requests to service providers M through the IMS where MS\(m\) = NS\(n\)
4. Obtain service availability confirmation time \(Trp_m\)
5. Calculate \(\Delta T_r = Trp_m - Tp_m \ (r = 1,2,3,\ldots m)\)
6. Find Minimum \(\Delta T_r\)
7. Priority set to Service provider M = \(r\) offering services to HMU over interface MJ\(m\)

The operation of handoff using FBA over MIPv6 is shown with a state diagram in Fig 3.6 below. The state diagram reflects the following

- The Mobile node MN chooses his priority to have service E1 with HA1 and request for the same, HA1 in correspondence with CN establishes connectivity and renders service to MN
• This session event E1 is active and MN is on mobility, at the point of losing connectivity to HA1, handoff notice is issued to HA1 and in turn it is notified to CN (HMU).

• CN (HMU) will broadcast the request (E1 service binding) asking other HAs for willingness to render the same type of service E1 and awaits for the acknowledgement.

• Both HA2 and HA3 will respond. The priority resolution is used to check who has responded FIRST, since HA3 has responded FIRST in this case (shown as T10 in Fig 3.6), the CN will send CoA (IP address) of HA3 to MN for handover execution. Then handoff happens from HA1 to HA3 and the session event E1 is continued with connectivity of HA3.

The FBA algorithm offers quick response times reducing handoff overheads. The FBA algorithm is a fair and unbiased approach to resolve multiple issues related to various providers integrating over a MIPv6 based core network.

The FBA algorithm could also be seen as an energy efficient model of handoff implementation. The performance of the FBA with respect to handoff latency evaluated and presented in the subsequent sections.
3.4 Simulation Test bed Environment

For investigating the performance of the proposed architecture, we constructed a MIPv6 [64] based testbed as shown in Fig 3.7. The test bed is built upon Vmware virtual simulation environment. We have organized our experiments as follows to investigate the HMU performance for horizontal handoff, vertical handoff and multi interface handoff.
The convergent network test bed was realized on the Vmware Workstation 7 embodying the virtual network adapters/Routers configuration tool.

We have used three Interfaces where each Interface is a Network with different Domain (To exhibit different NWOs and HAs).

HA1 is WLAN configured,
HA2 is Ethernet LAN configured
HA3 is WLAN configured
All the three interfaces and CN/HMU are configured to work on varied subnet, gateway and DNS addresses.

The Test Bed is built on Intel and Microsoft environment (Rarely we find implementations of MIPv6 on Microsoft environment; major implementations are done on Linux environment).

The functionality of HMU is built on .NET frame work.

The additional updates were introduced into the MIPv6 using Visual C++ for implementation of FBA algorithm over MIPv6.
3.4.1 VMware platform for simulation [64] [65]

The greatest advantage of using VMware is its ability to set up advanced virtual networking environment with multiple virtual machines (VM). VMware Workstation 7 facilitates to build different networking set-ups on these virtual machines. With appropriate technique and configurations, you can customize and create multiple network architectures on each VMs and make them to speak to each other based on our requirements.

The Fig 3.8 shows the different virtual network components of workstation 7 that all enables to build a virtual network environment. Within VMware Workstation we can configure the following virtual Networks [64] [65] under the following category:

- Bridged networking
- Network address translation
- Host only
Bridged networking:

In this setup, each VM considered as independent client on the network. These VMs are configured to have connected to physical network on the host system. Each VM gets allotted an IP addresses from DHCP server or can be configured manually with IP addresses. In this configuration, the clients (VMs) can speak directly with the host machine.

Network Address Translation (NAT)[64]

In this type of configuration, each VM can be configured to use the IP address and MAC address of the host machine to communicate with outside network. This setup will provision the VMs to
communicate with the other clients on the network where this host machine is connected. This makes those clients connected to network as host systems instead of its network clients. This method will permit multiple VMs to communicate by means of single IP address. Using DHCP of Vmware, the IP addresses are assigned dynamically to NAT enabled VMs. In this setup the communication with outside network is possible only through VMs

- Host-only networking[64]:

In this setup, the host machine has the provision to configure private sub-net for the VMs for which no outside network access is essential. Here the VMs can communicate only with their host system and other VMs connected to same host. However, The VMs cannot communicate with the network to which host is connected. Using DHCP internal server option in Vmware, IP addresses can be assigned to all VMs connected to host. Even IP addresses can be assigned manually without using this option.

Any of the above configuration setup can be used based on the network requirements. We have used “host only” configuration in simulation test-bed. In Host only, the following are the configurations used for virtual network switch, virtual network editor configuration, and network configuration for Guest Operating system.
The Virtual switch is most important component of the whole setup. This switch connects all the VMs and other associated network components to form a virtual network. This switch is software based and has the same features and functionality as that of physical switch. VMware workstation 7 has the provision of total 10 virtual switches for windows hosts and 255 on Linux hosts. Any number of ports can be configured on windows hosts, whereas 32 ports can be configured per virtual switch on Linux hosts. The virtual switches are termed as Vmnet0, Vmnet1 and Vmnet8 respectively by default. We can use extra virtual switches such as Vmnet2, Vmnet3, Vmnet4 etc...as shown in the Table 3.2

<table>
<thead>
<tr>
<th>Name</th>
<th>Vmnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bridged</td>
<td>VMnet0</td>
</tr>
<tr>
<td>Host-only</td>
<td>VMnet1</td>
</tr>
<tr>
<td>NAT</td>
<td>VMnet8</td>
</tr>
</tbody>
</table>

Table 3.2 Vmnet configuration

VMware workstation has the provision to create multiple virtual network adapter of any type on virtual machine. Using Virtual network editor wizard we can configure the adapters on guest operating system as an AMD PCNET PCI adapter or as an Intel Pro/1000 MT Server Adapter. On Windows Vista and Windows 7 guests, it is an Intel Pro/1000 MT Server Adapter.
Networking Configuration for Guest Operating Systems[65]

The default virtual network adapter emulated in a virtual machine is either an AMD PCnet device (vlance) or an Intel 82545EM device (E1000). However VMware also offers a Para-virtualized network driver, called VMXNET that can be used with the vlance device. We can use specific drivers to configure different virtual adapters based on the access technology requirements.

<table>
<thead>
<tr>
<th>Device configurations</th>
<th>Other configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>512MB</td>
</tr>
<tr>
<td>Processor</td>
<td>1</td>
</tr>
<tr>
<td>Hard disk(SCSI)</td>
<td>6GB</td>
</tr>
<tr>
<td>Network Adapter 1(HMU /CNO)</td>
<td>Bridged</td>
</tr>
<tr>
<td>Network Adapter 2( HA 1 /NWO1)</td>
<td>Bridged</td>
</tr>
<tr>
<td>Network Adapter 3( HA 2 / NWO2)</td>
<td>Bridged</td>
</tr>
<tr>
<td>Network Adapter 3(HA 3 / NWO3)</td>
<td>Bridged</td>
</tr>
<tr>
<td>USB Controller</td>
<td>Present</td>
</tr>
<tr>
<td>Display</td>
<td>Used host Settings for Monitor</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3 Test Bed configuration details

We can configure the adapter’s interfaces to work as wireless adapters, Bluetooth, WLAN, etc..., we have used four Network
adapters and following are the specification each adapter as showed in the Table 3.3.

3.5 Simulation Operations

- A FTP application is considered for simulating transfers over the virtual physical channels.
- Handoff was induced to switch at regular interval of 30s on the mobile Node using timers across the varied virtual networks considered over which the FTP data was transacted ensuring only one active connection available at any instant to the MN.
- To evaluate the performance during handoff operations, we have used a bandwidth monitoring utility capable of monitoring the bandwidth across all the interfaces of communications with respect to the mobile Node. Bandwidth measurements are achieved by analysing the packets transacted over the communicating interfaces. These measurements are utilized to derive bandwidth usage graphs to the MN.
- These graphs provide the latency exhibited during vertical and horizontal handovers. The bandwidth utilized is directly proportional to the packets transacted over a particular interface.
3.6 Results Evaluation

We have organized this section to evaluate the performance of the HMU to handle Horizontal Handoff, Vertical Handoff and Multi-Interface handoff. All the evaluations carried out are based on the test bed shown in Fig. 3.7. The handoff latency is calculated based on Eqn(1).

3.6.1 Horizontal Handoff Evaluation

Horizontal Handoff is a handoff between interfaces of similar network technologies. From our test bed it is clear that a handoff across HA1 to HA3 and vice versa could be considered as a horizontal handoff. The handoff is initiated every 30 seconds by movement of the MN from HA1 to HA3 and vice versa. The bandwidth is monitored on the MN using the Bandwidth Monitor utility developed. The results obtained are tabulated in Table 3.4 and the bandwidth graph considering the packet capture utility results are shown in Fig 3.9.

<table>
<thead>
<tr>
<th>Handoff-Interfaces involved</th>
<th>Interface Association Time(s)</th>
<th>Connection Reestablishment Time(s)</th>
<th>Handoff Latency(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WLAN(HA1)-WLAN(HA3)</td>
<td>1.001832005</td>
<td>0.011280959</td>
<td>1.013112964</td>
</tr>
<tr>
<td>WLAN(HA3)-WLAN(HA1)</td>
<td>1.008099015</td>
<td>0.009289195</td>
<td>1.01738821</td>
</tr>
</tbody>
</table>

Table 3.4 Horizontal Handoff Simulation Results
The average Handoff Latency obtained is about 1.015245 Sec for horizontal handoff.

![Bandwidth Chart](image)

**Fig 3.9 Horizontal Handoff Evaluation**

### 3.6.2 Vertical Handoff Evaluation

In this section, we investigate the performance of the HMU across interfaces of different technologies. Vertical Handoff could be described as a handoff across the high speed HA2(LAN) to HA1 and vice versa. Considering our test bed we could also evaluate vertical handoff across high speed LAN to HA3 and vice versa. For evaluation the MN is positioned within the HA1 range or HA3 range maintaining LAN connectivity initially. The handoff is initiated every 30 secs as done in horizontal handoff.

The results obtained for vertical handoff across the LAN interface to HA1 and vice versa are given in Table 3.5. The bandwidth monitored on the MN during the handoff is as shown in Fig 3.10.
<table>
<thead>
<tr>
<th>Handoff Interfaces Involved</th>
<th>Interface Association Time</th>
<th>Connection Re-Establishment Time</th>
<th>Handoff Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAN(HA2)-WLAN HA1</td>
<td>1.0016446</td>
<td>0.012349488</td>
<td>1.013994089</td>
</tr>
<tr>
<td>WLAN(HA1)-LAN(HA-2)</td>
<td>1.008090653</td>
<td>0.009954815</td>
<td>1.018045468</td>
</tr>
</tbody>
</table>

Table 3.5 Vertical Handoff Simulation Results between LAN (HA2) and WLAN (HA1)

Fig 3.10 Vertical Handoff Evaluation between LAN (HA2) and WLAN (HA1)

We also evaluated the vertical handoff capability of the HMU across LAN (HA2) to HA3 and vice versa. The handoff latency values obtained in seconds are shown in Table 3.6.
Table 3.6 Vertical Handoff Simulation Results between LAN (HA2) and WLAN (HA3)

<table>
<thead>
<tr>
<th>Handoff Interfaces Involved</th>
<th>Interface Association Time</th>
<th>Connection Re-Establishment Time</th>
<th>Handoff Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAN(HA2)-WLAN (HA3)</td>
<td>1.009009653</td>
<td>0.0106963</td>
<td>1.019705953</td>
</tr>
<tr>
<td>WLAN(HA3)-LAN</td>
<td>1.002173368</td>
<td>0.01129682</td>
<td>1.013470188</td>
</tr>
</tbody>
</table>

Fig 3.11 Vertical Handoff Evaluation between LAN and HA1

Fig 3.11 shows the bandwidth monitored on the MN during a handoff across the LAN to HA3 interface initiated after 30 seconds and vice versa initiated after 60 seconds.

From the results obtained (in Table 3.5 and Table 3.6), it could be concluded that the HMU architecture for mobility management
effectively manages vertical handoff having a low average handoff latency delay of about 1.0163 seconds.

### 3.6.3 Multi Interface Handoff Evaluations

In this section, we would evaluate the performance of the HMU to handle multi interface handoffs for mobility management. The HMU in this test would maintain the handoffs of the MN across HA1 to LAN to HA3 to HA1. The handoff latency timings obtained are tabulated in Table 3.7.

<table>
<thead>
<tr>
<th>Handoff Interfaces Involved</th>
<th>Interface Association Time(s)</th>
<th>Connection Re-Establishment Time(s)</th>
<th>Handoff Latency(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WLAN(HA1)-LAN(HA2)</td>
<td>1.000980158</td>
<td>0.01996507</td>
<td>1.020945228</td>
</tr>
<tr>
<td>LAN(HA2)-WLAN(HA3)</td>
<td>1.000899158</td>
<td>0.010643338</td>
<td>1.011542496</td>
</tr>
<tr>
<td>WLAN(HA3)-WLAN(HA1)</td>
<td>1.008105303</td>
<td>0.008475396</td>
<td>1.0165807</td>
</tr>
</tbody>
</table>

Table 3.7 Multi interface Handoff Simulation Results

![Fig 3.12 Multi Interface Handoff Evaluation](image)
The handoffs were initiated at intervals of 30 seconds across various interfaces. The bandwidth is monitored through the utility on the MN during the various handoffs; the data is captured and represented graphically in Fig 3.12.

Our experimental evaluation represented through the results obtained makes it very evident that the proposed HMU architecture effectively handles various handoff operations with minimal handoff latency delays for mobility management in 4G networks.

**3.7 Summary**

In this chapter, we were able to demonstrate a novel approach for MM that suits 4G networks. The implementation of the MM architecture is done on windows environment. The proposed HMU for MM is shown to handle handoff operations with low latency of below 1second.