TOWARDS A RELIABLE SEAMLESS MOBILITY SUPPORT IN HETEROGENEOUS IP NETWORKS

A thesis submitted for the degree of Doctor of Philosophy

By

Shoaib Khan

Supervisor: Dr. Jonathan Loo

Co-Supervisor: Dr. Thomas Owens

School of Engineering and Design
Electronic & Computer Engineering,
Brunel University

January 2009

Abstract

Next Generation networks (3G and beyond) are evolving towards all IP based systems with the aim to provide global coverage. For Mobility in IP based networks, Mobile IPv6 is considered as a standard by both industry and research community, but this mobility protocol has some reliability issues. There are a number of elements that can interrupt the communication between Mobile Node (MN) and Corresponding Node (CN), however the scope of this research is limited to the following issues only:

- Reliability of Mobility Protocol
- Home Agent Management
- Handovers
- Path failures between MN and CN

First entity that can disrupt Mobile IPv6 based communication is the Mobility Anchor point itself, i.e. Home Agent. Reliability of Home Agent is addressed first because if this mobility agent is not reliable there would be no reliability of mobile communication. Next scenario where mobile communication can get disrupted is created by MN itself and it is due to its mobility. When a MN moves around, at some point it will be out of range of its active base station and at the same time it may enter the coverage area of another base station. In such a situation, the MN should perform a handover, which is a very slow process. This handover delay is reduced by introducing a “make before break” style handover in IP network. Another situation in which the Mobile IPv6 based communication can fail is when there is a path failure between MN and CN. This situation can be addressed by utilizing multiple interfaces of MN at the same time. One such protocol which can utilize multiple interfaces is SHIM6 but it was not designed to work on mobile node. It was designed

for core networks but after some modification in the protocol , it can be deployed on mobile nodes.

In this thesis, these issues related to reliability of IPv6 based mobile communication have been addressed.

Acknowledgements

Research track that I perused was impossible to achieve without help from my supervisors Dr. Jonathan Loo, Dr. Wenbing Yao and Dr. Thomas Owens. I would like to express my sincere gratitude to my supervisors for their constant guidance, invaluable advice and suggestions throughout my research and studies.

Shoaib Khan

List of Figures

List of Tables:

List of Acronyms:

3G	Third generation
3GPP	3rd Generation Partnership Programme
AAA	Authentication, access and accounting
ACK	Acknowledgement
ADSL	Asymmetric digital subscriber line
AES	Advanced encryption standard
ANP	Anchor point
ANWR	Access network wireless router
AP	Access point
API	Applications programming interface
ARQ	Automatic repeat request
ASA	Access Service Authoriser
ASP	Access Service Provider
BAA	Bootstrapping Authorisation Agent
BC	Bootstrapping Client
BCA	Bootstrapping Configuration Agent
BGP	Border gateway protocol
BT	Bootstrapping Target

BTS	Base transceiver station
CA	Collision avoidance
CD	Collision detect
CSMA	Carrier sense multiple access
CoA	Care-of Address
CoT	Care-of Test
CoTI	Care-of Test Initiate
CN	Correspondent Node
dHA	Designated Home Agent
DoS	Denial of service
DSL	Digital subscriber line
DSSS	Direct sequence spread spectrum
GGSN	Gateway GPRS support node
GPRS	General packet radio service
GSM	Global system for mobile telecommunications
GUI	Graphical user interface
HoA	Home Address
HoT	Home of Test
HoT	Home of Test Initiate
HA	Home Agent
HIP	Host Identity Protocol
IETF	Internet Engineering Task Force

IFS	Inter-frame space
IN	Intelligent network
IP	Internet protocol
IPSec	IP security
ISP	Internet service provider
IST	Information society technologies
ITU-T	International Telecommunication Union
LAN	Local area network
IeDR	Inter-domain handover
IeTR	Inter-technology handover
IaTR	Inter-technology handover
IaDR	Intra-domain handover
IeLM	Inter-link mobility
IaLM	Intra-link mobility
LIN6	Location Independent Network for IPv6
MAC	Medium access control
MN	Mobile node
MSA	Mobility Service Authorizer
MSP	Mobility Service Provider
NAS	Network Access Server
NAT	Network address translation
PCI	Peripheral component interconnect

PDA	Personal digital assistant
PoP	Point-of-presence
QoS	Quality of service
RTCP	RTP control protocol
RTP	Real-time transport protocol
RTS	Request to send
RTT	Round-trip time
SC	Service control
SCP	Service control point
SCTP	Stream Control Transmission Protocol
SIP	Session Initiation Protocol
SGSN	Serving GPRS support node
SNMP	Simple Network Management Protocol
sHA	Serving Home Agent
SHIM6	Site Multihoming for IPv6 Intermediation
SLA	Service-level agreement
SMS	Short messaging service
SP	Service provider
UWB	Ultra-wideband
VoIP	Voice over IP
VPN	Virtual private network
W3C	World Wide Web Consortium

WAP	Wireless application protocol
W-CDMA	Wideband code division multiple access
WEP	Wired equivalent privacy
WiFi	Wireless fidelity
WIMP	Weak Identifier Multi-homing Protocol
WLAN	Wireless local area network

Chapter 1

1 Introduction

The goal of cellular mobility standards has always been to provide global connectivity without involving network layer. General packet Radio Service (GPRS), Wideband Code Division Multiple Access (WCDMA) and Universal Mobile Telephony System (UMTS) are three standards that provide Link layer mobility. Access to IP networks in case of Link layer mobility is through one specific router. This feature of link layer mobility causes problems such as inefficient routing or triangular routing. This and many other problems associated with Link layer mobility can be resolved by using Network layer mobility.

Next Generation networks (3G and beyond) are evolving towards all IP based system where the aim is to provide global ubiquitous coverage. Major 3G standardization bodies such as 3GPP and 3GPP2 are adopting IETF's protocols i.e. Mobile IP (MIP) and SIP for supporting various service scenarios. These services include VoIP, online gaming, white boards and video conferencing. Wireless communications is a fast growing area. Recent addition to the family of wireless technologies is WiMAX which offers metropolitan scale wireless broadband with support for mobility. Every new technology opens new research issues. With many competitive options in wireless family, the content providers are left with more liberty in terms of selection of a particular technology for delivery of services to the users.

Consider a scenario in which there are multiple access technologies such as WiFi, WiMax and 3G Networks. In this situation, mobile users would like to freely move across different technologies without experiencing interruptions in connectivity. One possible solution to this problem is proposed in authors' research papers [1], [2]. These papers provide mechanisms which can be used to roam freely across heterogeneous networks. However, providing mobility in heterogeneous networks is not the end of story. Next step is making this IP based mobility reliable enough to match the Teleco requirement of five nines [66]. The aim of this research is to achieve this goal.

There are many challenges associated with providing a reliable Mobility service in IP based networks. These challenges are described later in this chapter but first mobility protocol is introduced.

1.1 Mobility Protocol

In MIPv6, a Mobile Node is always expected to be reachable by its Home Address whether it is attached to home link or some foreign link. This Home Address is an IP address assigned to mobile node with prefix that is advertised on its home link. When Mobile Node is located on the Home link, it can receive packets on its assigned Home address. While in foreign network, it is also addressable at a care-of address. Care-of address is an IP address associated with a mobile node that has a subnet prefix from some foreign network. The association between a mobile node's home address and care-of address is known as "binding". While away from home network, a mobile node registers its primary care-of address with a router on its home link, requesting this router to function as the "home agent". The mobile node performs this binding registration by sending a "Registration Request" message to the home agent. The home agent replies with a "Binding Acknowledgement" message. Any other nodes communicating with mobile node are referred to as "correspondent nodes".

Mobile nodes can provide information about their current location to correspondent nodes and home agent, using CoTI/CoT, HoTI/HoT Messages. The procedure is known as return rout ability test. It is performed between the mobile node, home agent, and the correspondent node in order to authorize the establishment of binding. Packets between mobile node and correspondent node are either tunneled via the home agent, or sent directly if a binding exists in the correspondent node for current location of the mobile node. Mobile IPv6 tunnels payload packets between mobile node and home agent in both directions. This tunneling uses IPv6 encapsulation. Where these tunnels need to be secured, they are replaced by IPSec tunnels.

A generic scenario of mobility is illustrated in Figure 1.1. This figure is explained as under.

Home Link: The network link that is assigned Home subnet prefix. From this home prefix MN obtains its Home Address.

Home Address: The address assigned to the MN when it is attached to Home Link and through which MN is always reachable regardless of its location on internet.

Home Agent: It is a router on Home link that maintains all the registrations of MNs which are away from their home link. This agent receives all the data destined for HoA.

Mobile Node: This is IPv6 enabled node which can change its point of attachment to internet.

Foreign Link: Any link that is not a Home link for MN is called foreign link.

Care of Address: This is an IPv6 address which is temporary and changes with MN's point of attachment to internet.

Corresponding Node: Any IPv6 node which can communicate with MN.

Home Subnet Prefix: The IP subnet identity that corresponds to home address.

Foreign Subnet Prefix: Any IP subnet identity that is not Home subnet prefix.

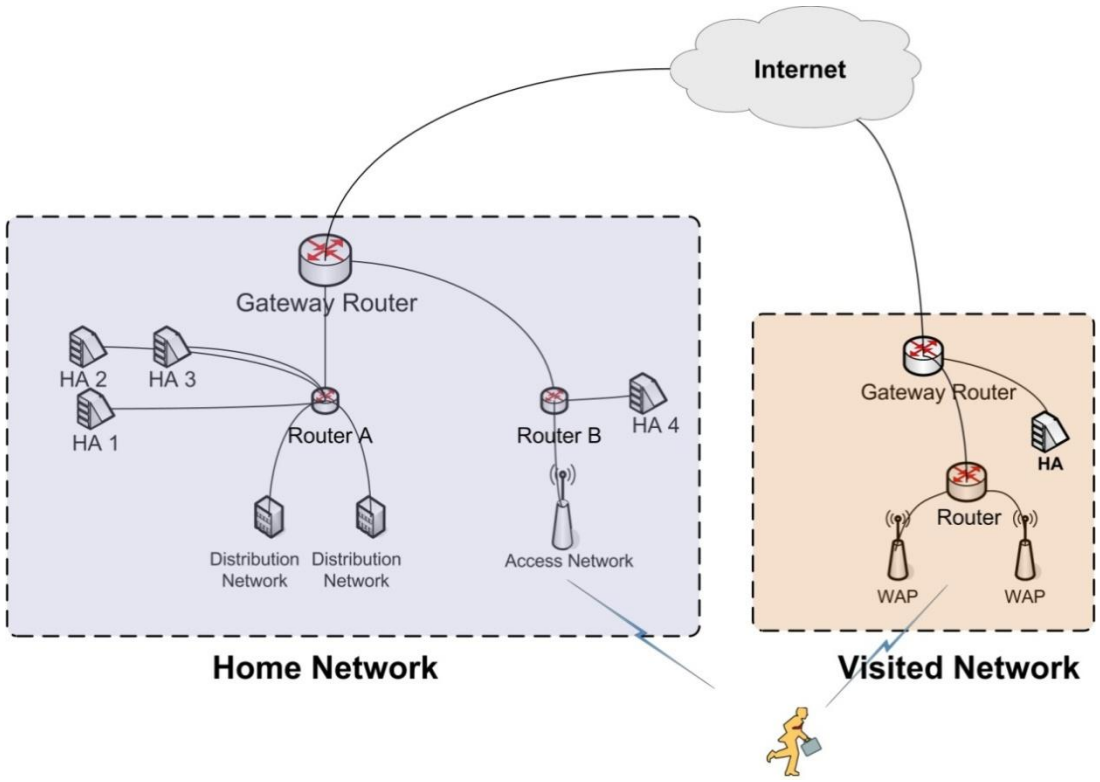

Figure 1.1: Generic Mobility Scenario

Home Registration: The registration of MN's CoA with Home agent on Home link.

Binding Update: The message sent by MN to notify Home agent regarding any change in its CoA

Binding Acknowledgement: The message sent by HA to acknowledge Home registration.

Mobile Prefix Solicitation: This message is sent by MN to HA to request its Home Link prefix which can be used for obtaining HoA

Mobile Prefix Advertisement: This message is sent by HA at inform MN about the home link prefix.

1.2 Research Challenges

In this section, the research challenges associated with communication reliability in of mobile environments are discussed.

Reliability of Mobility Protocol

Mobile IPv6 is a well-designed protocol but one of its shortcomings is its reliability. The protocol mechanism itself makes the mobility agent a very critical point. A HA can serve multiple MNs. If a HA crashes, communication to all MNs which is routed through that HA will get disrupted. This makes HA a single point of failure. This problem has been realized by IETF and is still under discussion in MEXT working group.

Home Agent Management

In future, it is anticipated that there would be a large scale deployment of mobile nodes which would require multiple HAs. In such a scenario, there is a need for an adequate management framework for HA. There is no existing mechanism that defines such a framework.

Fast and Secure Handovers

When mobile node moves, at some point the mobile node will reach the physical border of its current wireless network and will enter an area covered by another wireless network. In such a situation, mobile node must switch to the new wireless network. This switching is called

handover. This handover process is very slow in base MIPv6. Fast handover for MIPv6 (FMIPv6) has been developed as an extension protocol to reduce the handover latency and packet loss inherent with MIPv6. The RFC for FMIPv6 is categorized as “Experimental” by IETF because it is not in compliance with MIPv6 standard. One major compliance issue is that MIPv6 standard requires that all the messages exchanged between MN and HA should be secured which in case of FMIPv6 are not. In addition to this, FMIPv6 handovers are based on anticipation which is not very accurate and can result in failed handover.

Communication Path Reliability

A common scenario is when two mobile devices are communicating via MIPv6. To increase the reliability of this communication path there is a need to introduce redundancies. One approach is to make use of all the available interfaces of mobile device. A research challenge to this approach is how to detect path failures and how to switch interfaces. There is no existing mechanism to quickly detect path failures and switch interfaces in mobile environment.

1.3 Contributions

The contributions of this thesis are:

- A novel HA reliability architecture is proposed which can quickly detect failures and perform HA assignment and HA load sharing. The mechanism is based on a central controller called HA Manager. Failure detection is performed by listening to periodic heartbeat messages for HA. These heartbeat messages are L2 frames with embedded higher layer information related to specific HA. Based on this embedded information HA load sharing can be performed. Since this load sharing is based on

feedback mechanism, it outperforms existing solutions. The architecture also proposes a failure recovery scheme which is based on modified mobile node registration with HA. Overall, the proposed solution increases the reliability of HA. This research is part of FP6 project, ENABLE [3] and this architecture was considers as an early stage solution.

- A new framework for HA management is proposed which can perform self-healing, self-configuration and self-optimization. This proposal is based on IBM's vision of autonomic computing. This framework also proposes a new data delivery mechanism based on L2 frames. By using L2 frames, the monitoring data can be delivered quicker without affecting network performance.

- FMIPv6 is modified to make it compliant with MIPv6 RFC. This modification includes addition of new procedures which makes the protocol secure without affecting its performance. For verification of the procedures, the actual testbed is build at Brunel University. This research is part of FP6 project and the testbed was presented in final demonstration meeting of the project.

- A new improved handover anticipation mechanism for FMIPv6 is proposed. Under certain test condition the success rate of anticipation ranges from 30% to 35% in case of standard FMIPv6. After modification, under similar test conditions this success rate is increased to 60%. This new anticipation is based on prediction of MN's trajectory.

- A new mechanism for communication path redundancy is proposed. This scheme is based on modification of Reach Ability Protocol (REAP) which was designed for failure detections in core networks. After some modifications, it can be used on

Mobile Node. This scheme involves communication path failure detection and recovery mechanism. The mechanism allows a prominent multi-homing benefit i.e. reliability to be realized by quickly detecting failures in the path between MN and CN and recovering from them. It is shown that the proposed scheme out-performs existing MIPv6 failure detection and recovery techniques.

1.4 Thesis Outline

This thesis consists of six chapters and covers the research work motivated by the challenges outlined in section 1.2. Each chapter has its own introduction, related work, proposed scheme, results and conclusions. The thesis has been organised as follows:

In **Chapter 2** a new architecture for HA reliability is proposed. Generally, reliability schemes introduce redundant HA to the network. In the proposed scheme there are no redundant HAs. This is achieved by introducing a new HA registration scheme which allows MN to get registered with two HAs. In addition to this, a new HA failure detection scheme based on L2 frames is introduced. This L2 frames is embedded with higher layer information. With this approach more detailed information can be transported over the network without generating significant traffic. This failure detection scheme can detect the failure in less than 40ms. In addition to this, the proposed architecture also evaluates three possible load sharing schemes. In the results section of this chapter the failure detection and load sharing schemes are compared with existing solutions.

Chapter 3 presents a framework for HA Management based on Autonomic Computing concepts. By introducing this scheme, HA system

would be able to monitor itself and perform self-healing and self-optimization. This chapter also presents a review of some commonly used schemes in industry for server/service monitoring. Some modules of the proposed scheme are simulated in NS2 and the results are presented in this chapter. At the end of this chapter, a generalized solution is developed which can be used to perform autonomic management of different network services.

In **Chapter 4** modifications to FMIPv6 is presented. These modifications include addition of security during handovers and improvement of handover anticipation. The handovers are secured by exchange of authentication keys. MN requests the authentication key from network before initiating a handover. Once the key is acquired, it is presented to the new access router during handover procedure. Next the access router checks the validity of this key. If the key is validated, MN is allowed to use the network. This scheme is validated through actual testbed implementation. In the results section, the performance of secure-FMIPv6 is compared with non-secure version of FMIPv6. In addition to this, a new handover anticipation scheme is presented to assist FMIPv6 in access point selection. This scheme is based on prediction of MN's trajectory based on its previous three locations. In the results sections this scheme is compared with standard FMIPv6 procedures.

Chapter 5 presents a mechanism that adds redundancies to the communication path between MN and CN. The link failure detection is based on modification of REAP protocol which was originally designed for core networks. If there is a failure on active link the communication switches to the redundant link. For switching from one interface to another, MN sends a binding update message to HA. This binding update message contains the CoA associated with the interface to which it desires to switch. In the results section this scheme is compared with standard MIPv6 interface switching.

Finally, the summary of the thesis' contribution along with the scope of future research directions are presented in **Chapter 6**.

1.5 List of publications

1) **Khan, S.** Loo, J., Kiani, A., K., Yao, W.,"Home Agent management based on autonomic Computing" Submitted to Elsevier Journal, Computer Networks

2) **Khan, S**. Loo, J., Kiani, A., K., Yao, W.,"Testbed implementation of Secure FMIPv6 " submitted Elsevier Journal, Network Security

3) Kiani, A., K., **Khan, S.** Yao, W., "SHIM6 based Failure Detection and Recovery Mechanism in Multihomed MIPv6 Networks" WWRF17, 2006

4) Kiani, A. K., **Shoaib Khan**, Dr. Wenbing Yao, "A Novel Mechanism To Support Session Survivability in Heterogeneous MIPv6 Environment" IEEE ICET 2006

5) . "Supplementary Interworking Architecture for Hybrid Data Networks (UMTS-WiMAX)". In Proceedings of Wireless World Research Forum 16th, shanghai china, 2006

6) **Khan, S.,** Khan, S., Mahmud, S.A. and Al-Raweshidy, H. 2006. Supplementary Interworking Architecture for Hybrid Data Networks (UMTS-WiMAX). In Proceedings of the International Multi-Conference on Computing in the Global Information Technology, Anonymous IEEE Computer Society Washington, DC, USA, 2007

7) **Khan, S**., Kiani, A. K., Cecelja, F., and YAO, W "Home Agent Load Balancing in Mobile IPv6 with Efficient Home Agent Failure Detection and Recovery" IEEE ICET 2006

8) Khan, S., Khan, S., Mahmud, S.A. and Al-Raweshidy, H. 2006. "A Comparison of MANETs and WMNs: Commercial Feasibility of Community Wireless Networks and MANETs" Proceedings of the 1st international conference on Access networks Athens, Greece 2006

Chapter 2

2 Reliability of Mobility Protocol

It is anticipated that next generation networks will be all IP based. For Mobility in IP networks MIPv6 is considered as the prime protocol. MIPv6 is a very well designed, well implemented and very flexible protocol. However, it has some deficiencies. One of the main issues overlooked during its design is reliability. The protocol mechanism makes mobility agent a very critical element. This chapter focuses on improving the reliability of mobility protocol by making this "single point of failure" more reliable. This research is part of FP6 project ENABLE.

2.1 Introduction

Mobile IP (MIPv6) is originally defined in [4] as an extension to IPv4 for mobility support. The evolution of MIPv6 from MIPv4 solved the problems of triangular routing and ingress filtering. In addition to this, the new protocol introduced built-in route optimization, multiple HAs support and IP security. With time and discussions on mailing-lists, original draft of MIPv6 [5] is standardized in RFC 3775 [6]. Today, MIPv6 is the most mature and widely accepted protocol for mobility support in IPv6 networks.

In MIPv6 each Mobile Node (MN) is identified by its home address (HoA) regardless of its current point of attachment to the Internet. When MN is away from its home network, a temporary address called care of address (CoA) is assigned to it. While away from its home network, MN sends a registration request to HA which informs HA about MNs current location

on Internet. HA stores this information in a table called binding table. HA also advertises its own link-layer address for HoA of MN. Thus, all packets sent to HoA are received by the HA which are then tunneled to MNs current location using its CoA.

An HA can serve multiple MNs. If HA crashes, communication to all MNs which is routed through that HA will get disrupted. This makes HA a single point of failure. According to MIPv6 protocol MN can only discover failure of HA in following two scenarios.

I. When MN moves from its current location to a new location and sends BU to the failed HA and does not received a corresponding BA. In this situation, MN will retransmit BU until timeout expires. At this point MN will detect failure of HA
II. When Global IP address prefix of an HA expires, MN sends a Mobile Proxy Solicitation message to HA. If MN does not receive Mobile Proxy Advertisement then MN will retransmit this message until timeout expires. At this point HA failure will be detected.

According to MIPv6, MN after sending first BU or Mobile Proxy Solicitation message to the failed HA, will wait for an initial timeout which is set to 1 second in case of BU and 3 seconds in case of Mobile Proxy Solicitation. This timeout period will be doubled for each subsequent BU or Mobile Proxy Solicitation message until the max time of 32 seconds is reached. This causes MN to send 6 BUs or 5 Mobile Proxy Solicitation to the failed HA before the final timeout occurs.

So the total failure detection time in case of MIPv6 is,

Detection time (base on BU timeout) = 1+2+4+8+16+32 = 63 seconds

This detection of failure causes significant delay. In addition to this, BU or Mobile Proxy Solicitation messages are not periodic message. MN will send BU only if it has to register a new CoA or when current registration

expires. Similarly MN will send Mobile Proxy Solicitation message only when its serving HA address is about to become invalid. Once the failure of HA is detected, it must perform Dynamic Home Agent Address Discovery (DHAAD) to find another suitable home agent on home link. MN will suffer a significant packet loss before regaining connectivity if MIPv6 standard failure detection and recovery procedures are employed. To increase the reliability of HA, there is a need for more sophisticated failure detection scheme. Later in this chapter a novel HA failure detection scheme is proposed which can detect HA failures within 40ms without introducing significant network traffic.

In addition to failure detection, another issue in an HA deployment is fair load sharing. MIPv6 allows the deployment of multiple Home agents in the network. In the Next Generation of Networks (NGN), it is expected that there will be a huge number of MNs supporting MIPv6. With large number of MNs, multiple HAs are required which presents the problem of fair load distribution. There is a crude load sharing scheme in MIPv6 which is based on anycast addressing. In this scheme it is assumed that all HAs have same load handling capabilities. It is a sort of round robin scheme which cannot guarantee fair load sharing. Anycast addresses may allow a router to pick different anycast destinations for every packet but this does not mean that load will be shared fairly between the HAs. Another major problem in this scheme is that it does not provide any administrative control. In this chapter, a novel scheme for load sharing is presented which can guarantee fair load distribution among HAs.

To increase the reliability of MN's communication, the reliability of HA must be increased. This can be achieved by adding redundancy to HA setup. In case of multiple HAs, simple one-to-one redundancy is not a cost effective solution. Ideally, in a situation where there is failure of a single HA in a group, the remaining HAs should be able to take over.

This chapter presents a solution that will enable us to quickly detect and recover from failures. Along with this quick recovery mechanism, the proposed solution can perform a fair load sharing.

In the remaining part of the chapter, an intensive review of existing HA reliability solutions is presented. Subsequently, a detailed architecture of proposed solution is presented. Finally some test results are presented along with chapter summary.

2.2 Evaluation of Existing HA Reliability Approaches

In the research community different solutions for HA reliability have been developed but the most promising ones are outlined and assessed in this section.

2.2.1 HA Redundancy Protocol

This scheme [7], known as “Home Agent Redundancy Protocol” (HARP), is an extension to Mobile IP for providing Home Agent redundancy. In HARP, one or more HARP peers act as a single shared Home Agent. In every HARP peer the neighboring HARP peers information is stored. Each peer forwards any Mobile IP registration messages it receives to its peers via “HARP udp Forward” message. This allows peers to share registration information. In addition to this, when one HA boots or recovers from any failure, it can retrieve the complete registration information from another peer.

HARP peers are multi-homed in a sense that they have two IP addresses. The HARP peer address is used for peer communication, e.g. the exchange of binding information; the "Mobile-IP subnet" address is used as HA address known by MNs. HARP peers are located in different networks but they are within the same routing domain and intra-domain routing protocols like OSPF and RIPv2 are running on the network. Both HARP peers announce the same Mobile IP home prefix. Depending on the routing

metric, Mobile IP packets are routed to either peer. In case of a HA failure, the failing HA will not announce the prefix anymore and OSPF or RIP will change routes towards the available HAs.

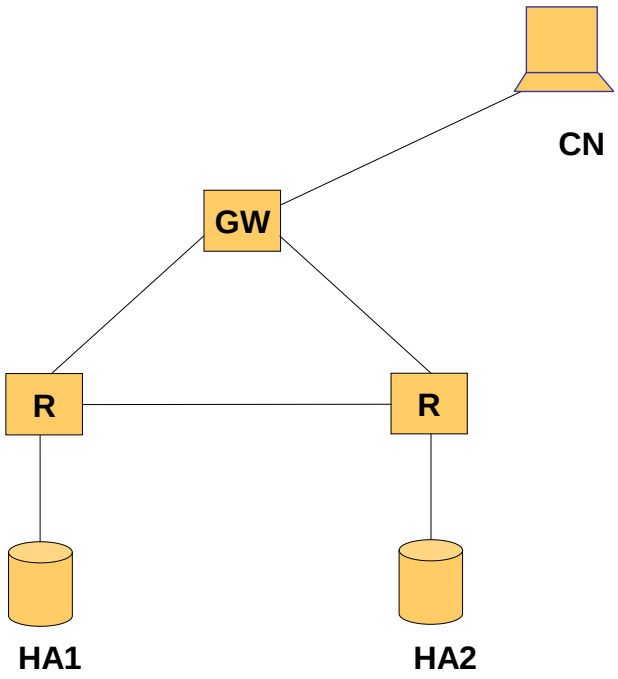

Figure 2.1: HA Redundancy Protocol

This protocol was designed for MIPv4 and it does not take into consideration IPSec SAs established between MN and HA, but for a transparent failover the IPSec states have to be shared as well. Hence, it cannot be used for MIPv6 enabled networks without major modifications.

2.2.2 Fault Tolerant Mobile IP

In [8] a solution for HA reliability is presented. MN sends a registration request to HA1, which then forwards it to HA2 for binding

synchronization. HA2 acknowledges this by sending a BA to HA1. After this, HA1 sends a BA to MN. In case of HA1 failure, HA2 performs gratuitous ARP to take over the failed HA1. In the gratuitous ARP process a host informs other hosts of a new MAC address by sending ARP message. The document does not specify the mechanism used for HA failure detection itself. Once HA1 returns after a crash, it asserts itself through gratuitous ARP and takes back its previous IP address and contacts HA2 to obtain its mobility binding.

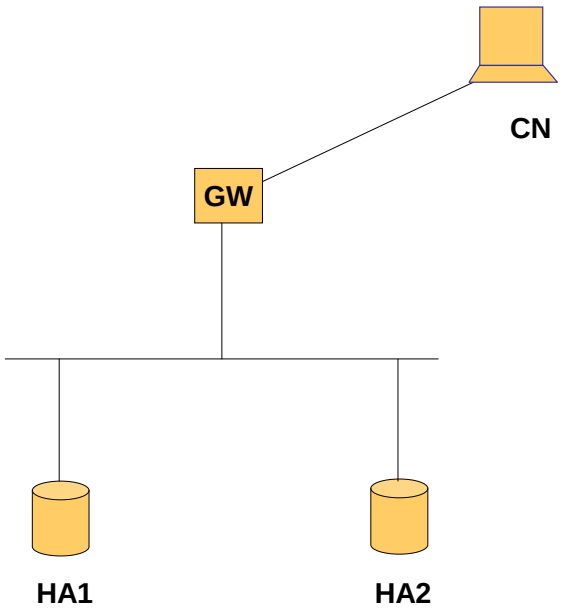

Figure 2.2: Fault Tolerant Mobile IP

This scheme does not discuss IPSec SAs between MNs and HAs. It synchronizes the binding tables between HAs but not IPSec SA states so there would be no seamless switchover possible. It uses gratuitous ARP to replace a failed HA. This is a slow process and as a result mobile clients would suffer from service interruption.

2.2.3 Efficient Fault-Tolerant Protocol for Mobility Agents in Mobile IP

This scheme is presented in [9]. It introduces a “stable storage”.

Registration process which involves logging binding information to a stable storage before sending BA to MN. In case of failure, the least loaded HA takes over the failed HA using gratuitous ARP and reads the stable storage. Failures are detected by not receiving RAs from the HAs.

This scheme does not discuss the synchronization of IPSec SAs, established between MNs and HAs. It has no fair load sharing scheme and delayed failure detection. It introduces a novel idea of stable storage but it uses gratuitous ARP which is a slow process.

2.2.4 HA Redundancy in Mobile IP

This proposal was presented in [10]. This scheme is based on having Standby and Active HAs, the selection of which is beyond the scope of the scheme. Active HA is responsible for registering MNW and synchronizing binding information with Standby HAs. Thus binding cache is synchronized on Active and Standby HAs. Standby HA is responsible to download the binding cache from the Active HA either before it takes over the Standby HA role or immediately after becoming a Standby HA. Multiple Standby HAs per Active HA are supported by this scheme. Messages exchanged between HAs are secured. This draft only provides the scheme for redundancy. It does not suggest any scheme for failure detection.

This scheme does not define a way to select Active / Standby HAs. Failure detection is also not tackled in this scheme. Hence, from ENABLE's perspective it is not a complete scheme.

2.2.5 Local HA to HA protocol

The proposed protocol was presented in [11] and it is for communication between HAs. Each HA maintains a list of home agents on the home link by listening to RAs, in which H bit is set to 1, indicating that it is a HA. Failure detection is based on Hello messages, sent by each HA

periodically. Binding cache is synchronized by all HAs. The HA which actually processes the MN registrations is considered as primary HA of that specific MN. Each HA is supposed to broadcast BU messages whenever it processes and updates its binding cache due to a MN registration request. HA can query any other HA for information regarding a particular MN binding. The primary HA can send Home Agent Switch Request message to ask a MN to switch from one HA to another. In case of failure of the primary HA, the backup HA can send Home Agent Switch Request message to all the MNs that are registered at the failing HA. The Home Agent Switch Request message must be secured via IPSec ESP in transport mode. If there is no existing IPSec SA between backup HA and MN, the HA must negotiate an IPSec SA first.

This protocol introduces "HA switch" concept which gives service providers the flexibility to request MN to switch to another HA. This approach is similar to Hard Switch mode of [HAREL]; however, the establishment of IPSec SAs between MN and backup HA during bootstrapping is not mandatory. Hence, the IPSec SA negotiation in failure case would add delay to the switch over process.

2.2.6 Introducing Reliability and Load Balancing in Mobile IPv6 based Networks

The reliable HA service offered by [12] is a "Virtual HA Reliability Protocol" which achieves this service by introducing the concept of a Virtual HA. In essence in VHARP, each HA has a unique link-local IP address but all the HAs have the same global IP address known as "Global HA address". This Global HA address is resolved only to one of the multiple HAs on the home link. This HA is the owner of the Global HA address and it is known as the Active HA. MN's and CN's are only aware of the Global HA address, which is considered HA address by MNs

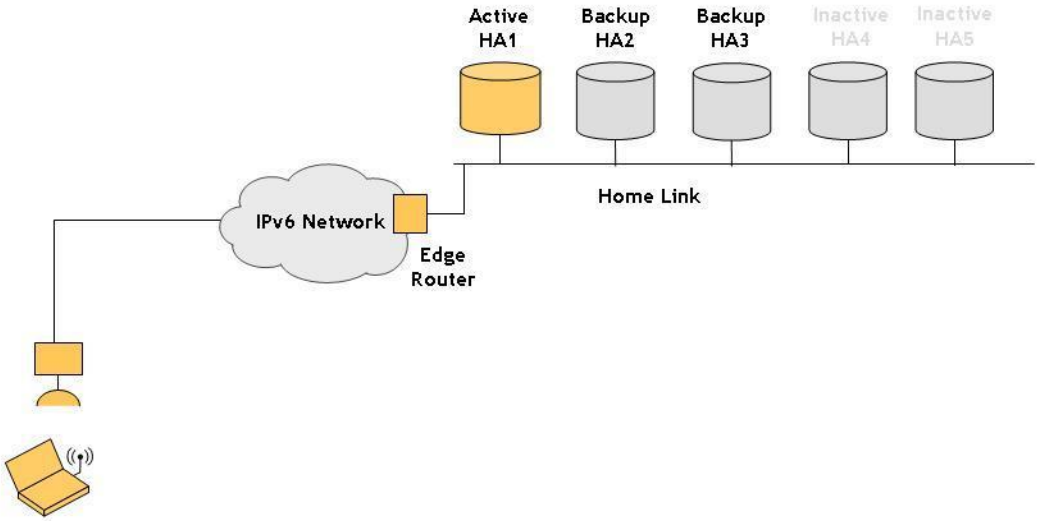

Figure 2.3: Single Virtual HA view of multiple HA's in VHARP

For state synchronization, VHARP introduces the concept of operating states for HA. In VHARP all HAs attached to home link do not operate equally. Each HA operates in one of three states, namely Active HA, Backup HA or Inactive HA.

In this scheme there can be only one HA acting as “active HA” on the home link at a given time. For synchronization VHARP uses more than one HA for storing registrations of a MN. Binding Synchronization is done over the network; however, no details are given about this synchronization process. VHARP modifies the MIPv6 binding cache originally defined in RFC3775 [6], with some new fields. These are “Binding Owners List”, “Number of Binding Owners” and “Load”. After successful binding synchronization, the active HA performs IPSec synchronization by multicasting the IPSec to all other HAs. However, no details of this synchronization are given.

Failure detection is achieved by means of a heart beat message, which utilizes a modified router advertisements (RAs) that is multi casted by each HA at a constant rate on the home link. In case of a HA failure, a

backup HA can provide service to the MN in a transparent way, as VHARP requires that for each MN there is at least two HAs holding its Binding in their Binding Cache. These HAs are known as the “Binding Owners” for the MN.

This scheme looks promising but there is no optimal mechanism defined for HA failure detection. The scheme relies on modified router advertisements, which is a slow process. Furthermore, there are no details given for synchronization process of bindings and IPSec SAs..

2.2.7 Virtual Router Redundancy Protocol

This approach is outlined in [13]. It is not strictly an HA reliability mechanism, but a protocol to provide virtual router functionality. However this could be applied to HA reliability due to the fact that two or more HAs in a HA redundancy set could have one set as the master router and the others as backup router(s), creating a virtual HA towards MN.

In VRRP, responsibilities of virtual router are assigned dynamically to one of the VRRP routers on LAN. The master router is the one that controls the IP addresses associated with the virtual router functionality. It also forwards packets sent to these IP addresses.

The main advantage of using VRRP is that fast switching to a Standby HA in an event of failure of the Active HA can be achieved. However, this approach does not consider at all the fact that the routers are running MIPv6; consequently there is neither binding cache synchronization nor IPSec synchronization specified in this approach. It must also be highlighted that for this protocol to work all HAs should be on a single subnet.

2.2.8 Home Agent Reliability Protocol

This approach is outlined in an Internet Draft of the IETF HA reliability

design team [14]. In this draft there are two different HA network configurations for standby HAs. Either all the HAs serving the same home network are located on the same link or standby HAs are located on a different link (named the recovery link).

Once a HA network configuration is chosen there are two possible redundancy modes, a Virtual Switch mode and a Hard Switch mode

Home Agent Virtual Switch mode: Home Agent Virtual Switch mode is transparent to the MNs. In this approach the Active and Standby HA share a virtual home agent address and perform synchronization of binding cache, AAA and IPSec information between them. The failure of the Active HA is detected by the Standby HA e.g. via a HA Hello message mechanism. Once the failure is detected, the Standby HA activates the virtual home agent address and starts immediately to act as Active HA. This is completely transparent to the MNs registered at the failed HA.

Home Agent Hard switch mode: Home Agent Hard Switch mode is not transparent to MNs. In this approach the Active HA and Standby HA have separate HA addresses, both maintain an IPSec SA with MNs, and the active HA synchronizes the required states, such as the binding cache and AAA information, with other standby HAs periodically. When a failure of the Active HA is discovered by a Standby HA, the Standby HA will use the pre-existing IPSec SA to notify the MN that there is a failure of its current Active HA via a HA Switch Message. Upon receipt of this HA Switch Message the MN sends a BU to Standby HA, which will in turn update the MIPv6 tunnel end points.

[14] Also allows for Active HA management. In the case where the Active HA may need to stop serving MN’s due to system maintenance. This specification provides for an administrator triggered HA switch between an Active and Standby HA by using the Switchback Request and Reply messages.

This approach is already relatively mature and provides mechanisms for failure detection, state synchronization, and HA switching. Moreover, the approach provides the flexibility for the provider to choose either the Virtual or the Hard switch mode, both having advantages and drawbacks. Finally, being developed by the IETF HA reliability design team, this approach is likely to become standardized.

The solutions discussed in sections 2.2.1 – 2.2.7 have deficiencies either regarding delayed failure detection, service interruption, or the non-availability of mechanism details, e.g. how to perform IPSec state synchronization.

The approach developed by the IETF HA reliability design team [14] has the flexibility to support both Hard Switch and Virtual Switch mode. Based on service provider preference, either of them can be deployed. Furthermore, the level of details in the specification is already advanced and, since the IETF is following this approach, it has most potential to become standardized.

Our proposed solution is different from above-mentioned approaches. Aim of this research is to make HA more reliable but at the same time not introduce any redundancies. In contrast to the solutions listed, in our proposed solution there is no redundant HA. All the HAs have active registrations and in case of single HA failure, remaining active HAs can act as backup HAs. This very nature of the scheme makes it cost effective solution.

2.3 HA Reliability Architecture

This section presents the proposed architecture for HA reliability. For this scheme to work a central controller known as the HA Manager is required that can detect HA failures and perform HA load sharing and assignment.

For this architecture, it is assumed that Mobility is a premium service and needs to be authorized by the Mobility Service Provider (MSP).

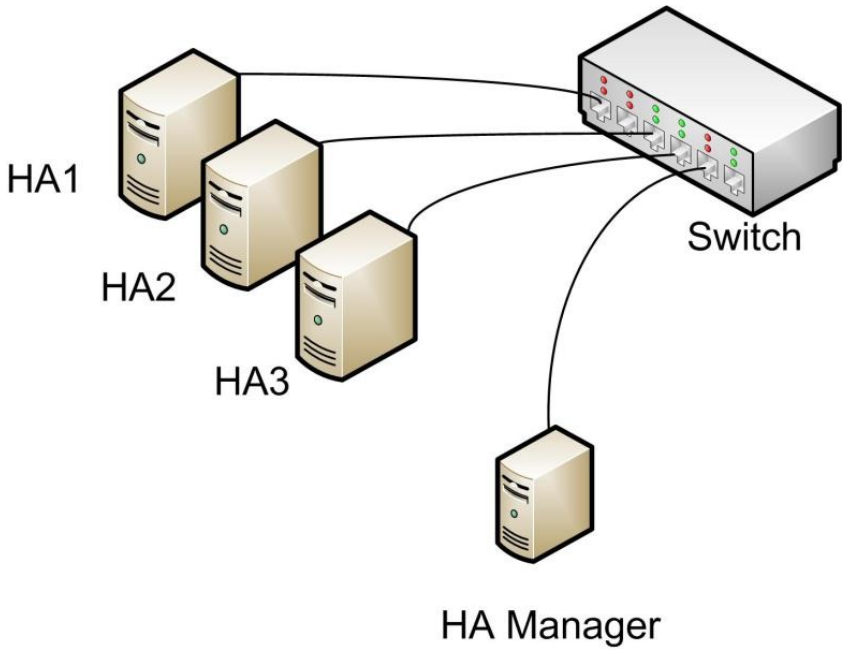

Figure 2.4: Typical HA deployment Scenario

If an MN is switched on in its home network, it does not require any HA and all the packets addressed to the MN on its HoA are routed directly to the MN. If the MN is switched on in a foreign network and is successfully authorized with Access Service Provider (ASP) then the next step would be to authorize its mobility service. For authorization, proposed protocols are either Diameter or Radius protocol. If mobility service authorization is successful, MSP AAA will send IP addresses of two HAs that can be used by MN. In a situation where MSP and ASP are the same, the whole process can be completed in one phase. In this case, mobility service will be authorized during the authorization request for network access.

HA Registration:

Once the mobility service is authorized, MN will send a modified

registration request to both HAs whose IP addresses are received from authorization server. In each registration request, MN will include IP addresses of both HAs. In registration request, first IP address represents primary HA for that MN and second IP address represents secondary HA. After completion of HA registration, MN will have two HAs, one primary and the other secondary. Using these procedures for HA registration, both HAs will know that for a particular MN there are two HAs

Failure Detection and Recovery:

In case of HA failure, it will be detected by HA manager in under 40ms. Once the failure is detected, HA manager will multicast “HA Failure” (HAF) message. It is assumed that all HAs are located on a single link, so multicasting is the most efficient way to broadcast this HAF message. Once this HAF message is received by HAs, they will check their binding table and look for MNs whose primary HA has failed and for which they are acting as secondary HA. Once all the MNs are identified, HAs will start sending HA switch message to those MNs.

HA Switching:

Once HA switch message is received by MN, it will switch its primary and secondary HA and communication can be resumed on the new primary HA.

HA Load sharing and load redistribution:

As part of failure detection process which is explained in the next section, each HA sends some higher level information to HA manager in a Layer 2 frame. This information includes number of active registrations and traffic throughput. HA manager stores this information in a table which is consulted for fair load distribution. Using this table, HA Manager knows the number of registrations on each HA and last 10 seconds average traffic value on their interface. Once HA Manager knows the number of registrations on each HA and traffic through each HA, it can decide the

least loaded HA based on developed cost function described in the following section. Load sharing mechanism starts with discovery / assignment of HA IP address. When HA IP address is assigned during mobility service authorization by Mobility Service Provider (MSP), AAA server will send additional attribute called HA-Address. Before sending the HA-Address attribute, AAA would check the next preferred HA with HA Manager using SNMP or a simple SQL query.

Load redistribution after the initialization stage is required in situations where an HA recovers from a failure or MNs suffering from a temporary connection loss in particular locality. Such a situation can result in un-fair load distribution. To resolve this issue, load redistribution among the HAs is required. It could be carried out by sending re-registration request to MNs that are using an overloaded HA as their primary HA. The IP address of a preferred HA could be included in the re-registration request. Upon receiving this request, the MN would send BU to new HA. Trigger for load redistribution procedure could be the detection of significant difference of load between different HAs. The MN whose HA will be re-assigned can be selected in accordance with certain criteria that may be developed based on the applications running on the MN and their QoS requirement etc.

2.3.1 Composition of the HA redundancy set

For economical reasons it makes no sense to have a one-to-one redundancy for all HAs. With separate dedicated backup HAs for all active HAs, the reliability solution could be realized as pure hardware backup and would not require any software-based backup mechanism. Therefore, for the proposed scheme a design assumption is that each HA needs to serve as backup for multiple HAs.

In an operational network it is reasonable to assume that HAs are all placed within a secure environment so it is not necessary to secure any

control protocol used between them. When HAs are in a secure environment, they can be located on single link. So in this scheme, it is assumed that all HAs are located on a single link. This allows us to use multicasting. With these design assumptions, we calculate the reliability of an HA deployment.

Let the random variable X represent the lifetime or the time to failure of an HA. The probability that HA survives until time t is called reliability R(t) of the component.

Thus

$$R(t) = P(X > 1) = 1 - F(t)$$

Where F is the distribution function of the HA lifetime, X. The HA is assumed to be working properly at time t=0 $[i.e \quad R(0) = 1]$and no HA can work forever without failure $[i.e \lim_{t\to\infty} R(t) = 0]$.

Consider a fixed number of identical HAs, N_o, under test. After time t, $N_f(t)$ HAs have failed and $N_s(t)$ have survived with $N_f(t) + N_s(t) = N_o$. The estimated probability of survival may be written as:

$$\hat{P}(survival) = \frac{N_s(\mathrm{t})}{N_o}$$

In the limit as $\mathrm{N_o} \to \infty$, we expect $\hat{P}(survival)$ to approach $R(t)$. As the test progresses, $N_s(\mathrm{t})$ gets smaller and R(t) decreases:

$$\mathrm{R(t)} \cong \frac{\mathrm{N_s(t)}}{\mathrm{N_o}}$$

$$= \frac{\mathrm{N_o} - \mathrm{N_s(t)}}{\mathrm{N_o}}$$

$$= 1 - \frac{\mathrm{N_f(t)}}{\mathrm{N_o}}$$

The total number of components N_o is constant, while the number of failed components N_f increases with time. Taking derivatives on both sides of the above equation, we get:

$$\acute{R}(t) \cong -\frac{1}{N_o}\acute{N}_f(t)$$

Where $\acute{N}_f(t)$ is HA failure rate. Therefore, as $N_o \to \infty$, the right hand side of this equation may be interpreted as the negative of the failure density function, $f\ (t)$;

$$\acute{R}(t) = -f\ (t);$$

Consider an HA deployment of n HAs with $X_i$denoting the lifetime of HA i *and* X denoting the lifetime of HA system.

Then

$$X = max\ \{X_1, X_2, X_3, X_4, \ldots..X_n\}$$

$$R_x(t) = 1 - \prod_{i-1}^{n}[1 - R_x(t)] \geq 1 - [1 - R_x(t)], \quad for\ all\ i$$

This implies that the reliability of an HA system is more than a single HA. Now calculating expected life of the system or the mean time to failure.

$$E[X] = \int_0^\infty R_x(t)dt \geq max_i\{\int_0^\infty R_{x_i}(t)dt\}$$

$$= max_i\ \{E[X_i]\}$$

Now if the X_i is an exponentially distributed function with parameter λ then

$$R_x(t) = 1 - (1 - e^{-\lambda t})^n$$

And

$$E[X] = \int_0^{\infty} [1 - (1 - e^{-\lambda t})] dt$$

Let $u = (1 - e^{-\lambda t})$; then $dt = \frac{1}{\lambda} du$.

Thus

$$E[X] = \frac{1}{\lambda} \int_0^1 \frac{1 - u^n}{1 - u} du$$

As above is sum of finite geometric series:

$$E[X] = \frac{1}{\lambda} \int_0^1 (\sum_{i=1}^{n} u^{i-1}) \, du$$

$$= \frac{1}{\lambda} \sum_{i=0}^{n-1} \int_0^1 u^{i-1} \, du$$

Note that

$$\int_0^1 u^{i-1} \, du = \frac{u^i}{i} |_0^1 = \frac{1}{i}$$

Thus, the MTTF of HA system with redundant HA is give by

$$E[X] = \frac{1}{\lambda} \sum_{i=1}^{n} \frac{1}{i} = \frac{ln\,(n)}{\lambda}$$

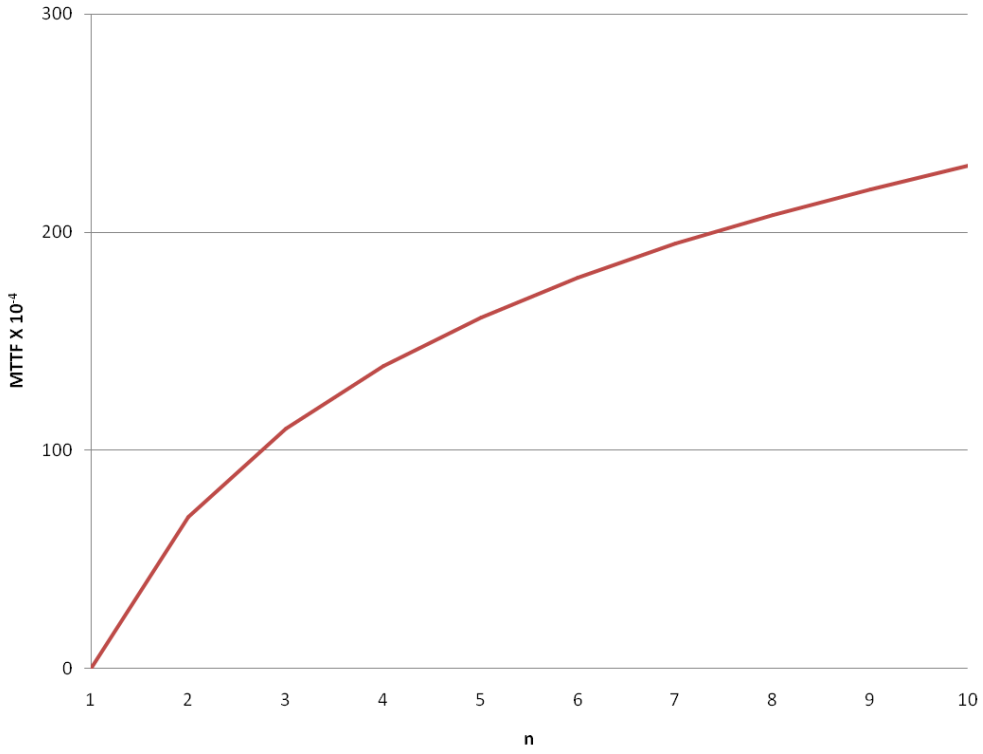

Figure 2.5: MTTF vs. number of redundant HAs

Figure 2.5 shows the expected life of an HA system as the function of n. It should be noted that beyond n=3 or 4, the gain in expected life is not very significant. Also the rate of increase in MTTF is $(\frac{1}{n\lambda})$.

This reliability model for HA system is accurate with the key assumption that if there is a failure of single HA, MN can switch the other HAs and this switching is 100% successful. Because of this assumption, it is not a very realistic model. In some cases it may not be possible to switch to a backup HA resulting in failure of recovery procedure. These failure are classified as un-recoverable and the probability that a given failure belongs to this class is denoted by $1-c$, where c denotes the probability of occurrence of recoverable failures and this is known as coverage parameter. Here is the mathematical model for such a system.

Let X denote the lifetime of an HA system with two HAs, one active and

the other backup. The failure rate of an active HA is λ. It is assumed that backup does not fail. Let Y be the indicator of failure class; this is

$Y = 0$ if failure is un-recoverable

$Y = 1$ if failure is recoverable

Then

$$P_y(0) = 1 - c \ \ and \ P_y(1) = c$$

To calculate the MTTF of such system, the conditional expectation of X give Y by noting that if a unrecoverable failure occurs the mean life of the system equals the mean life of the initially active HA. That is

$$E(X|Y = 0) = \frac{1}{\lambda}$$

And if a recoverable failure occurs, then mean life of the system is the sum of mean lives of two HAs.

$$E(X|Y = 1) = \frac{2}{\lambda}$$

Now total expectation of the system is

$$E[X] = \frac{1 - c}{\lambda} + \frac{2c}{\lambda} = \frac{1 + c}{\lambda}$$

Thus when $C = 0$, the backup HA does not contribute to system reliability and when $C = 1$, the expectancy of the system is affected.

Given that failure was recovered ($Y = 1$) the system lifetime, X, is the sum of two independent exponentially distributed variables, each with parameter λ. Thus the conditional probability distribution function of X given $Y = 1$ is two-stage Erlang density.

$$f_{X|Y}(t|1) = \lambda^2 t e^{-\lambda t}$$

If an unrecoverable failure occurs, lifetime X is simply the lifetime of initially active HA. Therefore

$$f_{X|Y}(t|0) = \lambda e^{-\lambda t}$$

Now the joint density is computed by $f(t,y) = f_{X|Y}(t|y)P_y(y)$ as

$$f(t,y) = \begin{cases} \lambda(1-c)e^{-\lambda t} & t>0, y=0 \\ \lambda^2 cte^{-\lambda t} & t>0, y=1 \end{cases}$$

And the marginal density of X is computed by summing over the joint density:

$$f_X(t) = \lambda^2 cte^{-\lambda t} + \lambda(1-c)e^{-\lambda t}$$

Therefore the reliability of the system is give by

$$R_X(t) = (1-c)e^{-\lambda t} + ce^{-\lambda t}(1+\lambda t)$$

$$= e^{-\lambda t} + cte^{-\lambda t}$$

$$= e^{-\lambda t}(1 + c\lambda t)$$

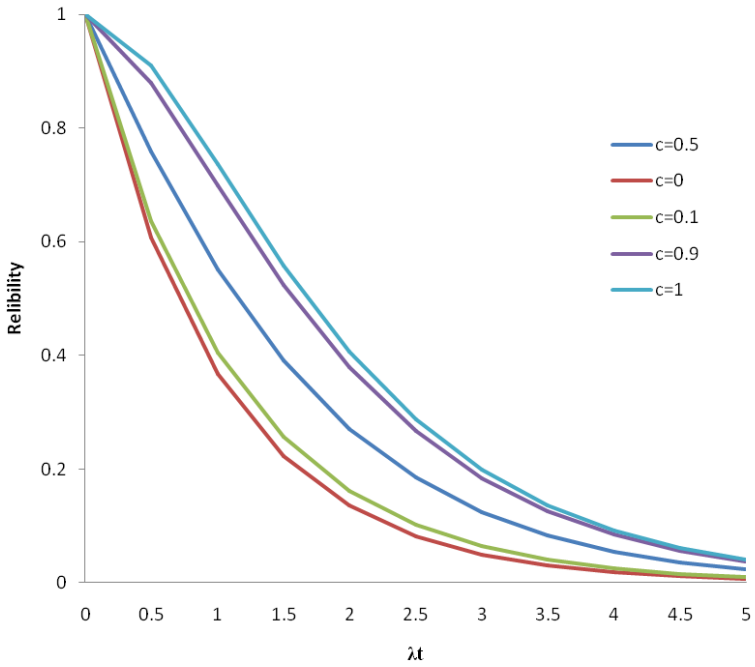

Figure 2.6: Reliability vs. coverage parameter

Figure 2.6 shows $R_X(t)$ as a function of T for various values of coverage parameter.

Typically, the reliability of a system (any piece of software or hardware equipment) is expressed in terms of its availability (**A**).

$$A = \frac{t_a}{t}. \qquad (2)$$

The formula states that the availability of a system is the percentage of time when the system is operational (i.e. the ratio between the time during which the system is available t_a and the total time t). The other two important parameters for availability are MTBF and MTTR. Mean Time between Failures (MTBF), as the name suggests, is the average time between failures estimated by the manufacturer of a piece of equipment. The average time between failing and the return to service is termed Mean

Time to Repair (MTTR). In an operational system, however, repair generally means replacing the hardware module, thus MTTR could be viewed as mean time to replace a failed hardware module. It must be noted that the lower the MTTR requirement, the higher the operational costs for the operator.

Availability can be expressed in terms of MTBF and MTTR by the formula given below.

$$A = \frac{MTBF}{MTBF + MTTR} \tag{3}$$

Availability is typically specified in nines notation. For example 3-nines availability corresponds to 99.9% availability. 5-nines availability corresponds to 99.999% availability.

Like software and hardware equipment, the best way to describe the degree of reliability of a service is through the concept of availability.

$$A = \frac{MTBD}{MTBD + MTTR} \tag{4}$$

Where MTBD is the Mean Time between disservices. Assuming that the mobility service is only dependent on the correct operation of the HAs.

In more similpfied terms, reliabilty is define as the ability of a system or component to perform its required functions under stated conditions for a specified period of time. Normally it is calculated in percentage of a system uptime (when its working properly) to total time. In case of HA agent it would be

$$\text{Availibility} = \frac{\text{HA}_{\text{UT}} - \text{HA}_{\text{DT}}}{\text{T}}$$

(5)

Where

HA_{UT} = Time when HA is working properly

HA_{DT}= Time when HA was not working properly.

HA_{DT} can be further split into two parts.

$$HA_{DT} = HA_{fdt} + HA_{rt} \quad (6)$$

Where

HA_{fdt} = failure Detection time

And

HA_{rt}= HA Recovey time

In most HA reliability mechanisms, emphasis is on second part of equation 6, which is the recovery time and first part is left un-touched. This proposed scheme presents a solution where overall HA down time is significantly decreased by reducing both failure detection time and recovery time, but main focus is on failure detection time. A novel load sharing scheme for HAs is also included.

The solution proposed for reliability is described in two parts. First, new radical failure detection scheme based on cross layer design is discussed and then actual reliability scheme is described.

2.3.2 HA Manager

In the proposed scheme, HA Manager would maintain a table called "Service Table". This table would contain three records. First record for HA address, second record for its bandwidth utilization and third record would be active binding on that HA. This table would be populated by

heartbeat messages received from different HAs. Data from this table would also be available to other entities on the network.

Another function of this HA Manager is to perform HA failure detection and initiation of recovery procedures which is explained in the next section.

2.3.3 HA Failure Detection

In this scheme, HA Manager has been introduced to the network. As the name suggests, it would be responsible for monitoring HAs. In proposed scheme, all HAs will send a heartbeat message to the HA Manager. When HA Manager receives this message, it will analyze the data payload and will update its service table. If three consecutive heartbeats are not received by HA Manager for a particular host, it is declared to be failed. This scheme is not quite different from the current practice but the novelty is within the message that is being sent. Heartbeat message would be based on layer 2 and will carry higher layer information in its payload. Since this scheme is based on layer 2, the interval of this heartbeat message can be greatly reduced without adding significant load to the network. Simulation results shows that we can easily reduce this interval below 10 ms but for this research, aim is to detect failure within 40 ms and for this 10 ms is enough.

Typical Deployment scenario is presented in Figure 2.7. In this Figure there are three HAs labeled as HA1, HA2 and HA3 which are monitored by HA Manager.

Procedures for scheme are as following:

- HA would send L2 frames to HA Manager at the rate of 100 frames/sec
- These L2 frames are generated by a thread of service daemon. In this case, HA daemon would be responsible to generate the Frames.

- HA Manager would process the frames and update its service table based on the information available in frames.
- If three consecutive frames from any particular node are missing, either the node or the service that is monitored will be declared failed.

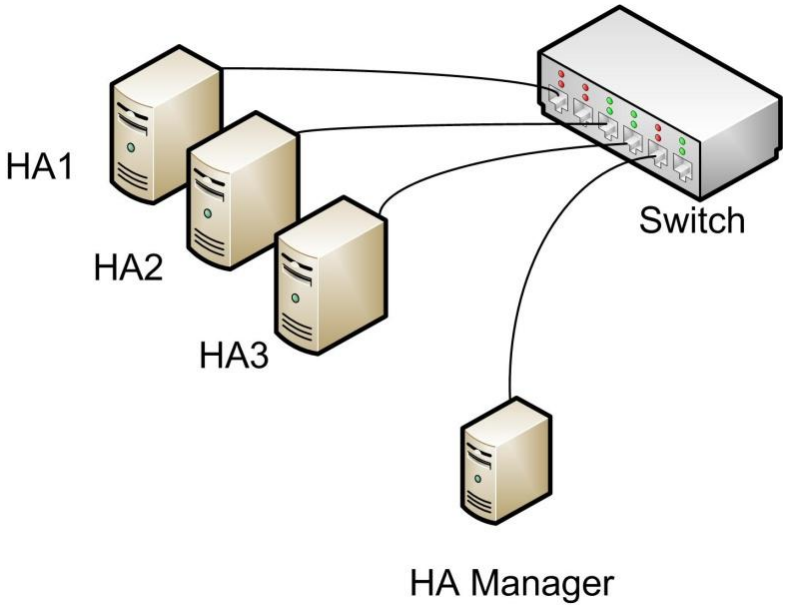

Figure 2.7: HA deployment Scenario

Packet Format:

The structure of this periodic heartbeat message is critical. If the packets are huge, they cannot be sent frequently as they would generate excessive load on network. If packet size is small, it cannot carry enough information. To address this problem, a radical approach based on cross-layer design is adopted. Application layer encoded information is embedded in payload of L2 frame. This unique method would resolve the dilemma by keeping the packet size small and still carrying enough information.

Figure 2.8 presents the actual packet structure that is used for heartbeat messages. The frame is designed to have the capability of using IPv4 and IPv6 transport mechanism if required. This frame can carry eight different parameters related to a single HA. For evaluation of the scheme, only two parameters are considered. 16 bits are assigned for each parameter. In the parameters section of packet, first 16 bits represents number of active registrations on HA and next 16 bit represent the bandwidth consumption is kilobits. Other possible parameters could be CPU utilization, RAM utilization, available disk space etc.

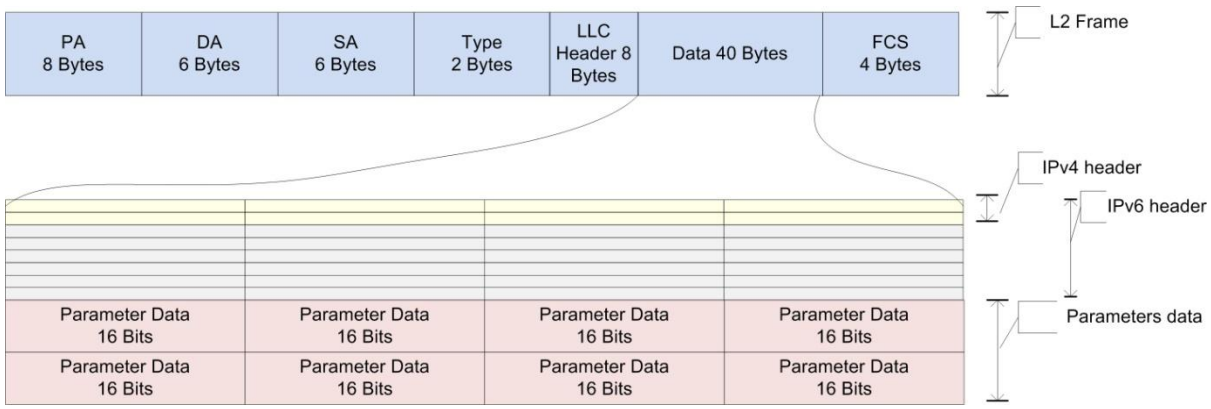

Figure 2.8: Packet Format

From this packet structure it is very clear that detailed information is being sent by HA utilizing very small bandwidth. If fact it used about 74 bytes or 592 bits. This frame supports up to eight different parameters.

In this frames, 256 bits are left blank which can be used by either IPv6 or IPv4 header. In situations where different HA are across different VLANs or different subnets, an IP transport scheme is required. Normally IP transport scheme is avoided to reduce load on network routers. Network Routers are usually slow as compared to Network Switches. If a single HA is sending 100 packets per second it will not affect routers in anyways but consider a situation in which there are ten HAs and each is sending 100

packets per second. In this case, routers need to process 1000 packets per second. This may affect performance of some small routers. To avoid such a situation it is preferred to use L2 frames and thus it will only engage network switches which are not be affected by this extra load.

To meet the requirements [66] of commercial telecom network of five nine i.e. 99.999% error/failures detection process should be quick. In our actual testbed, failure detection time was less than 40ms while utilizing network capacity by only 0.01 percent. Details of testing are mentioned in result section.

2.3.4 Modified Binding Update

A modified binding update procedures is proposed for this architecture. In this scheme, MN will register itself with atleast two HAs. One is primary HA and other is secondary HA. While sending binding update message to either of HA, IP address of both primary and secondary HAs should be included. This will make HA aware of MN's association with any other HA.

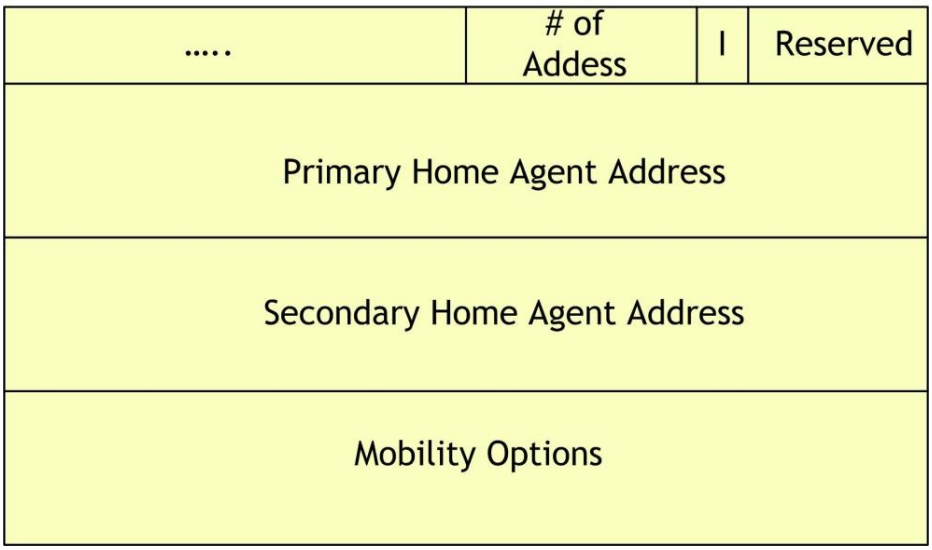

Figure 2.9: Modified Binding Update Message Header

2.3.5 Switching the HA at the MN

If there is a failure of primary HA for a certain MN, then its secondary HA will be able to request MN to switch its primary HA. This is performed via HA switch message. This message can optionally carry IP address of HA.

Figure 2.10: HA Switch Message Header

In case of failure of the primary HA for a certain MN, the secondary HA will send the HA switch message to MN. Once this message is received by MN, the secondary HA will become its primary HA and a registration request would be sent to the HA whose IP address is include in HA switch message. This new HA would become the new secondary HA.

2.3.6 HA loadsharing

This module is responsible for sharing load fairly between different HAs. In MIPv6 environment, dynamic HA allocation is not a feasible solution because if the assigned HA is switched frequently, it will create disruption in MN's communication. The switching process involves sending HA switch message, HA de-registration request, followed by de-registration acknowledgement, sending BU to the new HA and waiting for BA. Assuming the end-to-end delay from HA to MN is X milliseconds, then the switching time can calculated as follow.

$$T_{Switch} = T_{DeReg-BU} + T_{DeReg-BA} + T_{BU} + T_{BA} + 4X$$

This T_{Switch} ranges from 1-3 seconds because of de-registration and registration process which involves HoA link-address removal from one HA and its advertisement on another HA.

Since frequent HA switching is not feasible, the load sharing must be performed during HA assignment phase. The variables that are considered during the assignment phase are number of MNs registrations, traffic throughput and the load-handling capacity. With these variables there are three possible approaches for HA load sharing; first, based on number of registrations, second, based on traffic throughput and the third based on load-handling capability of HA. . Three different approaches have been explored and simulated. Simulations were run in MATLAB. First approach tries to equalize the number of registration on all HAs. Second approach tries to equalize the network traffic through each HA. Third approach tries to equalize the load handling capability of each HA. This load handling capability is calculated based of following cost function.

$$LCHA = \left\{ \frac{(Rmx \times Tmx) - (Rcr \times Tcr)}{(Rmx \times Tmx)} \right\} \times 100 \quad (7)$$

Where

LCHA= Remaining Load handling capacity

Rmx = Maximum Registrations

Tmx = Maximum Throughput

Rcr = Current Registrations

Tcr = Current Throughput

This cost function is simulated in MATLAB and was test for fairness. Advantage of load sharing based on this cost function is that it can be deployed for HAs having different load handling capabilities. For the first two approaches, load handling capabilities of HAs must be the same.

2.4 Operational scenarios for HA reliability

Consider a scenario of Figure 2.7. In this scenario, there are three HAs and a single HA Manager. Consider a situation where an MN is registered with HA1 and HA2. HA1 is the primary HA for this MN and HA2 is secondary HA. If there is failure of HA1, HA manager will detect it and declare that HA1 has failed by sending a HAF message on HA multicast address. Once this message is received by MN's secondary HA, it will send HA switch message to MN. This process is presented with help of a signaling diagram of Figure 2.11.

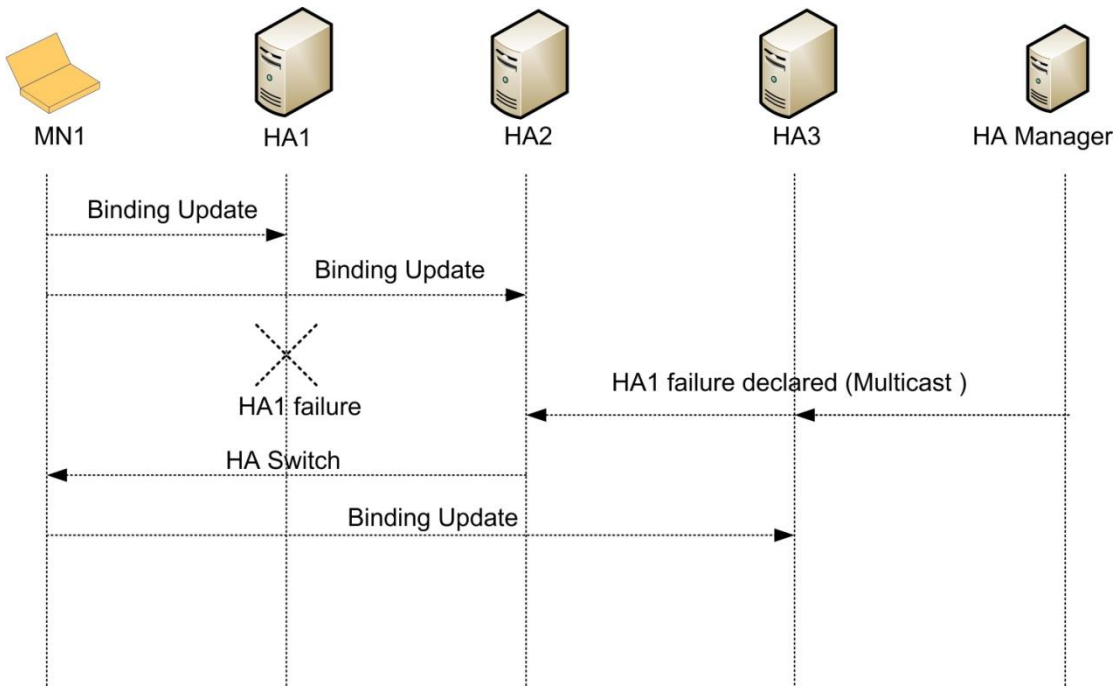

Figure 2.11: Typical HA Failure

As illustrated in this signaling diagram, MN registers with two HAs. The failure of HA is detected by HA Manager which then multicasts HAF message. When this HAF message is received by HA2, it send HA Switch message to MN.

MN Registration: The proposed scheme suggested a modified registration processes which would enable MN to get registered with two HAs.

Failure Detection: Let's consider the situation where MN1 has primary HA namely, HA1 and secondary HA HA2. Now if there is a failure of HA1 after time T, it will be detected by HA Manager. Once this failure is verified, HA Manager multicasts HAF message, on HAs multicast address indicating failure of a specific HA.

Recovery Procedure: Once HAF message is received by HA, it will check if it is acting as secondary HA for any MN whose primary HA has failed. If it finds one, it will send HA switch message with IP address of HA3. Once this message is received by MN1, it will discard its current primary HA and will nominate its secondary HA as its new primary HA. After this, MN1 will send a new registration request to HA3 using the modified Registration request message. This modified registration message will include IP address of HA2. Once the registration is completed, HA3 would know that it is acting as secondary HA for MN1. In this way MN1 can quickly recover from HA failure. Once this process is completed, MN1 would still have a primary HA and a secondary HA.

In a situation, were there is failure of both primary and secondary HAs for a particular MN, the procedure would not be able to recover from it.

2.5 Scheme Verification

For the verification of proposed scheme, it is simulated in two parts. First the failure detection system is simulated and then load distribution

schemes is simulated. For failure detection scheme, Network Simulator 2 (NS2) is used. For load distribution scheme, MATLAB is used as simulation tool.

2.5.1 Failure Detection Scheme

The proposed failure detection scheme is simulated in NS2. Different aspects for recovery scheme including failure detection time, effect of this scheme on network applications and its bandwidth utilization, are studied. The scenario developed in NS2 is shown in Figure 2.12.

In the figure there are three nodes. Node 0 is HA manager, Node 1 is HA and Node 2 represents a test node. In simulation run, HA was sending L2 frames to HA manager every 10ms. Test node can request both CBR and UBR data from HA.

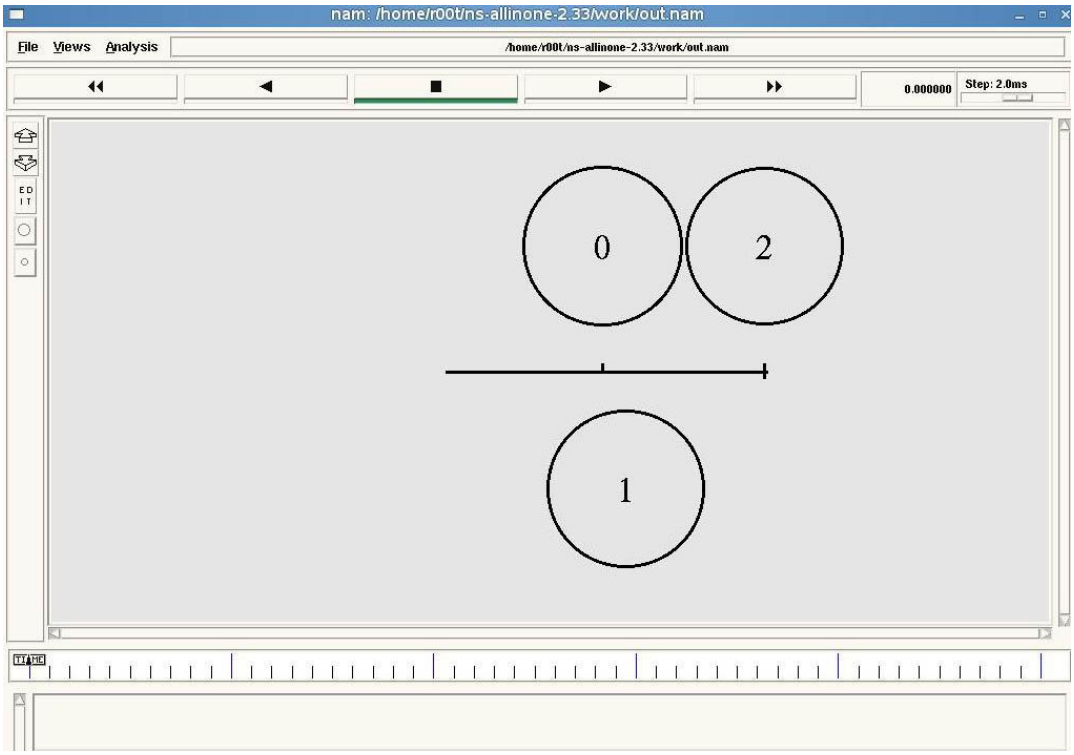

Figure 2.12: NS-2 Test Scenario Screenshot

For simulation, L2 frame with length of 74 bytes or 592 bits is created. Nodes were connected through a 100Mbit/sec Local Area Network (LAN) using IEEE 802.3 standard. Network latency was set to 4ms. This 4ms latency is a very huge delay in term of today's standards. These days different venders provide Ethernet switches with latency of 3-4 micro seconds. Examples are Cisco Nexus 7000 Series switches which can handle 480 million packets per second. Selection of 4ms was to study the effect of failure detection scheme is worst possible case.

2.5.1.1 Failure Detection time

Failure detection is based on missing three consecutive frames. HA will send L2 frames to HA Manager at rate of 100 frames/sec. HA Manager will process the frames and update the table based on the information available in frames. If three consecutive frames from any particular node are missing, either the node or the service that is monitored has failed. This theory was tested in NS2 and the results are presented and compared with MIPv6 [5], VHAHA [11] and VHARP [12] in Figure 2.13. In MIPv6, the failure detection time is around 63 seconds. In VHARP the failure detection time is about 3.2-3.5 seconds (data was sourced from [12]). In VHAHA the failure detection time is 4.2-4.8 seconds (data was sources from [11]). In contrast, the proposed scheme was able to detect the failure within 40ms. This faster failure detected is possible because of the new radical approach to detect failures with the help if L2 frames rather than L3.

For testing, HA failure is simulated (Node 1 in figure 2.12) by stopping the transmission of L2 frames. This was detected by HA manager and reported in nam file. This process is repeated about 20 times to calculate average failure detection time. The recorded time was between 34ms and 46ms. In theory this detection time should vary between 30ms and 40ms but because of network latency and end to end delay the actual variation is

between 34ms and 46ms. In comparison to exiting solutions, the proposed scheme provides far superior performance.

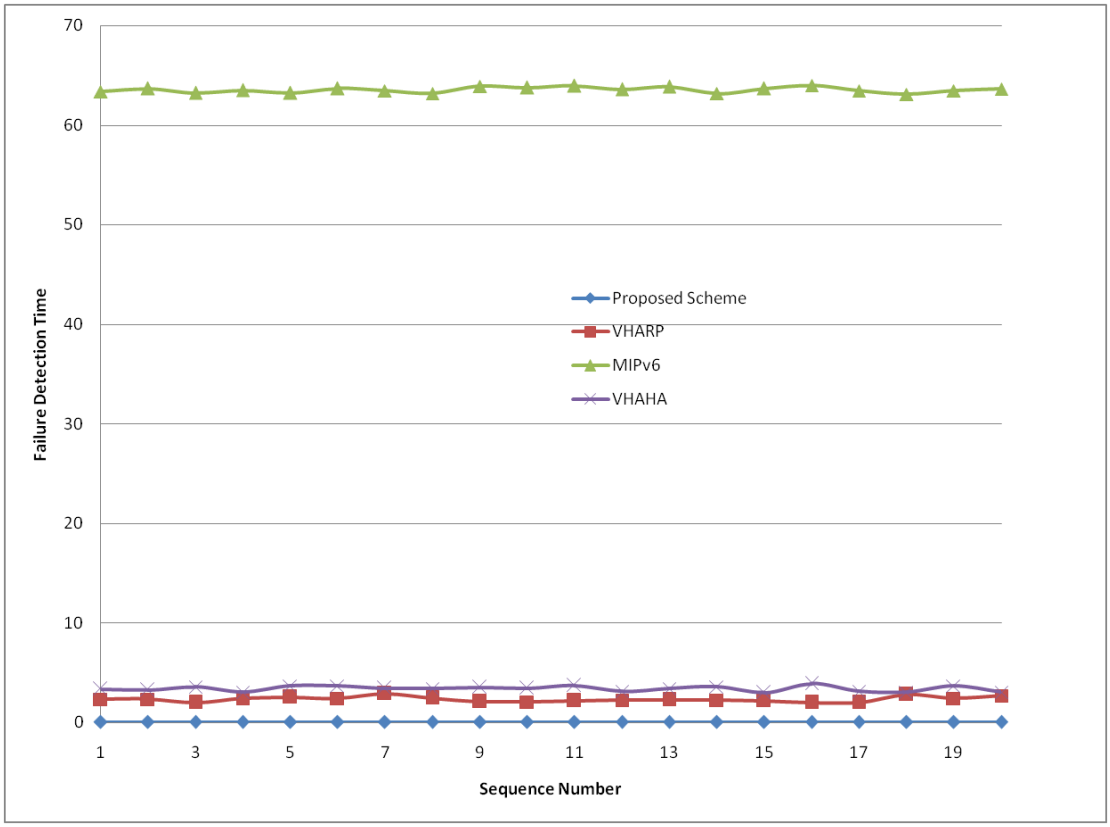

Figure 2.13: Failure Detection Time

2.5.1.2 Effect on network Applications

After analyzing failure detection time the effect of proposed scheme on network applications is studied. The application selected for testing is FTP. First end-to-end delay of FTP is analyzed without introducing L2 frames and then with introducing L2 frames.

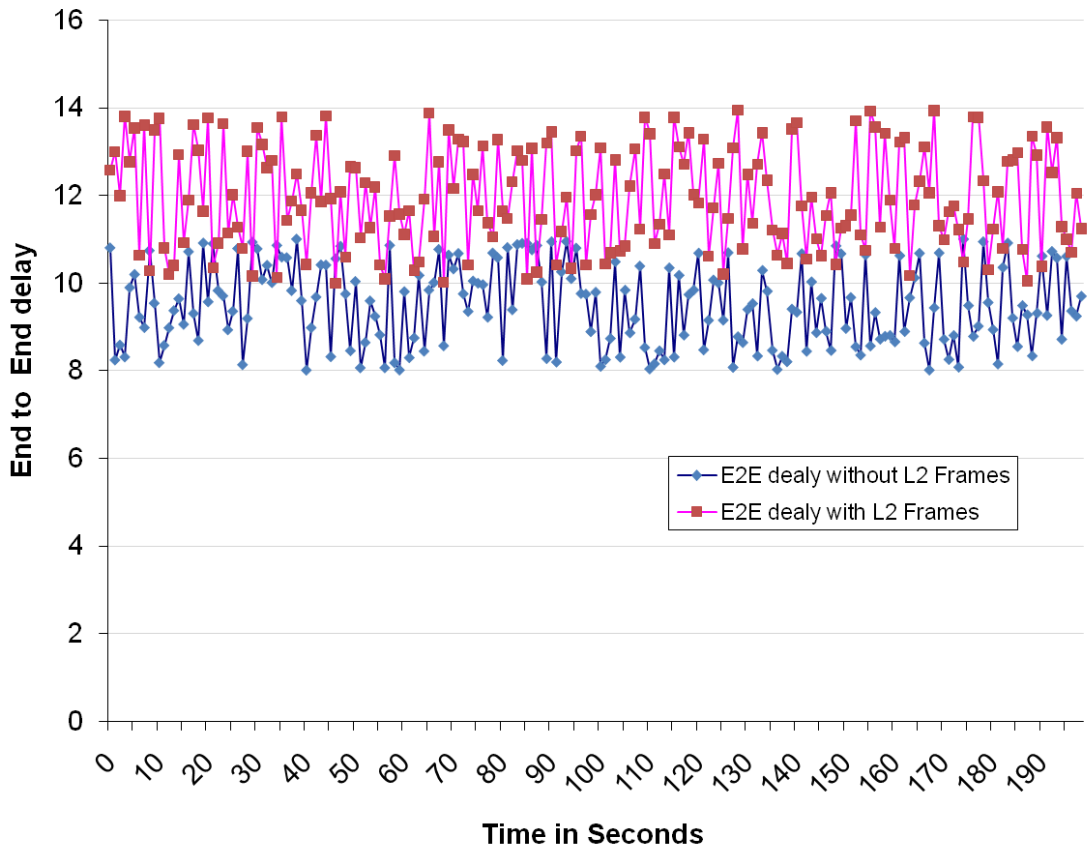

Figure 2.14: Effect of L2 Frames

In this plot x-axis represents time in seconds and y-axis represents end-to-end delay for FTP application. In the plot blue line represents end-to-end delay without introducing L2 frames and red line represents end-to-end delay with introducing L2 frames. Before introducing L2 frames the delay ranges between 8ms and 10ms and after introducing L2 frames it ranges between 10ms and 14ms. End-to-end delay increases but the increase is not very significant.

2.5.1.3 Bandwidth Utilization

After calculating failure detection time and effect on end-to-end delay we analyze bandwidth consumption. First theoretical bandwidth requirement is calculated.

Calculations based on 802.3 MAC

MAC Preamble & SD	8 Bytes
MAC Source Address	6 Bytes
MAC Dest. Address	6 Bytes
Length	2 Bytes
Payload	48 Bytes
CRC	4 Bytes
FRAME Size	74 Bytes
Inter Frame Gap	12 Bytes
Total Frame Size	86 Byes or 688 bits

Table 2.1: Mathematical Calculations

If this frame is sent every 10ms it would be 688x100= 68.8 k bits per second and on 100Mbit/sec it is still less than 1% utilization and in case of Giga bit Ethernet, it would be less than 0.1 percent.
Next the bandwidth consumption is tested in NS2 simulation. The result of this simulation in presented in Figure 2.15.

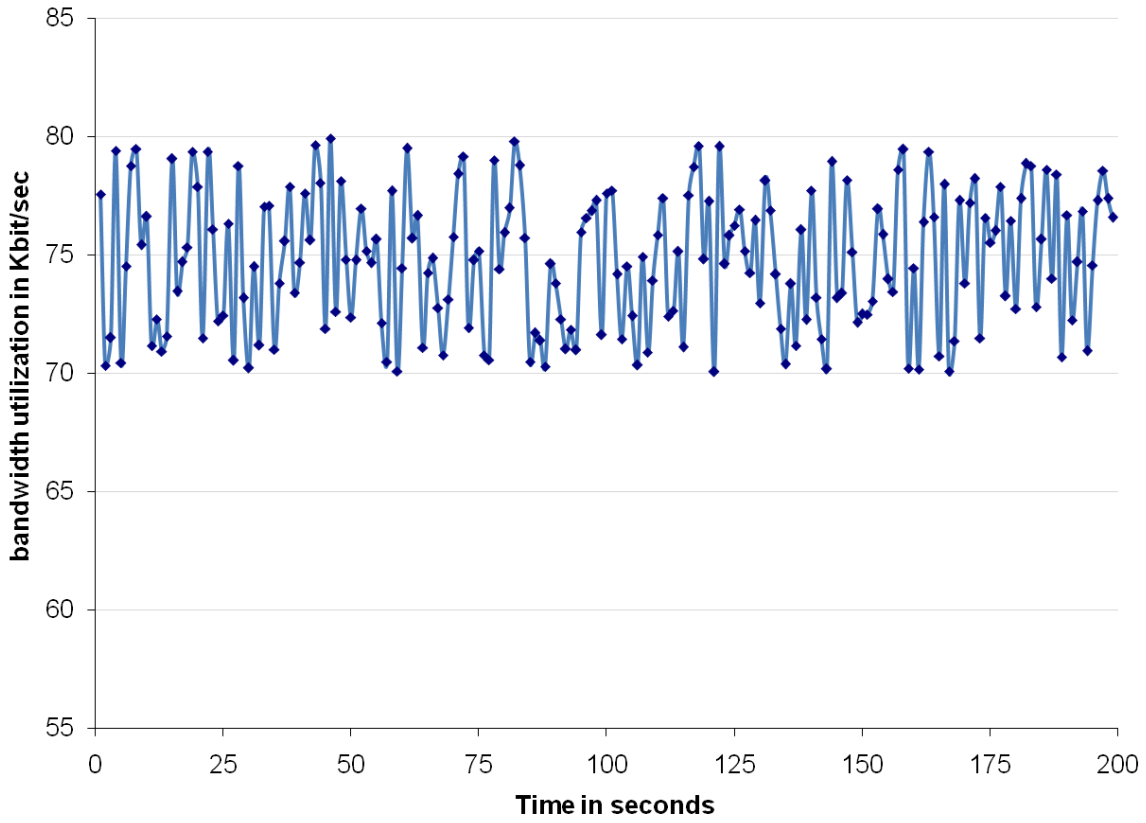

Figure 2.15: Bandwidth Consumption of L2 Frame

If these results are compared with the theoretical results, they are not very far from each other. The bandwidth consumption varies from 70kbit/sec to 80kbit/sec which is less than 1% of total available bandwidth.

2.5.2 Loadsharing scheme

As discussed earlier, three possible methods for HA load sharing are proposed. These three different approaches were simulated in MATLAB.

2.5.2.1 Load Sharing based on Number of Registrations

In this method it was same number of binding were kept on each HA without exceeding the maximum limits of both bandwidth and registered binding on all HAs. This was simulated in MATLAB. User's traffic was normally distributed from 0.1Mb/s to 1Mb/s with an average of around

0.5Mbits/sec. Simulation was run for 50 minutes. Average users registration rate was 2.5 registrations/min. Figure 2.16 shows that all HA have almost equal number of registrations. Figure 2.17 shows HA throughput vs. time. In this graph, there is a variation in throughput. The load sharing method based on the number of registration is very useful with the assumption that "standard deviations of user's traffic is small" remains valid. .

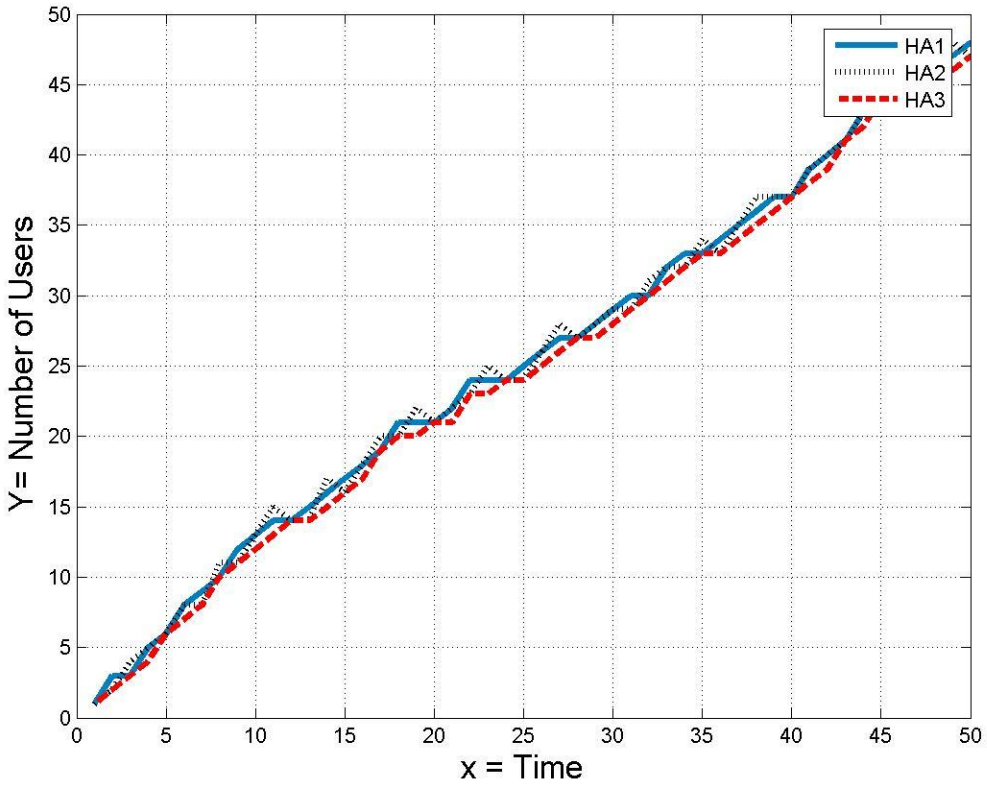

Figure 2.16: Number of users vs. time

The first approach was to equalize the number of MN registration on all HAs. For test case three HAs are considered. From this graphs in Figure 2.15 it is clear that there is no significant variation in number of MN registrations on each HA. This graph represents one side aspect. Now next graphs load/traffic through each HA is considered.

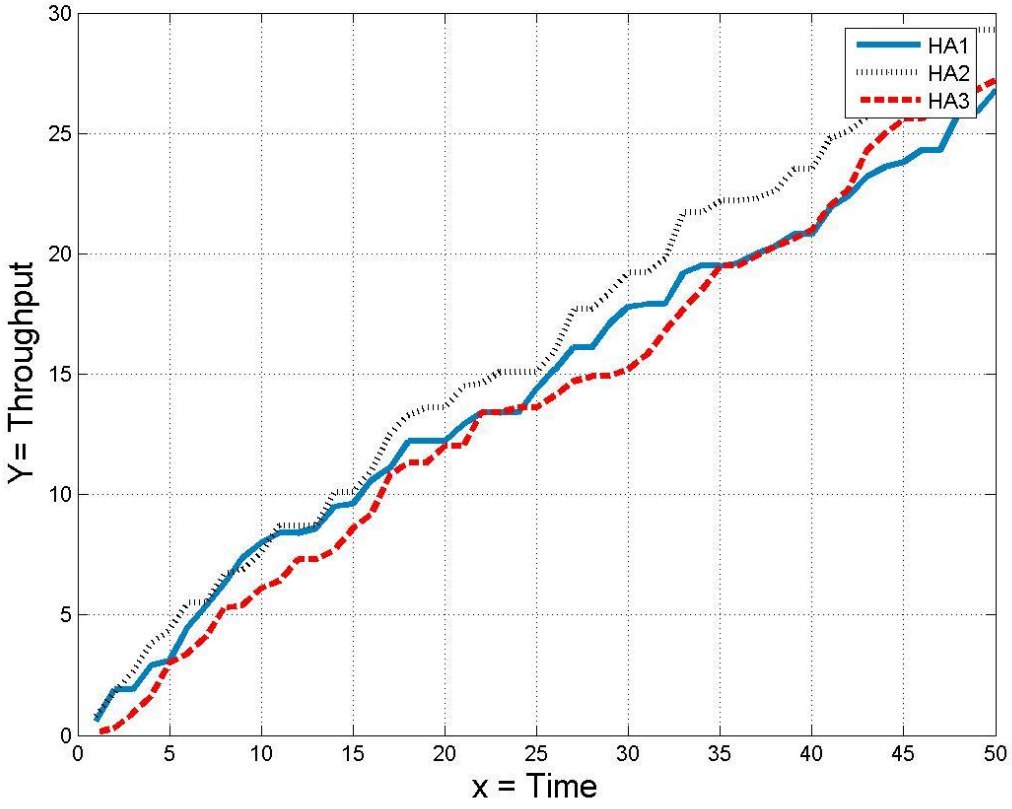

Figure 2.17: Bandwidth vs. Time

The plot of Figure 2.17 represents the variation in network traffic that is passing through each HA. As the number of registrations is equalized there is variation in network traffic. For simulation, load was normally distributed. In theory, if traffic generated by each HA has a normal distribution and if number of registrations are equalized, there should be no significant variation in the throughput of each HA. From the plot, this theory is validated.

Overall this approach should be very useful for load sharing if traffic generated by each MN is normally distributed but this assumption may not be always valid.

2.5.2.2 Load Sharing based on Bandwidth Consumption

In this method traffic generated on HAs is equalized. For simulation, MN traffic is normally distributed from 0.1Mbits/s to 1Mbits/s with an average of 0.5Mbits/sec. Simulation time is 50 minutes. On average, user registration rate is 2.5 registrations/min.

In this approach, emphasis was on overall traffic through each HA.

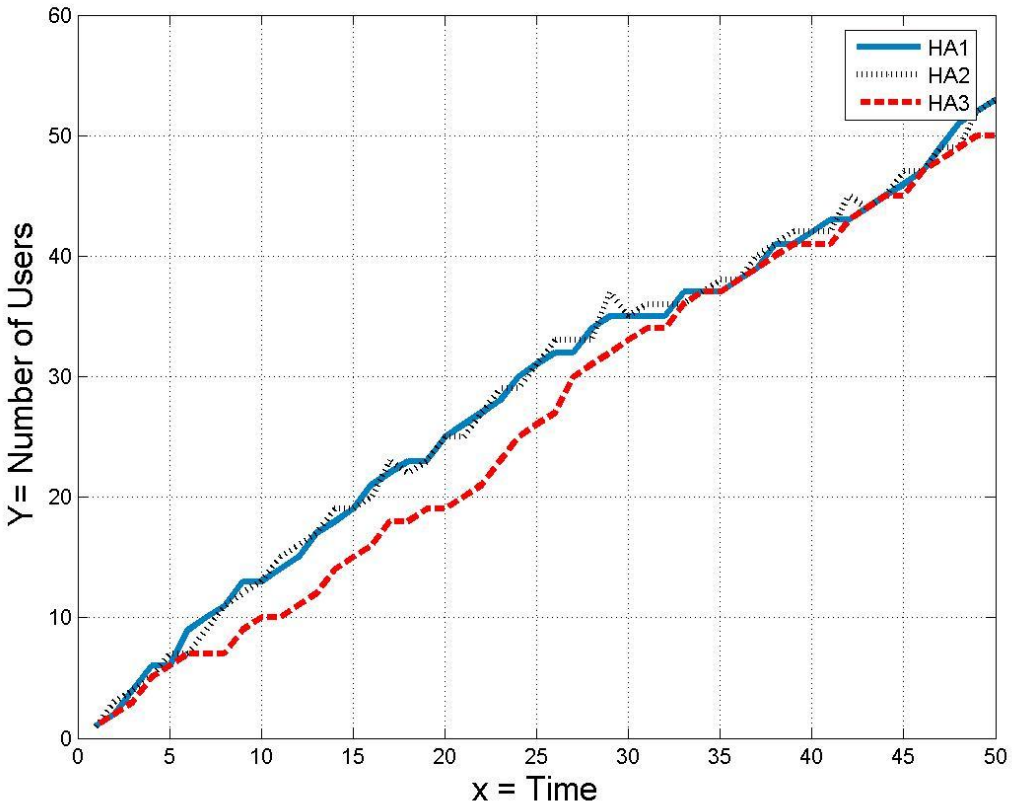

Figure 2.18: Number of Users vs. Time

The result of simulation is presented in Figure 2.18. In the Figure, there is significant variation in the number of registrations on each HA but with time, as the number of registrations increases, this difference is reduced. This is because the simulated traffic is normally distributed, and as the

number of registrations is increased, the average traffic generated by all MNs on different HAs is almost the same. Since load through HAs is equalized, numbers of registrations on each HA are automatically equalized.

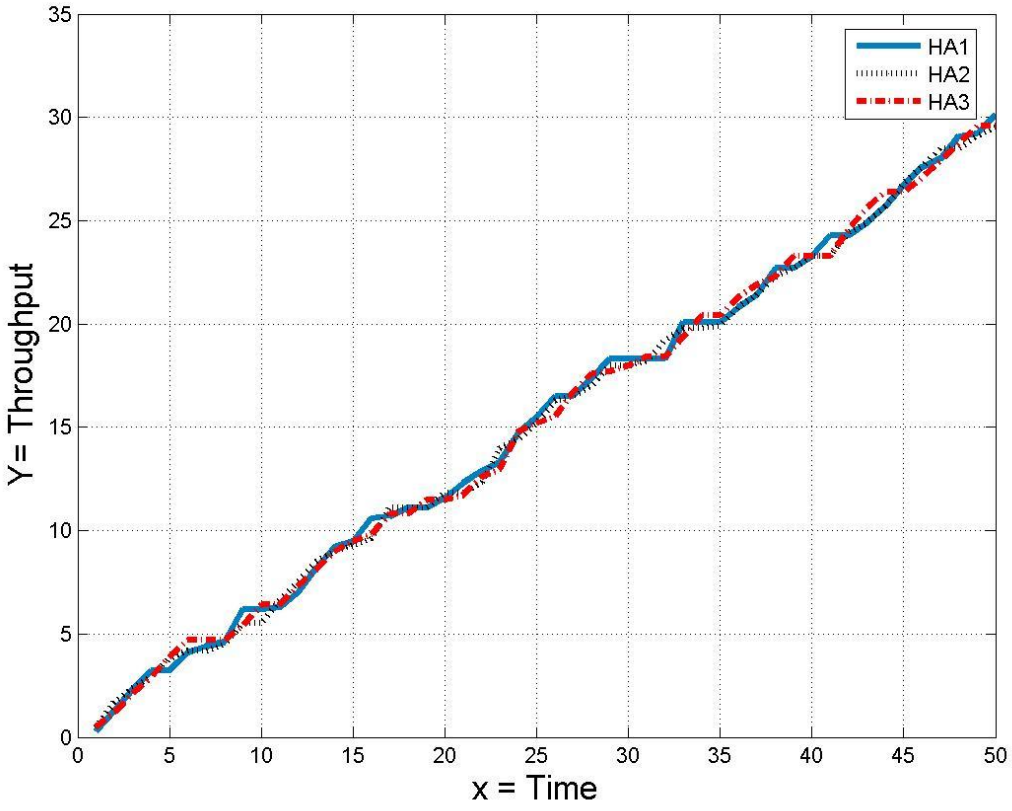

Figure 2.19: Bandwidth vs. Time

In this plot of Figure 2.19, there is very little variation between the traffic through each HA. This validates that proposed scheme can distribute network traffic equally across the deployed HAs. One drawback of this scheme is that it is based on assumption that variation in MN generated traffic is very little. But this assumption is not very realistic. In real life environment, some MNs may be just browsing the web and others may be streaming videos. In such situation this approach will not be very useful.

2.5.2.3 Load sharing based on both number of registration and bandwidth consumption

In this scheme, HA assignment is based on available load handling capacity. For simulation purpose, MN traffic is normally distributed from 0.1Mbits/s to 1Mbits/s with average of 0.5Mbits/sec. All HAs were considered to have same maximum throughput and maximum user handling capabilities.

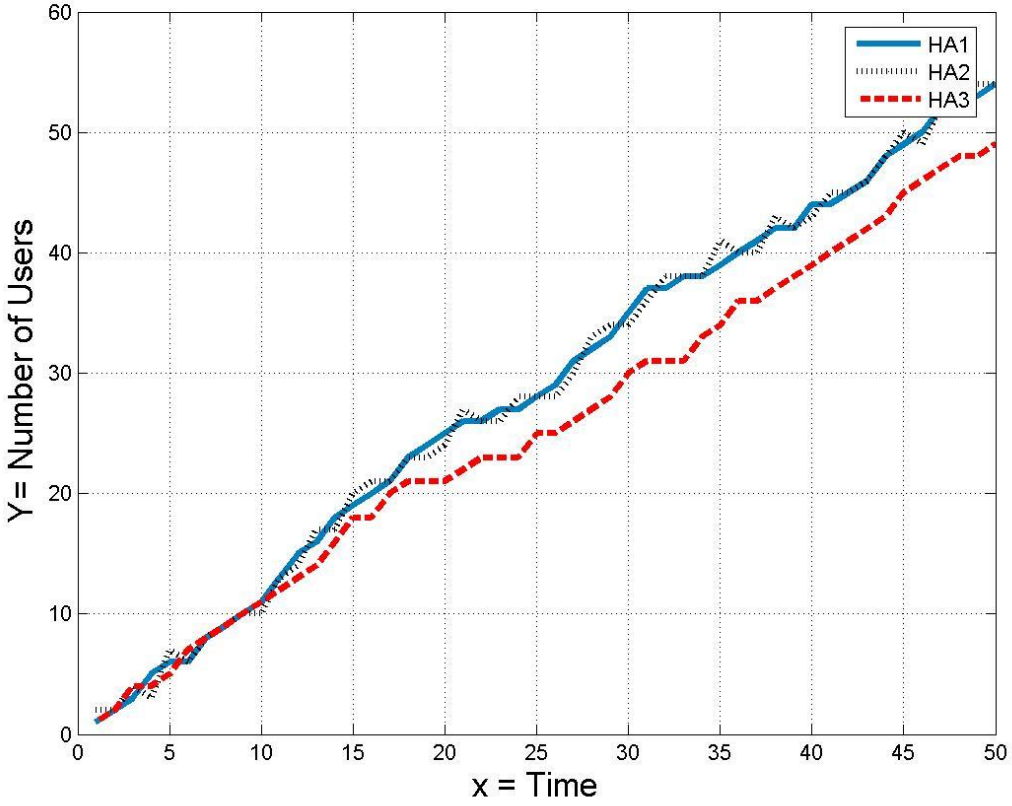

Figure 2.20: Number of Users vs. Time

In this approach load was distributed in accordance with the cost function developed earlier where load is distributed based on available load-handling capacity. For test case, three HAs were considered all

having equal load sharing capabilities. In plot of Figure 2.20, there is a variation in the number of registrations on different HAs.

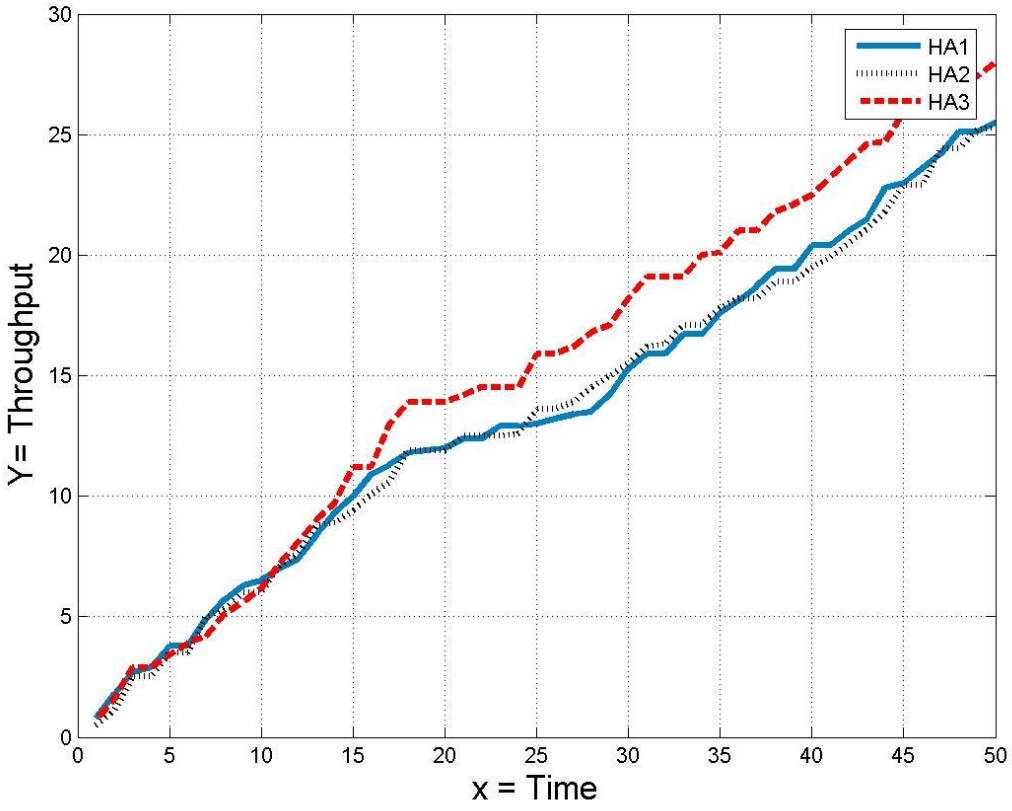

Figure 2.21: Bandwidth vs. Time

The graph of Figure 2.21 presents the variation in network traffic passing through each HA. Load handling capacity was equalized and as a result there is a variation in network traffic. For simulation purpose, load was normally distributed. Overall, this approach should be very useful for load sharing as it is independent of traffic variation between MN.

2.5.3 Comparison of the Loadsharing Schemes

In this section a comparison of three possible HA load sharing schemes is presented. The first scheme is based on equalizing total number of registrations. This scheme is suitable where all HAs have an equal load-handling capacity. This approach is more realistic because the HA will have to maintain IPsec tunnel with each MN and this IPsec tunnel creation requires high speed memory (normally the RAM) which has a limited availability on HAs.

Next scheme is based on equalizing the traffic throughput across different HAs. During simulation it is observed that this scheme has fairly distributed HAs network traffic but the downside is that it is based on assumption that variation in traffic generated by different MNs is not very significant. This assumption is not valid because in real life it is easily possible that a mobile user will use his mobile device just to check emails while another will be using his device for video conferencing.

The last scheme is based on equalizing the available load-handling capacity. This scheme is very useful in situations where different HAs have different load-handling capabilities but the complex nature of the scheme makes it difficult to implement. This scheme and the scheme based on equalizing number of registration have almost similar results.

Because of simple implementation and the effectiveness, the first approach is preferred over others.

2.5.4 Comparison with existing load sharing Schemes

In this subsection, the proposed HA load sharing schemes are compared with Deng's [22] and anycast load sharing of MIPv6. The technical report associated with reference 12 presents a simulation of Deng's load sharing scheme. In that OPNET simulation, 100 MN are considered. The simulation duration is 3000 seconds. MN registrations are simulated between 100s and 200s. Figure 2.22 presents the simulation results.

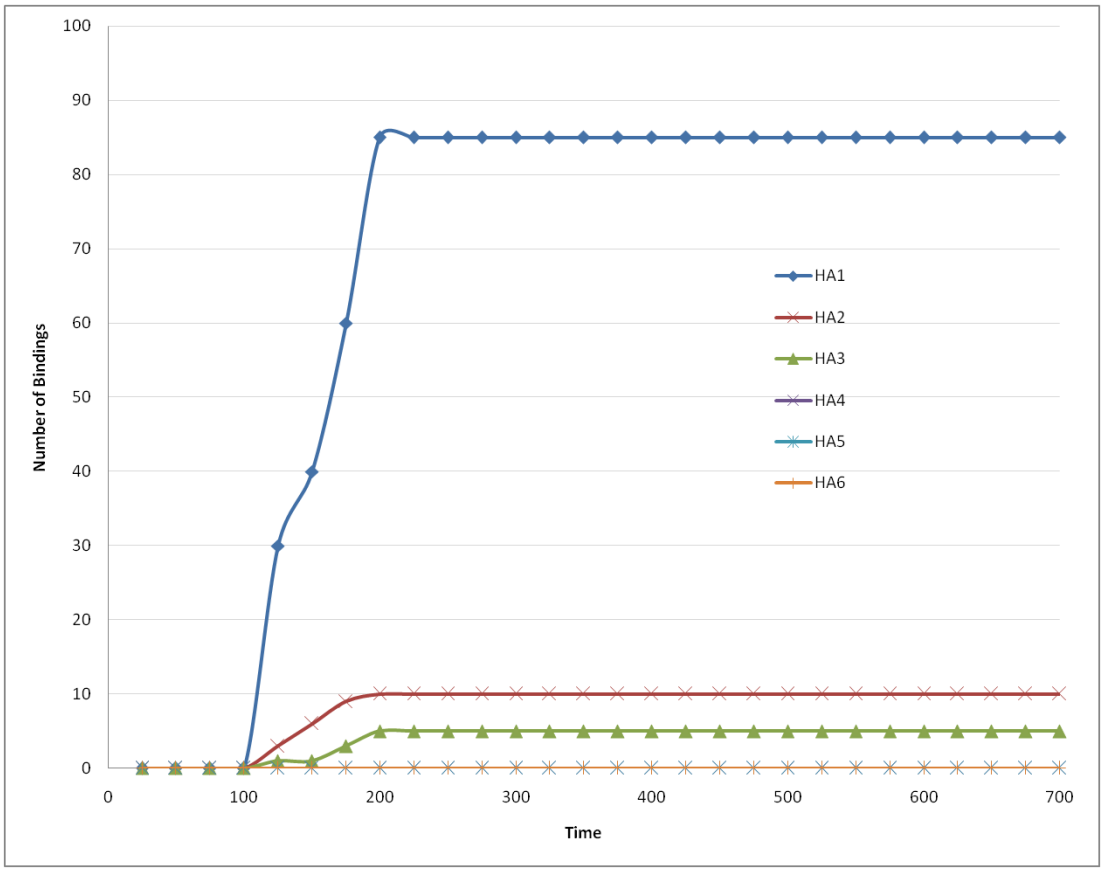

Figure 2.22: Deng Loadsharing for HA

In this plot it is obvious that Deng load sharing scheme fails to distribute load fairly among the HAs. After completion of simulation, HA1 registered 86 MNs, HA2 registered 10 MNs and HA3 registered 5 MNs while HA4, HA5 and HA6 did not register any MNs. The main reason for this poor distribution of load is that all MNs used DHAAD to explore HAs on home network. Once all HAs were explored, the most preferred HAs were selected by MN. If these results are compared with any of the three proposed schemes, it is obvious that the load sharing capabilities of the proposed schemes are far superior to that of Deng's load sharing scheme.

Next, the proposed load sharing schemes are compared with MIPv6 standard's load sharing. In MIPv6, load sharing is based on anycast addressing where MNs registration requests are distributed in round robin fashion. This process is simulated in NS2 with simulation time of 25 minutes. The registration rate is 15 users per minute. The plot in Figure

2.23 shows that this scheme is capable of sharing load equally among three HAs. If this scheme is compared with any of the proposed load sharing schemes then in term of results there is no significant difference.

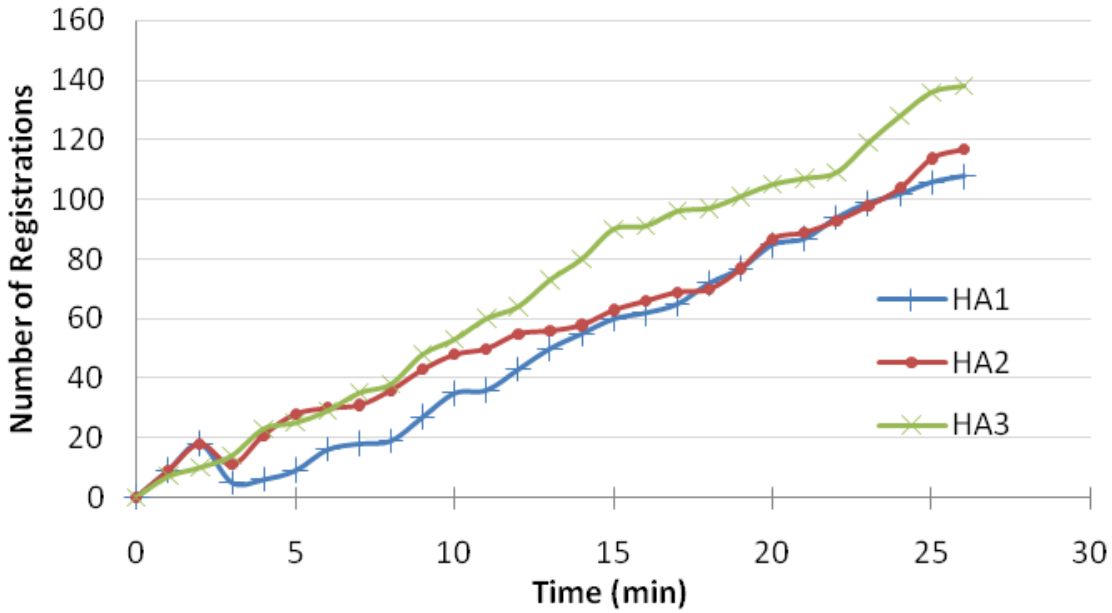

Figure 2.23: Anycast HA loadsharing

Now consider a scenario where load sharing is based on anycast addressing and there is failure of an HA. In such a situation, anycast addressing will continue to distribute HA registrations between all HAs. Any MN, whose registration request is forwarded to the failed HA, will not get registered and the MN will have to re-send the registration request. Now if the failed HA recovers, it will start processing the registration requests but initially it will have zero registrations. So at this point there will be a significant difference in the number of registrations on each HA. This scenario is simulated in NS2 where simulation time was 25 minutes with average registration rate of 15 users per minute. One of the HA was failed after 12 minutes into the simulation. The resulting distribution of MN registrations on different HAs is illustrated in Figure 2.24.

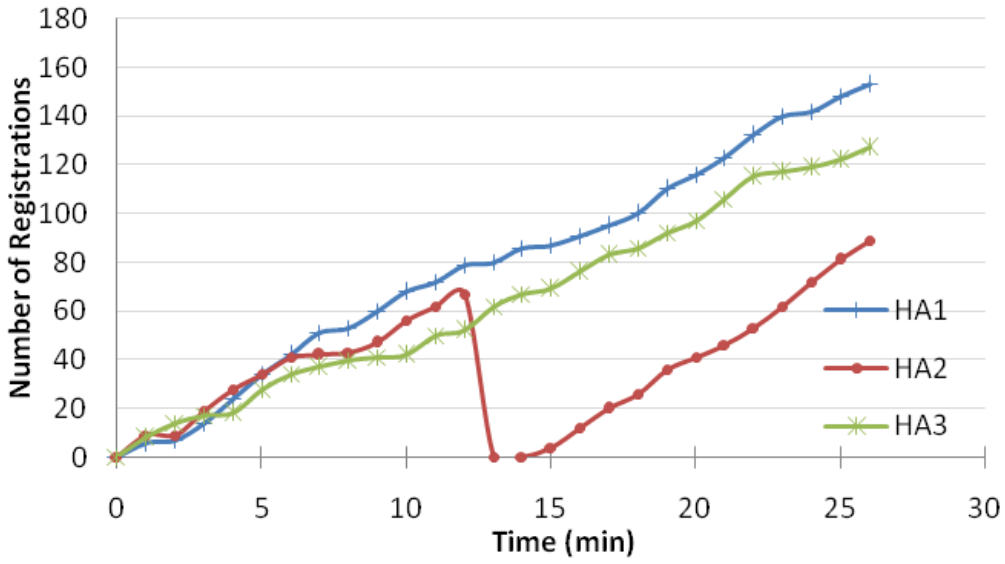

Figure 2.24: Anycast loadsharing with Single HA Failure

This plot shows the shortcomings of the load sharing scheme based on anycast addressing. After 25 minutes of simulation, there is a significant difference between the number of registrations among HA2 and HA3 and HA1. The reason for this difference in number of registrations is that in anycast load sharing there is no feedback mechanism to check the actual load status. In proposed schemes there is a feedback mechanism which checks the number of registrations and traffic throughput on each HA before assigning HA to any MN. If a similar scenario had occurred in any of the proposed schemes, the load sharing scheme would have started assigning the failed HA to all new registering MNs. Along with this, it could have forced some MNs from the non-failed HA to register with this recovered HA.

From the discussion it can be concluded that the proposed schemes provide superior HA load sharing as compared to either Deng load sharing or anycast load sharing of MIPv6 RFC.

2.6 Chapter Summary

Aim of this thesis is to make seamless mobility in IP based heterogeneous network more reliable. MIPv6 is considered the standard for mobility in IP networks. This mobility protocol has one weak point and that is its dependency on HA. Architecture of MIPv6 makes HA as single point of failure. This chapter presents an architecture that will make HA more reliable and thus will increase reliability of overall mobility protocol. Along with reliability the important issue of load sharing in case of multiple HA deployment is also addressed. Reliability is treated in terms of Mobility service availability. This “availability” can be increased by addressing two aspects, first to quickly detect an HA failure and second to quickly recover from HA failure. For the first part, a cross layer approach is used by embedding higher layer information into L2 frames. The reason for choosing L2 frames was to reduce the packet size which enables us to send it more frequently without involving routers. For second part, which is quick recovery, a scheme is devised in which there is a backup HA for every MN. In case where there is a failure of primary HA for a particular MN, its back-up HA will send a message to MN to switch its primary HA. This will reduce the overall service unavailability for a certain MN. This reliability approach will not match the performance of a scheme where there is one-to-one redundancy for every HA but that was not the aim. The aim of this research is increase HA availability without introducing any redundancies in HA deployment which will decrease overall equipment cost and will be a little greener on energy consumption.

The work presented in this chapter was considered as one of the early stage solution for IST project ENABLE but later on IETF approach [14] was modified to meet the special needs of ENABLE project.

In IETF working group Mobility EXTensions for IPv6 (MEXT), there are discussions going on about HA reliability. First draft for standardization

was supposed to be submitted in December 2008 but till date it has not been submitted. In the discussions two possible approaches for HA reliability based on the HA switching mechanism are considered. One type of HA switching is called “Soft Switch” where HA failure recovery is completely transparent to MN and the other type is called Hard Switch where the HA failure is not transparent to MN. In term of IEFT discussions, the solution presented here supports hard switch where MN is aware of HA failure on home link.

In ENABLE project, we have developed a solution based on “Soft Switch”. Details of that solution can be found is Deliverable three (D3 in [3]) of the project. The biggest problem to support soft switch is the requirement of synchronizing SAs between MN and HA across both Active and standby HA. The SAs are both time and IP dependent. Synchronizing them requires huge amount of data exchange. This very requirement makes it highly un-likely that it would ever be deployed in industry.

Chapter 3

3 HA Management

Previous chapter presents a model based approach for increasing HA Reliability by quickly detecting HA failure and by introducing quick recovery scheme. That scheme should work fine in a medium sized deployment of MIPv6, but for large scale deployment there is a need for a new framework for HA Management.

This chapter presents a state of the art framework for HA Management based on autonomic computing concept which introduces self-healing and self-optimization in an HA deployment. This self-healing and self-optimization aspect of the proposed scheme will increase the reliability of mobility Protocol which in turn will increase the overall reliability of the seamless mobility of MN in IP based seamless mobility.

At the end of this chapter this proposal is extended to support different network services like web services which means that this scheme can be generalized for any network service.

3.1 Introduction

In Next generation networks it is anticipated that everything would be IP based including telecom networks. For Mobility in IP based networks, MIPv6 is accepted as standard. It is expect that there would be a large scale deployment of MIPv6 and for this large scale deployment there is a need for a whole framework for HA management.

In the previous chapter a solution for HA reliability was presented along with a review of existing solutions but these solutions may not be suitable for large scale deployment. This chapter presents a framework for HA Management based on the concepts of Autonomic Computing. By introducing these concepts, HA system would be able to monitor itself and perform self-healing and self-optimization.

In remaining part of this chapter a review of existing management schemes used in industry for monitoring and detecting failure of a node/server is presented which is followed by the proposed framework. In the subsequent sections simulation results are presented and later on before conclusion an extension of this framework is presented which supports other services like web or SIP etc.

3.2 Related Work

A typical Sever management scheme includes three distinct functions. First is to get specific information from a server, second to detect any failures and third to report anomalies to the Management Interface. To explain existing mechanism a generic network scenario of Figure 3.1 is drawn.

Consider a typical network scenario of Figure 3.1. In this scenario there are three servers responsible for providing different services like Authentication, Email, SIP and Mobility services. There is a monitoring Node, responsible for monitoring the health of different servers. This figure would be used for introducing different schemes used in industry for server management. To authors' knowledge, there is no existing scheme for HA management. There are some solutions which address HA reliability which have been discussed in previous chapter. So the existing management schemes for difference servers/services, which are used in industry, are reviewed.

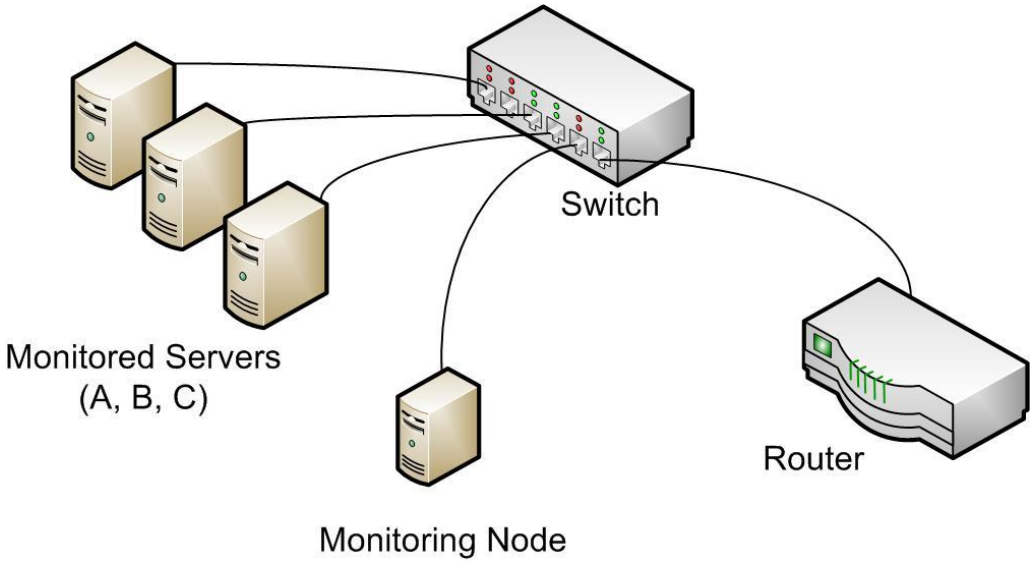

Figure 3.1: Typical Servers Deployment

There are three popular schemes used in industry for management of different services.

3.2.1 SNMP based monitoring

Simple Network Management Protocol (SNMP) [23] is a component of Internet Protocol suite as defined by IETF. SNMP is used in network management systems to monitor network-attached devices including routers, switches, firewalls and servers. It consists of set of standards for network management, including an application layer protocol, a database schema and a set of data objects. Using its standard procedures, it can deliver a very diverse set of information but this diversity is the very problem which makes it really slow. It is good for monitoring node's CPU utilization, memory and network traffic etc parameters but slow nature of this protocol makes it un-useable for monitoring networks attached

devices used in real time data. In this case, for a high availability of HA, there is a need to reduce the failure detection time. This reduction of failure detection time can be achieved by introducing a faster monitoring scheme. If SNMP is selected for HA monitoring, this will delay the overall monitoring process and hence affect the availability.

SNMP introduction:

In typical SNMP [23] based monitoring system there are a number of systems which are managed by one or more systems. A component of SNMP called agent runs on each managed system which reports information via SNMP to the managing system. SNMP agents expose management data to the managing systems as variables, such as CPU load, free memory, free diskspace etc. This protocol also allows active management tasks, such as modifying and applying a new configuration. The managing system can retrieve the information through the **GET**, **GETNEXT** and **GETBULK** protocol operations or the agent will send data without being asked using **TRAP** or **INFORM** protocol operations. Management systems can also send configuration updates or controlling requests through the **SET** protocol operation to actively manage a system. Configuration and control operations are used only when changes are needed to the network infrastructure. The monitoring operations are usually performed on a regular basis. The variables accessible via SNMP are organized in hierarchies. These hierarchies are described by Management Information Bases (MIBs).

SNMP procedures for simple network:

Here SNMP procedures are explained with reference to Figure 3.1. In this figure, there are three servers (Server A, Server B and Server C) which are the managed elements and a Monitoring Node is managing them. SNMP agent is running on Server A, Server B and Server C. Managing node,

which in this case is monitoring node can use **GET**, **GETNEXT**, **and GETBULK, TRAP** or **INFORM** primitives to collect the data from the servers. In case if **GET** primitive is used, when it is received by any of the managed element, the managed element will reply with required information. This information is organized in hierarchies called MIBs. Using these MIBs, the data is translated to Human readable form. Using this data, further control decision can be made by either network administrator or some automated script.

After explaining SNMP procedure, it is clear that this process is very slow for monitoring network attached elements. For today's real time data like videoconference or VoIP application, SNMP would be struggling to provide the monitoring data in-time.

3.2.2 WSDM-based Content Service Status Monitoring Scheme

This is web based node monitoring scheme developed in Fudan University, Shanghai [24]. Down side of this scheme is that it requires a whole new platform to work. So its implementation is not very easy. It was design to monitor health of any "Service Access Point". Data collection scheme is based on web queries and to support web queries monitored node should have a web server running on it.

Basic idea of web base monitoring is very simple. There are two nodes involved, one being the monitored node and other which is monitoring node. The monitored node hosts its data on its own web server. Each parameter available for web based monitoring has a specific URL. The monitoring node just sends a web query on specific URL to get the value of specific parameter.

This approach is very simple and fast but like SNMP based monitoring, it involves a send request and a reply request. Unlike SNMP, which can use both TCP and UDP, this web based monitoring can only use TCP. A TCP connection establishment requires three way handshakes, which make the overall process slow for frequent monitoring.

Web Based Monitoring procedures for simple network:

Here Web Based Monitoring procedures are explained in reference to Figure 3.1. There are three servers (Server A, Server B and Server C), which are being monitored by Monitoring Node. Web Service is running on Server A, Server B and Server C. Monitoring Node, can using simple web client to read the desired data from the servers. Since this scheme uses TCP for transport, a three-way handshake must be preformed to establish this required TCP connection. Once data is being read, further control decision can be made by either network administrator or some automated script.

In reference to Figure 3.1., Server A, Server B and Server C are responsible to keep the data up to date on their unique URLs. Again, how frequently this data is updated is yet another problem. In some networks transparent web cache is deployed which creates problem for this approach, as the data may be stored in these caches. There is a possibility that the monitoring node will read old data from cache. Normally administrators use this web based monitoring for one time readout.

3.2.3 Microsoft SMF

The Service Monitoring and Control [25] (SMC), a service management function (SMF) is responsible for the real-time observation and alerting of health (identifiable characteristics indicating success or failure) conditions in an IT computing environment. Down side is that it is not fast. The failure detection part of the scheme is very slow. Data collection process for monitoring purpose is not disclosed.

3.3 State of the Art: Autonomic Approach

IBM as a part of Autonomic Computing introduced the concept of self-healing back in 2003 [26]. Aim of introducing this concept was to reduce

computing systems administration cost and administration complexity. The vision of IBM was a computing system that can manage them in accordance with higher level of administration objective. Based on functions/requirements, Autonomic computing was further split into different domains including self-configuration, self-optimization, self-healing and self-protection.

3.3.1 Self-Configuration

This domain covers the configuration and integration related problems of computing system.

3.3.2 Self-Optimization

This domain covers the Operations and Maintenance aspect of computing system. An autonomic system should have the ability of running itself in peak performance all the time.

3.3.3 Self-Healing

These domains address the issue of failures in computing systems. Normally it can take up to weeks to identify the problem in a large computing system. An autonomic system should be able to identify the problem and try to isolate it.

3.3.4 Self-Protection

Autonomic system should self-protect in two senses. They should defend the system as whole against large-scale, correlated problem arising from malicious attacks or cascading problems that are not covered by self-healing function. They should also anticipate problems and take steps to avoid or mitigate them.

3.4 Architecture of Autonomic system

As mentioned in [26], an autonomic system consists of autonomic elements that are individual systems which provide services to humans. The behaviour of these autonomic elements is controlled by goals or objectives set by humans. These autonomic elements also interact with the environment.

The basic functional structure of one such autonomic element is shown in Figure 3.2. Here in this figure there is an autonomic manager and a managed element. This autonomic manager is responsible to controlling the managed element is accordance with higher level of objectives or goals.

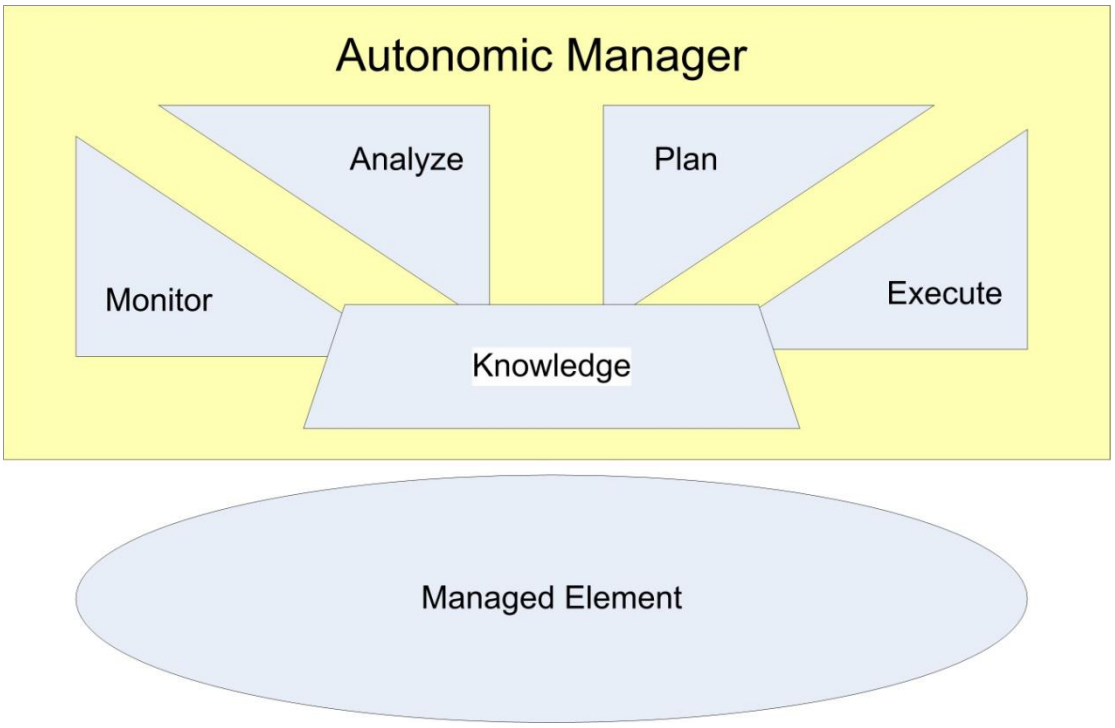

Figure 3.2: Autonomic Computing Basics

The managed element could be a utility or an application. The autonomic manager is responsible for collecting different information from sensors in

managed element. This function is labelled as Monitor in the figure. Once the data is collected it is analyzed by the Analyzer. Knowledge represents AI module, which contains a databank and is able to suggest best course of action. Plan function represents higher level objects. The Execute function is responsible for executing the suggested course of action.

An autonomic element is responsible for its own internal state, its behaviour and its interaction with environment. Its actions are governed by the goals or objective set by the administrator. There could be different levels of autonomic elements. At lower level it could be either a single disk drive or single workstation. At higher level it could be either a cluster of systems or a whole enterprise.

3.5 Analysis and Motivation

This was the vision of what an autonomic computing would look like but as with many other suggested proposals, this is not production version of such system. There are many challenges to this proposal. One such challenge is how these autonomic elements will interact with each other. Another challenge is homogenous policy definition. Another is management of these autonomic elements. Another is designing. Another is monitoring, what to monitor and how to monitor it. Another is installation and configuration of autonomic element and to make it aware of its environment. Another is resolving system wide issues ranging from hardware to software and translating them to unified language. Securing these elements is another outstanding issue. Last but not least, if all mentioned issues are resolve, the designing, testing and verification poses a challenge itself.

Main idea of autonomic computing was that all computing systems perform self-management and self-configuration but this idea remained

limited to a “Computing system”. Later on the concept of this autonomic computing was introduced to Networks.

For HA management framework, the architecture is kept in line with IBM vision of Autonomic computing. The details are discussed in next section.

3.6 Framework for HA Management

To explain the framework, it is essential to first discuss its building blocks.

- Event Monitoring
- Serious Incident Detector
- Reaction Module

3.6.1 Module: Event Monitoring

As shown in Figure 3.3, this consists of two sub-functions. One is responsible for actual RAW data delivery to a target and other for receiving the data. First module, which is responsible for sending data, can co-exist on server. It would collect data from server and will send it to the monitoring node. Second Module, which is responsible for receiving this data, resides in the Monitoring node.

Figure 3.3: Event Monitoring Module

This monitoring data packet structure is very critical. If the packet size is huge, it cannot be sent frequently. If it's small, it cannot carry enough information. To address this problem a radical approach is adopted based on cross-layer design. By embedding application layer encoded information in payload of a L2 frame this problem can be resolved. This unique method would resolve the dilemma by keeping the packet size small and still carrying enough information.

Here the actual packet structure is presented that would be used to send out the data. Please note that this frame structure is kept generic so it can be used for any application.

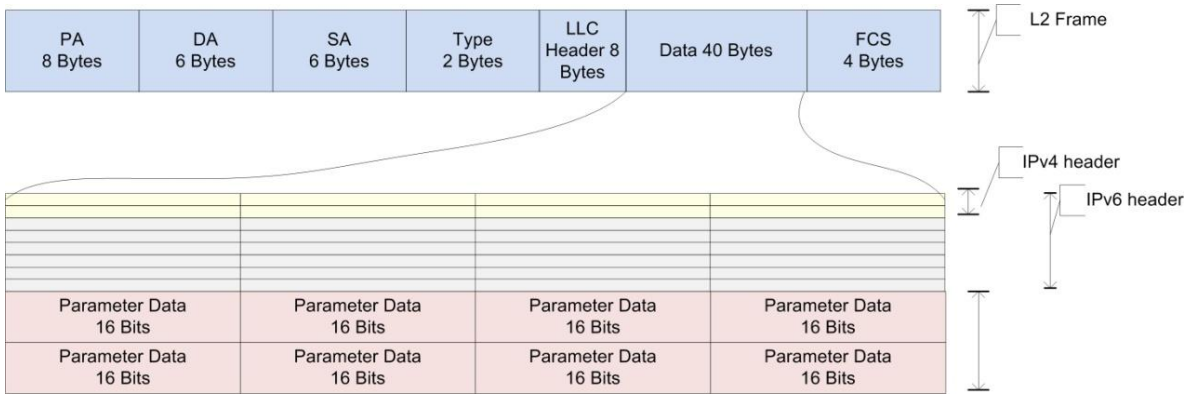

Figure 3.4: Monitoring Data Frames

This frame is engineered to have the capability of using IPv4 and IPv6 transport mechanism if required. This frame can send eight different parameters related to HA. For evaluation of the scheme, only two parameters are considered. 16 bits are assigned for each parameter. In the parameters section of packet, first 16 bits represent active registrations and next 16 bit represent the bandwidth consumption is kilobits. Other possible parameters could be CPU utilization, RAM utilization, available disk space etc.

From this packet structure it is very clear that detailed information is being sent by sensor in server, utilizing very small bandwidth. If fact it used about 74 bytes per frame. This frame designed supports up to eight different parameters in a single frame.

Apart from sending critical information, the data is used for actual failure detection. In the simulation, failure detection time was less than 45ms while utilizing network capacity by only 0.01 percent. Details of testing are mentioned in result section.

3.6.2 Module: Serious Incident Detector

This module will receive the data from receiver/collector in monitoring node. This data is in RAW form and needs classification. Data is classified based on server status code and severity of error. Once errors are classified, all the critical and fatal error on server would be forwarded to Reaction module.

Figure 3.5: Incident Detection Module

Apart from classification of errors, this module is also responsible for generating reports for administrator.

3.6.3 Reaction Module

Based on the data received from SID, it will try to diagnose/locate the problem. Once exact problem is identified, it would be forwarded to "Reaction Center". This module will first try to identify solution in its database where per-defined actions are listed by network administrator. If no solution is found, it will consult the AI module. AI module would try to generate a solution. In our case, due to lack of expertise in AI field, it has been left for future development.

Once appropriate solution is found, it would be forwarded to the "Translator" which is responsible for translation of solution to actual "command/instruction" for the server.

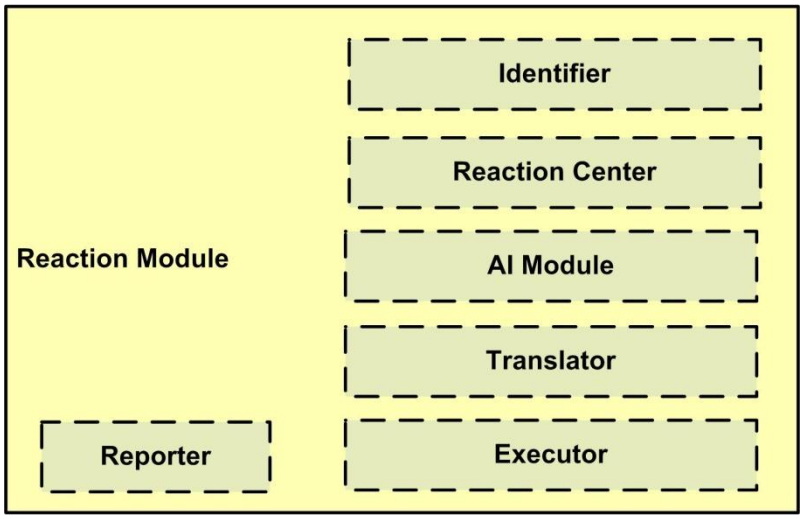

Figure 3.6: Reaction Centre

3.6.3.1 Problem/Fault Identification

This sub-module is responsible for performing five functions.

i. Locate Exact Server (its IP address or Host name)
ii. Identify Application and load it error table
iii. Look for error ID in its table
iv. Identify Problem
v. Forward this exact problem to “Reaction Center”

3.6.3.2 Reaction Centre

This sub-module is responsible for finding appropriate solution for any fault. It is a three-step process.

i. First Identify Platform and Application, for example Linux and Apache
ii. Look for possible reasons that can result in that error.
iii. Construct a solution, let it be called a “Task”, for Executer and forward it to Translator

For third step, either a pre-defined solution table is needed or some sort of AI module is required that could deduct conclusion.

3.6.3.3 AI module

This module is responsible for taking intelligent decisions based on previous events and feed data. Due of lack of expertise this AI module is not developed and is left for future development.

3.6.3.4 Translator

Function of this sub-module is quite simple. It would translate the designed task to actual command line code that can run on the server. Once translation process is complete, the actual code would be forwarded to executor.

3.6.3.5 Executor

This sub-module will receive the code from translator and execute it on actual server. It can used standard protocols line telnet or ssh for this task.

3.7 The Framework

This framework is based on the original concept of autonomic computing. In original concept, autonomic element (Figure 3.2) is the building block of an autonomic system. Autonomic element continuously monitors system behavior through “sensors” and make adjustments through “effectors”. By monitoring behavior through sensors, analyzing the data, then planning what action should be taken next and executing that action through effectors. This whole process makes a kind of “control loop” as mentioned in [27] Figure 3.7.

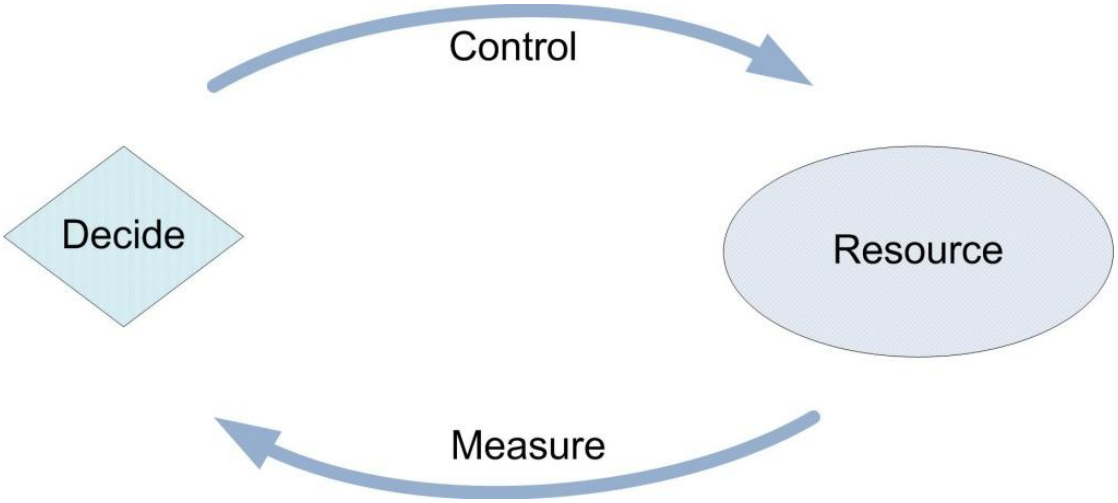

Figure 3.7: Autonomic Control Loop

In this framework a similar control loop is implemented based on three base modules (Figure 3.7) which are Monitoring module, Incident Detector and Reaction Module.

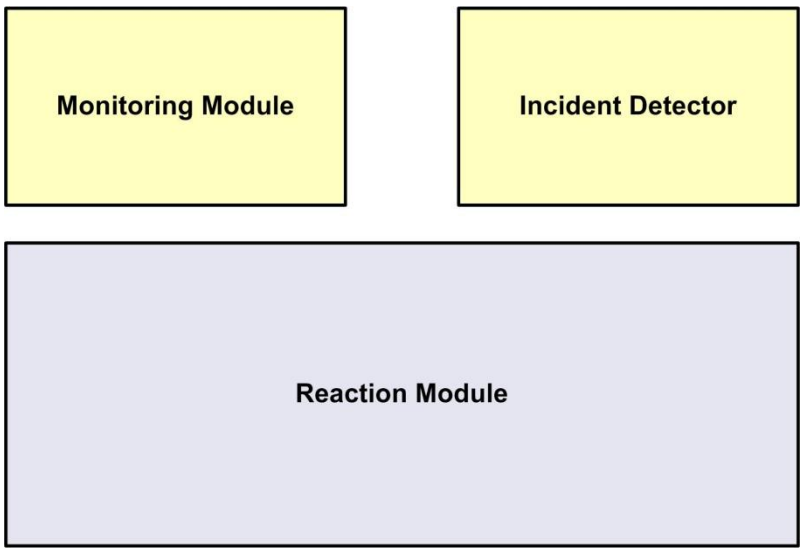

Figure 3.8: Proposed Architecture

The Monitoring module monitors the data and performs the function "Measure". The Incident Detector and reaction module combined, perform the functions "control", "Decide" and "Resource".

Functionally and structure of all these three modules has been explained in the basic building blocks. The whole framework is explained with help of a test case in next section.

3.8 Test Case

This section presents a typical scenario for deployment of proposed HA Management. Consider a situation in which there are three HA and one Monitoring node as show in Figure 3.9.

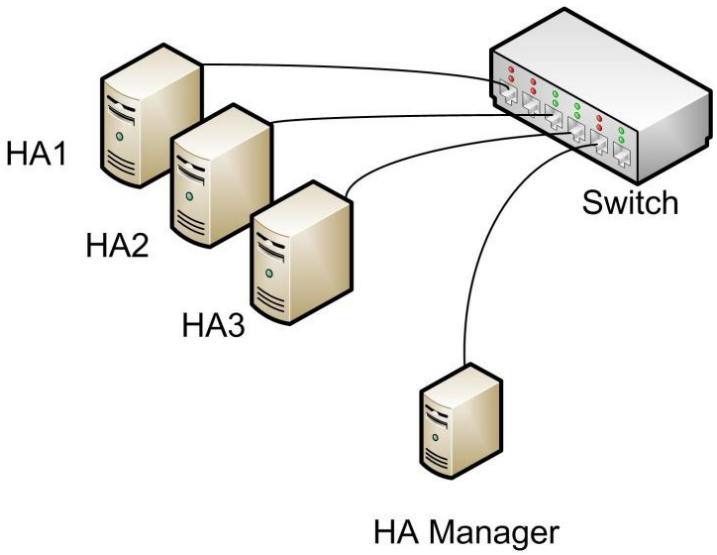

Figure 3.9: Typical HA deployment

In terms of Autonomic Computing, HA1, HA2 and HA3 are the managed elements and monitoring node is an autonomic manager as show in Figure 3.10.

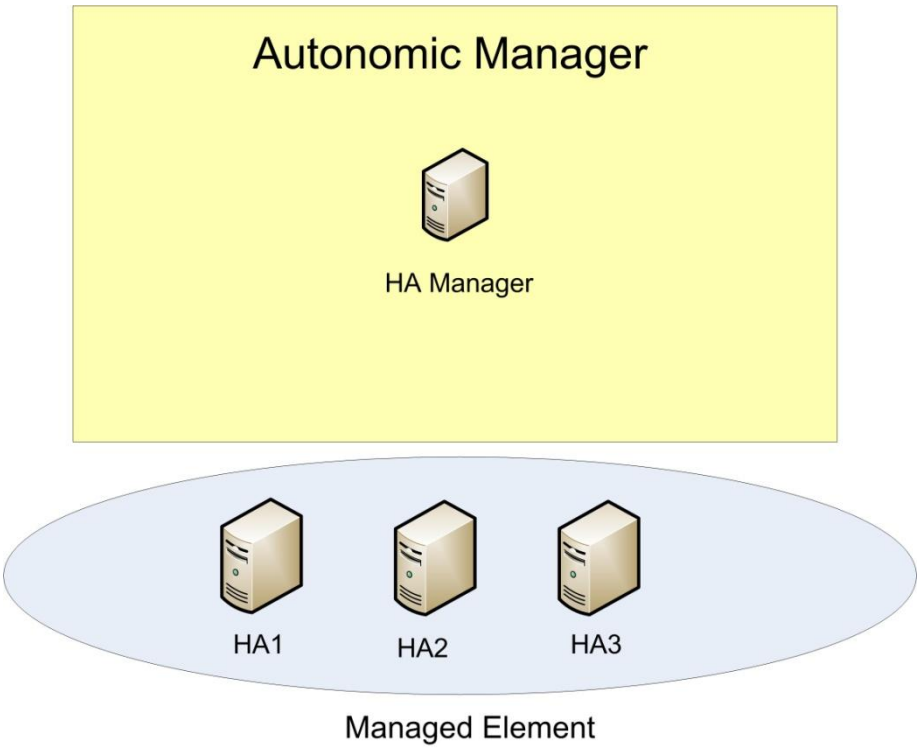

Figure 3.10: Typical HA deployment and Autonomic System

First module in our proposed scheme is Monitoring node. In this scenario, data sending part is located on each managed element that is HA1, HA2 and HA3. Data Receiver part is located in Autonomic Manager which in this case is HA Manager.

The Data sender part is sending data frames to autonomic manager in accordance with Figure 3.4. These frames are sent every 10ms containing parameters like number of registration, bandwidth consumption, CPU load, remaining RAM and disk space.

Consider a case where this framework is deployed in production environment where there are thousands of users registered for mobility

service. If HA allocation is random then after sometime, there would be un-equal number of registrations on different HAs. Let's assume that HA1 reported a high CPU load and high HA registration than its threshold. As humans, we can predict that because of this high CPU load and number of registrations, the user experience would not very good for the mobility service. Aim of developing this Autonomic Scheme was to enable this system to detect this problem and rectify this problem without allowing it to develop into some serious problems like HA failures or service outage.

Consider that frame as shown in Figure 3.11, is received by data receiver of monitoring module. Once the "data receiver" receives this frame, it is forwarded to the Incident Detector, which is the second module of this autonomic management. Incident Detector will read different parameter and if there is something alarming, it would forward packet to the Reaction Module and at the same time it would generate a report for administrator. In Incident Detector, Normal range for registered MN for this server is define as less than 2000 and CPU utilization is define as less than 90 percent. In case of this frame, there are two parameters shooting off the limits. One is CPU load and the other is number of registrations. So the Incident Detector will forward this to Reaction module.

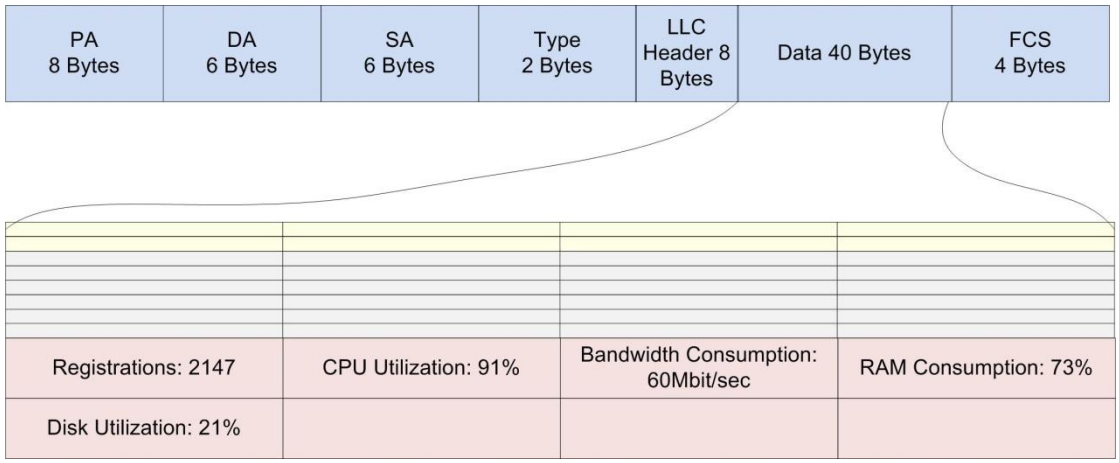

Figure 3.11: Embedded upper Layer Data

Once the packet is received by Reaction module, it will identify the exact server based on either MAC address in frame or IP address. Based on frame data, it will identify the service running on the server. Next step is to identify the problem.

In case of the frame mentioned in Figure 3.11, there are two problems that would be forwarded to the reaction center.

Reaction center would first identify the platform, which can be done in different ways. One simple approach would be to pre-configure the reaction center with the platforms. It is assumed to be detected as Linux. Next the Reaction center would try to rectify this problem with help of either pre-written routines or by consulting the AI module. As mentioned before, the AI module is not developed due to lack of expertise in the field. Consulting a pre-written routine, reaction center establish a solution that is to re-distribute some load (which in our case is mobile nodes) to other servers. This solution would be forwarded to the translator which would translate this solution to a platform dependent code which will be executed the by executers. For example code generated for a Linux based platform would be different from that of a Solaris based platform.

Once the executor has executed the code, it would instruct HA1, to re-distribute some of its load among neighboring HAs, which are HA2 and HA3. HA1 could use different methods like “HA switch” message as discussed in previous chapter to re-distribute the load.

3.9 Simulation Results

For simulation, the Event Monitoring module was tested in NS2. Other modules are software implementation and for the time being are left for future development. In NS2 simulation, the actual L2 frames with fixed length of 74 bytes were created. These frames represent the actual data frames that would be used for sending information from Data Sender to Data Receiver.

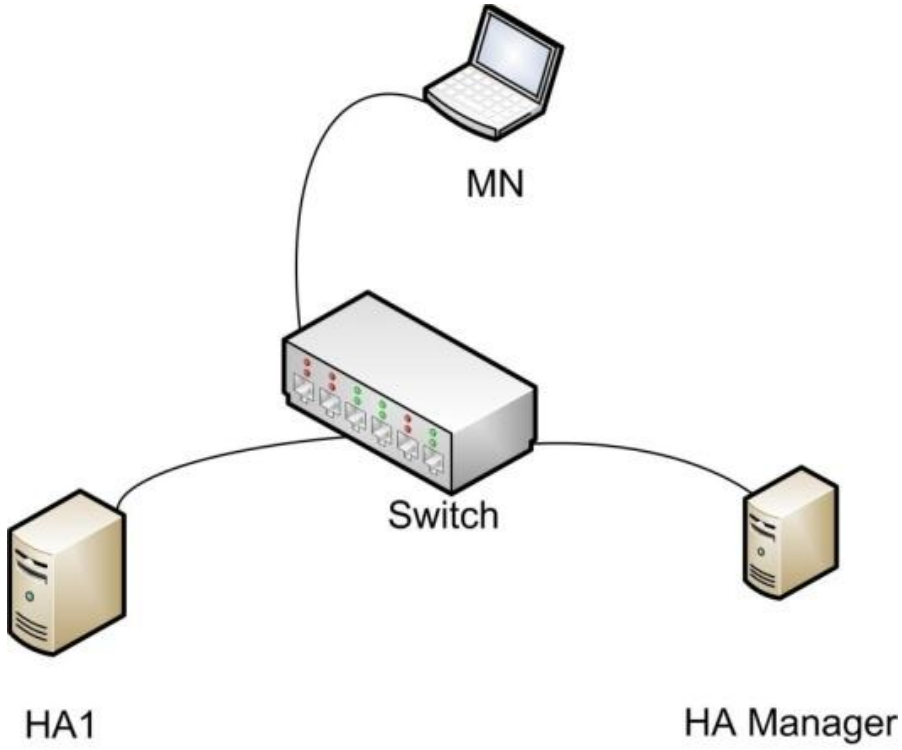

Figure 3.12: Test Scenario

A typical topology of Figure 3.12 is implemented in the simulation. HA1 was sending L2 frames to HA manager.

Following aspects were analyzed through NS2 simulation

- Bandwidth consumption of L2 frames
- Effect of these frames on application in terms of end to end delay

3.9.1 Bandwidth Utilization

First, the bandwidth consumption is analyzed. Here is the mathematical computation for theoretical calculations.

Calculations based on 802.3 MAC

MAC Preamble & SD:	8 Bytes
MAC Source Address	6 Bytes
MAC Dest. Address	6 Bytes
Length	2 Bytes
Payload	48 Bytes
CRC	4 Bytes
FRAME Size	74 Bytes
Inter Frame Gap	12 Bytes
Total Frame Size	86 Byes or 688 bits

Table 3.1: Frame Calculations

If this frame is sent every 10ms, then in one second there would be 688x100= 68.8 k bits per second and on 100Mbit/sec which is still less than ONE PERCENT Utilization and in case of Giga bit Ethernet, it would be less than 0.1 percent.

If these results are compared with the simulation results, they are not ever far from each other. The bandwidth consumption varies from 67kbit/sec to 77kbit/sec which is less than 1% of total available bandwidth.

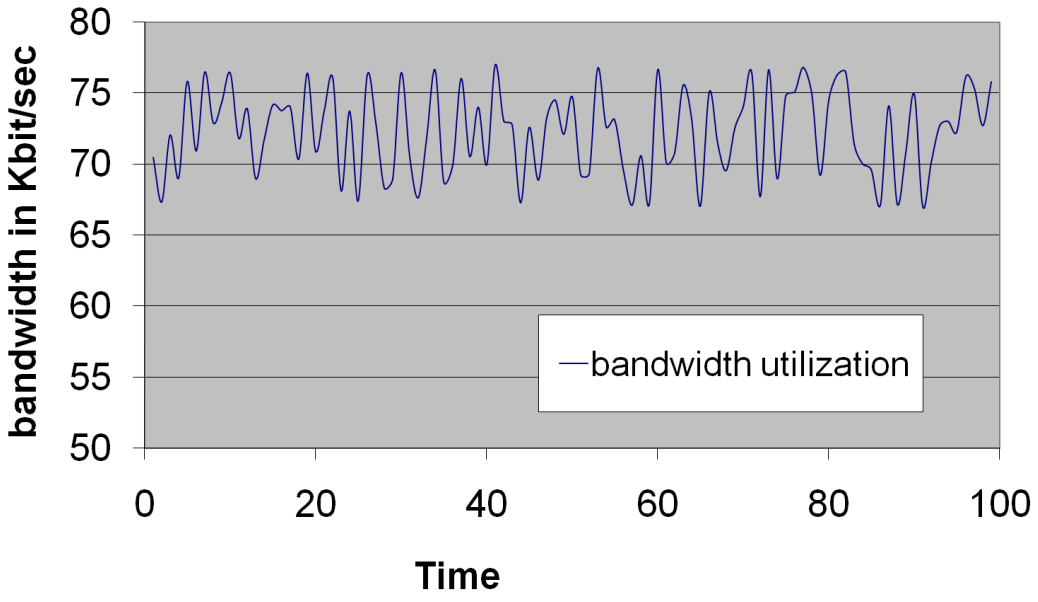

Figure 3.13: Bandwidth Utilization

3.9.2 Effect on network Applications

After analyzing the bandwidth consumption, effect on latency is analyzed. For this first calculated the end to end delay of the network without introducing L2 frames and then with L2 frames.

In the plot (Figure 3.14) black colored lines represent end to end delay without introducing L2 frames and red lines represent end to end delay after introducing L2 frames. Initially it was between 4ms - 7ms and after introducing L2 frames it was between 5.5ms - 9ms. Network latency has increased but this increase in not very significant.

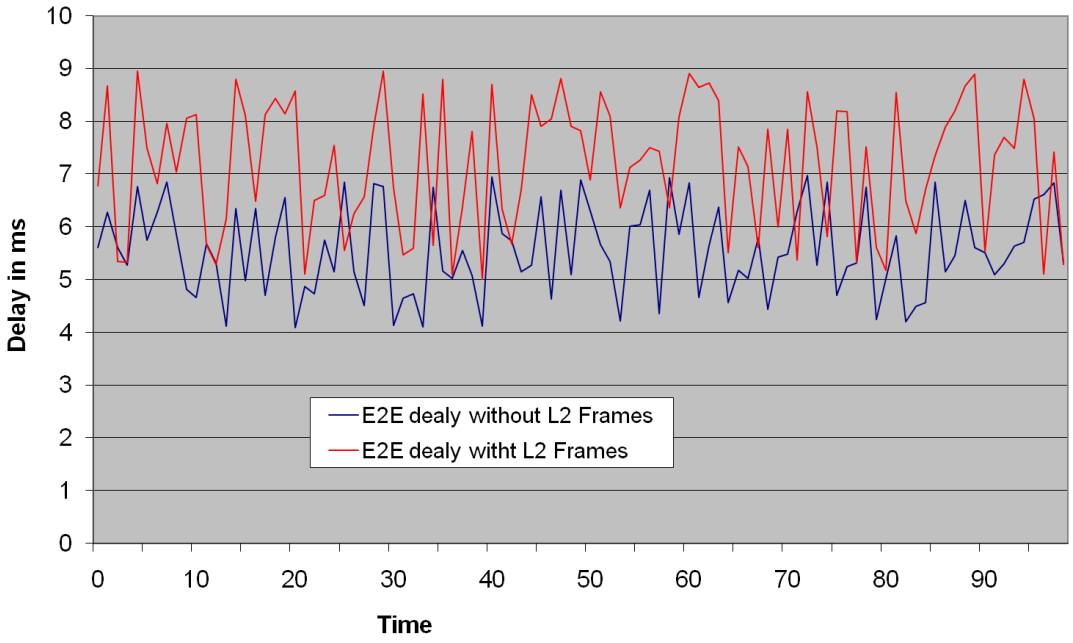

Figure 3.14: Effect on Network Applications

3.10 Extension of the framework to support generic service point

The proposed scheme can be adopted to support different type of applications. For this, modification of data which is sent to Monitoring node is required. Along with this modification Incident detector and the reaction module also requires modification. This can be achieved by adopting modular approach. In the Figure 3.15 the actual packet structures that can the used for other applications is presented.

This frame is engineered to have to capability of using IPv4 and IPv6 transport mechanism if required. In this frame, information regarding two services/applications running on a single machine can be sent. 16 bits are assigned for application identification. Standard port numbers are used for application identification. For example if service that is being monitored is a web server; 80 should be used as application ID. 8 bits are allocated

for application status, 5 of which are reserved. In fact only 3 bit status code is used for a server. which gives us a range of 0-7 in numerical. If status is 7 it means that the service is healthy and if it is Zero, it means that service is not running. 8 bits are reserved for error identification. For each application a lookup table containing up to 2^8 entries should exist in reaction center. If error number 99 is being sent but senor in server, it would be interpreted as 99th entity in lookup table for that application.

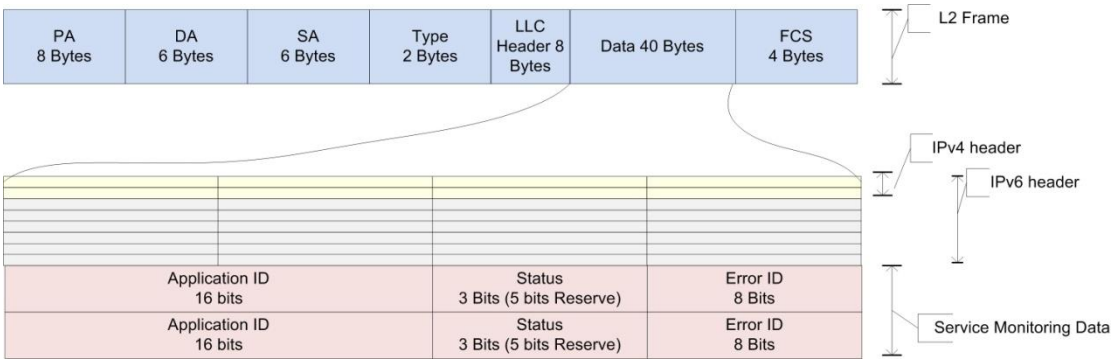

Figure 3.15: Generic Services Support

This explains the whole format. Consider an example where L2 frame carries application ID 80, status code of 4 and error ID of 99. When this frame is received by the monitoring node, it would interpret this as error number 99, which is disk space full, has occurred on web server is it is currently running with health index of 4.

From this example it is very clear that detailed information is being sent by sensor in server utilizing very small bandwidth.

If any monitored node doesn't send data for three consecutive times, it would be declared as failed.

3.11 Chapter Summary

With this architecture, self-monitoring, self-healing and self-management is introduced to a typical HA deployment. This self-management of HA system will increase the overall reliability and will try to maintain an HA deployment in optimal state. These features help in increasing the overall reliability of Mobility Protocol which is vital for seamless mobility of Heterogeneous Networks.

This chapter presents a novel approach of HA management based on autonomic computing. By adopting autonomic approach, a Home Agent system can be efficiently managed. This approach is more robust and can be extended to other application as described in the last section.

In this proposal there are three modules which are responsible for HA management. Out of three modules, one module used in this proposed architecture is simulated in NS2. Other two modules require software implementations and their “run time” depends on the server processing power and the software architecture of these modules. The critical entity for this scheme is the actual data delivery and monitoring. As mentioned before it was simulated in NS2 and the results are presented in the results section of this chapter. The results were similar to what was expected, especially the effect on latency.

Chapter 4

4 Fast and Secure Handover in FMIPv6

In next generation all-IP based networks, a large number of mobile nodes are expected to use MIPv6 protocol for mobility. In mobile environment one frequent cause of service disruption is handovers. When MN moves, at some point MN will reach the physical border of its current wireless network and will enter an area covered by another wireless network. In such a situation, MN must switch to the new wireless network. This switching is called handover. This handover process is very slow. Fast handover for MIPv6 (FMIPv6) [42] has been developed as an extension protocol to reduce the handover latency and packet loss inherent with MIPv6. The RFC [42] for FMIPv6 is categorized as "Experimental" by IETF because it is not in compliance with MIPv6 standard. One major compliance issue is that MIPv6 standard requires that all the messages exchanged between MN and HA should be secured. In FMIPv6 this is not guaranteed. In this chapter a modified FMIPv6 is presented which makes it in-line with MIPv6 standard. The major problem with adding security to FMIPv6 is that if you add additional steps to FMIPv6 protocol, it will become slow. But the main aim of designing FMIPv6 was to make the handover process fast. Hence, the challenge is to add security without adding any significant delay to the overall protocol performance. This challenge is recognised by IETF's "mishop" working group. The aim of this group is to work on designing "handover key" or HoKey. If FMIPv6 is combined with HoKey, the protocol will become compatible with MIPv6

standards. With this combination, FMIPv6 is secured without any significant delay. Thus, by reducing overall service disruption time it can be claimed that reliability of communication has increased which is in-line with the aim of this thesis. This work was part of FP6 project ENABLE, and the actual testbed was presented in projects final demonstration meeting at T.I. Labs in Turino.

In addition to the security issue, FMIPv6 also suffers from the problem of failed-handovers. FMIPv6 is based on anticipation of handovers which is not always accurate. Under certain test conditions, the success rate of correct anticipation ranges from 30-35%. In this chapter a new AP selection scheme is present which can increase this success rate up to 65%.

4.1 Introduction

In the vision of Next Generation Network, it has become mandatory for Mobile Node to constantly stay connected and seamlessly roam across different networks while enjoying the plethora of the 'all-IP' based services. A number of network layer mobility management solutions have been proposed. Amongst such candidate technologies, Mobile IPv6 has been widely accepted in the academics and industry. MIPv6 is standardized by the Internet Engineering Task Force as a viable option for delivering the required ubiquitous mobility services across an integrated heterogeneous network.

As discussed in previous chapters, there are many issues that can interrupt Mobile IPv6 based communication. In this chapter, one of the most frequent causes of disruption – the handovers is introduced. To author’s best knowledge, in IP based network there is no “Make before Break” style handover. One protocol, that comes close to making handovers seamless, is Fast Handover for Mobile IPv6 (FMIPv6).

In FMIPv6, static HA and HoA masks MN movements from CN. All packets sent to MN on it HoA are received by HA, which are then forwarded to MN current location using its CoA. However, when MN reaches the physical border of it current AP and enters an area covered by another AP, the MN must perform some standard procedure before regaining its connectivity. These procedures include scanning for suitable access point, switching access point, stateless address auto-configuration, Duplicate Address Detection (DAD), and finally updating binding on Home Agent. This sequence of procedures generally involves connection disruption and significant packet loss. FMIPv6 protocol is designed with the goal to reduce the handover latency and to assist MN in selecting a suitable Access Point. This protocol allows an AR to offer services to an MN in order to anticipate the L3 handovers. This anticipation is based on L2 triggers. The procedures of FMIPv6 are explained with help of a signaling diagram below (Figure 4.1).

FMIPv6 protocol enables an MN to request information on neighboring AP's and the subnets behind them. To do this, MN sends a "Router Solicitation for Proxy Advertisement (RtSolPr)" message. This solicitation may contain the ID of one or more APs, thus requesting subnet information corresponding to the AP. It may also contain a wild card signifying a request for all nearby ARs. In figure 4.1, when RtSolPr is received by old access router, it replies with Proxy Router Advertisement (PrRtAdv). When MN detects that a handover is required, it sends a Fast Binding Update (FBU) to its current router (pAR). This message contains MNs' CoA in pAR network (pCoA) and the access router (nAR) that MN is planning to switch to. At this point pAR sends a "Handover Initiate" (HI) containing link layer address of MN and pCoA. MN can optionally include nCoA. The nAR confirms the handover with a Handover Acknowledge (HAck) message that may provide further nAR specific details. Once the HAck is Received, pAR sends a Fast Binding Acknowledgement (FBAck) back to the MN, on pAR's link. At this stage,

MN is ready to switch AP. After MN moves to nAR, it sends a Fast Neighbor Advertisement (FNA) message and completes handover signaling. This type of handover, characterized by the FBAck received by the MN while it is in pARs networks, is called by the FMIPv6 a predictive handover. FMIPv6 protocol also defines a reactive handover scenario which basically represents the case where MN cannot predict a handover. In this case the FBU is sent from nAR's link after L2 handover is completed. It is usually encapsulated in the FNA. The nAR then forwards the FBU message to pAR, which is followed by HA/HAck messages. In FP6 project ENABLE, FMIPV6 was selected as prime protocol for handovers.

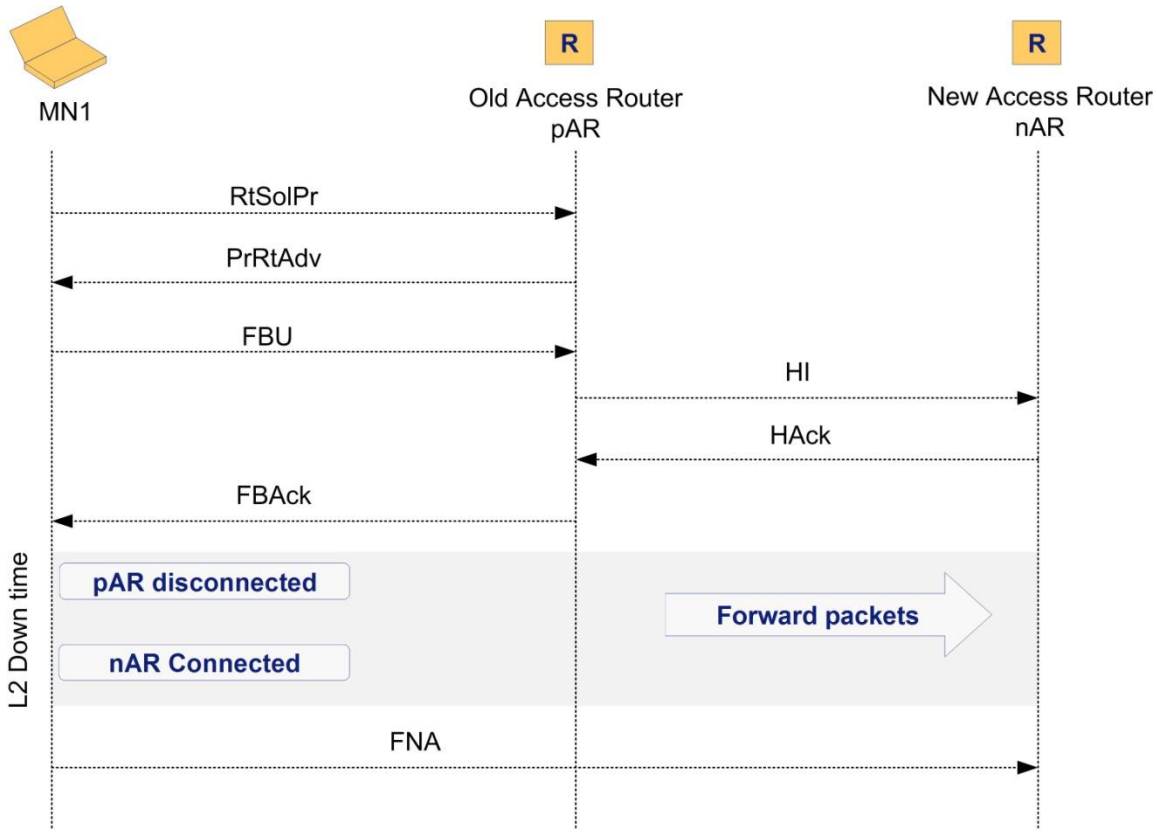

Figure 4.1: Standard FMIPv6 Signaling

A major problem in FMIPv6 is the un-secure design of the protocol. We, as partners, were assigned the task of securing FMIPv6. The details of which are discussed in Section 4.3 of this chapter. It was also a requirement to analyze if the added security will induce any delays to FMIPv6 protocol. For this analysis, a comparison with non-secure FMIPv6 implementation is required for which a review of existing FMIPv6 implementation is presented in next section.

In addition to this un-secure nature, FMIPv6 also suffers from another problem which can result in an increased number of total handovers and increased number of failed-handovers. In current FMIPv6 procedures, when MN detects a L2 trigger of “link going down”, it will go into active scanning mode. Once the scan is completed, it will select an AP with highest value of SNR. Tiebreaker for this selection is the scan sequence. If two APs have same SNR, then the one that is scanned first will be selected as next access point. This selection of AP based on highest SNR is not an optimal solution.

If MN does not select the optimal AP then there could be a considerable increase in total number of handovers during its mobility. To explain this, consider the scenario of Figure 4.2. In this Figure, there are four APs (A, B, C and D) and MN is moving from AP-A to AP-D as indicated by the doted arrow. Initially, MN is connected to AP-A. When MN moves toward AP-D, it starts scanning after crossing the “link doing down range”. After completion of scan, it will connect to AP-C because this AP will have the highest value of SNR. After some time when MN crosses the boundary of ‘link going down range’ of AP-C it will again go in scanning mode and this time it will select AP-D as its the AP with highest SNR value. At close inspection of Figure 4.2, it is observed that AP-A and AP-D have an overlapping transmission range and because of this overlap, there is no need for MN to perform a handover to AP-C. According to standard

FMIPv6 procedures, MN performs two handovers but in fact it only requires one.

A handover is classified as failed-handover when MN initiates handover with one AP and then moves to a coverage area of anther AP. In such situations, MN will not be able to benefit from FMIPv6, and its handover latency would be more than even the standard MIPv6 handovers. Through mathematical analysis later in the chapter, it has been proven that with non-optimal selection of AP, there is a considerable increase in probability of failed-handovers.

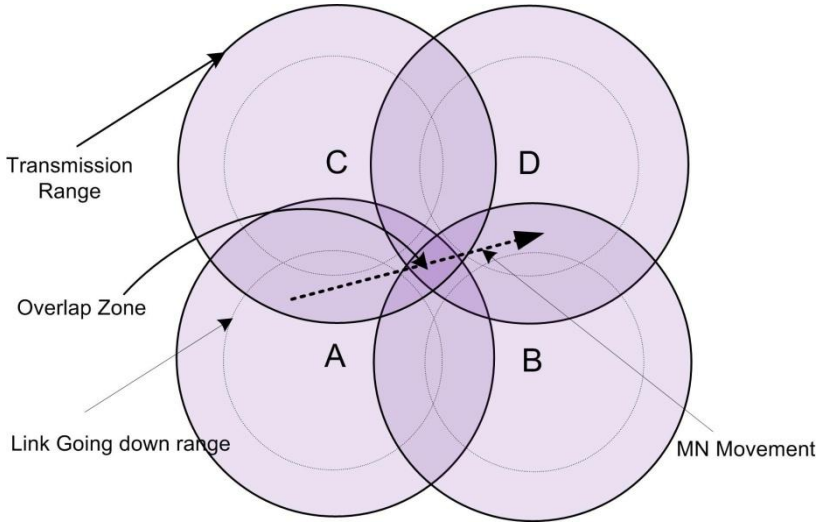

Figure 4.2: MN movement and Network Coverage

Consider a situation where MN moves from AP-A to AP-D. Assume that the trajectory that MN takes this time is through the centre of AP-A and AP-D. While following this trajectory, when MN enters the "overlap zone" and starts scanning, it will detect that all APs have the same value of SNR. At this point MN will select the AP which is scanned first. Now if MN selects AP-C for handover and actually move to AP-D then the handover will fail. Whenever MN enters an overlapping zone which is cover by two or more APs there is probability of failed-handover. Later in this chapter a novel scheme for an optimal selection of AP is proposed

which can reduce the overall number of handovers and at the same time it can reduce the probability of failed-handovers.

4.2 Literature Review

In this section some of the most recent and relevant work in the field of FMIPv6 deployment and implementation is discussed. In addition to this, some recent work related to optimal selection of AP during a handover is also presented.

In reference [43], the author is the actual developer of FMIPv6 protocol for Linux platform. In this paper, the performance evaluation of FMIPv6 over 802.11 WLAN is very comprehensive and detailed. For comparison of secure FMIPv6, this scheme was selected as benchmark. The developed testbed is almost the exact replica of what is suggested on fmipv6.org. Similar testbed was used by author for his paper. Author achieved a handover latency of 12.4ms with buffering and 42.7ms with buffering disabled on AR, in case of a predictive handover. In case of reactive handover, handover latency is about 2.45 seconds.

In reference [44], a proprietary implementation is evaluated. Experimentation involves wired handovers with emulated delay of 40ms. This delay was configured for handover. These values were chosen as they are closer to that of 3G systems. Similar to handover delays, link layer trigger are preconfigured which is again not a good approach. Triggers like link down were instant in a wired network but this is not the case for 802.11 based networks.

Another performance analysis of FMIPv6 is provided in reference [45]. Author's focus is on protocol overhead, wrong anticipation and buffer limitation on AR. They have shown that these vary largely depending on how close in terms of time the link layer trigger and actual link disruption is. They computed an optimal separation between them. They also studied

overhead which depends on application used. If you send larger packets as in the case of VoIP or Video on demand, percentage overheads would be very low as compared to a simple web browsing data.

The reference [46] also designs proprietary implementation of FMIPv6. Testbed was setup in back in 2001 when FMIPv6 RFC was not finalized, but this implementation has significant impact in IETF work. This experimental setup was a bit un-realistic and has been criticized in a number of papers. The procedures followed were also not appropriate. For example, L2 trigger were generated by command line so they were not actual triggers but just emulated signals. They also neglected candidate AP discovery which has a huge impact on overall handover latency.

All of the above implementations did not consider security issues of handover. Our goal was to make it secure. The architecture of secure FMIPv6 is discussed in section 4.3.

In reference [67], a fast handover scheme based on FMIPv6 in a vehicular environment is presented. This scheme relies on IEEE 802.21 frame work for AP selection. Unlike FMIPv6, where MN goes into scanning mode after receiving L2 trigger, the proposed scheme suggests contacting 802.21 Information Server (IS) regarding neighboring AP. IS replies with a single and most optimal AP. This is a good approach but the problem is that 802.11 is still under development. There are many pending issues like how to populate Information Sever, where it should be located, and how frequently the data on IS should be update. This means that unless 802.21 is standardized, the proposed scheme cannot be implemented. Later in this chapter a novel scheme to assist MN in selecting optimal AP is presented.

4.3 Architectural Overview of secure FMIPv6

For security of FMIPv6 based handover, in IETF discussion it was decided to secure the Fast Binding Update (FBU) message between AR and MN.

This is to protect against certain types of attacks. In ENABLE project this is exactly what was implemented. In the Figure below logical components required for securing FMIPv6 are given.

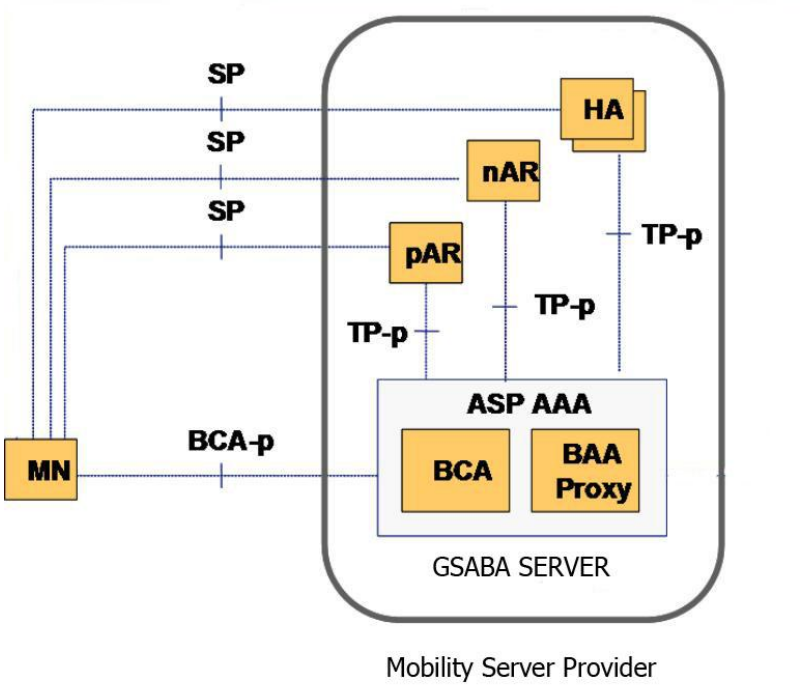

Figure 4.3: FMIPv6 Service Validation

In Enable architecture, FMIPv6 is considered as a premium service and is required to be authorized by either Mobility Service Provider (MSP) or Access Service Provider (ASP). With reference to Figure 4.3, for service authorization, either SP or BCA-p interface is used. SP interface was considered out of scope of enable project as it can be application specific. BCA-p interface was used for service authorization. It was based on Client Certificate. MN presents a certificate issued by MSA/ASA. Once it was authorized by MSA/ASA AAA server (called GSABA in ENABLE), MN was permitted to use FMIPv6 service.

After service authorization, Access Router/Access Points can authenticate MN based on Shared Keys.

4.3.1 Component Description

GSABA Server: This is responsible to facilitate authorization and authentication. Its detailed structure can found in ENABLE Project deliverable ZZ.

pAR and nAR: These are service point entitles. In our case these are FMIPv6 enabled access routers.

MN: This is FMIPv6 enabled Mobile node.

BCA: It is responsible for providing necessary bootstrapping information to the MN. For FMIPv6, this information would be the IP routable address of the pAR. In our case it was co-located with GSABA.

The BAA asserts the authorization statements. Based on MN's profile which is available in the authorizing domain, such statements are generated by the BAA. The statements and parameters need to be conveyed to the MN. In addition the BAA needs to authenticate the MN.

4.3.2 Interface Description

The Bootstrapping Target Protocol (Tp-p): This protocol is used between the BTs (i.e. pAR and nAR) and the BCA in the GSABA Proxy/GSABA Server to exchange FMIPv6 service related information and to authorize BT to provide FMIPv6 service to the MN. This information could be encoded in Attribute Value Pair (AVPs)[] or in XML. For implementation AVPs were used.

The Bootstrapping Protocol (BCA-p): This protocol is used to convey bootstrapping information to MN and inform the authorization decision taken by the BAA and BAA Proxy. The information could be encoded in

AVPs or in XML. Candidate transport protocols are EAP/PANA, DHCP, HTTP, and TLS.

The Bootstrapping Agent protocol: This is used between BAA Proxy and BAA to exchange information between BAA entities to enable them to make decisions and to deliver them to the BCA. This information could be encoded in AVPs or in XML.

The Service Related Protocol (SP): This protocol runs between MN and BT (i.e. the pAR and nAR)..

4.3.3 Message Flow

In this section, detailed message flow of the overall GSABA architecture with FMIPv6 is presented. The aim is to show how different entities interact to provide service bootstrapping. This message flow is divided into two parts. First part describes signaling involved during FMIPv6 service initiation and the second part describes signaling involved during a handover.

Assumptions made for this message flow are as follows:

- Mobility service provider is capable of providing FMIPv6 server.
- The GSABA Server is in the service authorizing domain and makes authorization decisions.
- MN has already completed bootstrapping and has successfully connected to one of the Access Point.

4.3.3.1 Messages Involved during FMIPv6 Service Authorization Phase

This section presents the message flow details of the overall GSABA architecture with FMIPv6 and aims to show how different entities interact to provide the service bootstrapping.

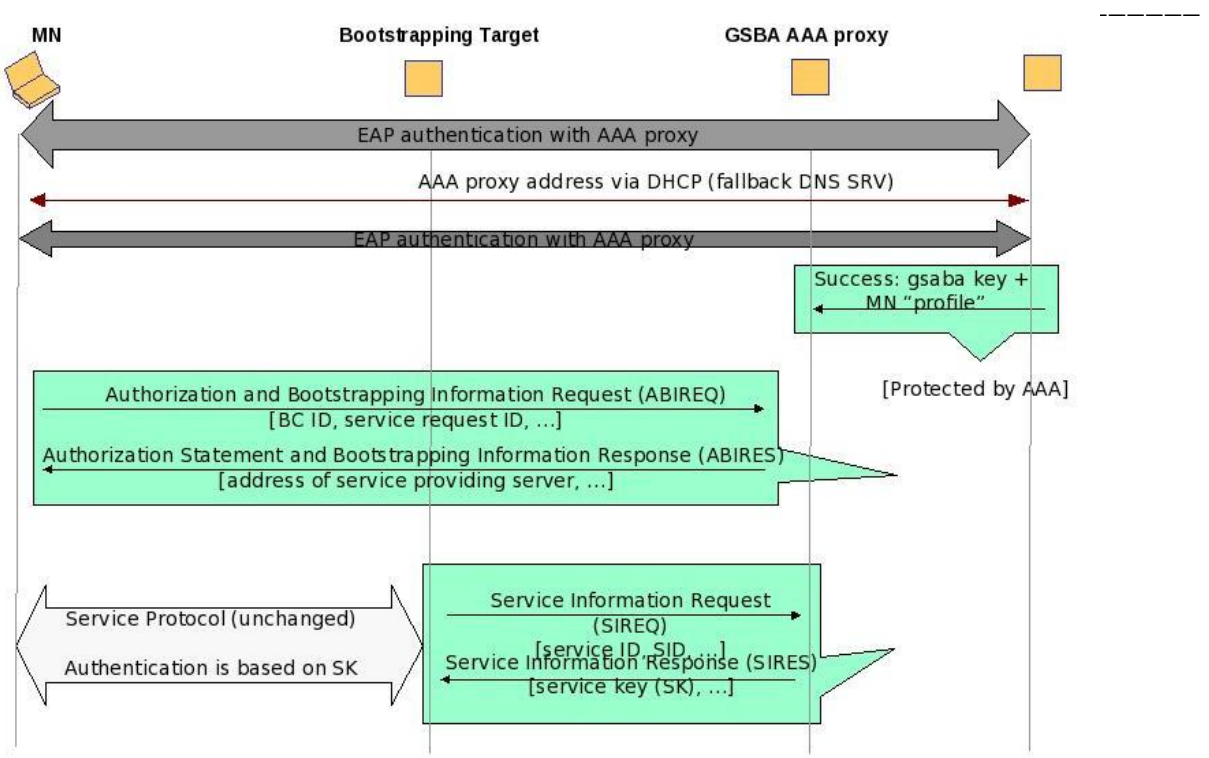

Figure 4.4: Mobility Service Authorization

GSABA Address Discovery: MN authenticates with Mobility Service Authorizer (i.e. the home domain which authorizes network access) in order to get authorization for network access. (This is done completely independent of the FMIPv6 service bootstrapping). It is assumed that the initial network access authentication uses EAP based authentication. At this stage MN knows GSABA server address.

Key Generation: During bootstrapping a new key is generated by the MN and the AAA server called GSABA key that might be derived from EMSK generated after a successful EAP method authentication (some guidelines for further key derivation by using EMSK as a root key can be found in [deliverable]). This key is used for the protection of communication between MN and the GSABA AAA server (i.e., the BCA-p interface).

BCID Generation: Additionally, a new identifier is generated by MN and GSABA AAA proxy with which the GSABA key (and therefore indirectly MN) is identified (called BCID for "Bootstrapping Client IDentifier").

ABIREQ Message: Services requests are done via the ABIREQ (Authorization and Bootstrapping Information REQuest) message. The minimum set of parameters included in the ABIREQ are the BCID, the identifier of the service that the MN wants to use (Service Request ID – SRID), and the corresponding identifier intended to be used on the SP interface (Service ID – SID). The SRID could either be a service name alone or, additionally a bootstrapping target address if known by the MN.

ABIRES Message: Upon receiving the ABIREQ message, the GSABA Proxy sends an ABIRES message with the authorization decision and minimum information conveyed to the MN about the service points (e.g. the IP address or FQDN of pAR). Additionally, the ABIRES can contain a key and an identifier (SID) to be used for accessing the service.

HK Key Derivation: With the information provided in the ABIRES, FMIPv6 service is bootstrapped as usual. For securing FMIPv6 signaling messages, a handover key (HK) is derived from the GSABA key shared by MN and GSABA proxy. Using the HOKEY [*] process, discussed in detail in the Next section.

4.3.3.2 Message Involved during FMIPv6 Handover

In this section the actual message flow for FMIPv6 handover is described. The flow is presented in Figure 4.5.

Step 1: Detection of new AR: MN performs periodic wireless scans. With the help of L2 trigger, MN starts looking for a candidate access router.

Step 2: RtSolPr: At this point MN sends a RtSolPr to pAR to resolve one or more Access Point Identifiers for subnet-specific information

Step 3: PrRtAdv: In reply to PrRtAdv, pAR sends PrRtAdv message to MN. This message contains subnet-specific information.

Step 4: nCoA Creation: From Subnet specific information about neighboring APs, MN creates nCoA.

Step 5: HKReq: Using this nCoA MN generates HKReq Message. This message is sent to nAR1. Aim of this message is to set up security association with candidate access router. In Enable project, it was decided that MN should establish Security Association with all available candidate access router before the actual switch.

Step 6: nCoA Check: After HKreq is received by nAR1, it should check if this newly created address (nCoA) can be used. If yes, then it proceeds to next step.

Step 7: SIReq: After having received HKReq and nCoA validation, it forwards this HKReq message to GSABA using SIReq message.

Step 8-9: HK and SIRsp: GSABA checks the MAC contained in HKReq from MN and if successful, derives a new handover key for new candidate access router and returns SIRsp message including the result code and new handover key as well as AAA nonce. This SIRsp message is sent to nAR1.

Step 10-12: HKRsp: Handover key is contained in SIRsp message. This key would be used to authenticate MN during a fast handover. Once this key is known, nAR sends the HKRes to MN.

Step 13: Selection of nAR2: By now MN has established security association with all the candidate routers and is ready to switch to any of the available candidate access routers. Final selection of candidate Access router is based on L2 trigger.

Step 15: HI Message: This is a standard FMIPv6 message, indicating that MN is about to initiate handover. This message is send to nAR by pAR.

Step 16: HAck message: If nAR is ready to accept connection from MN, it will acknowledge the HI message with HAck message.

Step 14: FBU: Once the candidate access router is selected, MN sends FBU message to pAR. These messages are already secured by previous handover key (pHK)

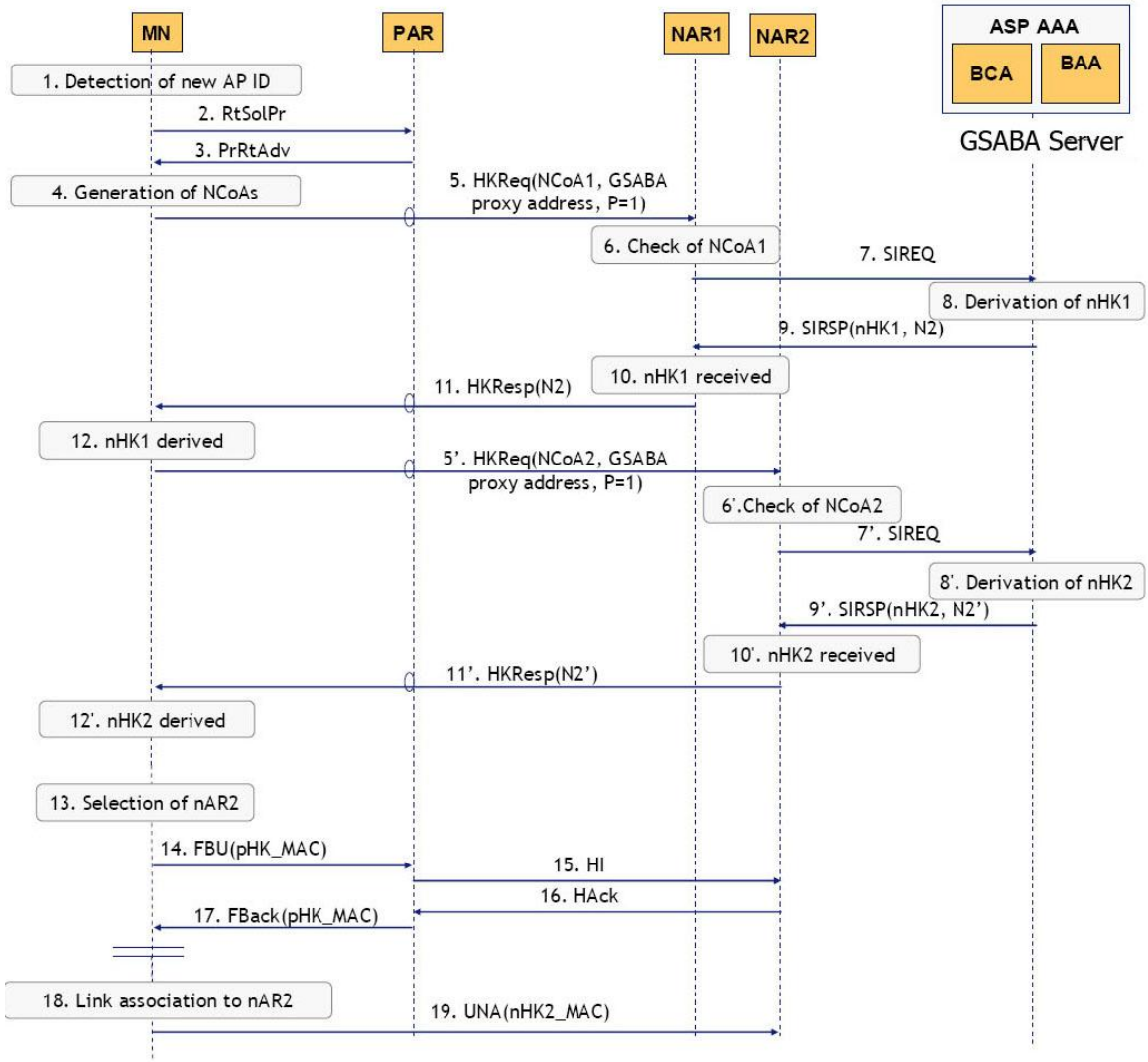

Figure 4.5:Message flow for FMIPv6 service (predictive mode)

Step 17: FBAck message: Once the pAR has received acknowledgment of HI message, it sends FBAck to MN. Again, this message is secured by pHK.

Step 18: L2 Switch: At this stage, MN performs actual L2 switch by changing its active access point to nAR.

Step 19: Once this step is complete, FNA message is sent to to nAR. This message is protected by handover key that was created in step 12.

After this step, handover is complete.

4.3.4 Securing FMIPv6 Signaling with HOKEY

In HOKEY, handover master key (HMK) is pre-shared between MN and AAA and should not be used to protect any data. Handover key (HK) and handover integrity key (HIK) are derived from HMK. HK is used to secure the messages exchanged between MN and AR while HIK is used to secure the signaling between the MN and AAA.

In GSABA, GSABA key is used as the HMK to derive the HK as well as the HIK. The HIK is used to protect the data between MN and GSABA proxy (i.e. ABIRES/ABIREQ message) via the BCA-p interface and therefore is in fact the BCA Key. The HK can be derived by:

HK = gprf+ (HMK, MN Nonce | AAA Nonce | MN ID | AR ID | “Handover Key”)

Or,

HK’ = gprf+ (HMK, MN Nonce | MN ID | AR ID | “Handover Key”)

The gprf+ is used to derive the HK (where “|” indicates concatenation), which is an ASCII string with 12-characters and no null termination. The MN nonce is generated by the MN and communicated to the AAA server in the HKReq message. The AAA nonce is generated by AAA server and

sent to the MN in HKRsp message. The MN ID is the NAI of MN and the AR ID is the IP address of AR as seen by the MN.

HK can be derived after MN attaches to an AR and HK can be generated during handover procedure. Thus, the probability for handover error induced by security mechanism will be reduced as it is possible for MN to leave its old link without FBU message.

There are many advantages for setting up a SA between the MN and the AR before the actual fast handover process:

- It allows using the AAA infrastructure that is already in place to establish session keys for securing FMIPv6 signaling messages.
- The handover key derivation does not impact handoff latency.
- The compromise of one AR or a particular handover key does not lead to the compromise of keys shared between the MN and any other AR.

After service authorization, when the MN wants to set up SA with AR, it sends a Handover Key request message (or HKREQ) to AR together with the GSABA Proxy's address (Step 5). The AR should send a SIREQ message to GSABA Proxy specified by the GSABA Proxy's address (Step 6). Upon receiving the SIREQ message from an AR, GSABA Proxy should check the user profile to confirm the authorization state and generate a new Handover key, and then return the handover key to the AR (Step 7). After AR receives MN's authorization state and handover key, it may return a Handover Key Response message (or HKRES) to the MN (Step 8). The messages HKREQ and HKRES are specified in [HOKEY].

While it is attached to the pAR, the MN may detect a new AP ID. MN sends RtSolPr message to pAR to check whether the newly detected AP is attached to a new AR (Step 10). When pAR finds out it is a new AR's AP,

pAR should return PrRtAdv message to the MN (Step 11). These two messages don't have to be secured.

The MN sends FBU message to pAR when it is willing to handover to nAR with FMIP service (Step 12). This message includes an authentication code (or MAC) generated using the previous handover key. The MAC may be transported re-using the Authentication Option specified in [RFC 4285] or defining a new FBU mobility header's mobility option. The MN may insert in the FBU the new HKREQ message to be delivered to the nAR for establishing the new handover key with it. After successful authorization/validation, the pAR sends the Handover Initiate (HI) message to the nAR [RFC 4068] including the new HKREQ mobility header addressed to nAR and the GSABA Proxy's address (Step 13). The nAR sends SIREQ to GSABA Proxy (Step 14), which requests the authorization state and the new handover key from GSABA Proxy. Subsequently, GSABA Proxy delivers the authorization statement and new handover key to nAR.

The nAR returns HAck message to pAR with the HKRES mobility header for MN and the new CoA which will be used by MN in the nAR domain. The tunnel between nAR and pAR for traffic forwarding is established (Step 16). The pAR sends FBAck message to MN and nAR. The FBAck message delivered to the MN has to include MAC and the new HKRES mobility header from nAR (Step 17). When MN attaches with the nAR, MN sends FNA message to nAR, which should be integrity secured by the new handover key established between MN and nAR.

4.3.5 AAA Integration

It must be noted that almost all telecommunication and Internet Service Providers (ISPs) make use of the Authentication, Authorization and Accounting Server (AAA server). To support roaming between accesses domains, AAA broker services have been deployed to accomplish peering

of various providers. Such agreements symbolize business relations and have an impact on AAA message routing. Due to this reason, our architecture leverages and integrates with the existing AAA infrastructure to reduce operational and deployment costs.

The GSABA Proxy is in essence an AAA server which consists of the BCA and BAA Proxy co-located inside it. The GSABA proxy will be located in the service providing domain (i.e. the network that actually provides the service). The GSABA proxy acquires the authorization statements for the FMIPv6 service from the GSABA via BA-p interface using AAA Proxy functionalities (which is same as any ordinary AAA server without any changes/modifications).

The GSABA server contains BAA component and is responsible for making the ultimate decision of authorizing the requested service (i.e. FMIPv6 in this case). The GSABA service will reside in the service authorizing domain (which is typically in the MN's home domain).

ARs :(i.e. The BT) as mentioned in previous section, AR which is the pAR and nAR in our case provides the FMIPv6 to MN via the SP interface. In addition, BCA component is connected via TP-p to obtain configuration and authorization information of FMIPv6 related to a specific MN.

4.4 Experimental Test-Bed Evaluation

The testbed built for secure FMIPv6 is shown in Figure 4.6. There are two access routers (AR), a mobile node (MN), a corresponding node (CN), Home Agent (HA) and Authentication and Authorization server.

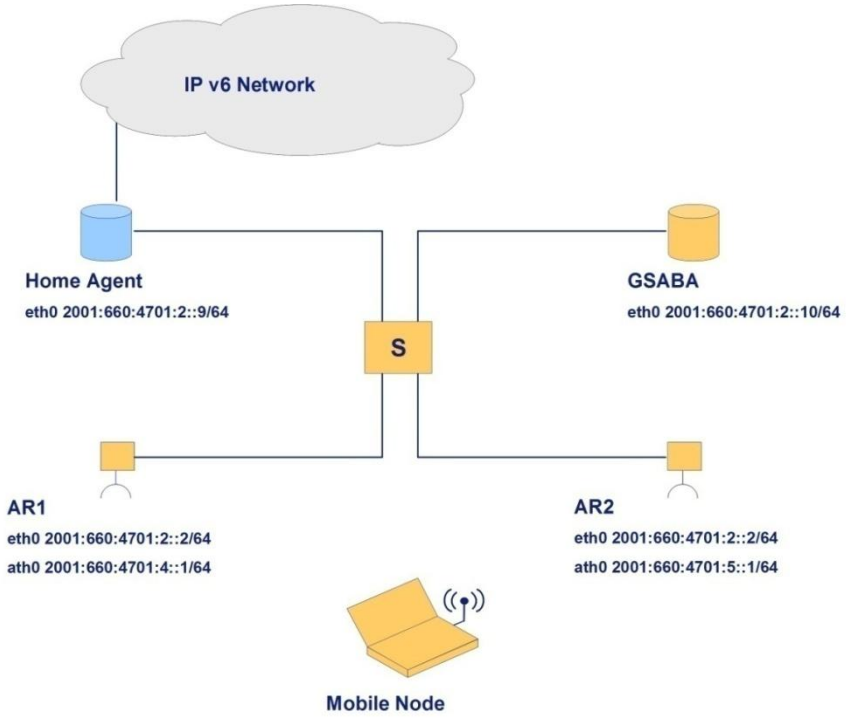

Figure 4.6: FMIPv6 Testbed

4.4.1 Detailed Description of Nodes

AR: ARs were based on highly optimized Linux based machines with Atheros chipset. Latest Linux kernel 2.6.23-rc3 available at that time was used. For Mobile IP, the recently developed Universal Mobile IP (UMIP) package was used. For routing, Quagga Routing Suite is used. The actual routing protocol used was RIPv3. Access routers were running the modified implementation of fmip-ar daemon developed by fmipv6.org. For route advertisement, Radvd was used.

MN: MN was Linux based machine with Atheros chipset. Again Linux kernel 2.6.23-rc3 was used. MN was also running UMIP and our modified implementation of fmip-mn daemon.

Home Agent: HA was Linux based machine with one Ethernet card. As with other nodes, kernel version 2.6.23-rc3 was used with UMIP. For route advertisement, Radvd was used. For routing, Quagga Routing Suite is used. The actual routing protocol used was RIPv3.

GSABA/Authentication Server: Authentication / Authorization server was again Linux machine. Softwares running on this machine were Apache (web server) with SSL support, modified FreeRadius server and Quagga routing daemon.

In Table 4.1 the exact specification of the each node used in testbed is given.

Network Entity	Hardware	OS	WLAN card
HA	Intel Core 2 Duo desktop	Linux kernel 2.6.23-rc3	N/A
pAR	Sony VAIO Laptop	Linux kernel 2.6.23-rc3	Atheros chipset
nAR	Siemens Laptop	Linux kernel 2.6.23-rc3	Atheros chipset
MN	IBM T43p Laptop	Linux kernel 2.6.23-rc3	Atheros chipset
CN	Intel Core 2 Duo desktop	Linux kernel 2.6.23-rc3	N/A
GSABA	Intel Core 2 Duo desktop	Linux kernel 2.6.23-rc3	N/A

Table 4.1: Testbed details

4.4.2 Test Description

This section presents the objectives of all the tests performed, their expected results, and comparison of actual and expected results.

The main objective of this testbed was to show that by adding security to FMIPv6, there would not be any substantial increase in handover delay as compared to standard FMIPv6 protocol. Results are presented. Exact time

measurement were based on Timestamps displayed on linux console provided by FMIPv6 suite when it was running in debug mode along with a packet sniffing tool, Wireshark.

4.4.3 Test Scenario

There are two scenarios in which a mobile node performs handover. Either it's anticipated which is called predicative or it's unanticipated which is called reactive handover.

4.4.4 Reactive Handovers

MN performs a Reactive handover when MN is not able to predict handover because of sudden drop/loss of signal from its attached AR. In such handover, a long service disruption time is expected because MN would perform scanning and after completion of scanning it would try to attach to one of the available access points. The scanning process takes about 1.3-5 seconds depending on available channels and access points. For all the reactive handover testing, mobile node was forced to change its point of attachment from one AR to another by shutting down the wireless interfaces of pAR.

4.4.5 Predictive Handovers

A predictive handover is performed when MN is able to anticipate a handover. This prediction is triggered when signal strength of an associated link drops below a certain threshold. In such a situation, MN moves to AR with highest signal strength from the last scan. For all predictive handover testing, MN was attached to one of the access router and then the Tx power of this access router was reduced, forcing it to move to nAR.

In both test scenarios, MN streams video from CN based on RTP protocol. RTP transport protocol was used to avoid the TCP retransmission that could affect the actual results. Buffering was disabled on Video client to

be able to perceive the handovers in the actual streaming. Data was collected on MN by WireShark. Timestamps were generated by FMIPv6 daemon.

4.4.6 Reactive Handover Test

In the reactive handover testing MN was connected to one of the AR and then while MN was streaming video, the wireless interface of that AR was switched off. Link failure was detected by MADWiFi drivers through missing beacons. Default value set in drivers was 7 beacons. In 802.11 the beacon interval is set to 100ms. So the device drivers detected the link loss after 700ms. The fmip daemon detected the link lost after another 350 ms and then it initiated the L2 handover. If fmip daemon had already completed the scan, it would try to connect to AR with highest signal strength. If there was no scan done before the link loss, the daemon would start a scan and would make to list of candidate access router. After establishing L2 connectivity, MN sent an FNA to nAR followed by FBU to pAR.

The experiment was repeated many time but the results were not consistent. Main reason for this lack of consistency in results is that the latest Linux kernel 2.6.23-rc3 is not capable of performing optimal L2 handover. Sometimes it very fast and sometimes it is not that fast. In future kernel releases, it has been noted that this problem will be resolved. As shown in Figure 4.7, it takes roughly 3-4 seconds to complete a reactive handover. This Figure represents a typical reactive handover. From the Figure it can be seen that time from when the interface of pAR was turned off to the time when actual sending of BU from MN to HA varies between 2.5-4 seconds. In this figure time between when pAR interface was turn off to the time when link loss was declared MADWiFi driver was about 700ms. This link lost was detected by FMIPv6 daemon in about 350ms. Once this was detected by FMIPv6 daemon it starts scanning. This scanning time depends on a number of factors which are

discussed later in this chapter. Finally when L2 link is up, BU is sent to HA. This time varies from 1ms-9ms. Table 4.2 gives the testbed results.

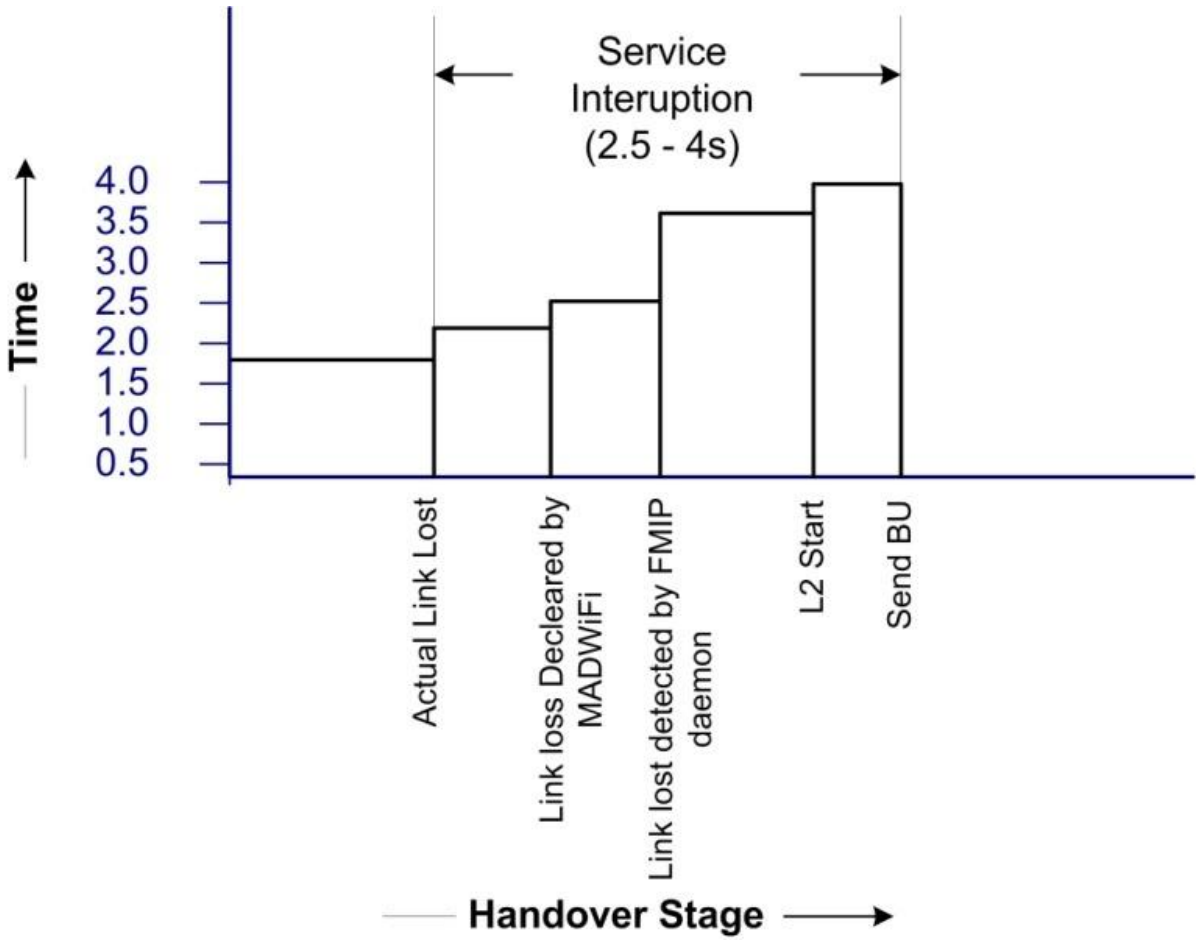

Figure 4.7: Handover Delay for Reactive Mode

This experiment was repeated multiple times and the timestamps are listed in Table 4.2. In this table average L2 scan time is 2.7 second. This is because all the available channels of 802.11a/b/g card were scanned. If all the channels of 802.11b were scanned, this scan time would be around 400ms. This could significantly reduce the handover latency.

Latency (ms)	RUN 1	2	3	4	5	Average
Link Loss Declaration by MadWiFi	785	734	789	738	721	753.4
Link Loss Detection by FMIPv6	351	362	373	341	336	352.6
L2 Scan time	2531	2621	1810	3124	3421	2701.4
BU sent to HA	2	4	1	9	8	4.8
Total Latency	3669	3721	2973	4212	4486	3812.2

Table 4.2: Testbed Results

4.4.7 Predictive Handover Test

The predictive handover of FMIPv6 is the most desirable part of the protocol. Handovers performed in this mode are almost seamless with minimum packet loss and service disruption time. Video streaming application was tested and there was just a glitch in the stream. In predictive mode, MN performed regular scans and whenever it detected that the SNR threshold has been reached it performed a handover. First MN sent HKreq to pAR. On receiving this HKreq, pAR sent SIReq message to Authentication Server which replied with SIres. Once this message was received by pAR it sent an HKRes message to MN. Using Ethernet link between two ARs, MN sent HKreq to nAR. On receiving this message, nAR sent SIReq to Authentication Server which was

acknowledged with SIres. Once this SIres message was received by nAR, it sent HKRes message to MN using the Ethernet link between nAR and pAR. While MN is connected to pAR it sent an FBU message to pAR. Next, pAR sent HI message to nAR. In response to HI message HAck message was sent to pAR and this was acknowledged with FBAck message by pAR. At this point MN actually established the L2 connectivity with nAR which takes about 16ms. The final message sent by MN is FNA to nAR.

Figure 4.8 shows the timers associated with Predictive handover. It takes about 16ms for a handover to complete. In the Figure, MN either has two interfaces, one for scanning and one for actual communication or fmip daemon has already performed the scanning.

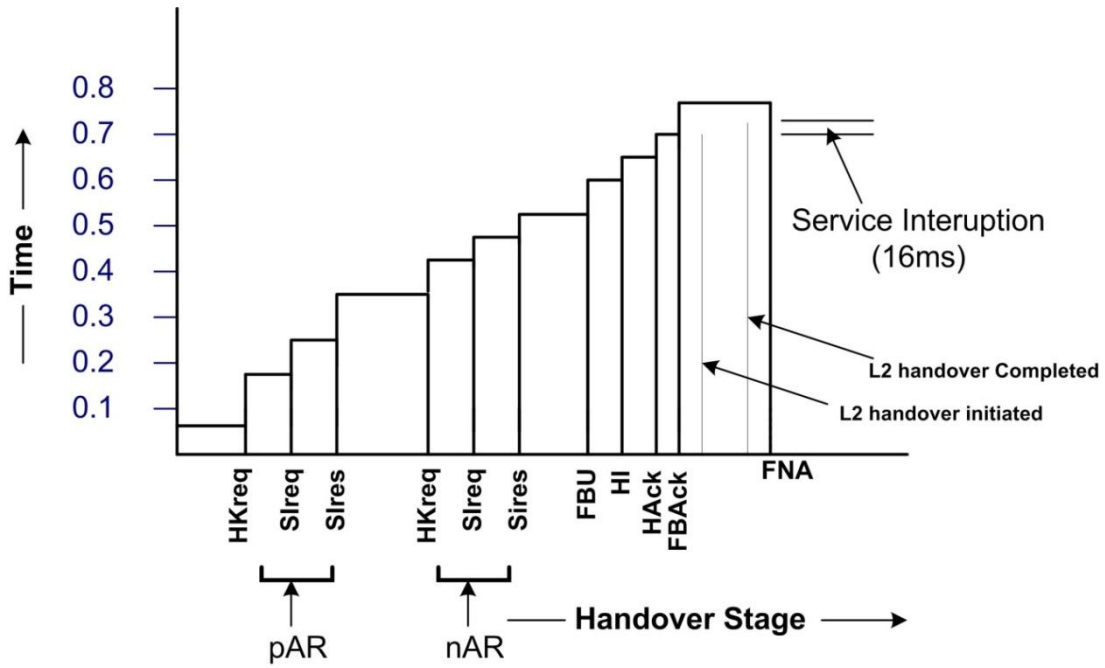

Figure 4.8: Handover Delay for Predictive Mode

In situations where MN does not have dual interface or if it has not already performed scanning then there would be an added scanning time which would be about 1.3 second.

4.4.8 Impact of Scanning on Handover

FMIPv6 does not provide a way for MN to discover candidate access router. By sending an RtSolPr to pAR, MN only requests the MAC addresses of Candidate access router. It doesn't tell MN if it is in the range of any of the Candidate access route. The only way for MN to discover a suitable AR is to perform periodic scanning.

During the tests it is observed that the scanning process takes about 1.5-3.5 seconds depending on the type of wireless card used and the scanning procedures employed. Two different types of cards were tested. One was capable of 802.11a/b/g and other was capable of 802.11b/g. For the first one, scanning process would normally take up 15.2 seconds and the other one would take about 2.7 seconds. After modifying MadWiFi drivers and setting the MaxChannelTime, scan time was decreased to 4.7s and 1.3 seconds respectively.

In case of reactive handovers, if MN has not previously performed scanning, after receiving SIres from pAR, the wireless card would go in scanning phase. During scanning, there is a significant packet loss in case of heavy traffic. Video streaming was used for testing and whenever MN would perform a scan, there was either a pause or glitch in the video. One way to resolve this issue is to have two wireless cards in MN. One for scanning and the other one for actual data communication. However, this is not an optimal solution.

4.4.9 Comparison of FMIPv6 with Secure FMIPv6

In reference [42], authors have achieved a handover time of 10.42 ms for predictive mode. On our testbed the standard time for same scenario was 12.2ms. Reason for this stems from the fact that author modified wireless

card's firmware which was not done for this testbed. Our secure FMIPv6 predictive handover had a latency of about 14ms. The delay was mainly because all the messages exchanged were secure. Secure message creation and its validation at the other end requires some computations which takes time. Overall, the overhead for added security is 2ms. If the firmware was modified, these delays may drop to 12.22ms.

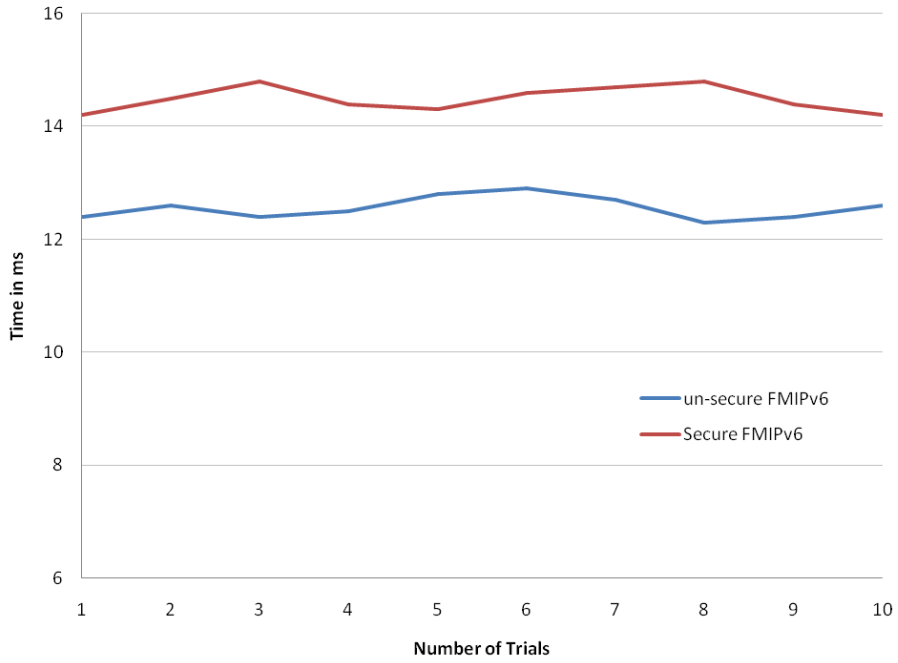

Figure 4.9: Secure vs un-secure FMIPv6

In Figure 4.9, a comparison of secure and un-secure FMIPv6 is presented. The data for un-secure FMIPv6 timings was collected from testbed with the added security. The data for secure FMIPv6 was collected from the testbed developed at Brunel University. In this chart, there is difference of about 2ms between the secure and un-secure FMIPv6. With overhead of just 2ms, it has been proved that FMIPv6 can be secured without adding any significant delay. Even for most demanding applications, this 2ms delay would not be a problem at all.

4.5 Improved Handover Anticipation

A handover is considered a failed-handover when MN initiates handover with one AP and moves into the coverage area of another AP. In such situations, MN will not be able to benefit from FMIPv6, and its handover latency would be more than even the standard MIPv6 handovers. Later in this section it has been proven with the help of mathematical analysis that non-optimal selection of AP results in significantly high probability of failed-handovers. Consider the situation in Figure 4.2 where MN moves from AP-A to AP-D. During this movement MN may enter an area called "overlap zone". In this zone, SNR values for all APs will be almost equal and MN can select any of the AP to handover to. At this point, if MN selects AP-C for handover and actually move to AP-D then the handover will be classified as failed handover. Whenever MN enters an overlapping zone which is covered by two or more APs there is a probability of failed-handover.

4.5.1 Probability of failed handover

To calculate the probability of wrong anticipation or failed handover the mobility of MN is modelled using two mobility models; Chiang's mobility model [68] and random walk mobility model [68]. First we consider Chiang's mobility model to calculate the probability of failed handover.

Chiang's mobility model utilizes a probability matrix to determine the position of a particular MN in the next time step, which is represented by three different states for position on *x-axis* and three different states for position on *y-axis*. State 0 represents the current position of a given MN, state 1 and state 2 represent two possible transitional states.

$$P(a,b) = \begin{bmatrix} P_{00} & P_{01} & P_{02} \\ P_{10} & P_{11} & P_{12} \\ P_{20} & P_{21} & P_{22} \end{bmatrix}$$

The probability matrix $P(a,b)$ is used where each entry represents the probability that an MN will go from state a to state b. The values within this matrix are used for predicting MN's location in next step on both *x-axis* and *y-axis*.

In Chiang's simulator each node moves randomly with a constant speed. The following matrix contains the values Chiang used to calculate x and y movements:

$$P = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.3 & 0.7 & 0 \\ 0.3 & 0 & 0.7 \end{bmatrix}$$

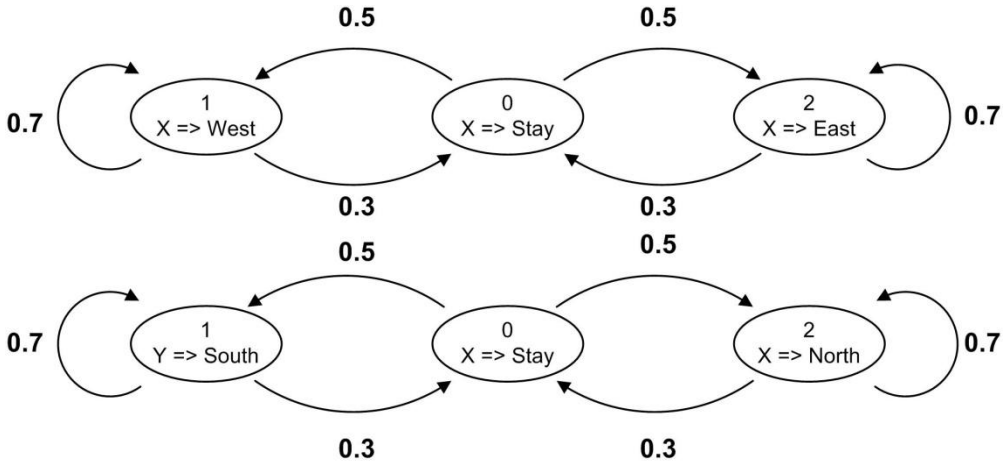

Figure 4.10: State Transition Diagram

These values are illustrated via a state transition diagram of Figure 4.10. With the values defined, an MN may take a step in any of the nine possible directions (i.e., i.e. north, south, east, west, north-west, south-west, south-east, and stay) as long as it continues to move (i.e., no pause time). At the same time, the probability of MN continuing to follow the same direction is higher than the probability of MN changing directions. Lastly, the values defined prohibit movements between the previous and next positions without passing through the current location. This

implementation produces probabilistic rather than purely random movements, which may yield more realistic behaviors. For example, as people complete their daily tasks they tend to continue moving in a semi-constant forward direction. Rarely do we suddenly turn around to retrace our steps, and we almost never take random steps hoping that we may eventually end up somewhere relevant to our tasks. However, choosing appropriate values of $P(a,b)$ may prove difficult, if not impossible, for individual simulations unless traces are available for a given movement scenario.

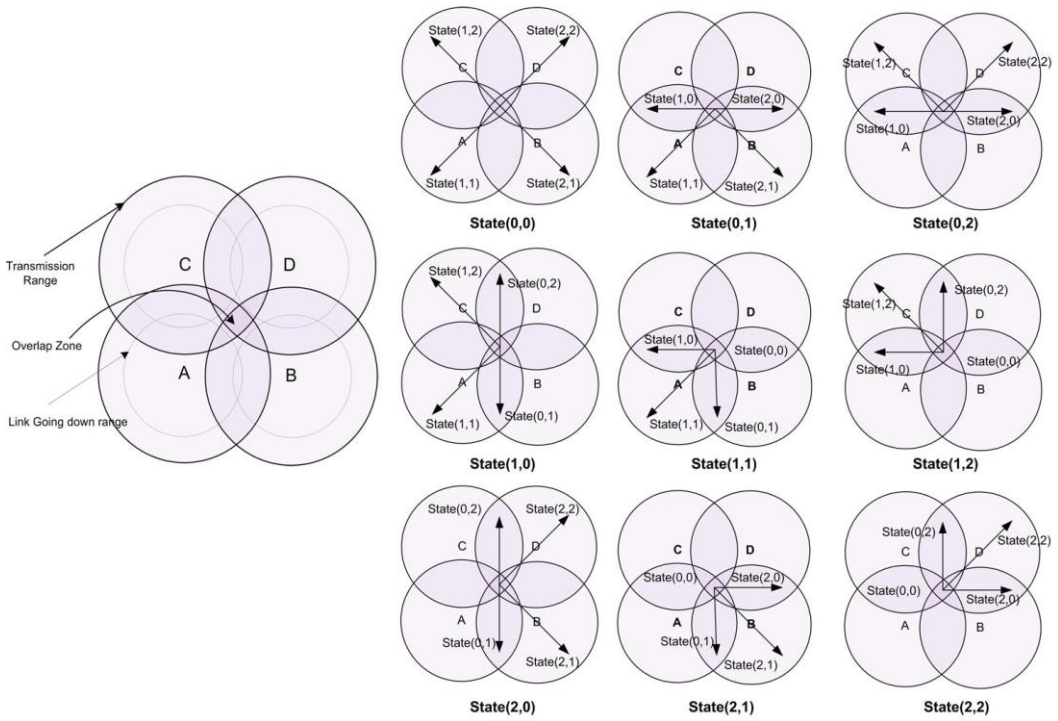

Figure 4.11: MN Possible States

For analysis, MN is initially located in the overlapping area as shown in Figure 4.11. MN randomly chooses one of initial states with the same probability and then changes to one of four next states as in Figure 4.11. It is assumed that handover will be successful only if MN moves to coverage area of AP-B otherwise it fails.

First handover success probability (Ps) is calculated and then handover failure probability (Pf) is calculated according to MNs initial and its transition probability. Table 4.3 shows the analysis results of Ps and Pf.

State(0,0)	Ps	Pf	State(0,1)	Ps	Pf	State(0,2)	Ps	Pf
State(1,1)	0	0.25	State(1,1)	0	0.35	State(1,0)	0	0.15
State(1,2)	0	0.25	State(2,1)	0.35	0	State(1,2)	0	0.35
State(2,1)	0.25	0	State(1,0)	0	0.15	State(2,0)	0.1125	0.0375
State(2,2)	0	0.25	State(2,0)	0.1125	0.0375	State(2,2)	0	0.35
State(1,0)	Ps	Pf	State(1,1)	Ps	Pf	State(1,2)	Ps	Pf
State(1,2)	0	0.35	State(0,0)	0.0675	0.0225	State(0,0)	0.065	0.0225
State(1,1)	0	0.35	State(1,0)	0	0.21	State(1,0)	0	0.21
State(0,1)	0.1125	0.0375	State(1,1)	0	0.49	State(1,2)	0	0.49
State(0,2)	0	0.15	State(0,1)	0.158	0.053	State(0,2)	0	0.21
State(2,0)	Ps	Pf	State(2,1)	Ps	Pf	State(2,2)	Ps	Pf
State(0,1)	0.1125	0.0375	State(0,0)	0.0675	0.0225	State(0,0)	0.0675	0.0225
State(0,2)	0	0.15	State(0,1)	0.158	0.053	State(0,0)	0	0.21
State(2,1)	0.35	0	State(2,1)	0.49	0	State(0,0)	0	0.21
State(2,2)	0	0.35	State(2,0)	0.158	0.053	State(0,0)	0	0.49

Table 4.3: Ps and Pf in different states

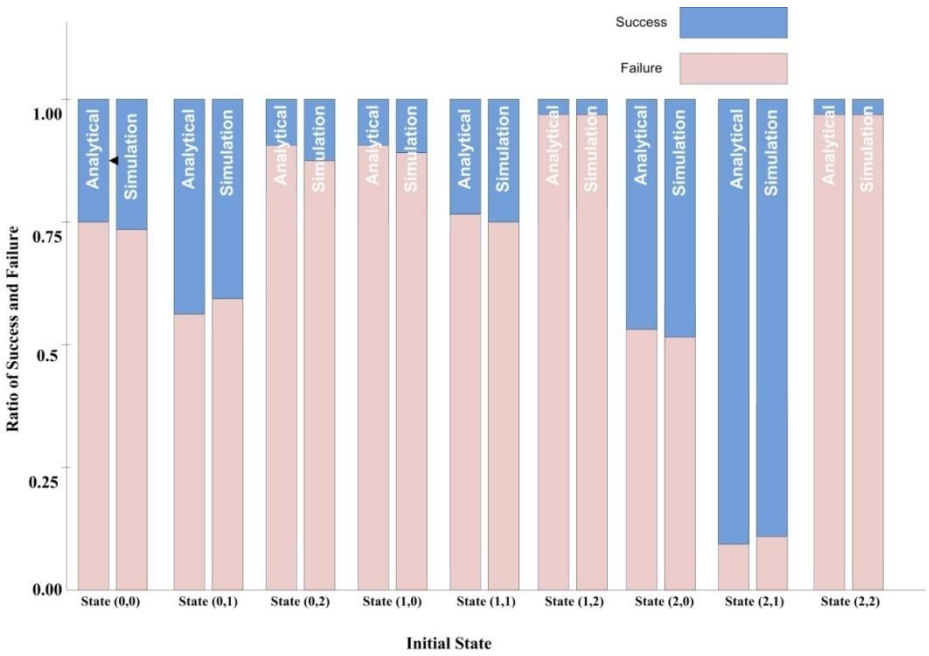

Figure 4.12: Ratio of Successful and Failed Handovers

Once the mathematical analysis was concluded, this analysis is compared with simulation results. In simulation, this probabilistic version of random walk was simulated with sample size of 1000. Figure 4.12 shows the comparison of simulation and analysis results. In these results, the simulation result closely resembles the analysis results. In simulation, 292 handovers were successful and a staggering 708 handover where failed handovers which can be translated as 29.2% success rate.

Next, the probability of failed handover is calculated using a modified version of Chiang's mobility model. Chiang's mobility relies on the fact that MN movement is not fully random; instead it is a directed movement. That is why in state transition diagram of Figure 4.10, the probability of maintaining its current state is higher. If the value of probability matrix is changed such that MN can move in to any state with equal probability, it will become a special case of Random walk model. Since the movement of MN will only depend on its current state rather than previous state. Thus model of random walk can be classified as Markov Process.

The state transition diagram is illustrated in Figure 4.13. State 0 represents the current position of a given MN, state 1 and state 2 represent two possible transitional states. With combination if each x and y states there are total of nine possible states (i.e. north, south, east, west, north-west, south-west, south-east, and stay)as illustrated in Figure 4.13. For the purpose of keeping analysis simple the speed of MN is assumed to be constant.

$$P = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0.5 \end{bmatrix}$$

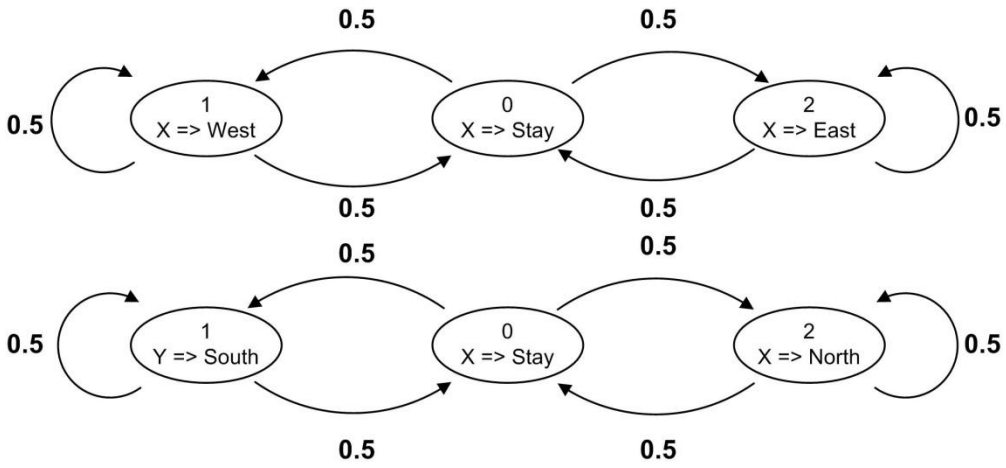

Figure 4.13: State Transition Diagram

To keep the analysis coherent with the previous analysis, identical test conditions are created. MN is initially located in the overlapping area of the coverage network as in Figure 4.11. Then MN chooses one of initial nine states with the same probability and then changes to one of the four possible states. It is assumed that initially MN was connected to AP-C (pAR) and then moves to AP-B (nAR) at once. In this case, handover succeeds only if MN actually moves to AP-B otherwise it fails. The probability of successful handover (Ps) and probability of a failed handover (Pf) is calculated based on the state transition diagram of Figure 4.13. Table 4.4 shows the analysis results of Ps and Pf.

Once the mathematical analysis is concluded, it is compared with simulation where this random walk model was simulated with sample size of 1000. Figure 4.14 shows the comparison of simulation and analysis results. In simulation, 340 handovers were successful and 660 handovers were failed handovers which can be translated as 34% success rate.

State(0,0)	Ps	Pf	State(0,1)	Ps	Pf	State(0,2)	Ps	Pf
State(1,1)	0	0.25	State(1,1)	0	0.25	State(1,0)	0	0.25
State(1,2)	0	0.25	State(2,1)	0.25	0	State(1,2)	0	0.25
State(2,1)	0.25	0	State(1,0)	0	0.25	State(2,0)	0.1875	0.0625
State(2,2)	0	0.25	State(2,0)	0.1875	0.0625	State(2,2)	0	0.25
State(1,0)	Ps	Pf	State(1,1)	Ps	Pf	State(1,2)	Ps	Pf
State(1,2)	0	0.25	State(0,0)	0.1875	0.0625	State(0,0)	0.1875	0.0625
State(1,1)	0	0.25	State(1,0)	0	0.25	State(1,0)	0	0.25
State(0,1)	0.1875	0.0625	State(1,1)	0	0.25	State(1,2)	0	0.25
State(0,2)	0	0.25	State(0,1)	0.1875	0.0625	State(0,2)	0	0.25
State(2,0)	Ps	Pf	State(2,1)	Ps	Pf	State(2,2)	Ps	Pf
State(0,1)	0.1875	0.0625	State(0,0)	0.1875	0.0625	State(0,0)	0.1875	0.0625
State(0,2)	0	0.25	State(0,1)	0.1875	0.0625	State(0,0)	0	0.25
State(2,1)	0.25	0	State(2,1)	0.25	0	State(0,0)	0	0.25
State(2,2)	0	0.25	State(2,0)	0.1875	0.0625	State(0,0)	0	0.25

Table 4.4: Ps and Pf in different states

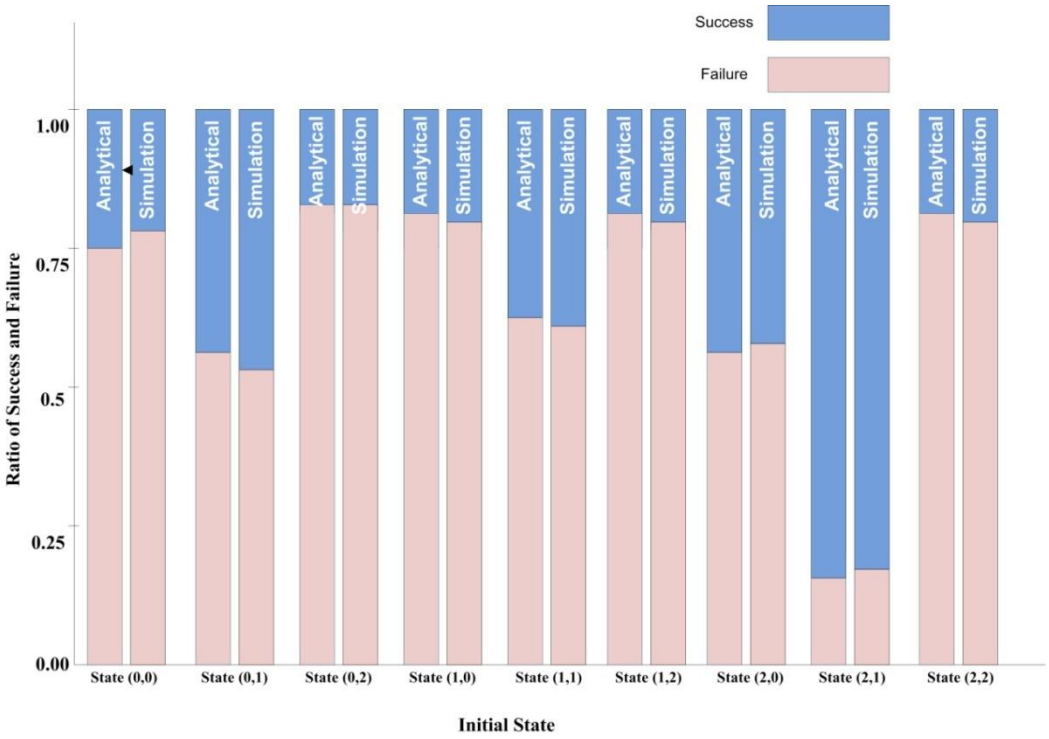

Figure 4.14:Ratio of Successful and Failed Handovers

4.5.2 Proposed Scheme for Next AP selection

Based on previous three locations of MN, the next position of MN can be predicted with reasonable certainty. Using "curve fitting" schemes, the equation of the MN trajectory can be derived from previous three values. By simply incrementing the values of either x or y variable in this equation, next possible location of MN can be predicated.

Since the proposed scheme utilizes MN's current position and location of APs. There are two possibilities to make this information available to MN. One possibility is that MN should be fitted with GPS receiver and co-ordinates of APs should be available to MN in the form of an offline database. Problems with this approach is that extra load is put on MN battery (constant running GPS receiver) and maintaining a potentially huge database. The other possibility is that the required information should be supplied by the network. The second option is preferred because it is easier to implement and does not put extra load on MN battery.

To provide the location information to MN, following changes to the network are required.

- All Access Points must have the same coverage area.
- All Access Points should broadcast there geographical co-ordinates based as part of their SSID

An example of such coverage area is illustrated in figure 4.15. There are 16 APs in this figure and their co-ordinates are illustrated in the figure. When MN is connected to any of the access point, the MN will know its geographic location based on SSID of connected AP. MN location information is only accurate to coverage area of AP but for scheme to work, this information is enough.

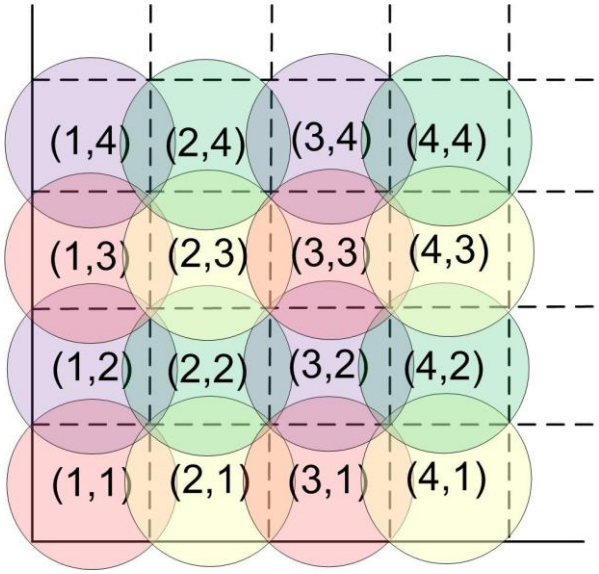

Figure 4.15: AP deployment scheme

In current FMIPv6 when L2 trigger of Link going down is received, MN goes into active scanning mode where it scans all available channels. Once the scan is completed, the AP with highest SNR value is selected. There is a good chance the MN will actually move to a different access point rather than the one with highest SNR.

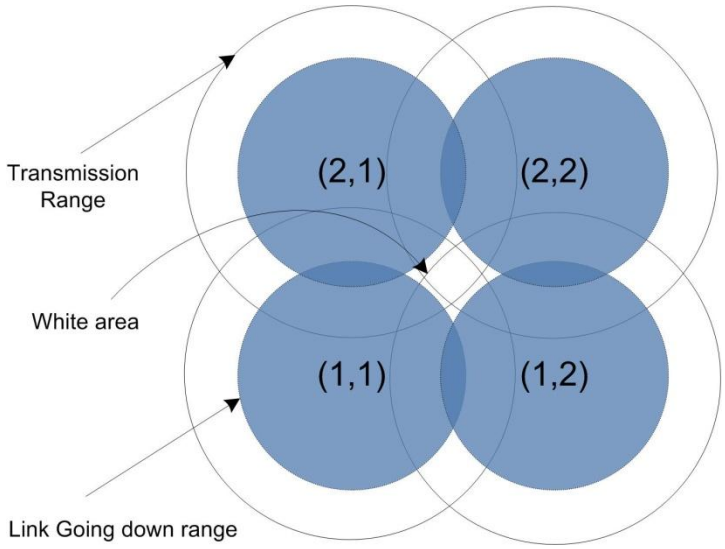

Figure 4.16: AP coverage area with overlapping zone

Consider network deployment as shown in Figure 4.16. There are four APs and all of them have equal transmission range. The range at which "link going down" trigger is generated by MN, is shown by blue circles. Consider a situation where MN is connected to AP (1,1) and it is moving in the direction of AP (2,2). Now when MN crosses the boundary of blue circle, L2 trigger will be generated and MN will go into scanning mode. At the end of the scan, MN will either select AP(1,2) or AP(2,1) because they are closer to MN position so they should have higher SNR value as compared to AP(2,2). In fact, it is very clear that MN is moving in the direction of AP(2,2), and MN should select AP(2,2) its next AP. In such situation, MN will select a wrong AP. With the proposed scheme MN will have a 'sense of direction' and will select the correct AP. This 'sense of direction' is based on MN's current location and uniformity of cell coverage plan. The location of MN is the geographical coordination of its active AP. Based on previous three locations MN, can predict its next location or coordinates of next AP. This prediction is explained in section 4.5.3.

The proposed scheme is now explained with help of a flow chart of figure 4.17. In this flow chart, when 'link going down' trigger is received, MN will check if it has a handover history from previous handovers. MN only needs three handovers to predict its next location. If there is no history available, MN will perform standard FMIPv6 procedures in which it will start active scanning and after completion of scan, AP with highest SNR will be selected. If MN has the previous history, MN will predict the coordinates of next AP. At this stage, MN will check if it has detected the presence of predicted AP from a passive scan, and if found then predicted AP will be selected as the new AP and handover procedures can initiate. If predicted AP is not found, MN will start active scanning and at the very instant MN detects the predicted AP it will terminate scanning and will initiate handover. If MN fails to find the predicted AP, it means that an error has occurred and the prediction was wrong. At this stage, MN will

fallback to standard FMIPv6 procedures; complete the scan and select AP with highest SNR.

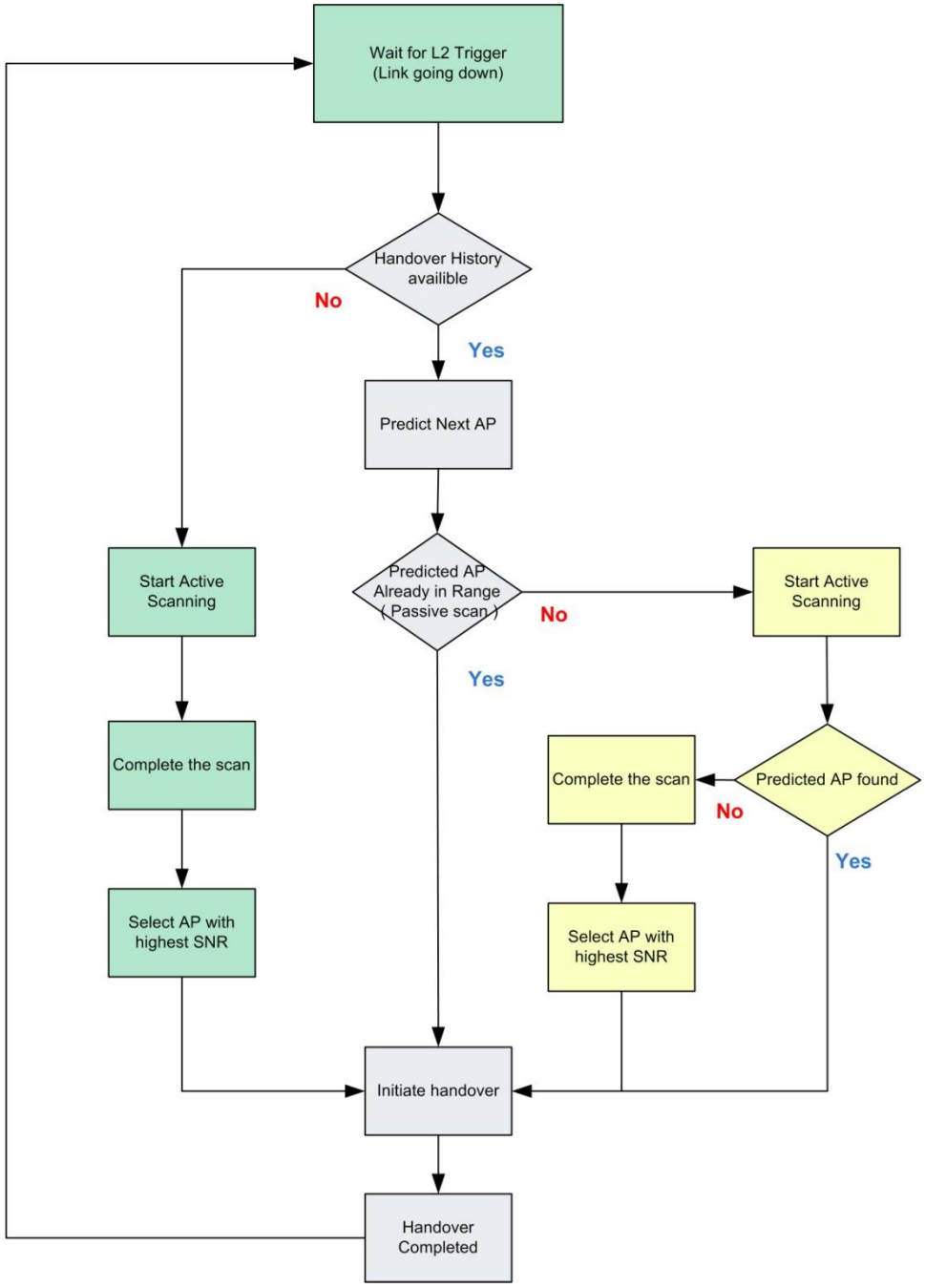

Figure 4.17: Flow Chart for the AP Selection

4.5.3 Estimation of next AP co-ordinates:

Using "curve fitting" schemes, the equation of MN trajectory can be derived from previous three values. Simply incrementing the value of either x or y variable in this equation, next value can be predicated. To explain the procedures for cure fitting, first linear curve fitting scheme, also called as linear regression is explained. Consider figure with four data points. If we have to pick the coefficients that best fit the line to data? Blue line makes a particular straight line and is a 'good' fit for the following reasons:

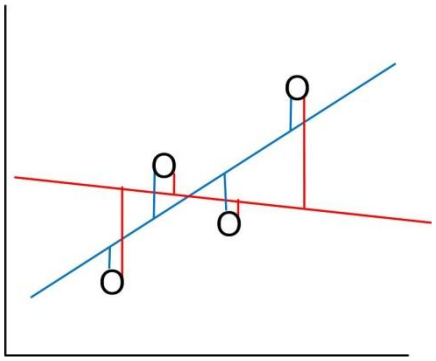

Figure 4.18: Linear Curve Fitting

- Consider the distance between the data point and the line.
- This distance is an expression of the 'error' between data and fitted line.
- The line that provides the minimum error is then the 'best' fit line.

Error in a curve fit:

To quantify error, it is assumed that positive and negative errors have same value and higher errors have big impact. In term of Mathematics, if we square the error values, the same effect can be achieved.

$$err = \sum (d_i)^2 = (y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 + (y_3 - f(x_3))^2 + (y_4 - f(x_4))^2$$

The fit is straight line, so now substitute $f(x) = ax + b$

$$err = \sum_{i=1}^{n} (y_i - f(x_i))^2 = \sum_{i=1}^{n} (y_i - f(ax_i + b))^2$$

The 'best' line has minimum error between line and data points. This is called the least squares approach, since the square of error is minimized.

$$err = \sum_{i=1}^{n} (y_i - f(ax_i + b))^2$$

Take the derivate of the error with respect to *a* and *b*, set each to zero.

$$\frac{\partial err}{\partial a} = -2\sum_{i=1}^{n} x_i (y_i - ax_i - b) = 0$$

$$\frac{\partial err}{\partial b} = -2\sum_{i=1}^{n} (y_i - ax_i - b) = 0$$

Solve for a and b so that the previous two equation both =0

These two can be re-written as

$$a\sum x_i^2 + b\sum x_i = \sum (x_i y_i)$$

$$a\sum x_i + b*n = \sum y_i$$

put these into matrix form

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \end{bmatrix}$$

Only this unknown is value of a and b.

$$A=\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} X=\begin{bmatrix} b \\ a \end{bmatrix}=B=\begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \end{bmatrix}$$

So

$$AX = B$$

$$or$$

$$X = A^{-1}B$$

Options to solve this equation are either matrix inverse or Gaussian Elimination.

$$f(x) = ax + b$$

$$or$$

$$y = ax + b$$

Polynomial Curve Fitting:

Consider Figure 4.19. In this figure, there is a trend in data points. Rather than using a straight line, a smooth curve can pass through these data point.

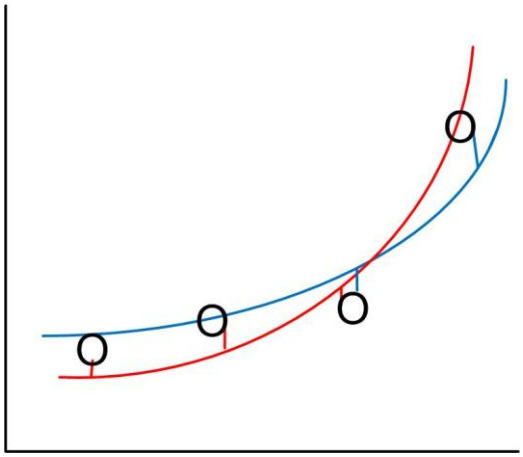

Figure 4.19: Polynomial Curve Fitting

The general equation of a polynomial is as follow.

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_j x^j = a_0 + \sum_{k=1}^{j} a_k x^k$$

Similar to linear regression, the best fitting curve is the one there the error is minimum.

The general expression for any error using the least square approach is

$$err = \sum (d_i)^2 = (y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 + (y_3 - f(x_3))^2 + (y_4 - f(x_4))^2$$

This equation in terms of equation (previous)

$$err = \sum_{i=1}^{n} (y_1 - (a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 + \ldots + a_j x_i^j))^2$$

Where n is number of data points, i is the current data point being summed and j is the order of polynomial.

$$err = \sum_{i=1}^{n} (y_i - (a_0 + \sum_{k=1}^{j} a_k x^k))^2$$

To find is the best fitting polynomial, this equation should be minimized.

Taking derivative with respect to each coefficient $a_0, a_k k = 1, \ldots, j$ set each to zero.

$$\frac{\partial err}{\partial a_0} = -2 \sum_{i=1}^{n} (y_i - (a_0 + \sum_{k=1}^{j} a_k x^k)) = 0$$

$$\frac{\partial err}{\partial a_1} = -2\sum_{i=1}^{n}(y_i - (a_0 + \sum_{k=1}^{j} a_k x^k)) = 0$$

$$\frac{\partial err}{\partial a_2} = -2\sum_{i=1}^{n}(y_i - (a_0 + \sum_{k=1}^{j} a_k x^k)) = 0$$

$$\frac{\partial err}{\partial a_3} = -2\sum_{i=1}^{n}(y_i - (a_0 + \sum_{k=1}^{j} a_k x^k)) = 0$$

$$\vdots$$

$$\vdots$$

$$\frac{\partial err}{\partial a_j} = -2\sum_{i=1}^{n}(y_i - (a_0 + \sum_{k=1}^{j} a_k x^k)) = 0$$

re-writing there $j+1$ equations, and putting into matrix form

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{j+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & \cdots & \sum x_i^{j+j} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_j \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \\ \sum (x_i^2 y_i) \\ \vdots \\ \sum (x_i^j y_i) \end{bmatrix}$$

or

$$A = \begin{bmatrix} n & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^j \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{j+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{j+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum x_i^j & \sum x_i^{j+1} & \sum x_i^{j+2} & \cdots & \sum x_i^{j+j} \end{bmatrix} X = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_j \end{bmatrix} = B = \begin{bmatrix} \sum y_i \\ \sum (x_i y_i) \\ \sum (x_i^2 y_i) \\ \vdots \\ \sum (x_i^j y_i) \end{bmatrix}$$

So

$$AX = B$$

or

$$X = A^{-1}B$$

Options to solve this equation are either matrix inverse or Gaussian Elimination.

Consider the example of Figure 4.20. In this Figure there are 12 APs. Initially MN was in coverage area of AP-A then it moved to coverage area of AP-B and later on it move in to coverage area of AP-G. Using this location history, next possible AP can be predicted. If curve fitting techniques are applied to MN previous locations then the equation of a polynomial that is a 'best fit' for tracking its trajectory can be derived.

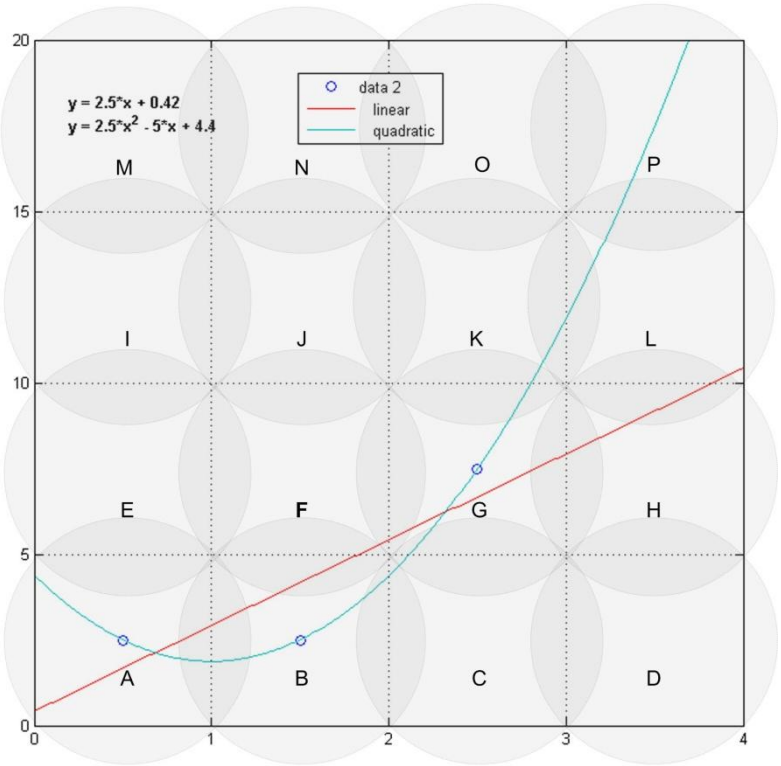

Figure 4.20: Next AP selection in MATLAB

In the figure two equations that are possible candidates for MN trajectory are derived.

$$y = 2.5x + 0.42$$

$$y = 2.5x^2 - 5x + 4.4$$

First equation is first order polynomial represented by red line in the figure. Second equation is a second order polynomial and is represented by blue line in figure 4.20. From visible inspection it is obvious that blue is much better than the red line.

When MN is in AP-G its coordinates are (2.5,7.5) which are actually the coordinates of AP-G. If we increment the values of x by one step and put its equation.

$$y = 2.5(3.5)^2 - 5(3.5) + 4.4$$

$$y = 17.5$$

Now the possible coordinates with incrementing x are (3.5, 17.5). This location is coverage range of AP-P and MN cannot directly move to AP-P for AP-G.

Now if we increment the value of y and pit it in equation

$$12.5 = 2.5x^2 - 5x + 4.4$$

Solving it for value of x

$$x = 3.05$$

So the possible co-ordinates with incrementing y are (3.05, 12.5). These coordinates are in coverage area of AP-L, so AP-L will be selected as the next access point for handover.

4.5.4 Evaluation of the scheme

For performance evaluation of this scheme, a simulation in Matlab was run. For this simulation, human mobility tracers were sourced from one thousand hours of GPS traces in various outdoor settings including college campuses, theme parks and metropolitan cities, were available. This data contains x-y coordinates of voluntaries from a fixed reference point. Data collection interval is every 30 seconds. Average walking speed of human during a normal walk is 1.2 meters per second], so in 30 seconds humans can cover distance of 36 meters. Because of this, the coverage area of an AP for this simulation was selected as 50m.

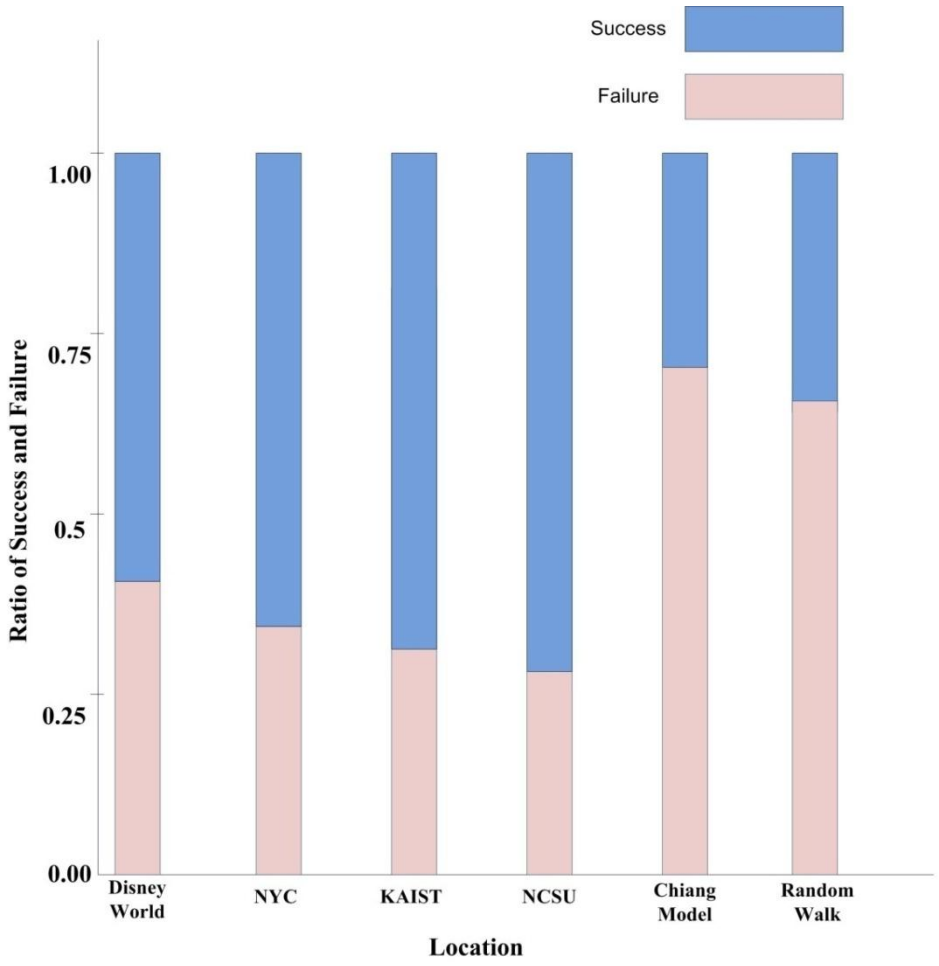

Figure 4.21: Ratio of Successful and Failed Handovers

The data from different locations including Disney world (theme park), New York City (metropolitan city), KAIST (campus) and NCSU (campus) was tested in simulation as the results are shown in Figure 4.21.

In this Figure 4.21, the ratio of successful and failed handovers is illustrated. In case of Disney work, 60% of the handovers are successful. In case of New York City, 62% handovers are successful. In case of KAIST and NCSU campuses, 65% of all handovers are successful. In contrast to mobility models where the success rate was 30-35%, the success rate in case of proposed scheme is 60-65%.

From these results, it can be concluded that the proposed scheme provides a higher success rate.

4.6 Chapter Summary

In the chapter it has been proved that adding security to FMIPv6 will not adversely affect FMIPv6 in any possible way. It is still fast and secure. During our experimentation on actual testbed, a seamless handover is observed despite all this added security. In case of reactive handover, the latency was between 2.5-4 seconds. Major contribution to this delay was from the scanning time which was between 1.8-3.4 seconds. In case of predictive handover, this time was about 16-17ms without including the scanning phase. In addition to added security, a novel scheme to assist MN in selection of optimal AP is also presented. This optimal AP selections scheme can increase the success rate of handover from 30-35% to 65%.

There is one major issue in the protocol which is not addressed. This is Candidate Access Router Discovery. The only way to discover Candidate Access Router is by scanning. This scanning time is the longest delay in overall handover. One possible approach that was adopted is to use two wireless interfaces on MN; one interface for scanning and one for actual

data communication. Another possible solution could be to use 802.21 information sever for Candidate Access Router discover.

Chapter 5

5 Communication Path Reliability

These days multiple interfaces like WiFi, Bluetooth, GSM and 3G are almost standard on all Personal Digital Devices. Aim of all these interfaces is to keep the device connected to any of the available access technologies.

One of the benefits of having multiple interfaces is that when active, they provide multiple communication paths between Mobile Node and the Corresponding Node. If there is failure on one path, Mobile Node can switch to an alternate path by switching its communicating interface. The switch from one interface to another causes service disruption. For many real-time applications this is undesirable.

When Mobile Node running Mobile IPv6 (MIPv6) [6] is equipped with multiple interfaces belonging to distinct access technologies, it is said to be multi-homed. The performance of a particular access technology may vary throughout a single communication session. If signal strength on a particular interface used for communication deteriorates beyond acceptable value, reliability becomes an issue. There needs to be a mechanism to quickly detect the failure and switch the traffic from failing communication path/interface to a different one.

This chapter presents a solution that can quickly detect path failures between Mobile Node and Corresponding Nodes. At the same time, it

provides a mechanism to automatically switch its active interface without any service disruption.

5.1 Introduction

Future of communication networks lies in the integration of different access technologies. This is probably the reason why more and more devices that support multiple interfaces are hitting today's market. Aim of having multiple interfaces on a single device is to keep it connected to any of the available access technologies. One of the benefits of having multi-interfaced device is to allow users to access multiple networks simultaneously and to manage mobility among them. Multiple active interfaces provide multiple communication paths between the Mobile Node and Corresponding Node. If there is a problem on path associated with one access technology, Mobile Node can switch to an alternate path by switching its active interface thus increasing reliability of communication session.

As the technology matures there are still many un-resolved issues with this interface switching. One of the most prominent issue is to enable session continuity during switching.

In IP based networks, mobility support is provided through MIPv6 [6]. Movement of a Mobile Node is hidden from upper layers through the use of two addresses. A permanent address in its home network called its Home Address and a temporary address in the visited network called Care of Address. Binding between these two addresses is kept in a router called a Home Agent. Hence it provides a level of indirection at the network to keep the address change transparent to Upper Layer Protocols (ULPs). When MN is away from its home network, packets are still sent to its HoA. HA intercepts these packets and tunnels them to the CoA.

An IP based Mobile communication can fail when there is a path failure between MN and CN. In MIPv6 when such failures take place, MN doesn't receive any indication. It, however, loses any connections routed through that interface. A situation in which there are multiple interfaces on MN, communication can be resumed by switching to another interface. Although some extensions of MIPv6 aim to reduce horizontal handoff latency, they all suffer from the fact that communication sessions break when the switch is made from one interface to another. In this chapter, a quick interface switching mechanism in multi-homed MIPv6 environment is proposed. This proposal ensures session survivability across outages. When MN is Multi-homed through multiple access networks, each interface corresponds to one of these networks. An MN thus has multiple CoAs.

Our approach is based on Site Multihoming with IPv6 Intermediation (SHIM6). However in place of locators, MN's HoA and CoAs are utilized. Main advantage of this is if path through one of the interfaces fails communication could continue using a different interface. Conceptually a SHIM6 sub layer is added above MIPv6. The locator set for MN consists of HoA and CoAs formed by appending the advertised prefixes with Interface Identifiers. Mapping between HoA and CoA is kept in MIPv6 layer. One of the HoAs or CoAs is used as an Upper Layer Identifier (ULID) of MN when communicating with a CN. In case this ULID becomes unreachable, communication can switch to a different interface by mapping the ULID pair to a different locator pair. Hence the locator switch is hidden from ULPs resulting in continuity of sessions. A SHIM6 state is maintained between MN and CN. During initial context establishment phase, a set of locators is exchanged between the communicating entities. This phase is also used to signal that both ends of communication support SHIM6.

5.2 Related Work

In research community, session survivability issue has been addressed at different layers of OSI models. Some are network layer solution like Shim6, HIP and LIN6, some are transport layer like SCTP and some are application layer solutions like SIP. In this section the above-mentioned solutions are analyzed and it is concluded that none of them fully satisfies the requirement of having multiple active interfaces while supporting seamless interface switching in a mobile environment.

5.2.1 Network Layer Proposals

In an IP based network IP address serves two purposes. It can be viewed as identifying a node (identifier) and also specifying the location of that node in the internet (locator). As IPv6 based networks allow a single device to have more than one globally routable IP address, this can lead to an undesirable situation where IP addresses can be taken to be serving as both identifiers and locators. Consider for instance a situation where a device has multiple interface and unique IP address on each interface. Such a device is said to be multi-homed. When another device on internet wants to communicate with this device it views this device as having several identities as well as locations (IP address) which is actually not the case. The idea behind this class of proposals is to employ a mapping function in the network between an identifier and a set of locators such that transport layer protocols see them as single device.

5.2.1.1 *Host Identity Protocol (HIP):*

HIP was first introduced in 2003 in an internet draft document [49]. In the protocol cryptographic identifiers called Host Identity Tags are used at application layer [50]. These tags are mapped to multiple locators at HIP sub layer (layer 3.5) which is introduced between transport and IP layer. HIT is 128-bit hash value of the host-identifier presented to the transport layer. Initial HIP draft didn't support multi-homing as it only allowed

hosts to use the same IP addresses which communicating peers exchanged during initial HIP exchange. An extension to the original draft [51] allowed hosts to change their IP address during communication.

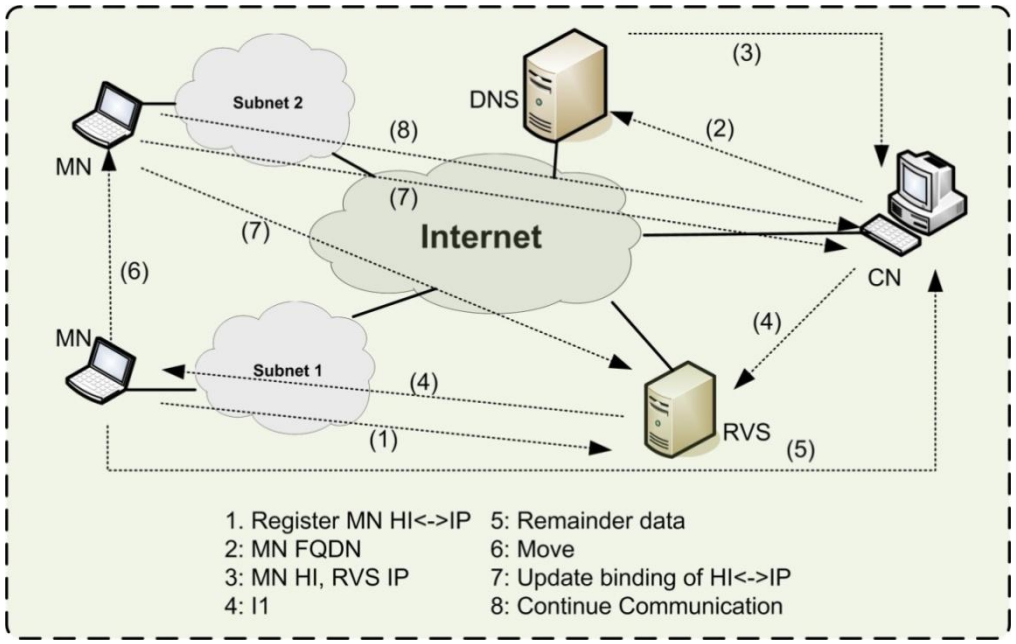

Figure 5.1:HIP protocol

When a host wishes to change its IP address, it informs its peers through HIP Readdress packet. Peer checks reach ability of the host by sending back HIP address check packet and subsequently host replies with Address Check Reply Packet. Hence the scheme suggests that transport layer sockets are not bound to IP addresses but instead are bound to cryptographic identifiers (HIT) resulting in dynamic changes to IP addresses during a communication session.

HIP protocol signaling exchange is given in Figure 5.1.

***5.2.1.2** LIN 6:*

An idea was presented in "LIN6: a solution to mobility and multi-homing in IPv6 [52]". Although the solution addresses both mobility and multi-homing in IPv6, the focus of research is on how the proposal tackles multi-homing problem.

LIN6 introduces two types of IPv6 addresses LIN6 generalized ID and LIN6 address. LIN6 generalized ID is a 128-bit long address which identifies a node in the transport and upper layers. An ID is assigned to the node and is known as LIN6 generalized ID. Application programs use LIN6 generalized ID to indicate target node whereas TCP establishes TCP connection between two LIN6 generalized ID.

LIN6 generalized ID consists of a 64 bit LIN6 prefix which is a predefined constant value and a 64 bits LIN6 ID which uniquely identifies node in the internet.

LIN6 address is also 128 bit long and is composed of 64bit network prefix and a 64bit LIN6 ID. Network prefix indicates the subnet to which the node is connected. It is attached to the head of LIN6 ID to construct LIN6 address. Hence LIN6 address identifies both the node and its point of attachment to the internet. With this approach, if a host changes its subnet only the network prefix part of LIN6 address has to be changed. Furthermore, this change doesn't affect existing TCP connections as they use LIN6 generalized IDs. The mapping between LIN6 ID and the network prefix is maintained in Mapping Agents (MAs). When a host changes its subnet it must inform its MA of the change. This is done through mapping update and reply messages. The record of hosts and their mapping agents are kept in DNS.

To illustrate how LIN6 solves multi-homing problem we refer to Figure 5.2. In the figure hosts A and B have LIN6 generalized IDs "LIN6_P + ID_A" and "LIN6_P +ID_B" respectively. Host B is a multi-homed host having two network interfaces. LIN6 address of host A is "P_A + ID_A" where P_A is the network prefix of node A, where for host B LIN6 addresses are "P_B_1 + ID_B" and "P_B_2 + ID_B" were "P_B_1" and "P_B_2" are the network prefixes corresponding to the two interfaces. Assume host A wants to communicate with host B. A TCP connection is established using the two LIN6 generalized addresses. If host A becomes aware of the

two network prefixes of host B and chooses "P_B_1" for communication the source and destination addresses in IPv6 header would become "P_A + ID_A" and "P_B_1 + ID_B" respectively. Now assume that the connection of host B to its network prefix "P_B_1" breaks. Host A would receive ICMP unreachable error. It can then choose "P_B_2" as the new destination prefix.

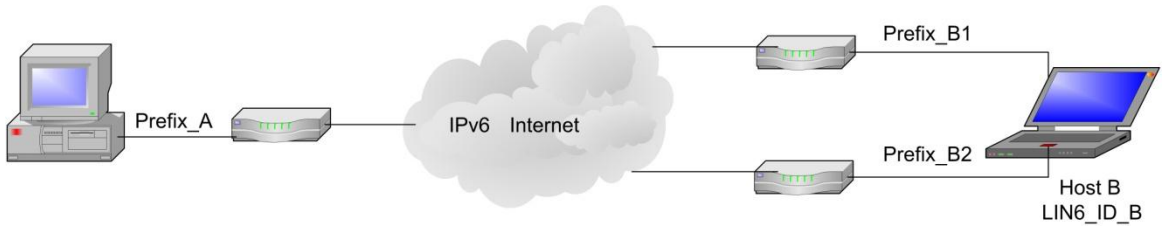

Figure 5.2: LIN6

5.2.1.3 Multi-homing without IP Identifiers (NOID):

Multi-homing without IP Identifiers [65] introduces a Shim layer between the IP layer and upper layers. The proposal uses DNS system for verification and to prevent redirection attacks. As opposed to layer 3 Shim approach which uses identifiers as ULIDs, NOID proposes Application Level IDs (AIDs) to be mapped to/from different locators. In order to perform this mapping M6 Shim layer maintains state called host-pair context. Mapping is performed consistently at both the sender and the receiver. Upper Layer Protocols (ULPs) see packets being sent using AIDs from end to end, whereas actually packets are sent through the network between locators.

When host A wants to communicate with host B it sends a request to DNS to resolve Fully Qualified Domain Name (FQDN) of host B. DNS replies with FQDN along with a set of locators associated with this FQDN. Context establishment phase consists of 4-way hand shake in which host A can learn of the set of host B's locators. This set is compared with the record from DNS and is verified by performing forward and reverse DNS lookups for each FDQN-locator pair. After the context establishment

phase, any combination of valid locators can be used. M6 Shim layer performs the mapping between locator sets and AID. AID is usually chosen to be the first record sent by DNS from the original FQDN lookup.

A main drawback of this approach is that it requires host to have both forward and reverse DNS entries which is sometimes not comprehensible.

5.2.1.4 Weak Identifier Multi-homing Protocol (WIMP/WIMP-F):
Weak Identifier Multi-homing Protocol (WIMP) [64] introduces a wedge layer between IP and ULPs. The approach is similar to NOID except that non-routable AIDs are used which are not part of IPv6 headers.

Before any communication takes place WIMP requires context establishment to occur between two hosts. It uses one way hash chains generated by each end host to authenticate messages exchanged. Host initiating the context establishment receives a set of locators from the corresponding host. It will then map the locators in IPv6 header and the AID. A “tag” is inserted into the IPv6 header to indicate the WIMP context state to the receiving host. Locator sets can be updated after the initial context establishment. WIMP is a scheme mainly defined for mitigating the security threats for multi-homed host. It protects from redirection attacks as the host receiving the redirection request can verify that the host sending the redirection request holds the successor values of a hash chain.

In addition to the above mentioned proposals for locator/identifier split some other proposals came forward. One such proposal involved splitting the IPv6 address into two 8-byte parts [53]. The least significant 8 bytes were used to uniquely identify the interface whereas the most significant 8 bytes were to encode information about the interface in global internet. The proposal met some resistance due to security issues it raised and had to be abandoned.

As mentioned earlier, an important requirement for multi-homed Mobile Node with interfaces bound to multiple access technologies is to be able to switch between them seamlessly. The above-mentioned network layer solutions fail to fulfill this requirement. In all the proposals listed, sessions have to be broken and re-started when switch is made from one interface/IP address to another.

5.2.2 Transport Layer Proposals

In the proposals described in previous subsection, identification problem was tackled by hiding IP address changes from transport layer. In the transport layer proposal, the idea is to modify actual transport layer to support IP address changes.

5.2.2.1 Stream Control Transmission Protocol:

Stream Control Transmission Protocol (SCTP) is presented in [54] to support the host multi-homing at either end of connection. Prior to any connection taking place a multi-homed host informs a corresponding server about all the IPv6 addresses associated with it in a message INIT chunk's address parameters. The server responds with INIT_ACK message containing all of its IPv6 addresses. Each IPv6 address is set to correspond to a transmission path between the two. One of these IPv6 addresses is taken to be the primary path that would be used in normal circumstances. During communications each SCTP connection sends heart beat messages to its peer along with all the transmission paths not currently used for data. The peers or host respond by heart beat ACK chunks.

Each transmission path has a state: active or inactive [54]. A path is active if it has recently been used and an ACK is received. If ACK repeatedly fails the path is deemed inactive.

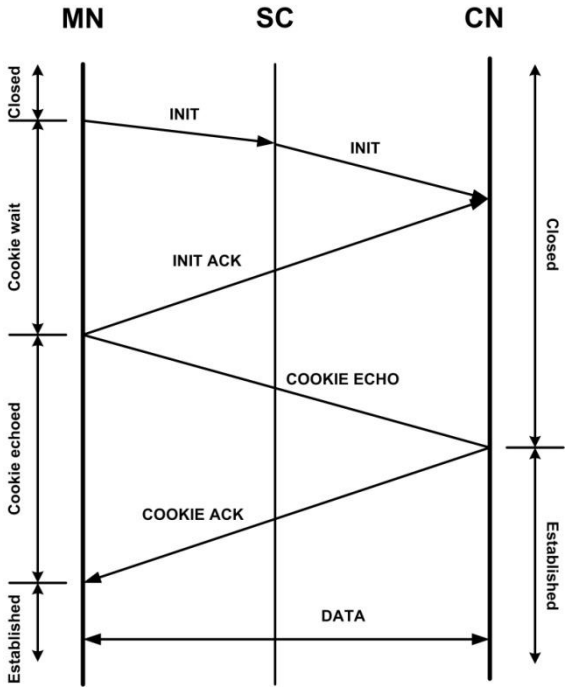

Figure 5.3: SCTP Signaling

End to End Multi-homing:

An approach to modify transport layer (TCP or UDP) or application layer protocols is presented in [56]. The idea is to assign multiple addresses to an interface and let application or transport layer decide which addresses to use. The proposal requires no changes to the routing protocols instead, multi-homing support is provided at end systems. TCP changes are proposed whereby an application can pass multiple addresses to the transport layer. To identify all the addresses of the destination [56] proposes to use DNS lookup.

Although this approach has the advantage that no changes are required in existing routing protocols there are many issues left un-resolved. There is no mechanism presented to allow hosts to choose which address to use as source address. There is no explanation as to how an application or transport layer reacts to any loss of connectivity. There is also no location management entity defined for Mobile Node's location.

5.2.3 Session Layer Proposals

There are certain proposals, which support the idea of providing mobility management at application layer. These proposals are based on Session Initiation Protocol (SIP) [63]. However, there are many flaws with such proposals.

First of all, it is difficult to maintain a long SIP session. These solutions also require additional equipment such as register servers to be deployed. Applications would also require major modifications in order to deploy any of these solutions.

5.3 Proposed Architecture

In order to explain the proposed architecture, its building blocks are discussed first.

Overview of Builing Blocks

- **SHIM6:** This is actual protocol that hides IP address changes. This protocol was designed to work on Edge routers in networks but with some modification, it is adopted to work on Mobile Node
- **REAP Protocol**: This protocol is designed to detect any link failures in core network. Again some changes have been proposed after which it can be used on Mobile Node.

In the following sub-section, these building blocks are described in more detail.

5.3.1 SHIM6

In 2004, Multi6 working group within IETF proposed a solution called SHIM6 [59] to support multi-homing in IPv6 networks. The proposal

introduces a SHIM6 sub layer in the IP stack of end hosts. These hosts reside in a multi-homed site where multiple IPv6 network prefixes are advertised. This results in more than one IPv6 address for each host. One of these addresses is presented to upper layers as Upper Layer ID (ULID). When host decides to communicate with a peer, it initially uses its ULID as the source address. Sometime later, a SHIM6 state can be setup between the host and peer. This state is later used to switch to a different locator pair should the original ULID pair stop working. SHIM6 approach suggests that instead of introducing a new namespace, the identifier or ULID should be chosen from the set of locators. This has an advantage of supporting callbacks and referrals in a long communication session.

Figure 5.4 illustrates where SHIM6 sub layer is added within the protocol stack. Conceptually, SHIM6 header is an extension to IPv6 header. It is placed before all the other extension headers i.e. AH, ESP, fragmentation and destination options header.

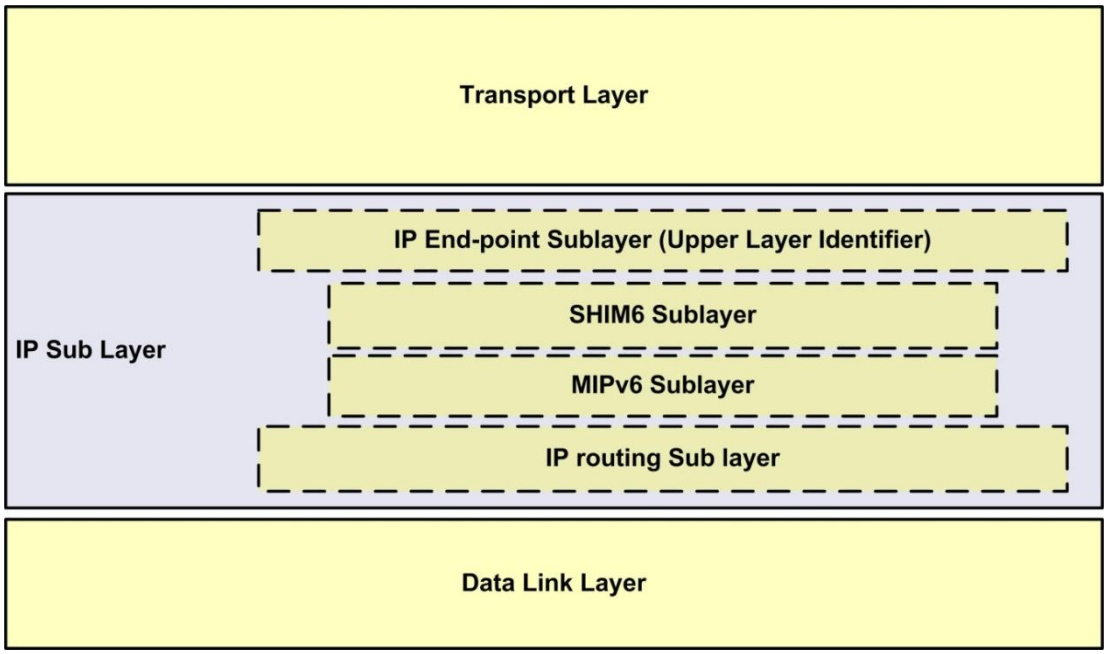

Figure 5.4: Placement of SHIM6 sub layer

Multi-homed hosts would generally choose a locator as source ULID according to default address selection mechanism [60] while destination ULIDs are generally chosen to be first locator obtained from DNS server through destination host lookup. As will be discussed later, our scheme differs somewhat from this in terms of source ULID selection. As in our scheme, the ULID chosen has to be a serving HoA or CoA of the MN. After SHIM6 context is established, the communication can fail over to a different locator pair in case the first pair goes down. A locator pair selection algorithm is presented in [57].

When application on a host decides to contact a peer using Upper Layer Protocol (ULP), it sends packets using IP addresses chosen by Default Address Selection mechanism. This is known as initial contact. At a later stage ULPs on either the host initiating contact or the peer might decide to establish a SHIM6 context to make the communication survive locator failures. This triggers a four-way SHIM6 context establishment exchange. This exchange includes I1, R1, I2, and R2 messages. Details of these messages can be found in [59]. A list of locators to be used by each host is exchanged during this phase.

When a failure takes place SHIM6 failure detection or some other ULP layer reach ability detection mechanism will detect it. SHIM6 uses Forced Bidirectional Detection (FBD) mechanism to detect the failure. Once failure is detected, one or both ends of communication would start probing for different set of locators. This involves sending probe messages containing different locator pairs [58]. When one of these pairs is found to be working, SHIM6 rewrites the packets and tags them with SHIM6 payload extension header. The header contains receiver’s context tag. Context tag is used to identify the context state. It enables the SHIM6 sub layer at receiver to place the right addresses in IPv6 header before forwarding the packets to ULP. This keeps the locator pair changes transparent to ULPs.

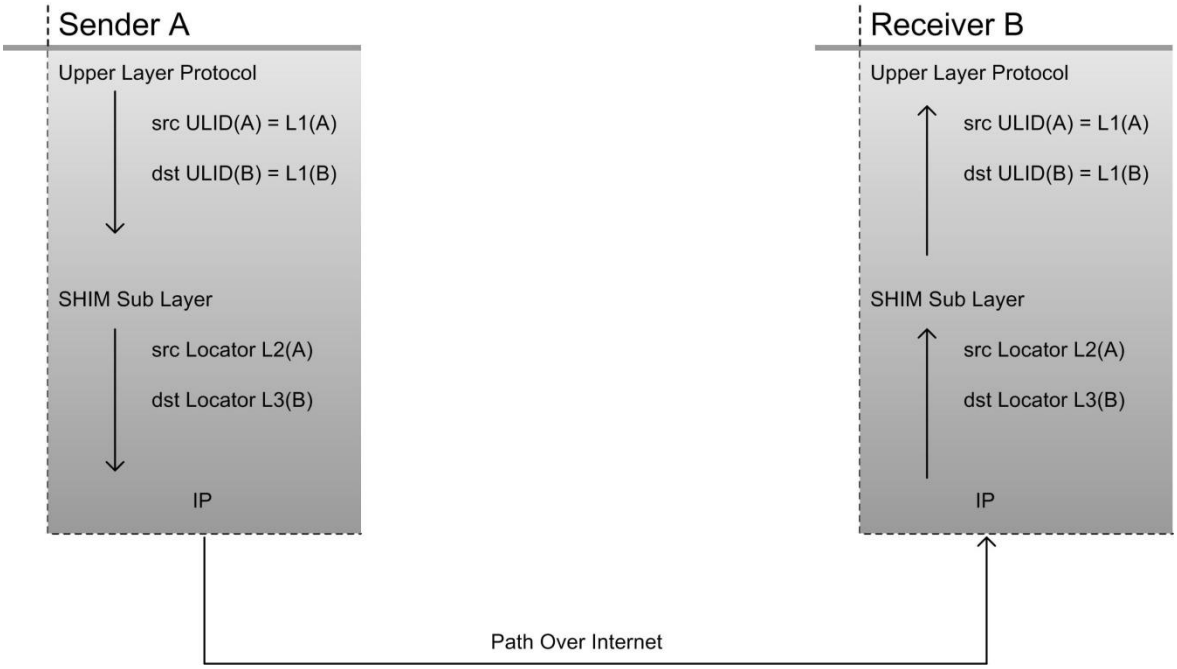

Figure 5.5: Shim mapping when locators are changed

A host can specify locator preferences and communicate them to its peer. It can also indicate any changes in the preferences during communication. When a context state is no longer needed, SHIM6 can garbage collect the state. A recovery message is defined to indicate that there is no context state. In addition, SHIM6 supports context forking where a ULP can specify that a context can be forced into two contexts each associated with a locator pair. This is useful when applications running on hosts want to use more than one locator pair for communications.

5.3.2 REAP Protocol

REAch ability protocol (REAP) as it is known, can be divided into two processes: a failure detection process and a locator pair exploration process. Failure detection can further be divided into three processes. These are tracking local information, tracking remote peer's information, and verifying reach ability. Different metrics can be used to monitor local information about addresses such as whether default router associated with them are still reachable, or whether their associated interfaces are up.

Information about remote peer's addresses is also necessary. Update messages used in SHIM6 protocol are employed to indicate any changes in remote peer's address list. Finally, it is necessary to verify reach ability of an address pair. REAP uses a technique called Forced Bidirectional Detection (FBD) for this purpose. Reach ability is confirmed when data traffic in one direction is complemented by either data traffic, transport layer acknowledgements or REAP keep alive messages. The same procedures are employed except the value of "keep alive" and "send timeout" in our proposal are significantly less than those defined in REAP. This is due to the fact that the original protocol was designed for site multi-homing where network failures are not frequent. However in cases where MN is multi-homed, communication can be disrupted quite frequently resulting in regular session break. To avoid this, timeouts have to be re-calculated. If after transmission of data traffic, a host does not receive any return traffic for the duration of send timer, a full locator pair exploration is initiated. Host which first detects disruption sends probe messages using different address pair in turn to its peer indicating state of connectivity. Reception of probe message triggers an identical exploration process in the peer. When the host receives probe messages back from the peer, it can conclude that peer has received its probe message and that peer's probe messages have also got back to the host. Hence return routability is inherently supported by REAP. Communicating hosts test all the available address pairs until a working pair is found. This testing is performed sequentially with exponential back-off. This process again defines two timeouts. An Initial Probe Timeout, which is the interval between initial attempts to send probes and Max Probe Timeout, which is the interval after which probe interval doesn't grow. In addition, the protocol specifies that Initial probes can be sent four times before exponential back-off procedure is started.

5.4 Test Scenario for the Proposal

In this section the proposed scheme is explained with help of a generic network diagram and a message flow chart. In Figure 5.6, MN is a multi-interface device with WiFi and WiMAX interfaces. MN is in coverage area of both networks and it is connected to both of them. MN runs one instance of MIPv6 and has one HoA, two CoA (one on each interface) and one HA. So it is a multi-homed device.

MN can communicate with CN using many interfaces. Let's consider a case that communication with CN is started with HoA on WiFi interface.

After initial communication, SHIM6 context is established with CN by exchange of I1, R1, I2 and R2 messages. Once the context is established, CN is aware of MN multiple CoAs.

Now for MN, it has two paths to communicate with CN and the current communication is on path associated with WiFi link. Consider a situation where there is a failure on this path. Normally in this case, there would be communication disruption and MN has to first detect path failure that would be when the communication is disrupted and second it would re-initiate communication by sending BU to HA with CoA2. But with the proposed scheme, this failure would be detected with REAP protocol. Once failure is detected by modified REAP protocol running on CN, it starts sending probe messages indicating to MN that it is not receiving any traffic. These probe messages are sent through a different access network and using a different locator. MN replies with probe message of its own and thus a working address pair is explored. This working pair is set as new address pair for communication. SHIM6 sub layer in MN maps the ULIDs to this address pair and adds receiver (CN's) context tag to the packets. This way address change is kept transparent to ULPs.

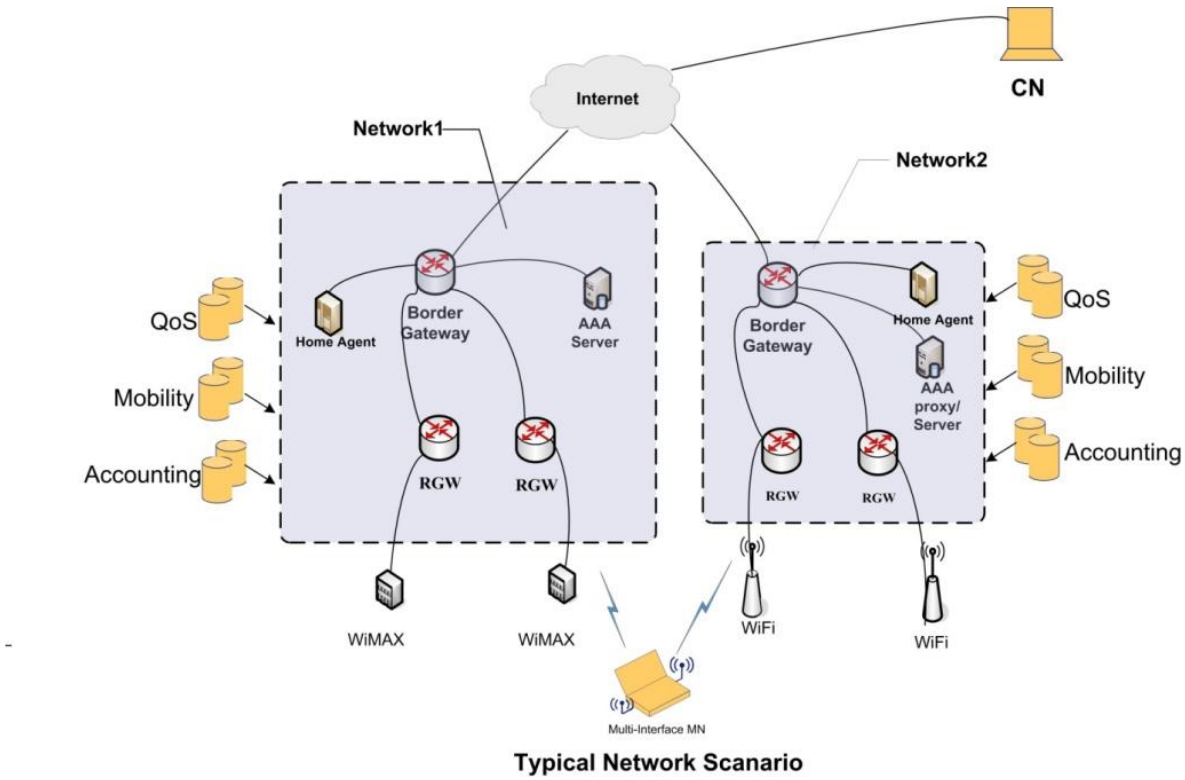

Figure 5.6: Multihoming scenario for MIPv6

The original REAP protocol was modified and the timers were adjusted according to the Table 5-1. In this table, **Send Timeout** represents the time after which if keep alive message is not received by any node, it will start probing for other working locater-pair. **Keep alive** interval of 150ms represents that a keep alive message will be sent very 150ms if there is not traffic exchange between two SHIM6 enabled nodes. **Initial Probe time** of 200ms represents that when a node sends a probe message, it will wait for 200ms before sending another probe message. The maximum probe attempts on a single locater-pair are set to be four and this is represented in table by **Number of probes**. Final entity in this table is **Maximum probe timeout** which represents for how long a node can keep on exploring a working locater-pair. Once this timer is exhausted, the node will stop sending probe messages.

Send Timeout	500ms
Keep alive	150ms
Initial Probe time	200ms
Number of probes	4
Maximum probe timeout	2.5s

Table 5-1: Modified Timers for REAP

5.5 Actual testbed implementation and Results

In this section the actual testbed and the results are presented. In the testbed, the proposed mechanism of failure detection and recovery was tested. A multi-interface node was connected to a number of foreign networks. It registered with a home agent for mobility services. There was another node acting as video streaming server. **VMware** visualization technology was used to simulate these nodes. All the nodes were based on Linux with UMIP for MIPv6 support. The results were collected by packet sniffing tool, WireShark.

5.5.1 The Testbed

For testbed development a minimum of four nodes are required. One acting as MN, one acting as HA, one acting as CN and one acting as router to interconnect different network prefixes. The actual testbed is illustrated in Figure 5.7. As drawn in this figure, Mobile Node with three interfaces is created. All these interfaces are Fast-Ethernet interfaces. The IP addresses assigned to these three interfaces are 2001:660:4701:3::128/64, 2001:660:4701:4::128/64 and 2001:660:4701:5::128/64. In these IP addresses “2001:660:4701:4” part represents the network prefix. So all the interfaces have different network prefixes. HA had just one Fast-Ethernet interface with prefix of “2001:660:4701:1” and IP address of “2001:660:4701:1::1/64”. Corresponding node was a multi-interface server with two fast-Ethernet interface with prefixes of “2001:660:4701:10” and

"2001:660:4701:11". This CN was acting as video streaming serving using VLC software to stream server in either RTP or UDP transport mechanism. Finally a Linux based router is required to interconnect different network prefixes. It was running RADVD protocol suit to perform routing.

Instead of using actual machines, VMware visualization technology was used to visualize these nodes. In VMware, four virtual machines were created and they were interconnected via virtual network. The VMware virtualization technology was selected because it can speed up overall testing. Linux, MIPv6 and SHIM6 were installed on one machine and then this machine was replicated three times to generate other required nodes. The details of these machines are described in following table (Table 5-2) and detail network diagram in illustrated in the Figure 5.7.

Network Entity	OS	Number of NIC	Network Prefix
HA	Linux kernel 2.6.23-rc3	1	2001:660:4701:1
MN	Linux kernel 2.6.23-rc3	3	2001:660:4701:3 2001:660:4701:4 2001:660:4701:5
CN	Linux kernel 2.6.23-rc3	2	2001:660:4701:10 2001:660:4701:11
Router	Linux kernel 2.6.23-rc3	6	2001:660:4701:1 2001:660:4701:3 2001:660:4701:4 2001:660:4701:5 2001:660:4701:10 2001:660:4701:11

Table 5-2: Hardware/software specifications

In this figure there are four network entitles, three acting a nodes and one acting a router to interconnect different network prefixes.

In the actual testing MN was requesting a video stream from CN. Both MN and CN were multi-interfaced and were supporting SHIM6 protocol along with modified Reap protocol. MN was streaming video from CN.

After initial communication, SHIM6 context is established with CN by exchange of I1, R1, I2 and R2 messages. Once the context is established, CN is aware of MNs' multiple locators.

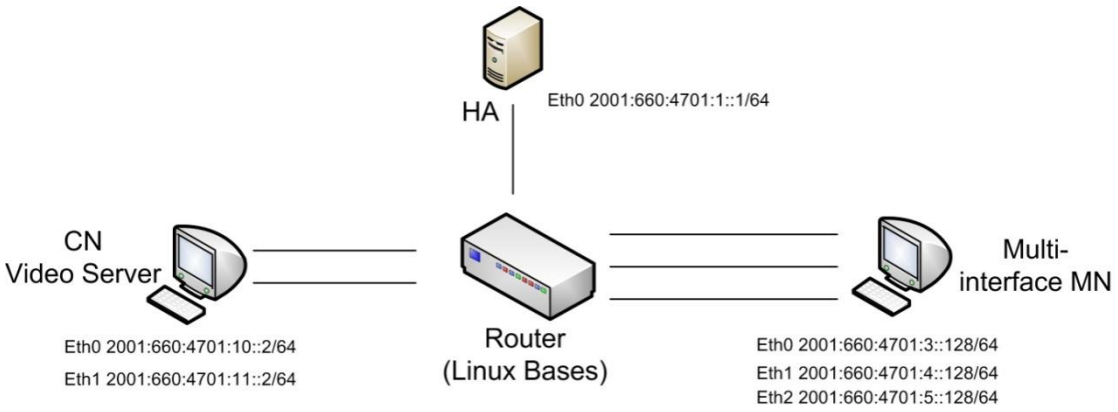

Figure 5.7: VMware Testbed

Now for MN, it has six paths to communicate with CN. One of these paths is selected for communication. By default, it would use the path that was used to initial contact as its primary path. Consider a situation where there is a failure on this path. Normally in this case, there would be communication disruption. MN detects this failure when communication is disrupted but with proposed scheme, this failure would be detected with REAP protocol.

For simulating a failure an active interface (on which actual data was transmitted and received) of MN was forced to non-operational state.

Once failure is detected by modified REAP protocol running on CN, it starts sending probe messages indicating to MN that it is not receiving any traffic. These probe messages are sent through a different access network and using a different CoA. When MN receives this probe message, it replies with probe message of its own and thus a working address pair is explored. This working pair is set as new address pair for communication. SHIM6 sub layer in MN maps the ULIDs to this address pair and adds receiver (CN's) context tag to the packets. This way address change is kept transparent to ULPs.

5.5.2 Results

During experimentation on testbed, traffic throughput, traffic disruption time, and packet loss were logged by wireshark for two scenarios: one with modified REAP enabled and the other with modified REAP disabled and the results were compared.

The log file contains timestamps and the actual packet header. There are other network sniffers available, but Wireshark was selected because this is the only tool that supports real-time network capture and display.

Traffic Disruption Analysis

We obtained results for average disruption time for traffic. There were ten trial runs and the results of modified REAP and simple MIPv6 failure detection and recovery procedures were compared in Figure 5.8. From the results it can be concluded that the disruption time experienced by traffic during failures is significantly reduced when modified REAP is employed. The average disruption time for MIPv6 is around 1.5 seconds where as for modified REAP its around 800 msecs.

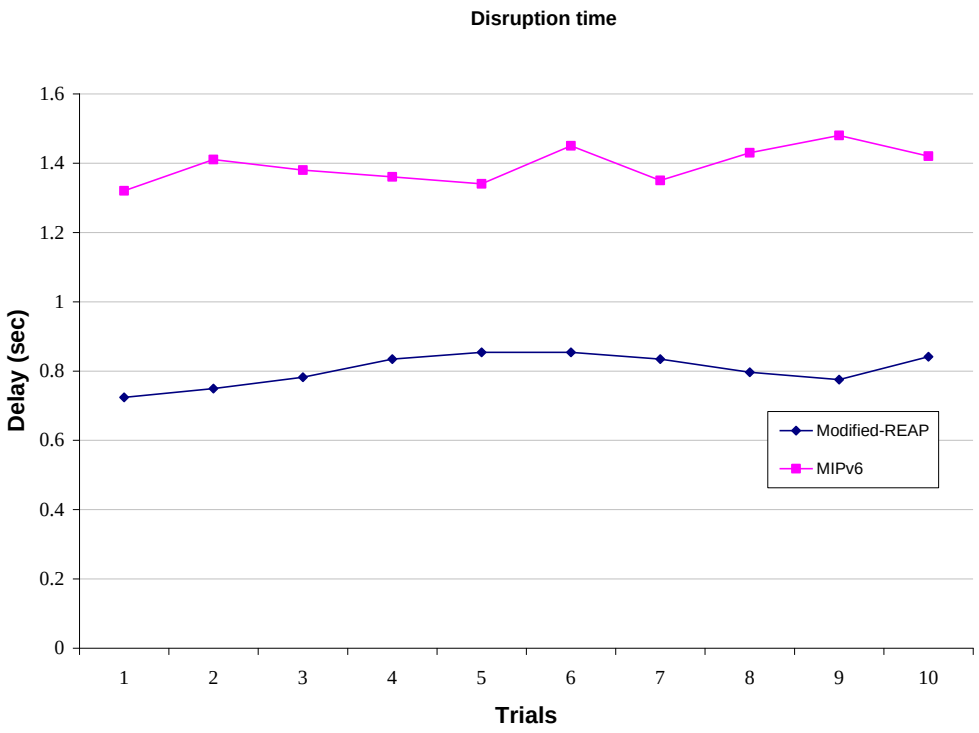

Figure 5.8: Disruption Time

Traffic Throughput Analysis

Next, we compare the traffic throughput. This is the traffic throughput at CN when communicating with MN. The comparison is presented in Figure 5.9.

The results show that the effect of failure on throughput is much more visible in the MIPv6 case than in modified REAP. The traffic flow is much more smooth for when modified REAP is employed.

It can be concluded that during switching failure detection and recovery is much quicker when modified REAP is employed. This is significant as for most applications, session continuity is guaranteed. This also guarantees reliable communication.

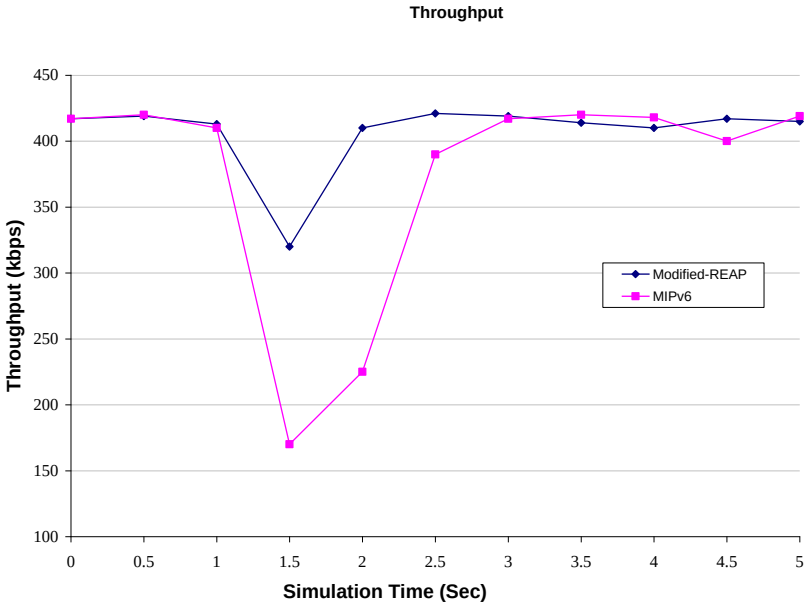

Figure 5.9: Throughout vs. Time

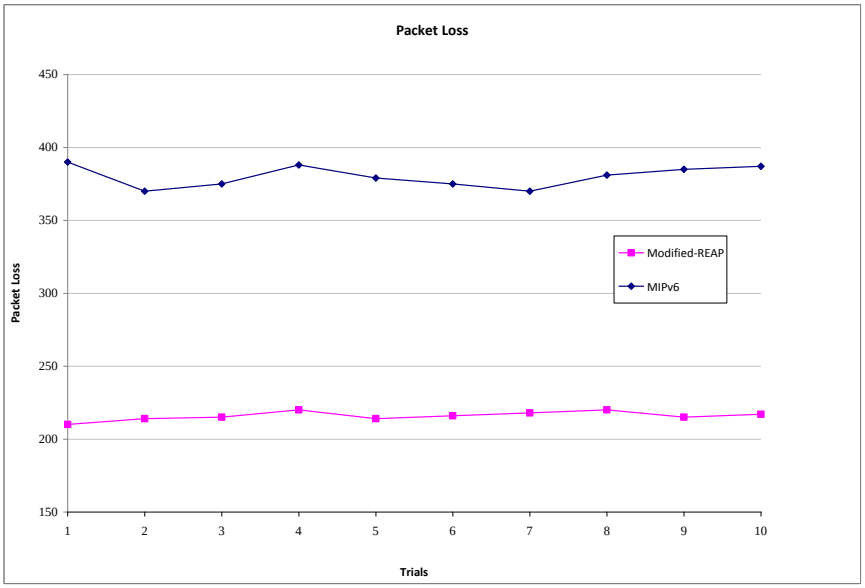

Figure 5.10: Packet Loss

Packet Loss Analysis

Using the same setup, we ran some tests to calculate the packet loss during the switch. Figure 5.10 gives the plot. The packet loss is plotted

for ten trial runs. Again, the result shows how modified REAP outperforms MIPv6 failure detection and recovery.

5.6 Chapter Summary

In this chapter, a SHIM6 based scheme to detect communication path failure in MIPv6 environment is introduced. The failure detection was based on expiry of “send time out” and then recovery was performed by switching to another active interface. This switching is transparent to applications. The main advantage of this is when failure through one interface occurs communication can switch to another interface without affecting most of the on-going sessions.

By introducing REAP based path failure detection scheme to mobile communication can greatly improve user experience in case of multiple Network interface. During experimentation, the path failure in case of multiple interface were detected within 800ms and, and 80% of recoveries took just 860ms. This recovery time is about half a second and most of today’s applications are resilient to such a delay. Some applications like VoIP and Video conferencing have very little chance of surviving such a delay.

The testbed was developed in a wired environment where prediction of link failure is next to impossible. For wired environment, REAP have to detect failure which takes minimum of 500ms because of the “send time out” timer. Once the failure is detected, the recovery procedures can start by exploring a “working pair”

In case of wireless networks, prediction of link failure is possible. For example, the Received Signal Strength (RSS), which is usually translated to Signal to Noise Ratio (SNR), can provide a good prediction. If RSS

value is close to the threshold, then it means that there is very good chance that this particular link will fail pretty soon. Rather than waiting for link failure to occur, if the exploration of new “working pair” is started before actual link failure the service disruption can be reduced to about 5-10ms. This approach could lead to virtually seamless interface switching.

One disadvantage of this SHIM6 based approach is that both CN and MN have to run another protocol stack which mean extra processing load. Another disadvantage is that MN will have to keep all interfaces active all the time.

Chapter 6

6 Conclusion and Future Works

This chapter highlights the research contributions presented in this thesis. Future work in this area is also described in the last section of this chapter.

6.1 Concluding Remarks

This thesis studies the reliability of mobility support in IP based network. For mobility support in IP networks, MIPv6 is considered a standard. This mobility protocol suffers from two issues: first, the reliability of HA and second, Home Agent load sharing. In research community, a few schemes are proposed to address these issues. These schemes were analyzed and it is concluded that they fell short on some important aspect. We propose a new HA reliability architecture. The existing reliability schemes introduce redundant HAs in the network. In our proposed mechanism we eliminate do not introduce any redundant HAs. MN is simultaneously registered with two HAs through the new HA registration scheme. A new HA failure detection scheme is also introduced. It involves introduction of L2 frames. Higher layer information is embedded in these L2 frames. Hence detailed information can be conveyed without introducing significant load on the network. Failures are detected through this scheme in less than 40ms. Three possible load sharing schemes are also evaluated in the thesis.

Finally, our proposed failure detection and load sharing schemes are compared with existing solutions.

Chapter 3, presents a framework for HA Management based on concept of Autonomic computing. By introducing this concepts, HA systems are able to monitor themselves and perform self-healing and self-optimization. There are three modules involved in the proposed solution. Two of them are pure software implementations, which are not developed due to lack of expertise in the field. Third module was simulated in NS2 and different aspects of the scheme were analyzed. L2 frames are used to transport monitoring data. Main advantage of using L2 frames is that more information can be carried without generating excessive traffic on the network. At the end of the chapter, this framework is extended to other network application/services. Overall this approach should increase reliability of seamless mobility in heterogeneous network as it increases the reliability of HA and it tries to maintain it in optimal state.

For seamless mobility, handovers are the biggest obstacles. In **Chapter 4**, a mechanism for handovers in WiFi networks is presented. This mechanism is based on FMIPv6 but as FMIPv6 is not in full compliance with MIPv6 RFC some modifications were incorporated. As part of FP6 project **ENABLE**, author was responsible for modifying and demonstrating a working testbed of FMIPv6 that is in compliance with MIPv6 RFC. Actual testbed was presented in final demonstration meeting of the project. This was the first ever implementation of RFC compliant FMIPv6 protocol. Aim of building the testbed was to analyze the effect of added security procedures on overall FMIPv6 protocol. After extensive testing, it was concluded that there is no significant difference between handover times of non-secure FMIPv6 and secure FMIPv6. In addition to this, an improved handover anticipation scheme was presented in the chapter to assist FMIPv6. This anticipation is based on prediction of MN trajectory. Under certain test conditions, the success rate of handovers in

standard FMIPv6 ranges from 30% to 35%. With the proposed scheme, this rate is increased upto 60%.

In **Chapter 5**, author has presented a scheme by which MIPv6 enable Mobile Node can utilize its multiple interfaces to seamlessly switch between different networks. Based on modified REAP protocol, it is possible to detect communication path failure in MIPv6 environment. In case of a failure, author presents a recovery scheme by switching to different active interface. This switching is transparent to applications. The main advantage of this is when failure through one interface occurs, communication can switch to another interface without affecting on-going sessions.

6.2 Future Research Direction

To further enrich the research work presented in this thesis, the following suggested research directions could be pursued in future.

- Transparent switching of HA without increasing network overheads. HA reliability is still outstanding issue. The IETF working group "MEXT" is responsible for standardizing HA redundancy. In their charter, transparent HA switching is still listed as an un-resolved issue.

- Concept of autonomic computing and networking requires standardization. There is plenty of work already done on standardization but still some of work needs to be done especially towards defining a homogeneous policy.

- FIMPv6 only works in WiFi networks. It can be extended to heterogeneous network.

- Another aspect of FMIPv6 where further research is required is Candidate Access Router selection procedure. Scanning phase takes

too much time. There could be a further reduction in handover latency if we can reduce or fully eliminate this scanning time. Options are to either use 802.21 information server or Autonomic Networks resource database.

- Multi-homing in core networks is not a new concept but in case of mobile environment it has emerged recently. These days multi-radio interface PDAs, Laptops and even Smart phones are common site. In our proposed architecture, only one interface is active at a given time. There is a possibility to use multiple interfaces at the same time. For example, using WiFi for streaming video and WiMax for VoIP. Interface selection on its own is a research area.

- The possibility of using link layer triggers to initiate “interface switching” procedures in a multi-homing scenario needs further research. For example if MN has two active interfaces and received signal strength on one interface is very close to threshold, MN should switch to the other active interface because there is a good chance that MN will soon lose its connectivity.

Bibliography

[1] Khan, S., Khan, S., Mahmud, S.A. and YAO, W. 2006 Supplementary Interworking Architecture for Hybrid Data Networks (UMTS-WiMAX). In Proceedings of Wireless World Research Forum 16th, shanghai china

[2] Khan, S., Khan, S., Mahmud, S.A. and Al-Raweshidy, H. 2006. Supplementary Interworking Architecture for Hybrid Data Networks (UMTS-WiMAX). In Proceedings of the International Multi-Conference on Computing in the Global Information Technology, Anonymous IEEE Computer Society Washington, DC, USA,

[3] Enabling Efficient and operational Mobility in Large Heterogeneous IP Network. http://www.ist-enable.org/

[4] Perkins, C., "IP Mobility Support," IETF-RFC (Proposed Standard) 3344, August 2002.

[5] Perkins, C., Johnson, D. and Arkko, J., "Mobility Support in IPv6," IETF Draft, draft-ietf-mobileip-ipv6-24 (work in progress), August 2003

[6] Perkins, C., Johnson, D., and Arkko J., "Mobility Support in IPv6," IETF-RFC (Proposed Standard) 3775, June 2004.

[7] Chambless, B. and J. Binkley, "HA Redundancy Protocol," IETF Draft, draftchambless-mobileip-harp-00.txt (work in progress), October 1997.

[8] Ghosh, R. and G. Varghese, "Fault Tolerant Mobile IP," Washington University,Technical Report (WUCS-98-11), 1998.

[9] Ahn, J. and C.S. Hwang, "Efficient Fault-Tolerant Protocol for Mobility Agents in Mobile IP," in Proc. 15th Int'l Parallel and Distributed Processing Symp., pp. 1273 -1280, 2001.

[10] Leung, K. and M. Subbarao, "HA Redundancy in Mobile IP," IETF Draft, draftsubbarao-mobileip-redundancy-00.txt (work in progress), June 2001.

[11] Wakikawa, R., Devarapalli, V. AND Thubert, P. 2004. Inter home agents protocol (HAHA). draft-wakikawa-mip6-nemo-haha-01.txt), Internet Draft, IETF

[12] Faizan, J., El-Rewini, H. AND Kaalil, M. 2005. VHARP: Virtual home agent reliability protocol for mobile IPv6 based networks. Wireless Networks, Communications, and Mobile Computing.

[13] R. Hinden "Virtual Router Redundancy Protocol for IPv6", draft-ietf-vrrp-ipv6-spec-08 February 23, 2007".

[14] R. Wakikawa, "Home Agent Reliability Protocol", draft-ietf-mip6-hareliability-02.txt (work in progress), July 2007

[15] Jue, J. and D. Ghosal, "Design and Analysis of a Replicated Server Architecture for Supporting IP-Host Mobility," Cluster Computing Special Issue on Mobile Computing, vol. 1, issue 2, pp. 249-260.

[16] Vasilache, J., and H. Kameda, "Threshold-Based Load Balancing for Multiple HAs in Mobile IP Networks," Telecommunications Systems, vol. 22, issue 1-4, pp. 11- 31, January-April, 2003.

[17] Heissenhuber, F., Fritsche, W., and A. Riedl, "HA Redundancy and Load Balancing in Mobile IPv6," in Proc. 5th International Conf. Broadband Communications,Hong Kong, 1999.

[18] Narten, T., Nordmark, E. and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)," RFC 2461, December 1998. Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification,"IETF- RFC 2460, December 1998

[19] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture," RFC 2373, July 1998.

[20] Kent, S. and R. Atkinson, "IP Authentication Header," RFC 2402, November 1998

[21] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)," RFC 2406, November 1998

[22] Deng, H., Zhang, R., Huang, X. and K. Zhang, "Load Balance for Distributed HAsin Mobile IPv6," IETF Draft, draft-deng-mip6-ha-loadbalance-00.txt (work in progress), November 2003.

[23] Droms, R., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," draft-ietfdhc-dhcpv6-28 (work in progress), November 2002.

[24] CASE, J., FEDOR, M., SCHOFFSTALL, M. AND DAVIN, J. Simple network management protocol (SNMP).

[25] FU, Z.H.L.J.W.W.M. AND ZHONG, Y.P. A Novel WSDM-based Content Service Status Monitoring Scheme in Content Federation Architecture.

[26] PULTORAK, D. Microsoft Operations Framework (MOF). The Guide to IT-Service Management 1, 190-203.

[27] KEPHART, J., CHESS, D., CENTER, I.B.M.T.J.W.R. AND HAWTHORNE, N. 2003. The vision of autonomic computing. Computer 36, 41-50.

[28] DOBSON, S., DENAZIS, S., FERNÁNDEZ, A., GAÏTI, D., GELENBE, E., MASSACCI, F., NIXON, P., SAFFRE, F., SCHMIDT, N. AND ZAMBONELLI, F. 2006. A survey of autonomic communications. ACM Transactions on Autonomous and Adaptive Systems (TAAS) 1, 223-259.

[29] GU, X., STRASSNER, J., XIE, J., WOLF, L. AND SUDA, T. 2008. Autonomic Multimedia Communications: Where Are We Now? Proceedings of the IEEE 96, 143-154.

[30] Agarwala, S., Chen, Y., Milojicic, D., and Schwan, K. 2006. QMON: QoS- and Utilityaware monitoring in enterprise systems. In Proceedings of the 3rd IEEE International Conference on Autonomic Computing (ICAC). Dublin, Ireland.

[31] Agrawal, D., Calo, S., Giles, J., Lee, K.-W., and Verma, D. 2005. Policy management for networked systems and applications. In Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management. 455–468.

[32] Bennani, M. N. and Menasce, D. A. 2005. Resource allocation for autonomic data centers using analytic performance models. In Proceedings of the Second International Conference on Autonomic Computing (ICAC). 229–240.

[33] Bhola, S., Astley, M., Saccone, R., and Ward, M. 2006. Utility-aware resource allocation in an event processing system. In Proceedings of 3rd IEEE International Conference on Autonomic Computing (ICAC). Dublin, Ireland, 55–64.

[34] Candea, G., Kiciman, E., Zhang, S., Keyani, P., and Fox, A. 2003. JAGR: An autonomous self-recovering application server. In Proceedings of the Autonomic Computing Workshop. 168–177.

[35] Garlan, D. and Schmerl, B. 2002a. Exploiting architectural design knowledge to support selfrepairing systems. In Proceedings of the 14th international conference on Software engineering and knowledge engineering.

[36] Guo, H. 2003. A bayesian approach for autonomic algorithm selection. In Proceedings of the IJCAI workshop on AI and autonomic computing: developing a research agenda for selfmanaging computer systems. Acapulco, Mexico.

[37] Littman, M., Nguyen, T., and Hirsh, H. 2003. A model of cost-sensitive fault mediation. In Proceedings of the IJCAI workshop on AI and autonomic computing: developing a research

[38] agenda for self-managing computer systems. Acapulco, Mexico.

[39] Lymberopoulos, L., Lupu, E., and Sloman, M. 2003. An adaptive policy-based framework for network services management. In Journal of Network and Systems Management. Vol. 11. Special Issue on Policy-based Management.

[40] Strowes, S., Badr, N., Dulay, N., Heeps, S., Lupu, E., Sloman, M., and Sventek, J. 2006. An Event Service Supporting Autonomic Management of Ubiquitous Systems for e-Health. In

[41] Proceedings of Intl. Workshop on Distributed Event-Based Systems.

[42] R. Koodli, Ed, “Fast Handovers for Mobile IPv6,” IETF-RFC (Proposed Standard) 4068, August 2005.

[43] Ivov, E. AND Noel, T. 2006. An Experimental Performance Evaluation of the IETF FMIPv6 Protocol over IEEE 802.11 WLANs. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC’06,* Anonymous , 568–574.

[44] Kempf, J., Wood, J. AND FU, G. 2003. Fast mobile IPv6 handover packet loss performance: measurement for emulated real time traffic. *2003 IEEE Wireless Communications and Networking, 2003.WCNC 2003* 2,

[45] Pack, S. AND ChoiI, Y. 2003. Performance analysis of fast handover in mobile IPv6 networks. *Lecture notes in computer science* 679-691.

[46] Koodi, R. AND Perkins, C.E. 2001. Fast handovers and context transfers in mobile networks. *ACM SIGCOMM Computer Communication Review* 31, 37-47

[47] Hangino, J. AND Snyder, H. 2001. IPv6 multihoming support at site exit routers. Request for Comments 3178,

[48] Johnson, D. AND Deering, S. Reserved IPv6 subnet anycast addresses, IETF RFC 2526, March 1999.

[49] Nikander, P., Arkko, J. AND Henderson, T. 2004. End-Host Mobility and Multi-Homing with Host Identity Protocol. *draft-ietf-hip-mm-00 (work in progress)*

[50] Huston, G., Wasserman, M., "A Summary of Proposals to Support Multihoming for IPv6", IETF Internet Draft draft-huston-multi6-proposals-00.txt, June 2004.

[51] Huitema, C., Draves, R., "Host-Centric IPv6 Multihoming", IETF Internet Draft draft-huitema-multi6-hosts-01.txt, June 2002(Work in progress)

[52] Teraoka, F., Ishiyama, M., Kunshi, M., and Shionozaki, A., "LIN6: A Solution to Mobility and Multi-Homing in IPv6", IETF Internet Draft draft-teraoka-ipng-lin6-01.txt, August 2001

[53] Mike O'Dell, " 8+8-An Alternate Addressing Architecture for IPv6", IETF Internet Draft, draft-odell-8+8-00.txt

[54] Stewart, R., "Stream Control Transmission Protocol", IETF RFC 2960, October 2000

[55] Dunmore, M. 2005. *Evaluation of Multihoming Solutions*

[56] Ohta, M., "The Architecture of End to End Multihoming", IETF Internet Draft draft-ohta-e2e-multihoming-02.txt, July 2001

[57] Bragg, N., "Routing Support for IPv6 Multi-homing", IETF Internet Draft draft-bragg-ipv6-multihoming-00.txt, November 2000

[58] Bates, T., Rekhter, Y., "Scalable Support for Multihomed Multi-provider Connectivity", IETF RFC 2260, January 1998

[59] Nordmark, E., Bagnulo, M., “Shim6: Level 3 Multihoming Shim Protocol for IPv6”, IETF Internet Draft draft-ietf-shim6-proto-12.txt, February 2009.

[60] Crawford, M., “Router Renumbering for IPv6”, IETF RFC 2894, August 2000

[61] Aura, T., “Cryptographically Generated Addresses”, IETF RFC 3972, March 2005

[62] Bagnulo, M., “Hash based Addresses”, IETF Internet Draft draft-ietf-shim6-hba-04, October 2007

[63] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E., “Session Initiation Protocol”, IETF RFC 3261, June 2002

[64] Ylitalo, J., Torvinen, V., Nordmark, E., "Weak Identifier Multihoming Protocol Framework (WIMP-F)" IETF Internet Draft draft-ylitalo-multi6-wimp-01, June 2004

[65] Nordmark, E. “Multihoming without IP Identifiers”, IETF Internet Draft draft-nordmark-multi6-noid-01.txt, October 2003.

[66] “Carrier Grade Voice over IP” Daniel Collins - McGraw-Hill Education

[67] Qazi Bouland Mussabbir, Wenbing Yao, Zeyun Niu, and Xiaoming Fu “Optimized FMIPv6 Handover using IEEE 802.21 MIH Services,” MobiArch, ACM, 2006

[68] Camp, T., Boleng, J., & Davies, V. (2002). A survey of mobility models for ad hoc network research.*Wireless Communications and Mobile Computing, 2*(5), 483–502.