NOTE TO USERS

This reproduction is the best copy available.

MOBILITY MANAGEMENT FRAMEWORK FOR LOCAL MOBILITY

by

BRYAN HARTWELL

Bachelor of Engineering, Ryerson University

Toronto, 2004

A thesis
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Masters of Applied Science
in the Program of
Computer Networks

Toronto, Ontario, Canada 2004

Borrower's Page

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Mobility Management Framework for Local Mobility, Bryan Hartwell, M.A.Sc, Computer Networks, Ryerson University, Toronto 2004.

IP mobility solutions allow mobile nodes to roam while retaining connectivity to the internet. However, as these solutions evolve, mobile node implementations continue to undergo modification. Since mobile nodes represent hundreds of thousands of hosts worldwide, deploying new mobility protocols will become expensive.

The main objective of this project was to design a framework that decouples the mobile node from route repair, which reduces the implementation and deployment time of new solutions. The proposed framework reengineers existing IP mobility protocols in order to facilitate the transition for network administrators. The second objective of the project was to provide a prototype of the framework to gain acceptance for our design within the Internet community. The result of this work is a mobility management framework that not only reduces the effects of deployment, but also provides a standard interface to the mobile node.

Acknowledgements

I would acknowledge Dr. Jaseemuddin and Dr. Ma for introducing and guiding me through the world of computer networks and mobility.

I also acknowledge the Go-Core project team for maintaining, and continuing to develop, the publicly available *Mobile IP for Linux*.

Contents

List of Figures

Chapter 1 Introduction

IP mobility solutions hide the mobility of a host from upper layer applications. Hosts that experience mobility are referred to as mobile nodes, and are tightly coupled to the IP mobility solution. There are currently several alternative mobility solutions within the IP suite though a mobile node will be associated with only one. The purpose of this thesis is to design and implement a Mobility Management Framework that hides the operation of the local IP mobility solution from that of the mobile node. This report describes the purpose and operation of that Framework.

To achieve this, this document is structured as follows: Chapter 1 provides an overview of why a framework must be developed, as well as an overview of the actual mobility management framework proposed by the author. Chapter 2 is a literature review of existing mobility solutions, including proposed standards as well as works in progress (i.e. internet drafts). It provides a technical explanation of existing mobility solutions. Chapter 3 describes the design of our proposed Framework in detail.. Chapter 4 explains our implementation of the Framework by describing some key software modules, and also discusses the test case. We also included the handover protocol as an internet draft in Appendix A, which we intend to submit for standardization.

Section 1.1 provides an explanation of what a Mobility Solution is. An overview of how IP Mobility Solutions work is given in Section 1.2, and an explanation also is given for why a framework needs to be developed for these protocols. Section 1.3 gives a simple overview of the Mobility Management Framework.

1.1 Mobility Solutions

In IP, every host is uniquely identified by (at least) one unicast IP address that is used as a node ID and location ID. Hosts are connected to, as well as located by, routers. As such the router must know of all the hosts that are directly connected to it. If a router receives an IP datagram whose destination is not on one of its interfaces, then the router must know of the next immediate router along the best path to that host.

The router learns of this path through IP routing protocols, such as OSPF [2]. Yet this would require every router knowing the path to every unicast address; the address space of IPv4 is 2^{32}, while IPv6 is significantly bigger. Instead hosts form subnets that share a common address prefix. Thus routers need only know the prefix of a subnet, rather than the address of every host within that subnet. Figure 1-1(a) illustrates the path a packet follows from a correspondent node to the host using the method described above.

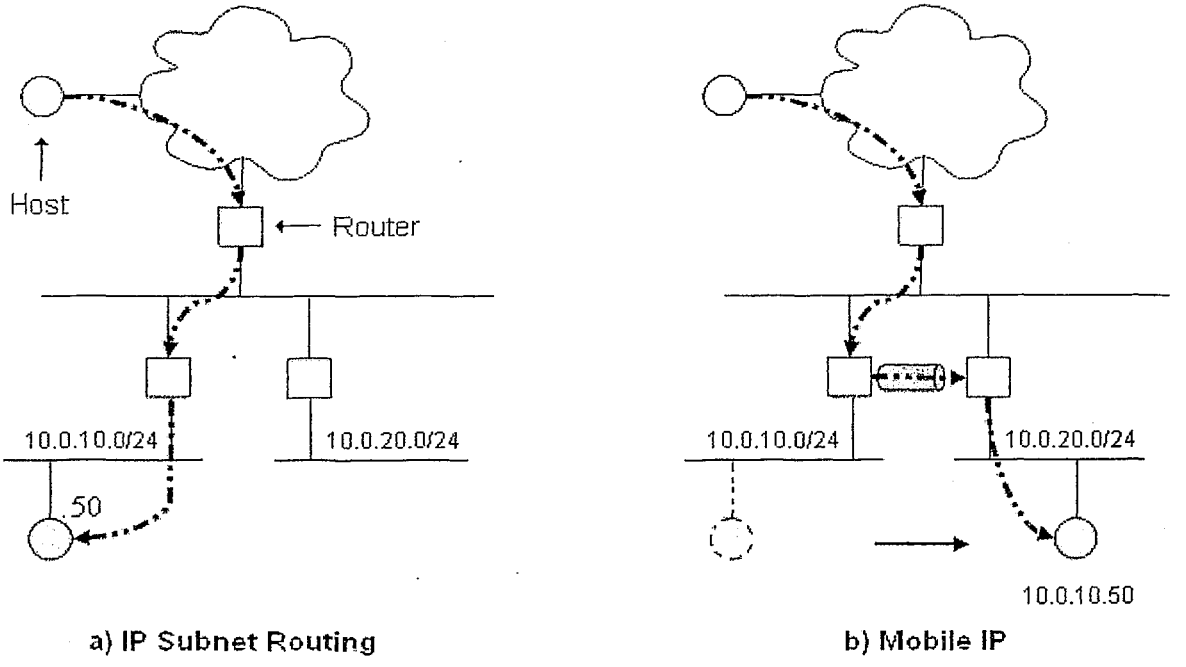


Figure 1-1: Routes to the Mobile Node

However, consider a host that can move to new subnets while in use. Using prefix-based routing this "mobile node" faces one of two problems:

1. Change its IP address to reflect the current subnet's prefix, which terminates all of its existing connections. Furthermore, other hosts will be unable to initiate contact with the mobile node without being informed of its new address.
2. Keep its current IP address. Packets bound for the mobile node will be delivered to the wrong subnet, and dropped upon arrival.

A mobility solution is a means of directing packets to the mobile node's current point of connection. While mobility solutions occur in a wide variety of designs, the Internet Engineering Task Force (IETF) has adopted IP mobility solutions. These mobility protocols are similar to routing protocols by allowing specific routers to know where the mobile node is using IP addresses. They are the foundation for the Mobility Management Framework, and in the remainder of this document the term "mobility solution" refers to these protocols.

1.2 IP Mobility Protocols

While a number of protocols are currently under development within the IP mobility suite, the current defacto standard is Mobile IP (MIP) [3] [4]. The basic premise of Mobile IP is that a mobile node maintains two IP addresses: a home address and a care-of address. Mapping one address to the other is referred to as Route Repair.

1.2.1 Route Repair

The mobile node uses its home address as a source address in all of its correspondence. This is a normal unicast address, such that when the mobile node is "home" packets can be delivered to it without the need of a mobility protocol. Note that correspondent nodes sending packets to the mobile node will direct them to the home address, whether or not the mobile node is physically there to receive them.

When the mobile node moves it acquires a care-of address. This is an IP address that allows the mobile node to be located in its current subnet. The mobile node sends this address to a router on its home subnet, called a home agent. While the mobile node is roaming, the home agent will encapsulate all IP datagrams bound to the mobile node and send it to the care-of address. This form of IP-in-IP is referred to as tunneling; the encapsulated IP datagram allows the mobile node to preserve its higher-layer connections. The path this tunnel enables is displayed in Figure 1-1(b). Every time the mobile node moves it registers a new care-of address with the home agent, who repairs the route to the mobile node by mapping the care-of address to the home address.

1.2.2 Transient Route Repair

It is also worth noting that when a mobile node moves, which is called a handover, there is a period during which it is unreachable. This is due to two reasons, the first being that it takes a certain amount of time to establish a physical association at the new link, however during this time the mobile node is completely unreachable. The second reason is that packets will continue to be delivered to the mobile node's previous point of connection until it successfully registers with the home agent. This registration initiates a primitive form of route repair as discussed above; the path to the mobile node is updated to reflect its new point of connection.

Due to the handover latency the mobile node is unable to receive or send packets. However there is a sub-category of mobility solutions called Fast Handovers that attempt to reduce this latency. They do this by providing transient routes local to the mobile node until it has completed registering its new care-of address. These routes are called transient because they have very short lifetimes and cannot be refreshed.

1.2.3 The Mobile Node

An important distinction between routing and mobility protocols is that only the former is transparent to the end node. That is, the host (fixed or mobile) is unaware of whether the local network uses OSPF, EIGRP, or some other protocol for route distribution. However mobile nodes must be involved in the mobility solution. Furthermore, if the mobile node does not support the same mobility protocol as its current point of attachment, it is the same as having no mobility solution at all.

It is the mobile node's responsibility to detect when it has moved. As such it also the mobile node's responsibility to register its care-of address(es) to instigate route repair. Thus with each new method of mobility, it is the mobile node's responsibility to initiate dialogue with any new node-types defined therein. The significance of this is that deploying a new (mobility) protocol, or simply changing any part of that process, would involve upgrading not only routers but also all of the dependent hosts. This would be costly, as well as deny service to end users until they can upgrade. Some vendors may even implement mobility in hardware, and these nodes would become obsolete. This results in the need for a process that minimizes changes to the mobile node.

1.3 Motivation and Overview the Framework Design

In the previous section we have introduced three actions that must occur to achieve mobility: handover, registration and route repair. However, current mobility protocols make little to no distinction between these processes. As a consequence mobile nodes are tightly coupled with the

network to an extent that it is involved in the route repair process, which is exclusively a network activity controlled, optimized and deployed at the discretion of the wireless operators. The above processes and their roles in IP mobility can be illustrated through an IP mobility model as shown in Figure 1.2. If we were to limit ourselves to current practice, this model would appear as a single block as shown in Figure 1.2(a). To achieve flexibility of deploying and developing route repair process independent to mobile nodes we need to design a framework that decouples the mobile node from the rest of the system. We can achieve this by providing the mobile node with a standard interface. Thus the operation of the processes on either side of the interface remains transparent to the other. This allows the mobile node to participate on the control path without having to be cognizant of how the forwarding path is established.

> Some mobility solutions create hierarchies of registration to reduce the overall registration latency. However the mobile node is aware of the entire hierarchy. Thus the process of registration remains within a single "layer", along with the obstacle of deploying new solutions.

Figure 1.2(b) shows the proposed Framework. It structures Mobile IP into three major processes:

- Handover
- Registration
- Route Repair.

Mobile IP may refer to the entire process as a handover. However it may also strictly refer to the handover event, i.e. when the mobile node changes its point of connection to a new access router and performs auto-configuration to establish connectivity on that link. This requires the mobile node to acquire a new care-of address which it must register with the appropriate mobility agents.

> In MIPv6, to register with the home agent the mobile node must send a Binding Update. Thus nodes that participate in the registration process within the framework are referred to as binding agents. This distinction is necessary because the framework's access router is a binding agent in both IPv4 and IPv6, but it is a tunnel-endpoint only in IPv4.

The Registration mechanism processes this request from the mobile node. We further refine registration into Home and Local Registration, as well as Fast Handovers. The significance of

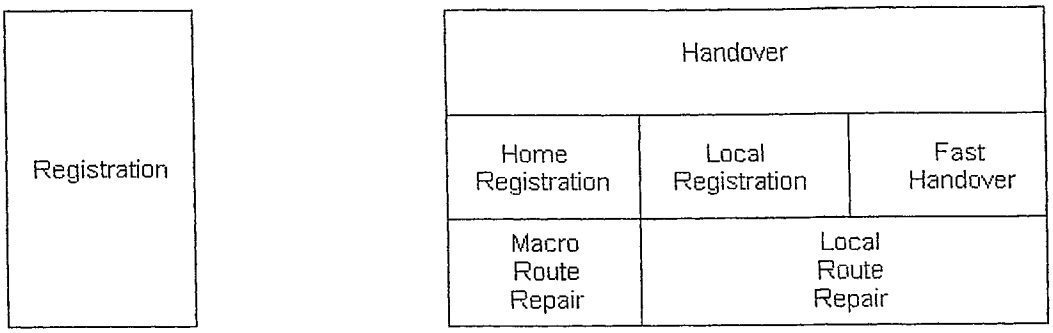


Figure 1-2: Mobility Processes

these individual processes will become apparent in the following chapters. Registration typically results in the act of route repair. Route repair is the establishment of a forwarding path to the mobile node. Improvements upon Mobile IP typically revolve around refining the route repair process. Thus relegating route repair to a separate activity eliminates the need to update the mobile node every time a new method of route repair is devised. As such, rather than dictate the design of this process, the framework reserves a place for route repair within the model but leaves its operation open to the requirements of the network administrator.

From Figure 1.2(b) we can see that Route Repair can be modified without affecting the Handover. This decoupling is desirable since Route Repair roughly corresponds to a few routers, while Handover may represent many mobile nodes. This segregation of processes achieves the objective of the framework.

1.4 Contributions

The problem statement was identified and the outline of the framework is provided by [1]. In this thesis we present the detail design of the Mobility Management Framework to address this problem, which the Framework accomplishes by decoupling the mobile node from route repair. As such the mobile node would receive upkeep only directly necessary to its function. This design includes mobile node and foreign agent operation. In this framework we focus on local mobility management and use Mobile IP for global mobility across multiple domains. We also produced two reference designs by modifying existing IP mobility solutions to operate within the Framework.

The second important contribution made in this thesis is a prototype implementation of the Framework in order to verify the success of the design. This would allow other researchers to

test the properties of the Framework. Towards this end the author has produced code for the mobile node, as well as the Local Registration protocol supported by the Foreign Agent. Using a simulation of the Route Repair API, the Framework operation was verified.

1.5 Summary

After each handoff a mobile node must register a care-of address with its home agent in order to continue receiving packets. The home agent intercepts packets bound for the home address and tunnel's the packet to the care-of address.

In current mobility solutions, the mobile node is tightly coupled with the domain solution. If they are not compatible, the mobile node will not be able to receive packets while away from its home subnet. Since the mobile node must also be modified to implement a new protocol, deploying more efficient solutions is time consuming and costly, if not altogether impossible.

The framework serves to make the domain solution transparent to the mobile node by separating the mobile node from the route repair process. This approach allows new domain solutions to be easily deployed.

Chapter 2 Mobility Solutions

This chapter provides a conceptual understanding of existing mobility solutions. The purpose of this is two-fold:

- To understand the need for a management framework
- The framework must be able to support the needs of existing and future solutions.

Every mobility solution uses a care-of address to "tunnel" the IP packet to the mobile node. Section 2.1 explains what these two terms mean, while the following sections focus on how individual mobility solutions register the care-of address with the home agent. Section 2.2 starts with basic Mobile IP. This section describes both MIPv4 and MIPv6 since concepts are drawn from both proposed standards. However we are only interested in a Mobility Management Framework for IPv6, therefore the rest of this document focuses on concepts and terminology used in IPv6. Sections 2.3 to 2.5 refer to works in progress; these are solutions that are meant to either work with, or replace, Mobile IP. Hierarchical Mobile IP and Multicast-based Mobility both use Local Registration, but in very different manners. Fast Mobile IP works with Layer 2 to hide handover latency.

The remaining two sections discuss IP mobility based solutions that are not currently supported by the Internet Engineering Task Force (IETF). Though the type of services they provide are not distinct from those described in the preceding sections, they demonstrate the diversity of mobility solutions. Since this chapter is a literature review of IP mobility protocols, readers who are familiar with this subject may wish to proceed to Chapter 3.

2.1 Tunnels and Care-of Addresses

Every host has a regular IP (home) address that is globally routable and to which all its correspondents direct their packets, including mobile nodes. The problem occurs when a mobile node connects to a new subnet that uses a different network prefix from that of its home subnet. To get around this the mobile node acquires a care-of address (CoA) that has a network prefix appropriate for that subnet. This address is then stored on a router, known as the home agent, that is located on the mobile node's home network. The home agent will intercept any packets destined for the mobile node's home address and forward them to the mobile node's care-of address. This process is known as tunneling; the IP packet from the home agent to the mobile node's care-of address contains the original unaltered IP packet as data, complete with the original header. This is the basic premise behind all mobility solutions.

While there are several different "types" of care-of addresses, the first two were introduced by MIPv4:

- Co-located Care-of Address: A unique, globally routable, address assigned to the mobile node.
- Care-of Address: A globally routable address assigned to one of the Foreign Agent's interfaces.

Figure 2-1 depicts tunnels for their respective care-of addresses. Recall that the IP packet for the mobile node is carried within the IP packet to the care-of address. When the mobile node uses a co-located care-of address the tunnel exits directly at the mobile node. If a care-of address is used, the tunnel exits at the foreign agent; the foreign agent must then deliver the packet to the mobile node.

> The foreign agent is a node found only in IPv4 mobility solutions. It is an access router located on the subnet that the mobile node is visiting. It participates in registration and may serve as a tunnel end-point. MIPv6 does not use foreign agents, although they are reintroduced by the framework for IPv6.

The different types of care-of addresses (CoA) may be referred to simply as a care-of address, such as in MIPv6 which uses only co-located care-of addresses. Readers must remain aware of the context in which they view the term, least they misunderstand how the address is being used.

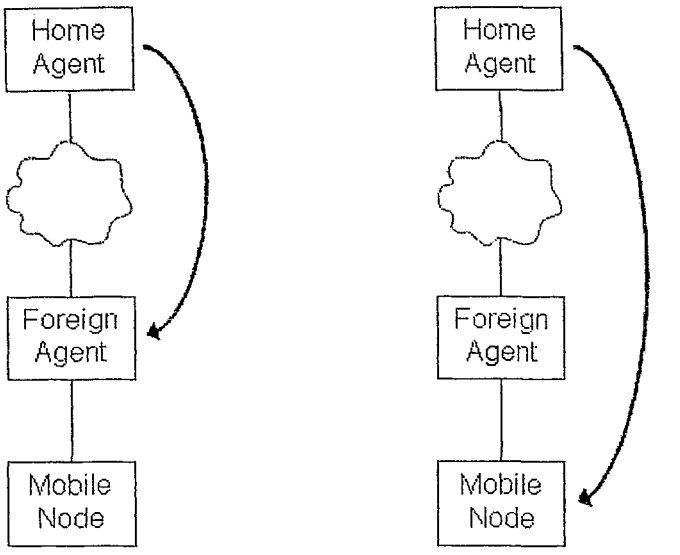


Figure 2-1: Tunnels and Care-of Address

The only significant difference between mobility solutions is how the care-of address is registered with the home agent, and the type of address that is registered. The remaining sections discuss various mobility protocols.

2.2 Mobile IP

The first proposed standard to address mobility was Mobile IPv4 [3]. The current standard is in fact the fourth RFC for that protocol; the first having been proposed almost 10 years ago, and with yet another revision being developed. A standard for IPv6 [4] is also currently in progress, and was recently accepted for the experimental phase.

2.2.1 MIPv4

Recall that a co-located care-of address is a unique address assigned to the mobile node, and packets are delivered directly to the mobile node via that care-of address. However in IPv4 the address space is limited; a wireless access network which may expect a high volume of visiting nodes may not be able to provide a unique, globally routable, address to every mobile node. If a single node could intercept packets for several local mobile nodes, then the address space could be preserved. This node is called the foreign agent.

The Foreign Agent is a router on the subnet that the mobile node is visiting. It will serve as a tunnel end-point for all mobile nodes on that subnet. That is, the mobile node will use an IP address of the foreign agent itself as its CoA. Any packets destined for the mobile node will be

delivered to the Foreign Agent, who inspects the internal IP header and delivers it directly to the Mobile Node.

A co-located CoA is acquired through statefull address configuration, such as DHCP [5]. The CoA is acquired from the Router Advertisement [6]. Router Advertisements are ICMP messages that are periodically broadcast over the subnet; they contain extra information about which node is the acting router on that subnet.

In MIPv4 these messages are called Agent Advertisements because the ICMP message also contains a Mobility Agent Advertisement option. This option defines whether the agent is a home or foreign agent, the maximum lifetime a requested registration could be valid for, whether registration is required, and some other useful values. This option will also contain one, or more, (care-of) addresses supported by the router. The mobile node also uses these advertisements to detect the fact that it has moved. If it receives an Agent Advertisement from a new router, then the mobile node must register a new CoA with the home agent.

Since the foreign agent may detunnel packets for a CoA, the mobile node will first send its Registration Request to the foreign agent. If there is anything wrong with the request, the foreign agent will immediately deny the request; otherwise the foreign agent appends some additional information and passes the request to the home agent. The home agent also determines whether the request is acceptable and returns a reply to the foreign agent. The foreign agent will also examine the response from the home agent, and if it does not satisfy the foreign agent's requirements, it will send a negative Registration Reply to the mobile node.

The mobile node can only resume communication using its new CoA after it has received a positive registration reply. If the mobile node uses its previous CoA then replies will be delivered to its old point of connection, and not to the mobile node. If the new CoA was rejected then replies would reach a similar dead-end. Recall that a co-located CoA terminates directly at the mobile node. Thus the registration request for this address can be sent straight to the home agent, bypassing the foreign agent. However the foreign agent can demand that the mobile node "register" with the foreign agent. This allows the foreign agent to exercise security and flow control on the subnet.

2.2.2 MIPv6

The primary difference between MIPv4 and MIPv6 is that the latter eliminates foreign agents. This is because in IPv6 the address space is virtually boundless, every mobile node can have one or more co-located CoAs without affecting the available address space. In fact the mobile node can automatically create a CoA based upon the network prefix and its L2 address with a negligible chance of address collision. This method of address configuration is known as "stateful address autoconfiguration". Typically a CoA is formed based upon a 64-bit network prefix and a hash of the mobile node's MAC address. Since there is some possibility of address

collision, the mobile node must perform Duplicate Address Detection (DAD) to confirm that it's chosen CoA is valid.

The mobile node acquires this prefix during Neighbour Discovery [7]. This is the IPv6 version of Router Advertisements (RA). Routers broadcast these for movement detection and address autoconfiguration. The router advertisements are slightly modified to include a few extra flags, such as whether the advertised node can act as a home agent. Otherwise the "foreign agent" is not involved in the MIPv6 registration process, and as such these nodes are referred to as access routers because they have no explicit mobility functionality. The mobile node simply creates a Binding Update and sends it directly to the home agent. Recall that in MIPv6, every CoA is a co-located CoA.

2.2.3 Correspondent Nodes

Correspondent nodes are provided for in IPv4 in a separate draft, while the MIPv6 draft discusses them directly. A correspondent node is precisely that: a node that corresponds with the mobile node. However in Mobile IP a correspondent node refers to a node that is mobile-aware. Using framework terminology, the correspondent node is a binding agent.

The advantage of including the correspondent node on the control path is the reduction of the hop count in the reverse path. Though many implementations often use "reverse tunneling", this is due to the fact that the packet must first travel to the home subnet, and only then towards the mobile node. If the correspondent node could receive binding updates from the mobile node then it could deliver packets directly to the mobile node. Figure 2-2 depicts both correspondence with a non-mobile-aware correspondent node (triangle tunneling), and with a mobile-aware correspondent node.

However, before a mobile node may bind with a correspondent node it must complete a process called Return Routability. This procedure reduces the security risk of a third party pretending to be either the mobile or correspondent node, as well as assuring the correspondent node that mobile node can actually be contacted at both its home and care-of addresses. This is accomplished through the exchange of four packets between the two nodes. The mobile node will start the process by sending a Home Init Test and Care-of Init Test message. The former is reversed-tunneled from the mobile node's home address via the home agent, while the later is sent directly to the correspondent node from the mobile node's care-of address.

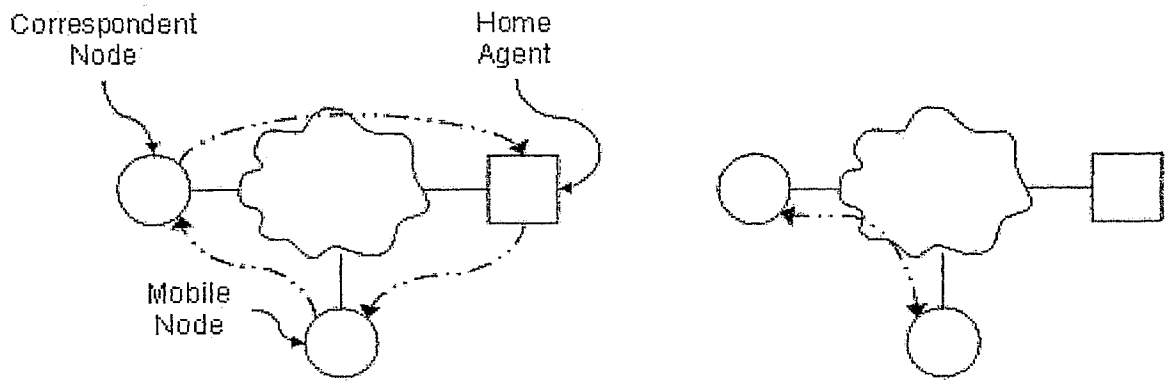


Figure 2-2: Correspondent Nodes

Each message initially sent by the mobile node will contain a unique random number called a cookie. The correspondent node will respond to each message individually, which must contain the appropriate cookie as well as a keygen token. The cookies are to confirm to the mobile node that the replies came from the correspondent node, while the tokens are used in authenticating the binding updated. Once this exchange of messages is complete the mobile node may then send a binding update to the correspondent node. Only when the correspondent node has received this binding update may it create a relationship with the mobile node.

The purpose of the return routability procedure is twofold. The first is to ensure that the correspondent node has a path to the mobile node through either the home agent or directly to its care-of address. The second is to restrict potential sources attacks to nodes that are on the path between the mobile and correspondent node.

2.3 Hierarchical Mobile IP

Quite often the home agent will be geographically distant from the mobile node. The longer the distance, the longer it takes to register a new care-of address. This can have a significant effect on the mobile node, particularly when Quality of Service is of importance. In addition to the registration latency there is also the consumption of bandwidth. Every time a mobile node moves it must update its home agent as well as all of its correspondent nodes. There are also proposals to provide multiple home agents and care-of addresses. This high volume of bursty traffic is not ideal for a wireless medium or the internet in general.

Local mobility management (LMM) [8] is a category of mobility solutions that attempt to both reduce registration latency and overall signaling. These protocols provide local route repair in addition to that provided by the home agent. This means overall shorter registration periods and may also result in a reduced number of registration messages.

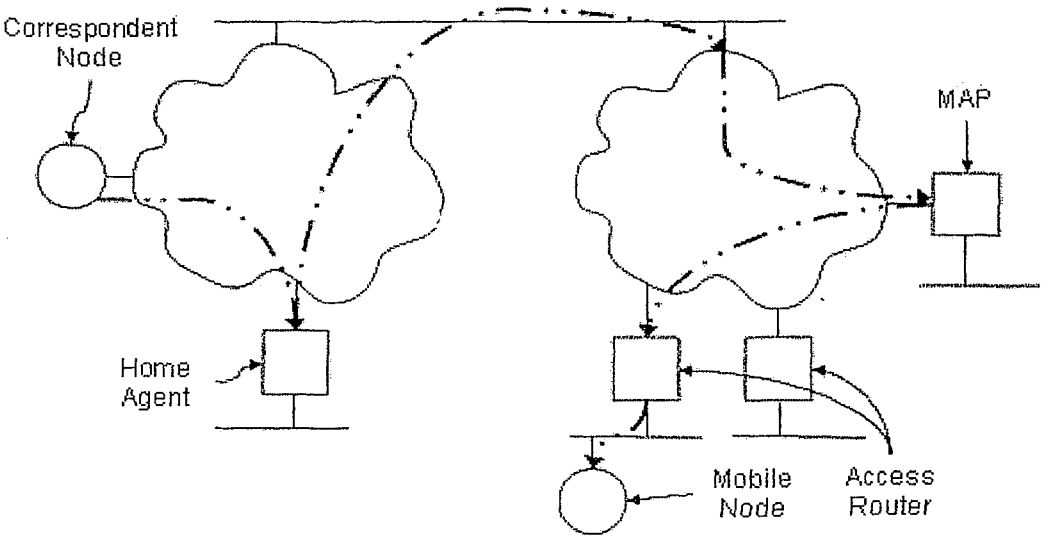


Figure 2-3: Hierarchical Mobile IP

Consider a situation where, despite movement of the mobile node, there was a care-of address that corresponding nodes could always deliver packets to. They would (almost) never need to be updated. Naturally, if interpreted literally, if this was possible then we would not need Mobile IP. Yet what if there was a mobility agent that provided services similar to that of a home agent, and was local to the mobile node? The effects of triangle routing would be minimized, and registration latency would be decreased. Also, since a binding update is sent only to the "local mobility agent", overall signaling and consumption of wireless-bandwidth is reduced.

Hierarchical Mobile IP (HMIP) [9] is a protocol which allows for local registration. It defines a Mobile Anchor Point (MAP) which binds a mobile node's regional care-of address to its on-link care-of address. Correspondents tunnel packets to the regional care-of address, which is located at the MAP. The MAP intercepts these packets and tunnels them to the mobile node's on-link care-of address. Figure 2-3 depicts the path a packet takes when a mobile node is registered with a MAP.

> HMIP is just one possible method of local mobility management (LMM). There is a set of guidelines for LMM [8], but they are geared towards IPv6. There was an IPv4 proposal, called Regional Registration; however it has fallen by the wayside as the focus shifted towards IPv6. In fact, even LMM implementations in IPv4 have been referred to as Hierarchical Mobile IP.

The MAP is a router, and as such advertises itself with router advertisements. It includes a MAP option which primarily indicates the prefix of a network adjacent to the MAP. A MAP may be positioned anywhere in the network; routers which receive the MAP option include it into its

own RAs, which eventually reach the mobile node. A MAP domain is the "area", I.E. a group of routers and mobile nodes, which will receive the MAP option. MAP domains may overlap, such that some access routers may forward multiple MAP options in their router advertisement. Thus the mobile node will receive a router advertisement from the access router which contains the standard options of Mobile IP, along with one or more MAP options.

Using the access router's network prefix, the mobile node creates a on-link care-of address as described in MIPv6. In the same manner the mobile node will also create a regional care-of address using a network prefix contained in the MAP option. The mobile node sends a binding update to the MAP to register both addresses. Any packets arriving at the MAP for the regional care-of address will be intercepted by the MAP and sent to mobile node's on-link care-of address. Afterwards the mobile node sends a binding update to the home agent and correspondent nodes about its regional care-of address.

From then on, whenever the mobile node moves all it needs to do is send a binding update to the MAP to register a new on-link care-of address. The MAP is closer so the registration latency is reduced, and the mobile node deals with a single binding agent which reduces signaling. Yet if the mobile node must acquire a new MAP then it must register the new regional care-of address with its home agent. One final thing to consider is that since the on-link care-of address is "hidden" from the mobile node's correspondents, this address could reside within private space. This property was of particular importance to IPv4 because address space was limited, but it can also provide some flexibility to IPv6 networks.

2.4 Multicast-based Mobility

We have seen solutions that provide the mobile node with a care-of address, based upon its current point of connection. However a different approach is host-based routing. That is, rather than routing based upon network prefixes, routing is based upon hosts. Thus each router would require an entry for every host; a packet arriving for the mobile node at the home network would be directed to an adjacent router, and that router would direct the packet to another adjacent router and so on to the mobile node.

There are two major problems with this approach. The first is that a router might be required to contain routes for thousands of hosts, rather than just a few networks. Second, potentially many routers would need to be updated when the mobile node moves. The only benefit is that the mobile's nodes address would not change; there would be no need for a home agent or to update correspondent nodes. Since this method of routing is somewhat similar to multicasting the next sub-section gives a brief overview of this topic, followed by a description of how it can be applied to mobility.

2.4.1 Multicasting

Multicasting is like subscribing to a newspaper. Many people in a given neighbourhood may want to receive the newspaper, but not everyone. Only people that have subscribed to the newspaper will get it. A multicast address is an IP address, with each unique address representing a different "newspaper". Each multicast address is supported by a server, packets sent to the multicast address first go to the server. The server then forwards the packet to adjacent routers that have joined the multicast address.

Hosts subscribe to a specific address by sending a "join" request to a router. Routers propagate the join request towards the multicast server. If a join request arrives at a router that has already subscribed then the join request does not need to propagate any further. Hosts unsubscribe in a similar fashion by sending a prune request to its router.

Figure 2-4 best demonstrates how packets are broadcasted on the multicast address in a network which has both subscribed and unsubscribed hosts. The shaded-shapes are nodes that have joined the multicast address, such that packets sent to this address are only broadcasted to these nodes. Nodes that have not joined the multicast address will not receive these packets.

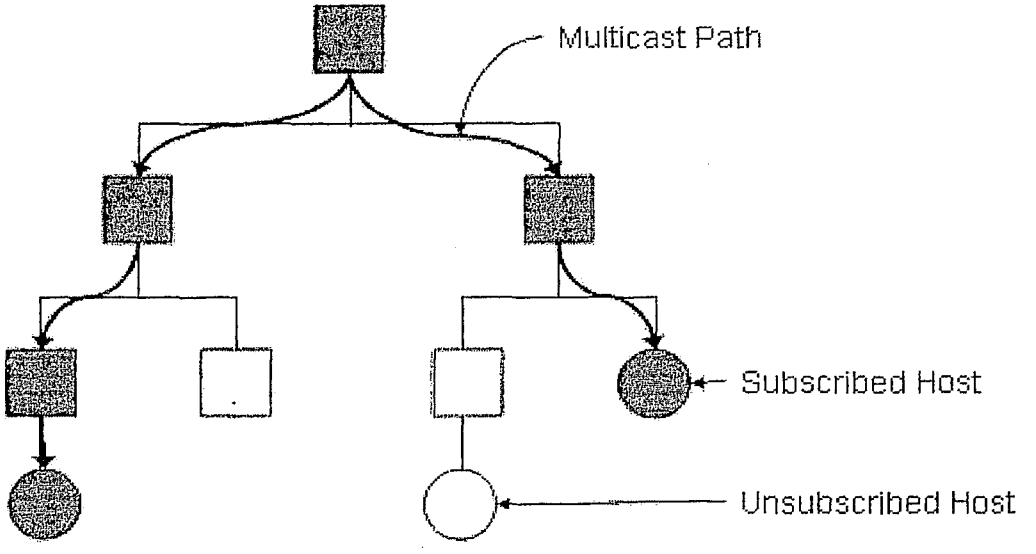


Figure 2-4: A Simple Multicast Network

2.4.2 Multicasting and Mobility

So how does a broadcasting protocol apply to mobility? Consider the method by which new hosts join the multicast address, and then take a look at Figure 2-5. This diagram shows a host subscribed to a multicast address that is moving between routers. To stay subscribed the mobile node must send a join message at the new router, except that this message does not need to

propagate all the way to the server. Now consider that the mobile node is the only subscriber to this multicast address; the multicast address is in effect a co-located care-of address [10]. "Binding updates" are quick because they typically do not need to travel more than a few hops from the mobile node. Also, since the address does not change, the home agent and correspondent nodes do not require binding updates. The mobile node cleans up after itself by sending a prune message to the previous access router, so that packets are delivered only to the mobile node.

Since IPv6 has a large address space, the care-of address can be globally routable. Yet it is also possible that care-of address may reside within private space. The latter can be accomplished by configuring the multicast server to act as a local mobility agent. The mobile node would configure a regional care-of address at the local mobility agent, and use the multicast address as a on-link care-of address. This would preserve the global availability of multicast addresses. The mobile node must still initially register the care-of address with each home agent and correspondent node. Afterwards the mobile node simply initiates route repair; it does not require further registration with any binding agents. Route repair in this case is achieved through prune messages between routers.

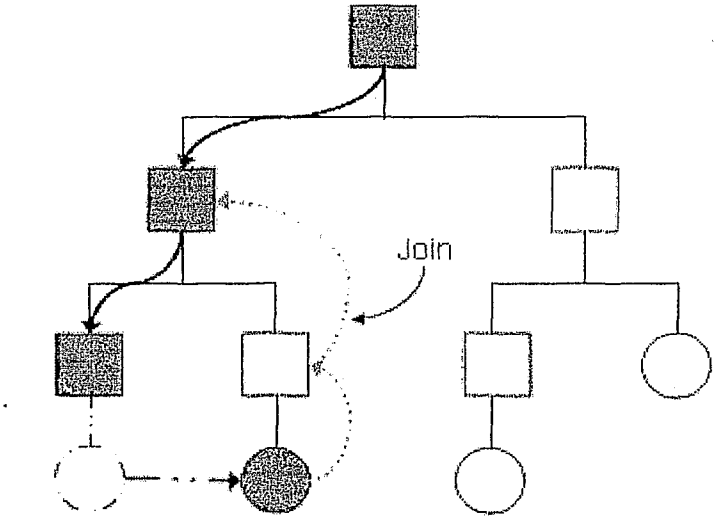


Figure 2-5: Multicast-based Mobility

2.5 Fast Mobile IP

Previous sections discussed mobility solutions that hide the movement of a mobile node from correspondent nodes, with respect to the exchange of data. Also discussed previously was the registration process. After a handover the mobile node must register its new care-of address, and during this period packets may arrive at the previous access router. Thus during the registration

process the mobile node can not receive, or send, packets. There is also the physical handoff itself, which can sometimes take lon[illegible]n registration. Fast Mobile IP (FMIP) [12] attempts to reduce or eliminate this latency.

2.5.1 Fast Handovers

To accomplish this reduced latency the previous access router will intercept packets bound for the mobile node and deliver them to its link on the new access router. The path these packets follow is demonstrated in Figure 2-6. Recall that this path is temporary. This is because sending packets directly to the new care-of address utilizes a shorter path. Secondly, using this method for mobility would likely mean multiple tunnels for every mobile node, consuming unnecessary bandwidth. Thus the fast handover must be independent of the regular binding process, and is referred to as transient route repair.

When the mobile node detects a handover it will send a Fast Binding Update (FBU) to the router it currently connected to. The previous and next access routers will exchange HI and HACK messages which establish the tunnel between. Success is indicated to the mobile node through a Fast Binding Acknowledgement (FBACK) message. If the mobile node detects the handoff after the fact, then the time to establish a tunnel between access routers must be shorter than the time it takes to register a new care-of address. Furthermore, the latency introduced by the handoff itself is not diminished in any way.

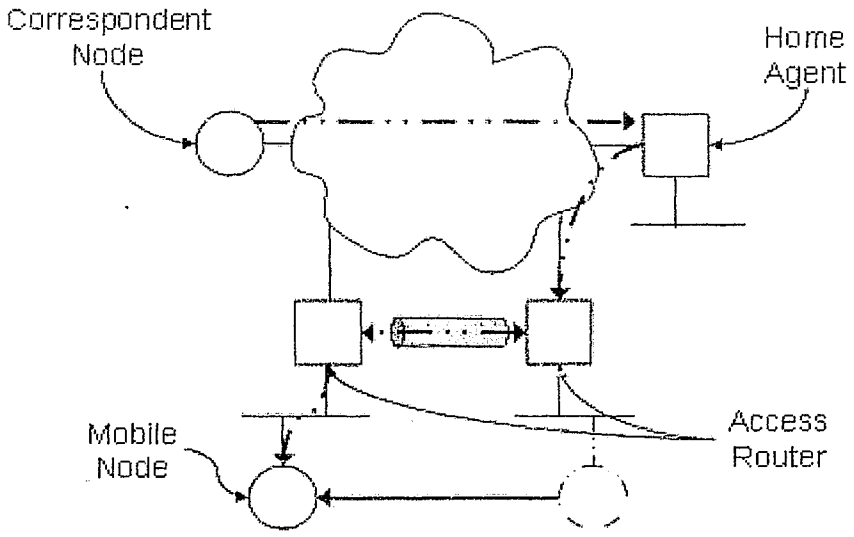


Figure 2-6: Fast Mobile IP

A handover can be predicted, or necessitated, by weakening or growing signal level, or the need to manage bandwidth. Thus a handover can be predicted by the mobile node, previous access router or the new access router. In fact, in cellular networks the handoff is always initiated by a router.

Before the mobile node can send a FBU to the "previous" access router, it must receive a Proxy Router Advertisement. This contains information about the next access router that the mobile node uses to construct the fast binding update and new care-of address. It is important to understand that in order to determine when to send or solicit for a proxy router advertisement, I.E. handover prediction, relies upon Layer 2. The Link Layer must provide triggers to Layer 3 indicating that handoff is about to, or has, occurred. Otherwise Mobile IP must rely on the Router Advertisement to detect movement, and this is not sufficient for handover prediction.

2.5.2 Local Handovers

Fast Mobile IP can work with any mobility solution as it does not interact with the registration process. However, some solutions can benefit from working directly with the fast handover solution. Fast Hierarchical Mobile IP [13] is such a proposal; it specifies that rather than establish a tunnel between the two routers, the mobile node should send the fast binding update to the MAP. The MAP then directs traffic to the new link, bypassing the old access router. Unfortunately this requires the mobile node to engage in additional registration with the domain solution, which contradicts the purpose of the framework. Since a local handover is more efficient than a fast handover, the framework must consider providing an interface for this operation (see Figure 1.2).

2.6 HAWAII

Like HMIP and Multicast-based Mobility (M&M), Handover-Aware Wireless Access Internet Infrastructure (HAWAII) [14] provides local mobility management. However M&M and HAWAII, as well as Cellular IP discussed in the following section, all have a common property that sets them apart from HMIP. They are all routing-based solutions; that is, within the access network they propagate host specific routes rather than tunnel packets.

In HAWAII, wireless access networks are referred as domains. The home domain is where the mobile node's home agent is located, while all other networks are foreign domains. Every domain is connected to the Internet core by a domain root router. Upon powering up the mobile node establishes a route within the domain by sending a path setup message to the domain root router. Each router between the mobile node and the domain root router inspects this message and creates a host-based route for the mobile node.

When the mobile node undergoes a handover it transmits a path update message to the previous access router. All of the routers in between create a host-based entry for the mobile node. Effectively packets get diverted to the mobile node at the cross-over router. This is the closest common router to the mobile node shared by the previous and next access routers. This method is used whether the mobile node is located in its home domain or in a foreign domain. The only difference between the two is that in the latter a care-of address is assigned to the mobile node. This care-of address allows packets to be located in the foreign domain via the domain root router.

HAWAII appears to be very similar to Multicast-based Mobility. In fact the only significant difference is that HAWAII defines its own path setup messages, while M&M uses general purpose join and prune messages.

2.7 Cellular IP

This protocol focuses on mobility within the wireless access network. Home agents and foreign agents are simply the gateway routers to each access network. Thus, similar to HAWAII, locating the mobile node within a network is the same regardless of whether the mobile node is in the home access network or a foreign access network. The first notable difference is that in Cellular IP the care-of address is not co-located. Packets are tunneled to the gateway router using the care-of address. The gateway router detunnels these packets, and within the access network the mobile node is identified by its home address. This is similar to the plain care-of address described in Mobile IPv4, and that is why the gateway router is sometimes also referred to as the foreign agent.

> Foreign domains supporting HAWAII will create host-based entries using the mobile node's care-of address.

The gateway router locates the mobile node by what is called Paging. Each router within the access network maintains a Paging Cache. This cache is populated with the source address of every packet it receives and the interface it was received on. These packets may be genuine traffic from mobile nodes, or the mobile node may periodically send empty packets towards the gateway router simply to advertise its presence as the cache entries have a limited lifetime. When a packet arrives for the mobile node the gateway router may have a host-based entry for it. If this is the case, the packets are simply forwarded through the network to the mobile node.

However if no entry exists the mobile node is paged via a paging packet. This packet is broadcast on the interface identified for the mobile node in the Paging Cache. This process is continued throughout the network until the mobile node itself receives the paging packet. When this happens the mobile node transmits a route-update packet to its access router. This packet

travels hop-by-hop towards the gateway router, with each router along the way creating a host-based entry for the mobile node. Paging Cache entries have a longer lifetime than Routing Cache entries. This allows paging-update packets to be transmitted less frequently such that idle nodes do not flood the network with control messages.

2.8 Summary

Mobility is achieved when a mobile node acquires a care-of address and registers this value with the home agent. This is the basic concept of Mobile IP. However this arrangement has problems of its own, such as the latency caused by the registration period, and by the handoff itself. Furthermore, maintaining bindings with several home agents and correspondent nodes can consume limited wireless bandwidth.

Thus other mobility solutions have been proposed such as Hierarchical Mobile IP, which is a type of local mobility management. Using care-of addresses, home agents and correspondent nodes direct traffic to a MAP, that in turn tunnels these packets to the mobile node. This reduces the registration period and consumption of bandwidth. However it does not reduce the latency caused by the handover.

Multicast-based Mobility is another proposal which provides the mobile node with a non-changing co-located care-of address. Thus the mobile node only needs to initiate route repair in the form of join and prune messages. While this greatly reduces the registration latency, like HMIP it does not eliminate the latency introduced by the handover.

There is a solution which can reduce, or eliminate, the handover latency. Since it also does not directly participate in registration, it can work concurrently with any of the mobility solutions discussed above. This protocol is called Fast Mobile IP.

The other mobility solutions have obvious advantages over Mobile IP; however there is little compatibility between them. While HMIP allows a mobile node that operates only MIP to function normally, if the local network resides within private space then the mobile node can't be reached. Though this is not likely, these nodes will not receive the benefits of HMIP or FHMIP without being reconfigured.

There are other mobility solutions with which a MIP-based mobile node would not be compatible at all, such as Multicast-based Mobility, unless the mobile node was re-configured. Thus the only way to obtain the benefits of a new mobility solution would be to modify all routers and mobile nodes. As discussed in Chapter 1, at best this process would be costly and time consuming. At worst, every node would have to be replaced.

Chapter 3 Mobility Management Framework

This chapter describes the design of the Framework, while the resulting Framework draft is contained in Appendix A. An (Internet) Draft is typically a specification for the operation of a protocol that provides a certain service, which in this case is mobility for a mobile node. Thus the Framework draft is the principal result of this project, since a specification may have several diverse implementations.

Section 3.1 begins with an overview of the Framework design. Sections 3.2 through 3.7 discuss the major processes that comprise the Framework. Sections 3.2 and 3.3 are a prelude to the Handover process, which is discussed in Section 3.4. Section 3.5 describes the interface between Handover and Registration. Sections 3.6 and 3.7 explain Registration and Route Repair respectively. Section 3.8 explains the operation of the mobile node, while Section 3.9 details the operation of the Foreign Agent. An analysis of Fast Handovers within the Framework is given in Section 3.10, while an example of route repair design is provided in Section 3.11.

3.1 Framework Design

The objective of the Framework is to facilitate the deployment of (mobility) protocols in the wireless network by eliminating the need to configure the mobile node. This requires disengaging the mobile node from the route repair process within the access network. However this does not necessarily mean that the mobile node remains ignorant of that process.

For instance the mobile node should know whether local mobility management is available on its current link; it is the details of how that service is provided that are hidden from the mobile node. To achieve this transparency we propose in the Framework a standard signaling between the mobile node and the access router, which is based on MIPv6.

To design a framework we need to perform the functional decomposition of the system. The Mobility Management System involves home agents, access routers and mobile nodes. Figure 3-1 shows the interaction of the above nodes.

The function of the home agent is to track the domain hosting the mobile node and redirect packets destined for the mobile node's home address to its current location in the hosting domain. The function of the access router is to provide link connectivity to the mobile node. The access router also performs mobility related functions, which are described below.

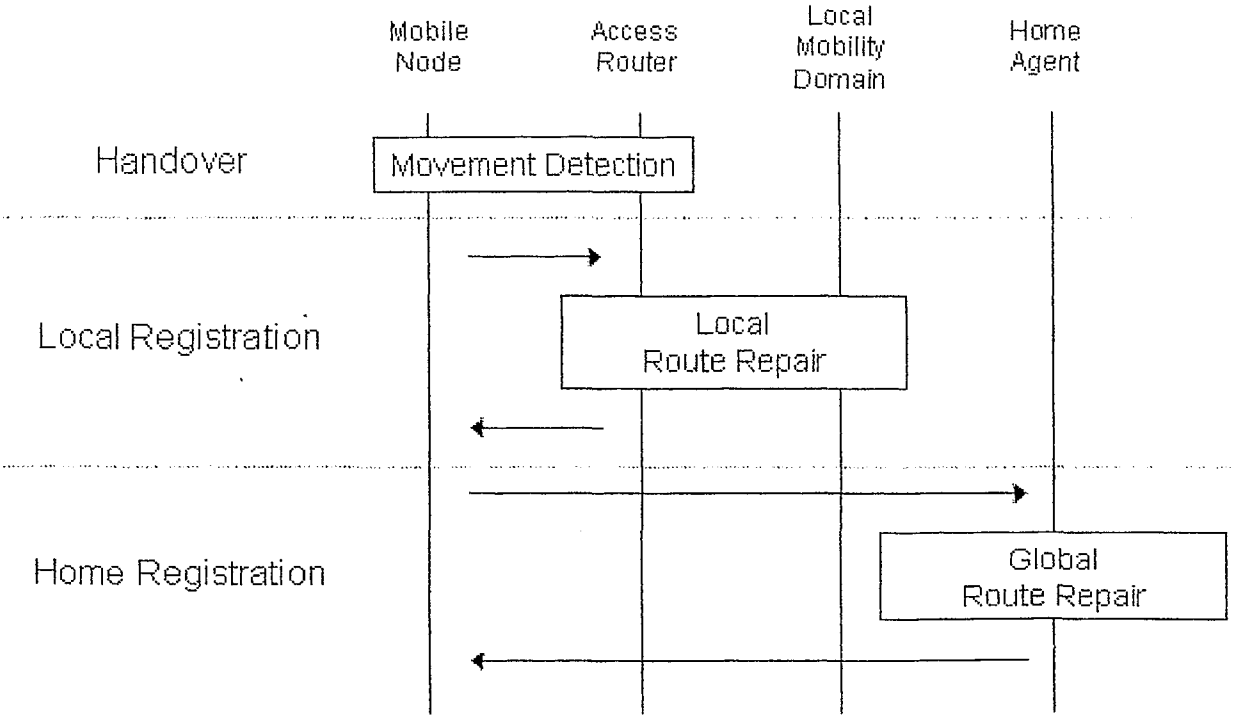


Figure 3-1: Elements of Registration

The mobility management system basically performs three functions to provide network connectivity to a mobile node when it moves from one access router to another. The three basic functions are:

- Handover
- Registration
- Route Repair

Handover is a process of detecting when a mobile node moves from the coverage area of one access router to another. It involves both link and network layer methods as well as their co-ordination for performance improvement. When this condition is satisfied the mobile node initiates a layer 3 handover, which is a process of reestablishing network connectivity at the new access router.

The mobile node will then initiate the Registration process, which in turn instigates Route Repair. Registration informs the access network and home agent that the mobile node has moved, while Route repair performs change in the forwarding path to ensure packet delivery to the new access router.

3.2 Location Management

A *domain* is usually the network administered and managed by a single wireless operator. We call this an *administrative* domain. We define the *mobility* domain to be the network that employs a single mobility solution. Thus an operator can employ different solutions in different networks that it administers. The solution controls the forwarding path to the mobile node.

The location of a node in an IP network is identified by the IP address. Mobile IP defines two IP addresses for mobile nodes that are visiting foreign domains. The *home address* identifies the mobile node's location in the home domain, which is known to all correspondent nodes in the Internet. A *care-of address* (CoA) is used to identify the mobile node's current location (i.e. subnet). This address may also be referred to as the *on-link care-of address* (LCoA).

> Historically the LMM Requirements draft used the term "local care-of address", hence the abbreviation "LCoA". However this draft no longer directly mentions care-of addresses as it may favour one protocol design over another.

We additionally define the *regional care-of address* (rCoA) as the address that the home agent forwards packets to. It does not directly specify the exact location of the mobile node in the domain. Rather it is used by the home agent to forward the packet to the correct hosting network.

The LMM solution employed in that network is responsible for delivering the packet to the mobile node's (on-link) care-of address.

3.3 Movement Detection – Domain Advertisements

In a cellular network both base stations and mobile nodes assess the handover condition based on continuous monitoring of the channel condition. In some situations, e.g. IEEE 802.11, the mobile node scans the channel frequency and latch on to the strongest channel signal it receives through a process known as *Scanning*.

Since Mobile IP resides on layer 3 it implements a mechanism to detect the physical handover using Router Advertisements (RA) that are periodically broadcast by every access router. A handover condition is satisfied when either the mobile node does not receive an advertisement from its current router within a specified period, or the mobile node receives an advertisement from a different access router.

To expedite layer 3 movement detection layer 2 triggers may initiate an early layer 3 handover process. A discussion on layer 2 handover and triggers is beyond the scope of this thesis.

. An access router periodically broadcasts router advertisements on the link, which is part of IPv4 Router Discovery and IPv6 Neighbour Discovery [6] [7]. Mobile IP modifies the router advertisement header and implements some additional options. The header contains additional flags that specify whether that access router also acts as a home agent, or whether it supports fast handovers. The options convey additional information, such as the Advertisement. This option advertises the frequency at which Router Advertisements are sent; this can be used to detect the absence of the access router.

We augment the same basic movement detection mechanism based on Router Advertisements as implemented in Mobile IP. For the Framework we decided to use a modified router advertisement because it involves a small change to a protocol that will already be in use by the router. Defining an additional protocol would consume bandwidth. Also, defining a new method of "mobility discovery" would also increase the complexity of the framework, where network administrators would be more inclined to use something they already understand.

The Router Advertisement header was modified to declare the services provided by the mobility domain that the access router belongs to. Hence we have we refer to this message as the Domain Advertisement. This is achieved by defining additional flags, which is discussed in the following section.

We also define the *Local Coverage Area* (LCA) option. The LCA is a generic option that contains domain specific information. This prevents domain specific options from being advertised to the mobile node, as this would create an implementation dependency to the solution deployed in that domain.

The LCA option is used to indicate the area over which a regional care-of address is valid. The option simply contains a binary identifier to define a coverage area; access routers that advertise the same identifier belong to the same coverage area. Thus a regional care-of address that is acquired in one LCA should not be used when the mobile node traverses to a different LCA. When this traversal occurs the mobile node must acquire a new regional care-of address.

The LCA identifier should be unique within a single administrative domain; hence it is locally defined and managed by a single operator. Note that the mobile node is only interested in the regional care-of address. The mobility domain may use any means to provide this address, such that the domain may not even have local mobility agents, and thus have a single coverage area (though it is uncertain how this could be accomplished). How this transparency is handled is described further on in this chapter.

The movement detection algorithm is based on straightforward comparison of information carried in the current domain advertisement with the previously received domain advertisement. If the router address in the new advertisement is different than of the address contained in the previous advertisement, then the mobile node has moved between access routers. Furthermore, if the new LCA identifier is different from the previous one then the mobile node has entered a new mobility domain. This involves inter-domain registration through Mobile IP.

3.4 Handover

The movement detection identifies the handover condition and initiates he handover process. Handover is a process that involves both the mobile node to establish network connectivity and resume packet delivery to the mobile node at the new link. It includes exchange of signaling between the mobile and access router, and configuration of the mobile with the new required addresses. It is followed by registration and route repair. Figure 3-2 shows the above Framework model.

As a result of a handover the mobile node acquires a new care-of address. The router advertisement instructs the mobile node to use either stateful autoconfiguration, such as DHCP, or standard IPv6 stateless autoconfiguration. Currently we require the mobile node to acquire a regional care-of address statefully during registration. However it is possible to define a Regional Prefix option to allow the mobile node to configure this address itself.

Careful readers may have observed that the term "handover" has been used in different contexts. Mobility terminology [11] has a very detailed section on the use of this term. The base definition is "when an active MN ... changes its point of attachment to the network ...". We have used this term to refer strictly to the act of transferring between two access routers, as described above. We have also used this term to refer the process reestablishing the mobile connectivity to the Internet; that is, registration and route repair. It is up to the reader to determine which definition applies based upon the context within which the term is used.

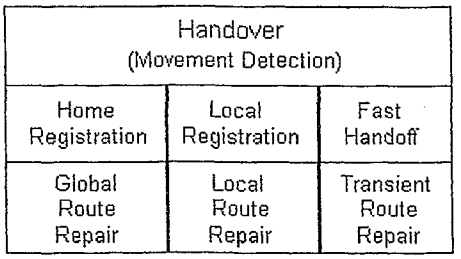


Figure 3-2: The Framework Model

The next step in the handover is to register the care-of address with the relevant mobility agents to resume packet delivery to the mobile node. There are three different types of registration:

- Home Registration
- Local Registration
- Fast Handovers

The mobile node performs Home Registration with the home agent to bind the regional care-of address to its home address. The home registration process also includes dynamic home agent discovery and prefix solicitation. To avoid middle man attack and other security fallouts the home registration must be initiated by the mobile node and must be accomplished using MN-HA security association.

Local Registration attempts to bind an on-link care-of address with to a regional care-of address. This allows packets tunneled by the home agent to rCoA to be delivered to lCoA. It has only local significance within the mobility domain. Further, since the local registration initiates local route repair, its detail should remain transparent from the mobile node. Therefore in the Framework we allow the new access router to perform the local registration.

Since the access router serves a significant role within the Framework we reintroduce the concept of a foreign agent. However we make a distinction between binding and tunneling agents. The former is active on the control path, while the later plays a part of the forwarding path. Unlike MIPv4, the foreign agent within the Framework is a binding agent only.

If the handover can be anticipated then a short-term route may be established between the old and new access routers. This would hide the latency introduced by both the layer 2 and layer 3 handover. Either the mobile node or access router may initiate the registration of this route using the process known as Fast Handovers. However, whether Fast Handovers are used the mobile node must still initiate either / both home and local registration at the new access router.

Following registration is route repair. We do not specify a single method of route repair, as that would defeat the purpose of the Framework. While the mobile node does not need to know the details of how route repair is accomplished within a mobility domain, it must know what types of services are provided by that domain. These properties are conveyed to the mobile node via a set of flags contained in the Domain Advertisement; this informs the mobile node on what course of action to take after a handover occurs.

3.5 Domain Flags

To provide a generic advertisement of available route repair services requires support for all existing types of care-of addresses. Preferably, the framework should also provide for future mobility requirements to minimize changes to the mobile node. In this regard, Chapter 2 was more than a literature review of previous mobility protocols. Each protocol discussed was selected because it introduces a different type of care-of address, whose operation must be included into the framework.

We previously mentioned that each type of service is advertised as a flag in the domain advertisement. The number of flags should be minimal to retain bits to allow for future definition of new services, and to reduce complexity for the protocol. Yet they should also provide for every possible type of care-of address. Figure displays all the possible types of care-of addresses discussed in Chapter 2. The first column provides a visual representation of the type of address(s) displayed in column two. Each block represents a tunnel terminal. Column three provides one possible example of a protocol that matches the displayed topology, though more may exist. Lastly, column four displays how the framework categorizes this care-of address.

Each row is referred to as a "scenario", and the letter in column 4 uniquely identifies each row. Consider cases B and D, where a foreign agent is present. Since this thesis is directed towards IPv6, and foreign agents are not used as tunnel terminals by MIPv6, these two cases can be ignored. Recall that even the though the Framework reintroduces the foreign agent, it sits on the control path only and does not affect forwarding.

Of the remaining four cases, A and E visually appear the same, while C and F look the same. That is because the second pair explicitly provides local mobility management, whereas the first pair does not. This is the first distinguishing feature, which we symbolize as the flag "I".

The second feature is that scenarios A and C have a changing co-located care-of address, while for E and F the care-of address is fixed. This is identified by the flag "C". Thus four possible domains are represented by only two flags. Figure 3-3 shows the value each flag should take for the given scenario.

These flags determine the action a mobile node must take in response to a new domain advertisement. In some cases local registration is not required, is other cases it is always required, etc.

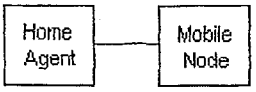
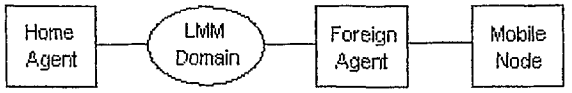
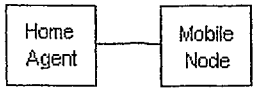
	changing co-located rCoA	MIPv6	A. C=0, I=0
	changing rCoA	MIPv4	B. C=0, I=0
	rCoA + changing co-located lCoA	HMIPv6	C. C=0, I=1
	rCoA + changing lCoA	Regional Registration	D. C=0, I=1
	non-changing co-located rCoA	Multicast-based Mobility	E. C=1, I=0
	rCoA + non-changing co-located lCoA		F. C=1, I=1

Figure 3-3: Types of Care-of Address

The C Flag

When set, this flag instructs the mobile node to acquire a new co-located care-of address, and that this address must be registered with the appropriate mobility agents. Note that this is the address that terminates at the mobile node. If the C flag is not set then the mobile node may keep its current (co-located) care-of address. This care-of address must be initially acquired via local registration, which will also require home registration so that the home agent may know this address. Thereafter the mobile node only needs to perform local registration to allow for route repair. Note that a non-changing care-of address is implicitly a form of local mobility management, such as that provided by Multicast-based Mobility (M&M). However M&M allows the mobile node to auto-configure its own care-of address while the Framework requires the mobile node to request for one.

The I Flag

This flag, when set, indicates the presence of a local mobility management protocol. This means that the mobile node may acquire a regional care-of address, via local registration, and

register this address with the home agent. Thereafter, while the mobile node remains within the same local coverage area, all it needs to do is perform local registration. An unset I flag advertises the absence of local mobility management. The mobile node acquires a single care-of address and registers it with the home agent. Recall that HMIPv6 allows the mobile node to form its own regional care-of address; in the Framework we currently require the mobile node to request the assignment of a regional care-of address during local registration to simplify the abstraction of domain operation.

3.6 Registration

MIPv6 specifies a Binding Update message which the mobile node uses to inform binding agents of its new care-of address. The equivalent message in MIPv4 is referred to as the Registration Request. While in the Framework we adopt the format of the Binding Update, we call the processing of this message Registration.

The binding update header, as defined by Mobile IP, contains some flags which act as instructions for the destination. Two of these flags are worth mentioning; the mobile node may specify whether or not it requires an acknowledgement (A), and it may also specify whether the destination should act as the mobile node's home agent (H). The Framework defines some additional flags for local registration, which MIPv6 does not take into account. They are as follows:

- Local Route Repair (R): Requests that the domain update its point of connection for the given care-of address(s).
- Regional COA Request (D): When the mobile node encounters a new local coverage area it must request a new regional care-of address from the domain.
- Local COA Request (C): Some domains may require the mobile node to acquire a local care-of address statefully. The mobile node uses this flag when it requires a new address.

These flags are used to request services of route repair based upon the needs of the mobile node. The detailed operation of these flags is discussed in more detail in the following sections.

3.7 Route Repair

Route repair is a process of setting up new a forwarding path as a result of node mobility. We identified three main types of route repair that occur during handover: global, local and transient.

The first two are handled by home and local registration respectively. The last is processed by fast handovers, which is discussed in another section.

Macro mobility causes route repair on a global level where the routing of packets happen between networks. We utilize the basic concepts of Mobile IP, which implements a packet redirection mechanism at the home subnet. In Mobile IP packets are direct to the home subnet and from there they are tunneled to the access network that the mobile node is currently connected to. Redirection through tunneling is a kind of *global route repair*. Its primary function is to establish the global forwarding path in the home agent's routing table. Home registration controls this process and it also allows the mobile node to dynamically acquire a home agent, and a home address.

Home registration is also responsible for Route Optimization as defined by MIPv6. Route Optimization entails informing correspondent nodes of the mobile node's care-of address. This allows correspondent nodes to send packets directly to the mobile node. However such optimization is not mandatory to provide mobility. Thus the operation of Route Optimization within the Framework is deferred to future projects.

Micro mobility (also called Local Mobility) causes local route repair within the access network. Local registration controls this process and relies upon the mobile node's regional and on-link care-of address. In our framework the foreign agent performs local registration, whose details remain transparent to the mobile node. This is the key idea that allows the operator to employ any local mobility management solution of its own choice. Since the mobile node doesn't need to know how the forwarding path is setup inside the access network, the local route repair can be performed solely in assistance with the foreign agents.

The local route repair is analogous to the intra-domain routing (IGP) in IP networks, hence it does not need security association between the mobile node and any mobility agents used by local mobility management (e.g. MAP in HMIP).

Local registration is initiated by the mobile node because the mobile node performs the movement detection between the mobility domains and controls the handover.

Fast handover is designed to lay down tunnels between old and new access routers for the packets in transit along the old forwarding path during the handover. We call this transient route repair as it attempts to establish short-term paths within the local domain. This requires the mobile node to send a (fast) binding update to either of the access routers, who then cooperate to create a bi-directional tunnel. These short-term paths are used to reduce the handover latency while the forwarding path inside the network is established.

3.8 Mobile Node Operation

Figure 3-3 shows domain abstraction through the use of the C/I flag pair. This section describes how the mobile node uses these flags, as while as the local coverage area identifier, during the handover process. The following subsections refer to the scenarios defined within Figure 3-3.

Within this section, unless stated otherwise, when a binding update is required to be transmitted we assume that all flags are not set except for the Acknowledgement flag (A). When a sending a binding update to the home agent the source address is its rCoA, while the home address is contained in an option within the mobility header. When the mobile node sends a binding update to the foreign agent it must use its link local address. The care-of addresses are stored as options.

3.8.1 Mobility States

Figure 3-4 is a state diagram of the mobile node based upon the design of the Framework discussed in the previous sections. Each state represents one of the four scenarios possible within IPv6, as defined in Section 3.5. State transitions are triggered by inbound Router Advertisements that contain the C/I flag pair and LCAI option. During the state transition the mobile node will engage one or both of local and home registration. It is important to understand that there is no relationship between each state (i.e. scenario), when a mobile node transfers between any two different states it is essentially restarting the mobility process.

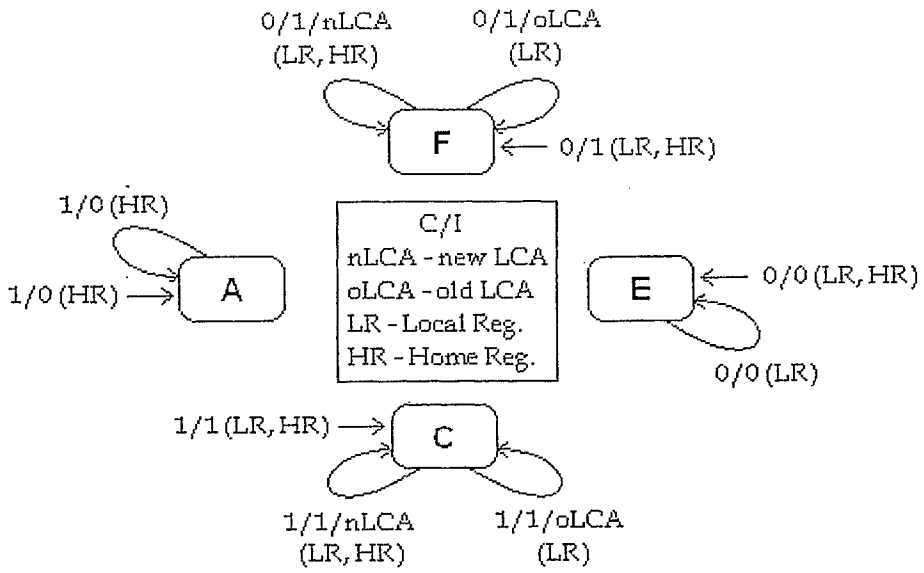


Figure 3-4: Registration State Diagram

3.8.2 Scenario A

Every time the mobile node receives this advertisement from a new router it must acquire a new care-of address. The mobile node may auto-configure an address based upon the advertisement or, where appropriate, it may instead pursue stateful address allocation such as DHCP. Once the mobile node acquires a care-of address it will send a binding update to its home agent with the H bit set, and an option containing its new care-of address. If the home agent rejects the binding update there is little that the mobile node can do, other than create bindings with individual correspondent nodes. However the mobile node can attempt to correct the error and try again. Note that this case is identical to MIPv6 and the foreign binding agent does need not to be part of the control path.

3.8.3 Scenario C

This scenario instructs the mobile node that it must change its care-of address, and that a regional care-of address (rCoA) can be acquired. The specific action taken depends upon the local coverage area identifier.

New LCA Identifier

Recall that the mobile node must acquire a new rCoA whenever it observes a local coverage area (LCA) identifier that is different from the one advertised by the previous router. Technically, receiving this DA for the first time is like receiving a new LCA Identifier, thus necessary actions for the two events are the same.

The mobile node must send a binding update to the foreign binding agent (FBA). This node should not act as a home agent, so H is not set. The domain will assign a rCoA to the mobile node, thus the D flag must be set. Since the mobile node can create its own local care-of address (lCoA) it does not need request one from the domain, as such the C flag must not be set. Finally the mobile node will want the domain to bind its lCoA to its rCoA (even though it doesn't have it yet). To request route repair the R flag must be set.

Local registration is complete when the mobile node receives a binding acknowledgement from the foreign agent. If it indicates success then the node registers its rCoA with the home agent. If the registration failed then the mobile node may auto-configure a care-of address and attempt to register this with the home agent, foregoing the benefit of local mobility management. If home registration fails, since the mobile node has a valid regional care-of address, it can still communicate within the local domain. The mobile node can also bind with correspondent nodes.

Old LCA Identifier

The mobile node only needs to change its local care-of address, and to register this address with the (local) domain. This means sending a binding update to the FBA with the R flag set.

The binding update must contain the lCoA, along with its rCoA as context. The mobile node does not need to initiate home registration as its regional care-of address has not changed.

3.8.4 Scenario E

Receiving this advertisement informs the mobile node that it can keep its current care-of address, and that it can use this address globally. Since a unicast IP address can not easily transfer between subnets the mobile node can not form its care-of address statelessly. Thus the first time the mobile node enters this domain the mobile node must acquire its care-of address from the domain statefully. This requires a local binding update with the C flag set. Furthermore, since this address can not be used in standard network-based routing, some form of local route repair is required. As such the mobile node must also set the R flag in the initial binding update.

Local registration is complete when the mobile node has received care-of address from the domain. The mobile node must then send a binding update to its home agent using its new care-of address. The H flag must be set. Note that in subsequent handoffs the mobile node does not need to update the home agent as its care-of address is not changing. However, as noted above, the mobile node must inform the domain so that it may properly route packets to the new point of connection. This requires a local binding update to the foreign binding agent with the R flag set, with the care-of address is contained as an option for context

3.8.5 Scenario F

The scenario advertised by this DA allows the mobile node to keep its local care-of address as well as acquire a regional care-of address (rCoA). Note that this is essentially a union of the previous two states. With each new LCA identifier the mobile node must acquire a new rCoA and lCoA. The mobile node will also want the two associated. This requires a local binding update to the FBA with the R, D and C flags set. Local registration is complete when the mobile node has received a binding acknowledgement containing both its rCoA and lCoA. The mobile node must then send a binding update to its home agent using the rCoA, and the H flag must be set. If the current LCA Identifier matches the previous one, then the mobile node only needs to send a local binding update containing its lCoA with the R flag set to the foreign binding agent. It does not need to send its rCoA as additional data since its lCoA is not changing.

3.8.6 Miscellaneous Operations

If local registration fails the mobile node can attempt to register with its home agent using a statelessly configured care-of address instead. However the mobile node must include a Type 0 Routing Header in its IP header, with the target as the Foreign Binding Agent. This will allow the FBA to inspect the binding update and make sure it does not violate any of the domain's requirements, such as attempting to register a site local address outside of the domain.

If registration with the home agent fails, the mobile node can still attempt to update its bindings with correspondent nodes. There little else that the mobile node can do if the home agent rejects the binding update other than to keep sending the same update in the hope that it will be eventually accepted.

MIPv6 defines additional actions such as dynamic home agent address discovery, as well as acquiring a home address. The framework allows for these actions, but does not modify them. For details readers must examine the MIPv6 document.

3.8.7 Example

A C/I pair of 1/1 is advertised. This indicates Scenario C, where a regional care-of address is provided and the local care-of address must be changed. The first time the mobile node sees this pair it must activate local and home registration. In consecutive advertisements with the same LCA the mobile node only activates local registration. If the advertised LCA identifier is different from the previous one (not necessarily an undiscovered identifier) then the mobile node must also activate home registration. If a different C/I pair is discovered then the mobile node will adhere to the new scenario's initial conditions. Subsequent advertisements of that pair will cause the mobile node to remain in that state while pursuing the necessary actions indicated in Figure 3-4.

3.9 Foreign Agent Operation

The Foreign Binding Agent is the sole node responsible for Local Registration. It also supports some functions related to the deployed Route Repair solution. Though it is beyond the scope of this document to discuss the implementation of every potential route repair protocol, we must define a standard interface between Local Registration and its neighbouring "layers". After we identify these services we specify the operation of Local Registration.

3.9.1 Local Registration Interfaces

We have already discussed the simple handoff and local registration interface in detail. The mobile node requests registration via a binding update while local registration responds through the binding acknowledgement.

There is a separate set of primitives defined for the (local) registration to route repair interface. This is because the interface resides within the foreign agent and as such a binding update is not entirely appropriate.

Typically all messages specified within a single document are defined within a single section. However this report follows the path of the framework design such that these primitives are best discussed here.

Local registration has three outbound request primitives: Initiate Route Repair, Request Regional Care-of Address and Request Local Care-of Address. There are also three corresponding inbound confirm primitives. They are effectively data structures that are passed between the two processes. The use of these primitives in Local Registration is described below.

Note that fast handoff is also implemented on the foreign agent, and that it also has an interface with route repair. However a discussion on this interface is deferred to the following section.

3.9.2 The Registration Process

The foreign agent will receive binding updates which is a request for services from a mobile node. The foreign agent maintains a temporary binding entry for this update until the all of the items in the request have been met, for which it processes each item individually by issuing requests to route repair. When all of the items have been dealt with a binding acknowledgement is delivered to the mobile node.

The services are requested by flags contained within the binding update. These are Local Route Repair (R), Regional COA Request (D) and Local COA Request (C). Additional data will be contained within the binding update depending upon which flags have been set, namely the addresses that require binding. Each flag directly corresponds to one of registration's request primitives; the C flag creates the Request lCoA primitive, the D flag creates the Request rCoA primitive and the R flag creates the Initiate Route Repair primitive.

The flags are processed sequentially in the following order: C, D and R. Any flags that were not set are skipped. The purpose for this order is that the lCoA may be useful in forming the rCoA. As such it is acquired first and passed as an argument for the Request rCoA primitive. Finally, route repair binds the local care-of address to the regional care-of address. Thus these two values must be acquired before route repair can be initiated.

Local registration was introduced to act as a buffer between the (Simple) Handoff and Route Repair processes. This arrangement provides route repair protocols an interface that can be changed, whether for operational or experimental purposes, without requiring a corresponding change in the mobile node.

The registration request is broken into individual actions to promote this level of flexibility. Furthermore, route repair will typically involve multiple nodes. Since all of the items in the binding update must be processed sequentially, by breaking the update into individual requests the foreign agent can process multiple binding entries instead of waiting for an entire binding entry to be concluded before moving on to the next.

3.10 Fast Handovers

We have previously discussed the four scenarios of the framework. Fast handovers as described by "Fast Handovers for Mobile IP" [12] may operate safely in all four cases even though Fast Handovers was designed specifically for Scenario A. This means that this protocol may run unmodified within the framework.

However, when local registration is present within the framework it may be more efficient to combine the two protocols [13]. This introduces the need to make the operation of the fast handover protocol transparent to the mobile node, which requires the design of a standard interface between the two. To do this we must assess each scenario, using unmodified Fast Mobile IP as a reference point.

Scenario A

- Fast Mobile IP was designed for this scenario, thus no special modifications are necessary.

Scenario C

Fast Hierarchical Mobile IP [13] is example of a fast handoff protocol for this type of scenario. However it specifies that the mobile node send the fast binding update directly to the MAP. This is unacceptable to the framework. Therefore the mobile node must operate as specified with one exception, it may include its rCoA in the FBU as context. The foreign agent may be configured to either operate "normally", or it may direct the tunnel agent sponsoring rCoA to bicast packets to the mobile node's previous and new care-of addresses.

Scenario E

Since this protocol employs host based routing or multicasting, there must be a common node located between NAR and PAR. Using the same principles as in the previous scenario, the foreign agent may instruct the access network at this location to bicast packets to the mobile node. Since the source address of the fast binding update is the previous care-of address (PCoA), there is no need to include extra data in the FBU. However, if the domain will be bicasting packets to PCoA then the mobile node must prepared to accept packets bound for this address, even on the new link, until indicated otherwise during local registration at the next access router.

Scenario F

This is simply a combination of the previous two scenarios. While it would be up to the protocol to determine the best method to bicasting, the mobile node should employ both modifications specified by the previous scenarios.

Based upon this analysis we determine that the mobile should operate as specified by Fast Mobile IP with the following modifications:

- If the mobile node has a regional and local care-of address, the mobile node should be allowed to include its regional care-of address in the fast binding update.
- The mobile node should retain "PCoA", as well as continue to receive packets destined for this address until indicated otherwise during local registration at NAR.
- The foreign agent determines whether to transmit the Handover Initiate (HI) message to the next access router, or to a mobility agent within the local mobility domain.

We demonstrate the control flow described here in Figure 3-5. Readers should be aware that this diagram is an overview only; it represents the predictive case only, and does not display detailed messages such as the Router Solicitation for Proxy Advertisement message.

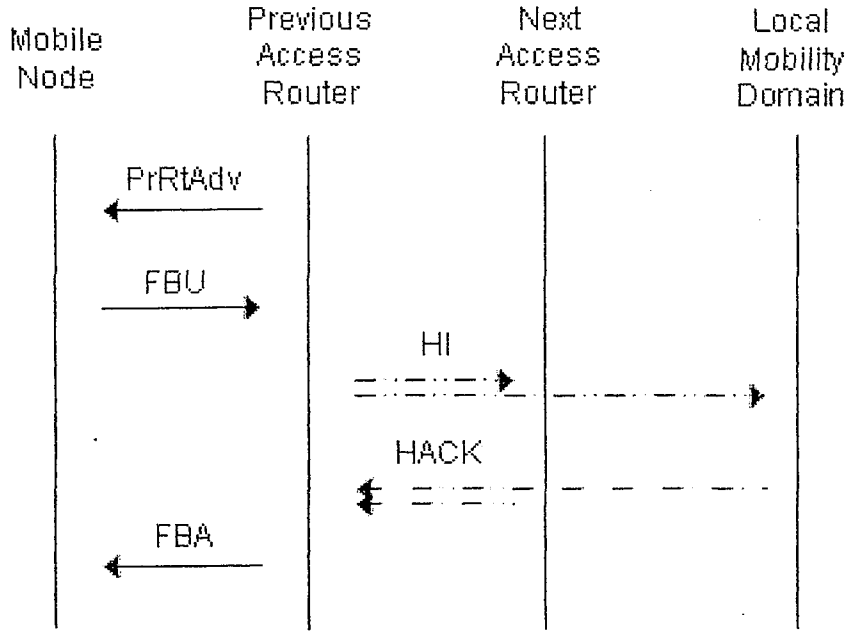


Figure 3-5: FMIP Control Flow

3.11 Security

Disengaging the mobile node from the route repair process within the local mobility domain in our framework does not create an additional security risk. IP mobility solutions use IPsec and authentication options between the mobile node and related binding agents. For either method to work requires an association between both nodes. For example, nodes in Figure 3-1 require the following associations:

- mobile node – home agent
- mobile node – foreign agent
- foreign agent – local mobility (agent) domain

The last association between the foreign agent and the local mobility agent is required for safe route repair. Thus while the foreign agent introduces an additional node in the overall control path we do not experience an increase of security associations in the Framework at the mobile node or home agent compared to protocols such as Hierarchical Mobile IP. The number of associations under conditions identical to Mobile IPv6 is also the same as the foreign agent is not engaged on the control path within that scenario. Furthermore when the foreign agent is a participant on the control path within the local mobility domain the network administrator can establish in advance additional security measures, while with a mobile node they must do this during the handover process and have no control over the security method selected.

3.12 Reference Design: Route Repair for HMIPv6

The detailed design of local route repair depends upon the local mobility management solution used in the network, which is beyond the scope of this thesis. However, we provide a case study to show how a route repair protocol would operate within the Framework. We discuss below the adaptation of HMIPv6 as a route repair solution.

3.12.1 Overview of HMIPv6

HMIPv6 is a local mobility management solution that requires the mobile node to acquire a regional and local care-of address. It employs a Mobility Anchor Point (MAP) to tunnel the packets that it receives from the home agent at the regional care-of address to the mobile node at the on-link care-of address. The MAP inserts the MAP option in its router advertisement that is propagated, at the network administrator's discretion, to certain access routers. These routers will include the MAP options in their own router advertisements to the mobile node. The mobile node uses this advertisement to auto-configure both the regional and local care-of address. It sends a binding update to the MAP to register these addresses in the MAP's routing table.

When the mobile node receives a binding acknowledgment indicating success it then sends a binding update to the home agent to associate its home address to its regional care-of address. Thereafter the mobile node only updates the MAP with new lCoAs so long as the mobile node continues to receive advertisements from that MAP. Since in this situation the regional care-of address doe not change, there is no need for the mobile node to update the home agent.

HMIPv6 allows overlapping MAP domains. In that case the mobile node receives multiple MAP options, one from each MAP. The MAP option contains a preference value that can be administratively configured. The mobile node registers with the MAP that has the lowest distance value and lowest preference (excluding 0).

3.12.2 Domain Abstraction

The C and I flags should be set in the Domain Advertisements of all foreign agents within HMIPv6 mobility domains to allow proper configuration of addresses at the mobile node. Furthermore, the Domain Advertisement should also include a LCA Identifier corresponding to the HMIPv6 MAP domain where the mobile node has acquired its regional care-of address.

The LCA Identifier is not the MAP option; hence in the Framework MAP advertisements are intercepted and processed by the foreign agent. The mobile node remains unaware of the existence of the MAP. In case of overlapping MAP domains the foreign agent receives multiple MAP advertisements and it will select the most suitable MAP for the mobile node. It then includes a LCA option corresponding to that MAP domain in its Domain Advertisement to the mobile node. Thus, in the case of HMIPv6, the Framework shifts the responsibility of MAP selection from the mobile node to the foreign agent.

We argue that the foreign agent should be the one that selects the best MAP because it is a network component that can maintain more elaborate knowledge of MAPs and use a corresponding set of criterion, e.g. reliability, performance, scalability, etc. to make the MAP selection. Furthermore, since the MAP option must not propagate to the mobile node, the MAP selection must remain with the foreign agent.

Note that overlapping mobility domains can only be supported if the Domain Advertisements can be dynamically configured. If they are statically configured then there may be no domain overlap. This is because the region care-of address must be valid wherever the corresponding LCA Identifier is advertised. An overlap will invalidate the regional care-of addresses beyond the union of the two local coverage areas. However, advertising multiple LCA options would alleviate this problem at the cost of bandwidth, as well as to shift the decision point back to the mobile node.

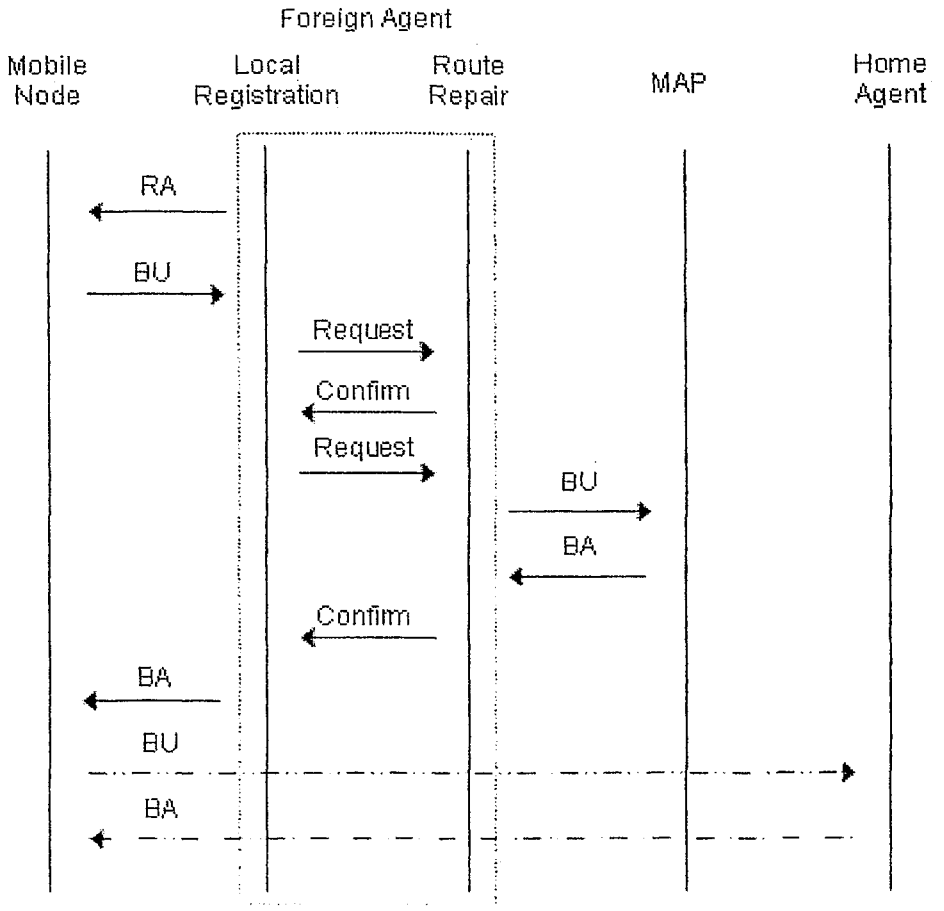


Figure 3-6: HMIP Control Flow

Figure 3-6 displays the flow of control of a local mobility domain that deploys HMIP. The mobile node does not need to know how the MAP operates, or whether one is even present. This abstraction is made possible by the Route Repair API hosted on the foreign agent. When local registration is complete the mobile node may then also engage home registration.

3.12.3 Local Registration

The access router is delegated the responsibility of performing local registration of the mobile node's local care-of address with the MAP. It starts local registration after receiving a binding update as a part of the MN-AR standard handover signaling. The binding update will either have the D and R flags toggled, or just the R flag depending upon whether the mobile node is new to the local coverage area. Since the mobile node can auto-configure its lCoA the C flag never needs to be set.

Local registration creates an entry for the update which stores requested actions, and the results of actions already processed. If the entry shows D=1 the foreign agent will issue a

Request rCoA primitive to route repair (i.e. HMIP). When local registration recives a Configure rCoA primitive it will store rCoA in the entry. The entry should also have R=1 which requires an Initiate Route Repair primitive sent to HMIP. Local registration will receive a Repair Acknowledgement in response which contains the status of the binding. When no further actions are required local registration will then create a binding acknowledgement for the mobile node. Once the binding acknowledgment is transmitted local registration can delete the binding entry.

3.12.4 Route Repair

HMIPv6 route repair must be able to handle the Request Regional Care-of Address and Initiate Route Repair requests. It confirms these actions using the Configure Regional Care-of Address and Repair Acknowledgment primitives respectively.

HMIPv6 caches MAP options at the foreign agent as discussed above. Thus when route repair receives the Request rCoA primitive it can directly inspect that cache and pick the MAP with the best characteristics. It will use this to form a rCoA for the mobile node, which it passes to local registration using the Configure rCoA primitive. The control path never leaves the foreign agent during this exchange. When route repair receives the Initiate Route Repair primitive the foreign agent must be able to identify the MAP using rCoA as context.

3.13 Reference Design: Route Repair for Multicast-based Mobility

We provide an additional reference design for Multicast-based Mobility (M&M) to demonstrate that the Framework is applicable to multiple methods of route repair.

3.13.1 Overview of Multicast-based Mobility

M&M provides local mobility management by allowing the mobile node to retain its care-of address after a handover. It achieves this by dedicating a multicast address to mobile node for use as a care-of address. The mobile node sends a binding update to the home agent only when it first receives this care-of address. Thereafter, when the mobile node moves, it simply needs to join at the new access router and to prune at the previous.

3.13.2 Domain Abstraction

M&M allows the mobile node to retain its care-of address but does not provide "explicit" local mobility management, i.e. it does not supply a regional care-of address to the mobile node. As such the C and I flags should both be set to 0.

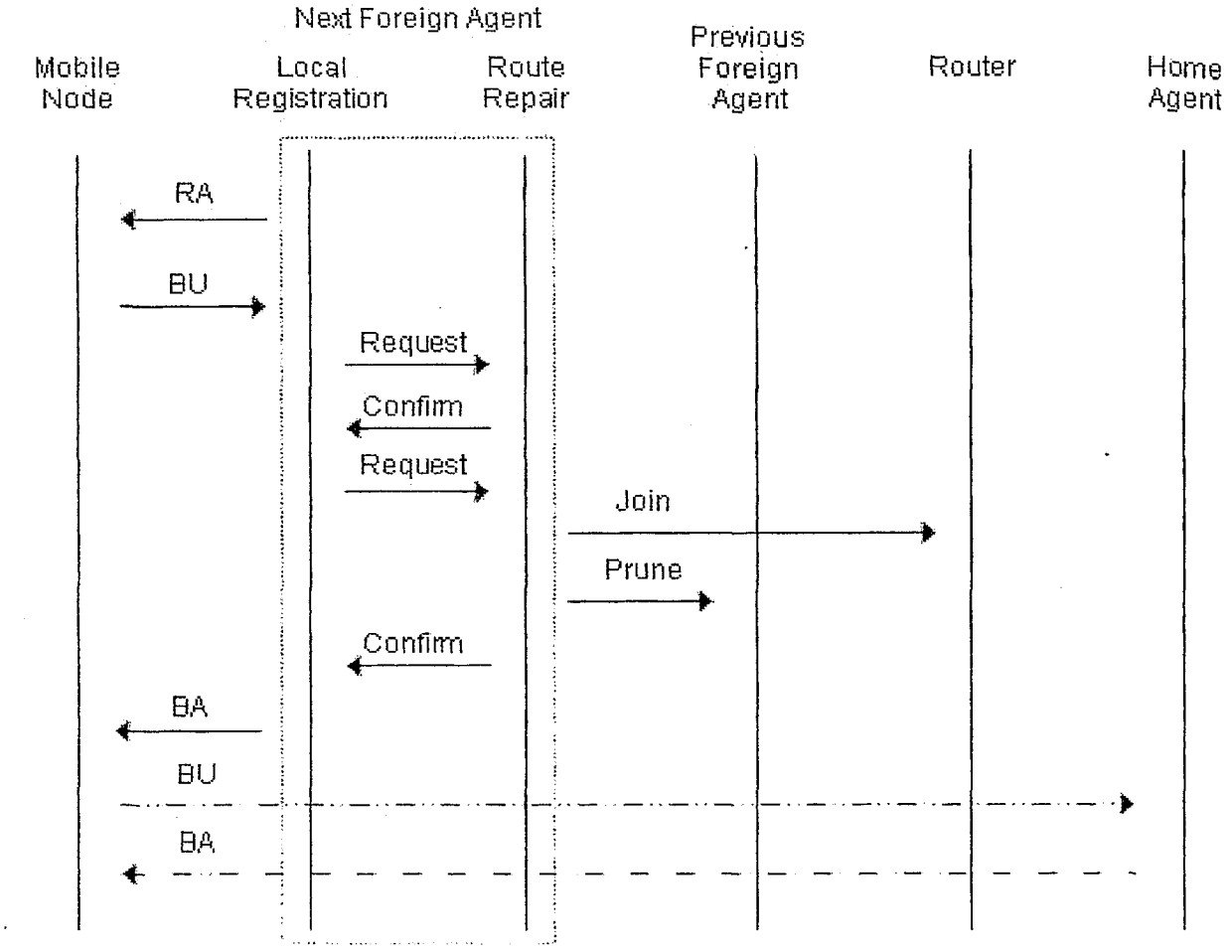


Figure 3-7: M&M Control Flow

Currently the Framework requires a LCAI to be advertised only when I=1. However this causes a problem when the mobile node crosses mobility domains that allow the mobile node to retain its care-of address (C=1). Under these conditions the mobile node is unable to determine whether it has entered a new mobility domain with the flags alone; as such must it will retain its current care-of address instead of acquiring a new one appropriate for the new domain. Thus the foreign agent must include the LCAI option in all router advertisements, and the mobile node must use this information in determining when to acquire a new care-of address as well as when to initiate home registration. This flow of control is displayed in Figure 3-7.

3.13.3 Local Registration

While M&M should not alter the operation of local registration, we discuss here the control of local registration while operating in conjunction with M&M. Local registration is activated by a

binding update from the mobile node. This request will have the R bit set, and may optionally have the C flag set.

Local registration creates an entry for the update which stores requested actions, and the results of actions already processed. If the entry shows C=1 then this process will issue a Request CoA primitive to route repair (I.e. M&M). When local registration receives a Configure C0A primitive it will store CoA in the appropriate entry. The entry should have R=1, which requires an Initiate Route Repair primitive sent t to route repair. Local registration will receive a Repair Acknowledgement in response which contains the status of the binding. When no further actions are required local registration will then create a binding acknowledgement for the mobile node. Once the binding acknowledgement is transmitted local registration can delete the binding entry.

3.13.4 Route Repair

M&M route repair must be able to process the Request CoA and Initiate Repair primitives. To achieve the first request there is an algorithm that M&M uses to auto-configure a care-of address; this algorithm is executed by the mobile node, as such this function must be relocated to the route repair process within the foreign agent for reasons discussed in the previous section. To perform route repair the access router simply needs to subscribe to the multicast address. However M&M will require a mechanism to determine the correct path to prune as the mobile node will not convey the identity of its previous access router as part of the binding update.

3.14 Summary

The framework is separated into processes, where each process roughly corresponds to the functions supported by a specific node-type. Defining a standard API between these processes allows the foreign agent to conceal the details of the local domain from the mobile node.

In turn, the mobile node conceals its mobility through the use of care-of addresses. The goal of the mobile node then is to have a valid care-of address. It was shown that the IP mobility protocol the local network operates can be abstracted into types of care-of address, and that these types can be reduced into two categories: the provision of local mobility management, and the provision of non-changing care-of addresses.

Router Advertisements are used by existing IP mobility protocols as a means of movement detection, as well as to advertise properties associated with that link. Thus it is natural to adopt and modify this advertisement to contain an additional two flags representing the care-of address properties.

The mobile node determines that a handoff has occurred through the advertisement of a new foreign agent. This indicates that its current care-of addresses (CoA) are no longer valid. It is the

mobile node's job to acquire the appropriate CoAs and register them with the proper mobility agents.

The mobile node uses local registration to initiate route repair within the access network. Local registration also assigns IP addresses appropriate for the network to the mobile node. Since these addresses will be registered with the home agent, local registration must precede home registration. Local registration does not guarantee successful route repair, or that IP addresses will be available.

Once local registration is complete, the mobile node may need to initiate home registration. Home registration handles "global" registration; it attempts to bind the mobile node's home address to its newly configured regional care-of address, and may also provide home agent and home address discovery.

The third registration protocol is fast handoff. It provides transient route repair, such that the mobile node may continue to exchange packets despite the latencies introduced by a handoff. After the handoff the mobile node must still initiate local and home registration where applicable. Like the other two protocols, the details of the transient route are transparent to the mobile node.

Route Repair is the actual act of establishing a route to the mobile node, whether that means binding addresses together within a single node or setting up the path node by node. This process also provides stateful address assignment. While the framework specifies the interface between this route repair and registration, it can be changed without also requiring a change in the mobile node.

Providing this transparency to the mobile node is the goal of the framework. Otherwise effecting those changes would be difficult to accomplish in networks with a large user base. The framework is also designed to adopt many of the characteristics of existing IP mobility protocols without compromising its main objective. This allows for an easy conceptual transfer from a non-structured mobility architecture to the Framework. It also provides for code re-use, which reduces development time.

Chapter 4 The Implementation

Another objective of this thesis is to design and implement a simple prototype that demonstrates the properties of a Mobility Management Framework. To achieve this goal the prototype must have the simple handoff, home and local registration protocols. For implementation purposes this requires producing software for a mobile node, home agent and foreign agent respectively. Since the remaining protocols are either optional or not critical to operating the Framework, they were relegated to future projects.

The Mobility Management Framework was designed for IPv6. There is publicly licensed code available for Mobile IPv6, which relies upon independent software for neighbour discovery. Since the Framework reengineered Mobile IPv6 it was logical to reuse code developed for this protocol.

Section 4.1 discusses the neighbour discovery software. Section 4.2 provides an overview of the unmodified Mobile IPv6 software. The Framework's implementation of the mobile node is explained in Section 4.3. The foreign agent is described in Section 4.4. Since the framework does not specify any operational changes for the home agent, this node is not discussed. The chapter concludes with a brief discussion of fast handovers.

The framework, and related projects, can be obtained at http://www.ee.ryerson.ca/~jaseem/.

We use the following convention to describe the source code wherever it is referred to in this chapter. The name in *italics* is the file in which the code can be located. The number prefixing the code is the line within that file that the code may be found on. The entire project was written in C.

ndisc.c

```
631    change_rtr = ndisc_mipv6_ra_rcv(skb);
```

ndisc.c

However in some circumstances the indentation of the original MIPL code is not printer friendly; it has been reformatted to keep commands and their documentation on a single line.

4.1 Neighbour Discovery

Mobile IP uses neighbour discovery for:

- Movement detection.
- Creating new care-of addresses.

To do this a node requires information about the link it's connected to, such as the network prefix, who the routers are, etc. Mobility agents periodically broadcast Router Advertisements to carry the above information.

4.1.1 radvd

The Router Advertisement Daemon, or radvd [16], was first written in 1996. Since then it has been modified to match modern discovery RFCs, as well as to provide support for mobility. Many researchers and developers use this application in their work, including the mobility software adopted by the framework.

While the router advertisement is only one aspect of neighbour discovery, it is the most crucial element with respect to mobility. In terms of the framework, and its implementation, this is also the only area of change within neighbour discovery.

Each router is configured via a static text file, as shown in Figure 4-1. A scanner parses the values in the file into variables. A structure is defined that mirrors the message format; the variables are simply assigned to the appropriate field. Additional structures exist for Options, and if configured they are appended to the message buffer. Once the message buffer is finalized it is written to a raw IP socket.

4.1.2 Domain Advertisement

We have introduced domain advertisements in the framework to broadcast the mobility services supported by the access network. In our implementation we modified radvd by:

- Adding the C flag
- Adding the I flag
- Adding the Local Coverage Area Identifier

The first two additions are easy; in each case it is simply a matter of toggling a single bit. This change entails adding an extra field to the RA structure. It also requires modifying radvd's lexer to identify administratively defined values in a configuration file.

radvd.conf

```
1     interface eth0
2     {
3       AdvSendAdvert on;
4       AdvIntervalOpt on;
5
6       MaxRtrAdvInterval 4;
7       AdvHomeAgentFlag off;
8       AdvSupportLMMFlag off;
9       AdvChangeCOAFlag on;
10
11      prefix fed0:10:0:2::/64
12      {
13        AdvOnLink on;
14        AdvAutonomous on;
15        AdvRouterAddr on;
16      };
17    };
```

radvd.conf

Figure 4-1: Configuration for radvd

Adding the LCA identifier requires defining a new option and its associated data structure. When present it must be appended to the message buffer, and the length of the message properly calculated and stored in the length field. In addition to recognizing this configuration, the grammar must check for the status of the I flag. If it is on, this value must be configured. If it is off then this value must be ignored.

4.2 Mobile IPv6 for Linux

Mobile IPv6 for Linux (MIPL) was first developed in 2000 and continues to be maintained to this day. This project uses MIPL version 0.9.5.1 [17], which is a 20,000 line patch to the Linux kernel. Since this document's 3,000+ lines comes close to 100 pages, the author decided not to include the Framework in its entirety. Instead it has been made available at the following website: http://www.ee.ryerson.ca/~jaseem/.

This section discusses the key aspects of the mobile node before the Framework, while Section 4.3 discusses the changes required to make the mobile node framework-ready. Though not originally part of MIPL, Section 4.4 describes the implementation of the Foreign Agent.

4.2.1 Movement Detection

Neighbour discovery is an independent process from movement detection. The kernel already has RFC 2461's [7] host specification embedded in its code; this code was produced by the same authors of radvd.

MIPL incorporates two changes in the implementation of neighbour discovery to make movement detection available within the kernel. First it allows neighbour discovery to initiate a movement detection process whenever a router is "discovered", as shown in Figure 4-2. Secondly it registers its own process with the kernel.

ndisc.c

```
632    change_rtr = ndisc_mipv6_ra_rcv(skb);
```

ndisc.c

module_mn.c

```
321    MIPV6_SETCALL(mipv6_ra_rcv, mipv6_mn_ra_rcv);
```

module_mn.c

Figure 4-2: Enabling Movement Detection

Figure 4-3 shows the entire movement detection process. The first function parses the router advertisement into a router structure usable by the rest of the mobility software. The second function, using the newly parsed router, determines if a handoff is necessary. This is ascertained by comparing the network prefix of the newly advertised router to that of the previously advertised router, whose data is contained in a record called "curr_router". If they are the same then the return value is set to "1" to indicate that a handover has occurred, otherwise the value "0" is returned.

Another aspect of movement detection is the advertisement interval. If the mobile node has not received a router advertisement from its current router within a specified interval it may be because the mobile node has moved. There are additional functions located in mdetect.c that handle this aspect of movement detection. However, since these functions are not changed by the framework, there is no further need to discuss them.

Figure 4-3: Movement Detection

4.2.2 The Binding Update List

Before proceeding with the Handoff discussion it is essential to understand the Binding Update List. Recall that the home agent binds the mobile node's home address to its care-of address. Thus when a packet when arrives at the home agent bound for the home address, the mobile node will intercept and tunnel that packet to the care-of address.

The Binding Update is implemented as a mobility header that instructs binding agents of the mobile node's home and care-of addresses. Whenever either changes it must update each and every binding agent which maintained that binding; for Mobile IPv6 this means home agents and correspondent nodes. The Binding Update List does not only contain binding updates, but also bindings actively maintained by binding agents on behalf of the mobile node. The definition of a binding entry is shown in Figure 4-4.

bul.h

```
43    struct mipv6_bul_entry {
44      struct hashlist_entry e;
45      struct in6_addr cn_addr;      /* CN to which BU was sent */
46      struct in6_addr home_addr;    /* home address of this binding */
47      struct in6_addr coa;          /* care-of address of the sent BU */
48
49      unsigned long expire;         /* expiration time of this entry */
50      __u32 lifetime;               /* lifetime sent in this BU */
51      __u32 lastsend;               /* last time when BU was sent */
52      __u32 consecutive_send;       /* Number of consecutive BU's sent */
53      __u8 flags;                   /* BU send flags */
54      __u8 seq;                     /* sequence number of the latest BU */
55      __u8 prefix;                  /* Prefix length */
56
57          /* Session Key*/
58      struct mipv6_mh_opt *ops;     /* saved option values */
59
60      /* retransmission info*/
61      __u8 state;
62      __u32 initdelay;
63      __u32 delay;
64      __u32 maxdelay;
65      struct mipv6_rr_info *rr;
66      unsigned long callback_time;
67      int (*callback)(struct mipv6_bul_entry *entry);
68    };
```

bul.h

Figure 4-4: Mobile Node Binding Entry

43-47

The first item is an entry's location within the binding update list; all actions related to locating the correct binding are based upon the correspondent/home address tuple.

The remaining lines are the addresses associated with the binding. There is the address of the binding agent, while the other two addresses are the home address and care-of address used in this binding.

48-52

The lifetime is the duration for the binding requested by the mobile node, while the expiration indicates when the binding will terminate as assigned by the correspondent node. The other two values deal with retransmission while the mobile node is waiting for a response.

53-59

These values are preserved in case the binding update needs to be retransmitted. The first three are found within the standard header. Note that of all four, only the sequence field will change after the initial binding update.

The last one points to the list of options, such as the Alternate Care-of Address option, that are to be contained in the binding update message. Though MIPL provides support for these options, it does not actually use them in the binding update.

60-68

The remaining fields are additional retransmission data. The state indicates the status of the binding: accepted, waiting or rejected. "Accepted" indicates an active binding, and "waiting" means a binding update has been sent and the mobile node is waiting for an acknowledgement. "Rejected" means that the binding agent does not understand the registration request and future updates should be avoided.

Lines 62-64 control how long the binding update may be delayed before (re)transmission. We ignore line 65 because it deals with return routability, or binding with correspondent nodes. The last two items reflect the "lifetime" of the binding, and what should be done when this time runs out.

The binding update list is managed through three essential functions: `mipv6_bul_add`, `mipv6_bul_delete` and `mipv6_bul_get`. In addition to these functions, we will observe auxiliary functions associated to each entry through their callback fields.

4.2.3 Handover Processing

If the movement detection process returns true then the kernel will initiate handoff processing. To make this possible within the kernel requires modifications similar to those discussed in Section 4.2.1.

When a handoff occurs the mobile node must update every node that currently maintains a binding on its behalf. MIPL breaks this into two tasks: updating home agents, and updating correspondent nodes. The handoff process is shown in Figure 4-5. The left branch shows home agent processing, while the right shows correspondent node processing. Since the framework does not currently deal with correspondent nodes, further discussion about these types of binding agents will be kept to a minimum.

The kernel `calls mipv6_change_router()` to prepare the handoff structure, which contains data related to the previous and next access routers. The router structure was previously initiated during movement detection. However it also stores the care-of address used by the mobile node at that router. Thus, to complete the handoff structure, a new care-of address is formed here. Finally, since the previous router is no longer accessible, the curr_router value receives the new router and all paths to the old router are deleted. The handover structure is used by subsequent functions to build a binding update entry.

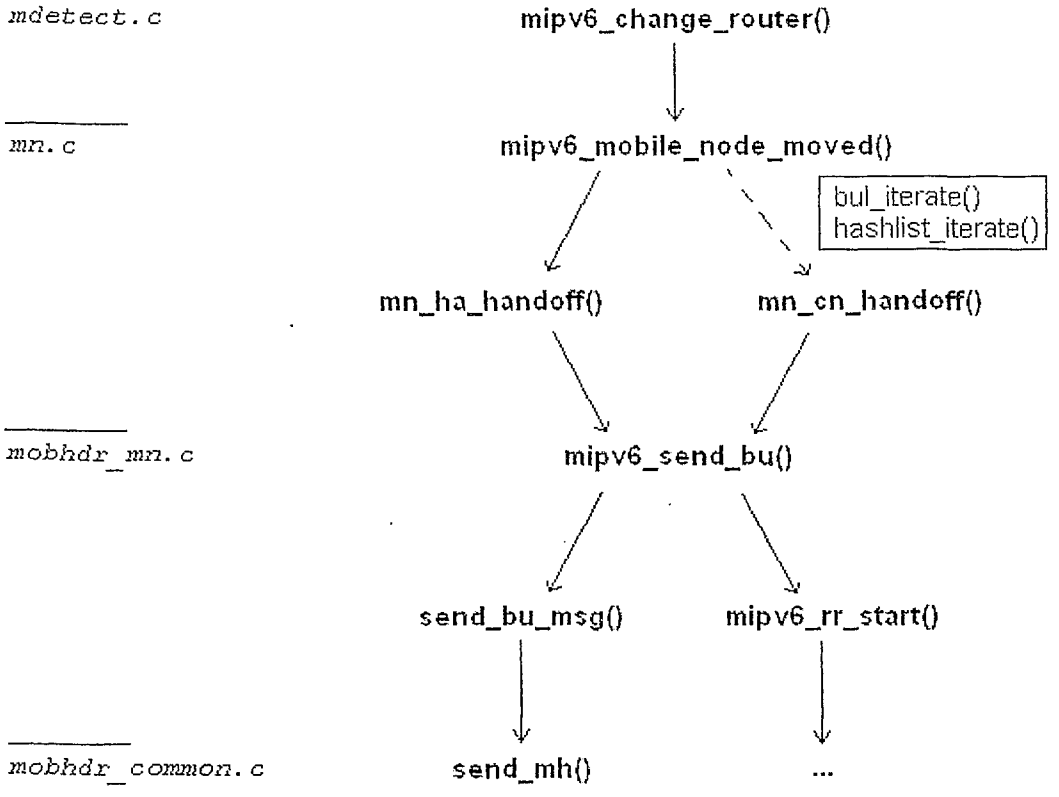


Figure 4-5: Handoff Process

The next function in the list, `mipv6_mobile_node_moved()`, is essentially a wrapper function to the following two functions, `mn_ha_handoff()` and `mn_cn_handoff()`.

Since the mobile node may have more than one home agent, handover processing must loop through each node. However, a mobile node can maintain only one home address per home agent and one care-of address per router. Therefore each home agent maintains only one binding on behalf of the mobile node. This relationship affects how the handover is processed.

For each home agent the function mn_ha_handoff() determines whether the advertised router is in fact that home agent. Note that when a mobile node returns home it still issues a binding update with the corresponding home agent. If the advertised router is not the home agent, and this is the mobile node's "first time" away from home, then `init_home_registration()` is called to prepare data for a new binding. If a binding already exists then it is fetched from the list. Most of the values remain constant with a few exceptions, most notably the care-of address.

The functions previously described within this section essentially assembled the data necessary to build a binding entry. This data is passed to the function mipv6_send_bu(), which either creates or updates an existing binding entry, and then determines the actions necessary to register that binding. If an existing binding has been rejected, no further updates for this particular binding should be sent. On the other hand, if too many updates have been sent within a given time frame then the mobile node should wait before sending another. If the binding is for a correspondent node, then the mobile node must initiate return routability. Return routability is not essential to the mobility of the mobile node, and thus analysis of this process is deferred to future projects.

Just prior to the sending of the binding update is the retransmission configuration. This is the first time we see the callback fields of the binding entry put into use. The callback_time registers when the callback function will be executed. The binding entry will point to one of the following functions based upon its status as described in Section 4.2.2:

- A handler to retransmit a request that has not yet been acknowledged.
- A handler to refresh a binding that is about to expire.
- A handler to remove an expired binding.

Since these functions basically influence communication with other nodes, we defer a more detailed description to the following section.

The function `send_bu_msg()` builds the binding update message based upon an existing binding. A structure resembling the format of the binding update header allows this function to easily build the mobility header. This header, along with the destination and source addresses are passed to `send_mh()`. While the latter function also accepts options, we mentioned previously that MIPL does not actually generate any.

The final function, `send_mh()`, is a general function that transmits mobility headers. Since we are only interested in the processing of binding entries, further discussion beyond this point is not necessary.

4.2.4 Binding Liaison

This module supervises correspondence with binding agents. Like any other protocol dependent upon IP, the mobility header must be encapsulated within an IP header. When demultiplexing, the IP layer must know to whom to pass each type of header. As such the mobility software must register itself with the kernel as shown in Figure 4-6.

mobhdr_common.c

```
1351  #if LINUX_VERSION_CODE >= 0x2052a
1352    if (inet6_add_protocol(&mipv6_mh_protocol, IPPROTO_MOBILITY) < 0) {
1353      printk(KERN_ERR "Failed to register MOBILITY protocol\n");
1354      sock_release(mipv6_mh_socket);
1355      mipv6_mh_socket = NULL);
1356      return -EAGAIN;
1357    }
1358  #else
1359    inet6_add_protocol(&mipv6_mh_protocol);
1360  #endif
```

mobhdr_common.c

Figure 4-6: Registering Mobility Protocol

As shown in Figure 4-7, mobility headers are passed to the function `mipv6_mh_rcv()`. Note that in the mobile node the primary responsibility of the Binding Liaison module is the handling of binding acknowledgements. There are additional mobility header types that the mobile node may encounter, such as the binding refresh request, however their function is not essential to the mobility of the mobile node and thus further discussion on these functions is not provided.

When a binding acknowledgement is received control is passed to the function `mipv6_handle_mh_ba()`. This function verifies that the binding acknowledgement is valid. If the authentication fails, or there are any undefined values, then the packet is dropped.

Once the binding acknowledgement has passed basic inspection, it is forwarded to the function `mipv6_ba_rcvd()`. If the status indicates an error the binding is deleted. Otherwise the binding is set to active status, and the lifetime is set to the value assigned by the binding agent. Once the callback function is configured the registration process is complete.

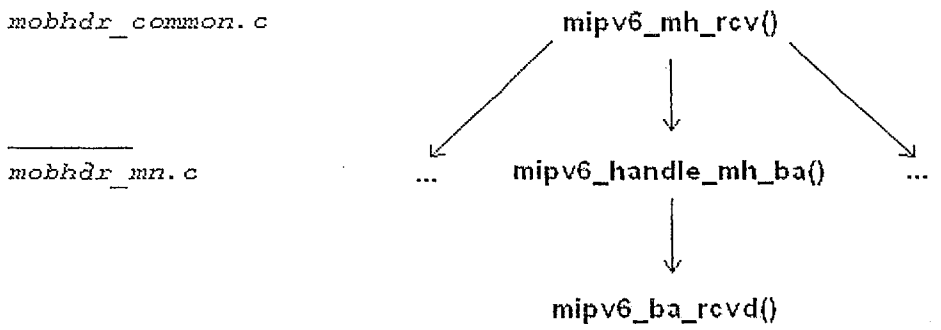


Figure 4-7: Binding Liaison Process

The callback function is set to bul_refresh() and the callback time is set to 4/5 of the lifetime. This gives the mobile node a small window before the binding expires to tell the binding agent that it is still using the same care-of address, and as such the current binding should be maintained.

In Mobile IPv6 each binding update entry has only one possible destination, which is the binding agent that registers the care-of address. However, due to the advent of local registration, the mobile node may correspond with two or more binding agents for a single forwarding path. Therefore the mobile node must be able to determine which of these nodes to send a binding update at each step during the handover process.

4.3 The Mobile Node

The model shown in Figure 4-8 visually depicts the components in each of the mobility agents. Readers will observe that certain modules exist on one or more agents. This does not necessarily mean that this module operates identically on each and every agent. Instead it may indicate that each mobility agent participates in the overall activity.

For example, both the home agent and foreign transmit domain advertisements as part of movement detection. However, only the mobile node is responsible for processing these messages to detect a handover event. On the other hand, each process maintains its own binding update list which does not directly interact with the equivalent list of any other process.

This section describes the changes necessary to make the mobile node framework compliant. Each subsection will discuss one of the modules required to operate this node.

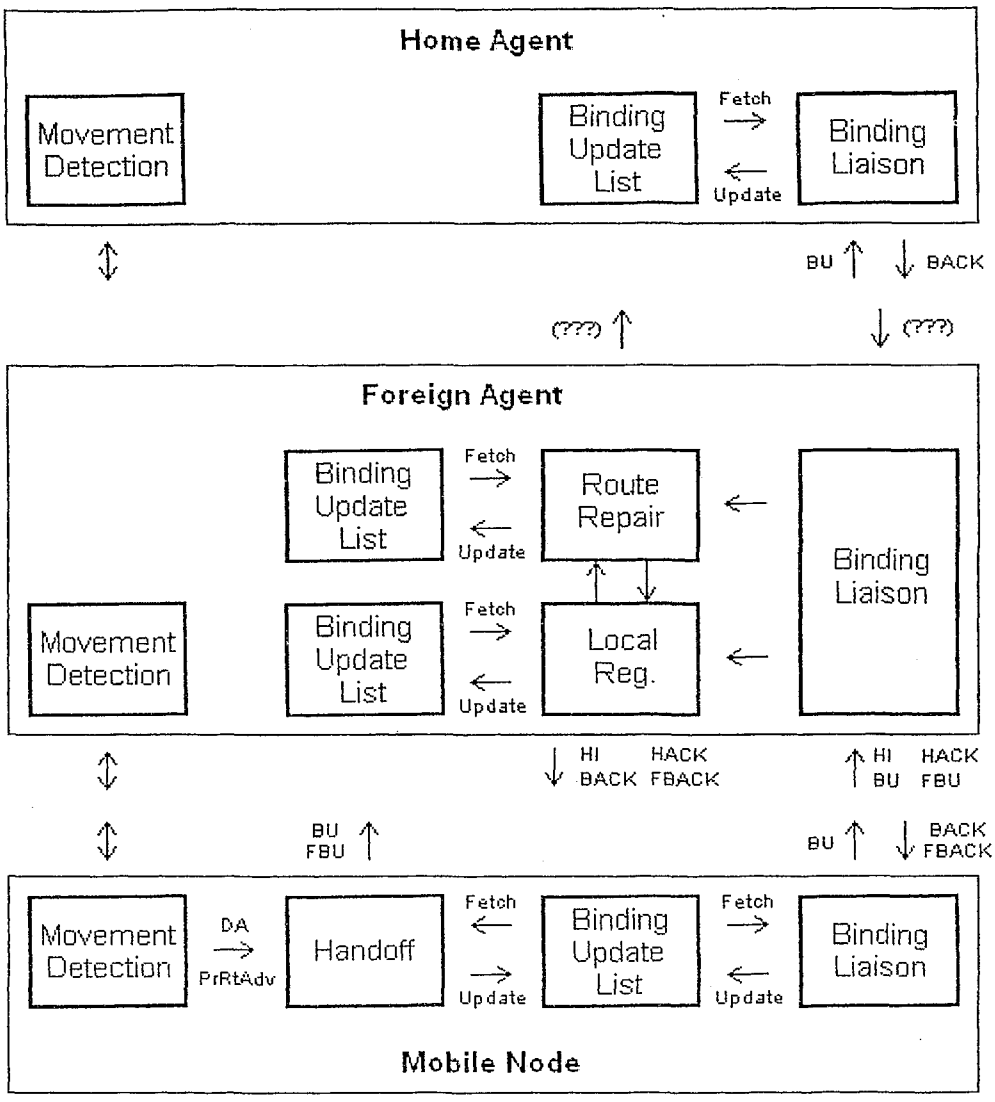


Figure 4-8: Modular Decomposition of Mobility Agents

4.3.1 Movement Detection

There are no significant changes to the processing of router advertisements as the domain properties do not directly affect movement detection. However the advertisement is parsed into the router structure, which is later used for handover processing.

Thus we must make two small modifications to the movement detection process to correspond with the changes we had made to the router advertisement (radvd). The first is a change to the router structure to store the domain specific attributes. The second change is to the function mipv6_mn_ra_rcv() to parse these additional attributes into the new structure. The modified router structure is shown in Figure 4-9.

mdetect.h

```
52    struct router {
53      struct list_head list;
54      struct in6_addr ll_addr;
55      struct in6_addr raddr; /* Also contains prefix */
56      __u8 link_addr[MAX_ADDR_LEN]; /* link layer address */
57      __u8 link_addr_len;
58      __u8 state;
59      __u8 is_current;
60      int ifindex;
61      int pfix_len; /* Length of the network prefix*/
62      unsigned long lifetime; /* from ra */
63      __u32 last_ns_sent;
64      __u32 last_ra_rcvd;
65      __u32 interval;
66      int glob_addr; /*Whether raddr contains also routers global address*/
67      __u8 flags; /* RA flags, for example ha */
68      struct in6_addr CoA;
69      int extra_addr_route;
70      __u32 lca_identifier; //for framework and LMA selection
71      char scene; //scenario advertised by router, quick reference
72      struct router *next;
73    };
```

mdetect.h

Figure 4-9: Modified Router Structure

4.3.2 The Binding Update list

Since the remaining modules operate upon the binding update list the first priority is to make the binding entry and related administrative functions framework-ready. However the only changes required are due to the need to support local mobility management.

If each binding was to represent a single care-of address we would require some mechanism to indicate the relationship between the local and regional care-of address. It would be much simpler to contain both addresses within a single binding. The range of the state field is expanded to include the status of both regional and local registration. Finally, the binding should also contain the foreign agent's link local address so that the mobile node will know where to direct binding updates for this entry.

Since only two fields are added to the binding entry structure there is little need to review the code here. Furthermore, only a small change is required to `mipv6_bul_add` to compensate for the extra fields.

4.3.3 Handover Processing

There are two fundamental differences in handover processing between Mobile IPv6 and the Framework for IPv6. The first difference is the mobile node's ability to handle different registration requirements. Secondly, as a result of this ability, is the mobile node's dependence upon the previous foreign agent.

We have observed in the function `mipv6_change_router()` that the handoff structure already contains information about the previous router. This is a fortunate coincidence because MIPL never actually uses this information. Also in this function is the creation of a care-of address for the new router. We let the mobile node form the care-of address and defer to subsequent functions to decide whether or not to actually use this value.

Once the handoff structure is completed control is passed to `mipv6_mobile_node_moved()`. Recall that this function is merely a wrapper function for `mn_ha_handoff()` and `mn_cn_handoff()`, and that the latter is a process that we are not currently interested in. Therefore we move our attention to `mn_ha_handoff()`.

This function checks if the mobile node is either at home or returning home, or whether the mobile node is away. If the mobile node is at home it either deregisters or does nothing. If the latter is true, then the mobile node assembles the data necessary to make a binding entry.

Regardless of the mobility protocol deployed in the access network the process of returning home remains constant. Thus no modification is required with respect to deregistration, i.e. Home Registration. However the process of Local Registration is heavily dependent upon the hosting foreign agent.

It is this portion of `mn_ha_handoff()` that calculates the values that are assigned to the binding update. Recall that Scenario A corresponds to Mobile IPv6, and as such no modifications are necessary to support the mobile node in this state. To preserve this code we detour the remaining scenarios to `mn_cef_handoff()`, which directly calls `send_bu_msg()` once it has assigned the appropriate values to the binding entry.

Note that the function `send_bu_msg()` constructs and transmits a binding update mobility header based upon the contents of a binding entry. As such it is called from many places within the mobility software. It may be called here during handover processing, it can be called during retransmission or it may even be called upon the reception of the binding acknowledgement to retransmit the update due to an error.

We mentioned previously that individual binding entries in MIPv6 correspond to a single binding agent, which basically requires `send_bu_msg()` to be a "copy and paste" function. The framework introduces an additional care-of address and binding agent to the entry. Thus we modify this function to assess where the binding update should be sent, and what values should be put into it. To achieve this it inspects the state of the binding entry to determine whether to send the binding update to the foreign agent to initiate local registration, or the home agent for home registration. This in turn determines the values assigned to the binding update.

4.3.4 Binding Liaison

When the mobile node receives a binding acknowledgement it sets the appropriate binding entry to active and the registration process is complete. This does not hold true for the Framework.

In the Framework the binding acknowledgement might only indicate the completion of the first step in a two step process, as under certain conditions the mobile node may require both local and home registration. Thus, when `mipv6_ba_rcvd()` updates the binding entry's status, it must check the state for whether further action is necessary. If registration is not complete then `send_bu_msg()` must be called to generate a new binding update based upon the entry's new status.

4.4 The Foreign Agent

Figure 4-8 depicts a simplistic model of the foreign agent. However, to achieve our objective, we are only interested in the components that support local registration. This is due to the fact that route repair may be any one member of an indeterminate set of protocols. This is in fact the second major benefit of the framework.

On the other hand we must be aware of route repair in the manner of how it interfaces with local registration. There would be little transparency between processes if we needed to redefine the interface for every new protocol.

This narrowing of focus resolves to the following modules: binding liaison, local registration, the binding update list for local registration and movement detection. However we have already covered the topic of movement detection in Section 4.1, such that no further discussion is required here.

4.4.1 Binding Liaison

The binding liaison on the foreign agent is merely the demultiplexing process described in Section 4.2.4. The IP layer passes the mobility header to this module, and from there it is determined whether the message should be forwarded to either local registration or route repair.

This requires code within the kernel, otherwise the mobility software on the foreign agent would have to snoop every IP datagram for those containing mobility headers. This process is similar to that described in Section 4.2.4, while the demultiplexed functions specific to the foreign agent are shown in Figure 4-11.

4.4.2 The Binding Update List

Each process has its own binding list that maintains information pertinent to that process's needs. In previous sections we have discussed the binding update list supported by the mobile node. It is a record of the care-of addresses it used and which binding agents supported them.

fa.h

```
23    struct fa_entry {
24      in6_addr mn;              //address of mn
25      __u16 sequence;           //sequence used in this entry
26      int f_repair;             //repair request status
27      int f_rcoa;               //rcoa request status
28      int f_coa;                //coa request status
29      in6_addr status;          //result of route repair
30      __u16 mn_seq;             //sequence used by mn
31      __u32 lifetime;           //lifetime for binding
32      struct in6_addr coa;
33      struct in6_addr lcoa;
34      struct bul_entry *next;
35    };
```

fa.h

Figure 4-10: Local Registration Binding Entry

Local registration requires its own binding update list to keep track of the progress of all current registration requests. The framework provides a conceptual structure of an entry in this list; this structure is shown in Figure 4-10. The draft also explains how it is to be used by local registration, which is discussed in the following section.

Administrative functions associated with the binding list do not change with the nature of the list. There is still the need for adding, deleting and fetching entries, which are already available in MIPL that we can reuse.

4.4.3 Local Registration

The operation of local registration is relatively simple compared to handover process maintained by the mobile node. This simplicity is due to the fact that Local Registration behaves the same regardless of the scenario deployed at that foreign agent. The control flow of this process is shown in Figure 4-11.

Local Registration can receive input as either a request from Simple Handoff, or as a response from Route Repair. Messages arriving from outside of the foreign agent will always be demultiplexed from IPv6 to the function `mipv6_mh_rcv()`. This function inspects the mobility header for further demultiplexing; for our purposes we are only interested in the functions `mipv6_handle_mh_bu()` and `mipv6_fa_handle_ba()`.

The former processes binding updates, which should be a registration request from a mobile node. Once the mobility header has been parsed into a structure it is passed to another demultiplexing function: `mipv6_bu_add()`.

This function checks the type of registration that has been requested and calls the appropriate function. Depending upon the configuration this allows a home agent to also act as a correspondent node, or it allows correspondent nodes to transmit binding acknowledgements indicating that home registration is not supported. We add a third function for local registration: `mipv6_bu_add_fa()`.

The third function creates a binding entry for the request, ignoring any binding updates from mobile nodes for which the foreign agent is already processing a request; once this is done registration is handled by the function `mipv6_fa_handle_reg()`. The entry is inspected for each request that has not yet been processed and polls `mipv6_fa_handle_repair()` for

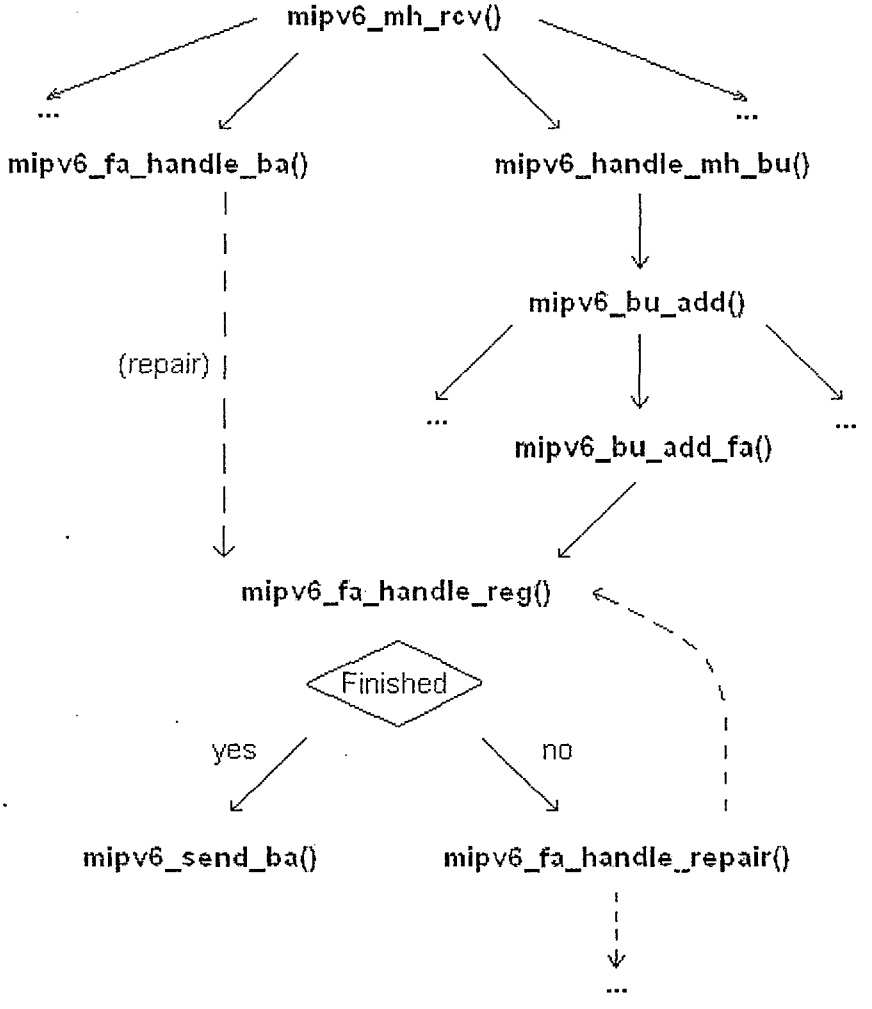


Figure 4-11: Local Registration

answers. If there is a reply it is stored in the entry and the next request is processed. However, the lack of a response from route repair indicates that the service was not available within the foreign agent and that action was required externally. At this point control leaves the mobility module and registration is resumed when Route Repair calls `mipv6_fa_handle_reg()`. Route Repair is activated by an inbound mobility header, which can be any header defined by the Route Repair protocol but would most likely be a binding acknowledgement.

Once all of the individual requests within the entry have been processed a binding acknowledgment is created for the mobile node. The entry is deleted from the binding list and Local Registration is complete.

4.4.4 The Route Repair Interface

We have observed that the interface between Local Registration and Simple Handoff requires the exchange of messages external to the node that supports them, while the interface between Local Registration and Route Repair is internal to the foreign agent. There is no need to construct a mobility header to be sent across the network.

Instead we define the Route Repair API, which currently contains a pair of well-known functions to handle the exchange of messages; these functions are called `mipv6_fa_handle_reg()` and `mipv6_fa_handle_repair()`. The former is called by Route Repair by passing a single structure containing the results of a requested action. The latter is called by Local Registration by passing two arguments. The first argument contains the request, while the second will point to the response.

4.5 Fast Handovers

We have previously discussed the operation of Fast Handovers in Section 2.5. That section describes the additional messages required to support fast handovers. We have also discussed the changes necessary to support fast handovers within the Framework in Section 3.10. It would not be too difficult to implement this functionality into the mobile node and foreign agent.

What has not been explained is the nature of how Mobile IP anticipates a handoff. Mobility Headers are a subset of IP, thus they operate on Layer 3. However a handoff is a change between physical links, a process that is controlled by Layer 2 (L2). Thus there must be a means of indicating anticipation, i.e. a trigger, from Layer 2 to Layer 3.

Currently there is no support in Linux device drivers for anticipating a handoff; developing these triggers is a project unto themselves. On the other hand MIPL uses **sysctl** to pass relevant arguments to the mobility module, such as a statefully configured care-of address provided by DHCP. This system can be used to pass simulated L2 triggers to the module.

4.6 Test Case

Our test case comprised of a mobile node and foreign agent. We did not need to test the relationship between the home agent and the mobile node as this operation was not altered by our implementation; it is simply required to prove that this operation is initiated as described below. Test results were recorded using Ethereal and the Linux kernel logs. Running Ethereal on each node demonstrates the control flow entering and leaving that node. Kernel logs record messages printed by the application which, while primarily used for debugging, indicate the occurrence of key events within a specific node. Using these tools we demonstrate the behaviour shown in Figure 4-12; note that this is only one mode of operation within the Framework and we do not reproduce all of them in this report.

Steps [1,2,7] were captured using Ethereal which verify the relationship between the mobile node and foreign agent. Steps [3-6] were recorded within the kernel logs verifying the Route Repair API. Finally Step 8, recorded using Ethereal, demonstrates that the mobile node initiates home registration after local registration is complete.

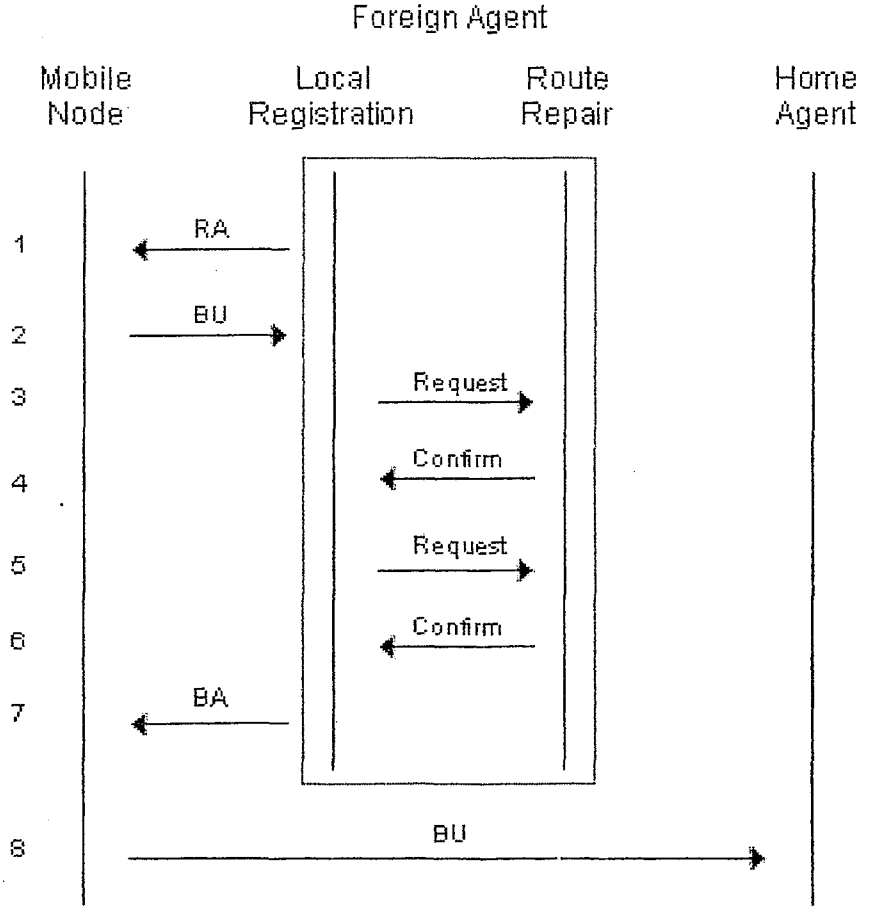


Figure 4-12: Test Case

4.7 Summary

The second objective of the thesis was to create a prototype of the framework designed in Chapter 3. This prototype was to contain the basic elements required in the framework to provide mobility to the mobile node.

These elements consisted of home registration, local registration and simple handoff. Common themes in Mobile IP that were not included are fast handovers and route optimization via correspondent nodes.

The selected components require support from the home agent, foreign agent and mobile node. Fortunately there is mobility software available under a public license that can be reused to build the Framework. This software is known as MIPL and it provides the Mobile IPv6 protocol. This means that it has implemented the home agent and mobile node.

Since the framework does not require changes to the operation of the home agent no work was necessary on this node. However the mobile node required modification to all areas of its operation to make it framework-ready.

While the foreign agent is not provided for in Mobile IPv6 there is Neighbour Discovery software also available under a public license. The application is called radvd and, with some minor changes, provides domain advertisements for both the home and foreign agent. However the binding agent portion of the Foreign Agent had to be developed from scratch.

Chapter 5 Concluding Remarks

Mobile IP is a proposed standard that allows mobile nodes to retain IP connectivity while roaming between subnets. In Mobile IP the mobile nodes retain their home addresses but change their care-of addresses while roaming from one subnet to another. The home agent intercepts packets destined to a mobile's home address and redirects them to its care-of address. Due to high registration latency of care-of address with the home agent Mobile IP is used for inter-domain mobility. Additional protocols are defined to manage mobility within a local mobility domain. Also enhancements to Mobile IP are defined to achieve seamless handover. These additional protocols serve either as alternatives or as enhancements to Mobile IP. Unfortunately these improvements cause their own problems with respect to deployment. Currently, if a mobile node is not upgraded to match the protocol supported by the access network then it cannot communicate with that network. This is a problem because the number of mobile nodes can number in the hundreds for private networks. For service providers they could number in the hundreds of thousands. It becomes especially problematic when the networking stack is implemented in hardware, such as for cell phones.

In this thesis we designed a Mobility Management Framework that solves the above problem by decoupling the mobile node from the route repair within the access network. This is primarily achieved by re-introducing the concept of a Foreign Agent to serve as an interface between the mobile node and the access network. The framework is based on the model analogous to Internet routing system. In Internet hosts interact with the network through DHCP and DNS and do not interfere with the routing infrastructure such as OSPF and BGP. In our framework, we defined a standard handover signaling based on Mobile IP and Fast Handover. We defined some extensions for Local Mobility Management. This interface makes the method of mobility within the access network transparent to the mobile node. Thus new protocols may be implemented and deployed with no impact upon the mobile node. The framework allows any mobility management solution to perform route repair within the access network. The framework provides wireless ISPs great flexibility in deploying mobility management solution that is optimal for their network without engaging mobile users. It also provides flexibility to the vendors of mobility management solution for they are not required to engage terminal vendors in the development and deployment of their solutions. It also saves terminal vendors from

customizing their terminal software and hardware for each mobility solutions. In summary, it facilitates all parties involved in the development and deployment of mobility solutions.

In our framework we introduced the abstraction of mobility domain to define the scope of a mobility solution. We modified the router advertisement (or mobility agent advertisement) to include Local Coverage Area Identifier (LCA) extension. The LCAI identifies a mobility domain. Our framework supports both tunneling-based and routing-based mobility solutions within a mobility domain, hence it allows a mobile node to either change or retain its care-of address within the mobility domain depending upon the solution employed within the domain. We also presented an example design of inter-working of the framework with HMIPv6.

We demonstrated the feasibility of our framework by implementing a prototype of the framework. To achieve this we reengineered a commonly used Mobile IPv6 software that is available as a freeware. The prototype contains the basic elements required to support mobility, and as such does not support optional behaviour such as route optimization or fast handovers.

Another significant outcome of this work is the creation of an Internet Draft (ID) to specify the operation of a Mobility Management Framework. This ID is used to develop an implementation of that Framework to demonstrate its properties. The implementation of route repair within the framework is an ongoing activity in the MoWIN lab. And in future when at least two route repair processes are implemented, then we can submit the ID to the relevant IETF working group for discussion and possible standardization.

Appendix A

This document is based upon an internet draft titled "Mobility Management Framework". This draft is a work in progress; the version which existed at the time this document was written is contained on the following pages.

B. Hartwell
Ryerson
M. Jaseemuddin
Ryerson
February 2004

Mobility Management Framework for IPv6

draft-hartwell-framework-00.txt

Status of this Memo

Abstract

This document describes a mobility management framework which defines a "layer architecture" for mobility solutions. This primarily serves to make domain solutions transparent to the mobile node. To achieve this, the framework proxies registration on behalf of the mobile node.

Domain transparency enables different protocols to be deployed within a network with no impact upon the mobile node. This is desirable to network administrators since domain infrastructure is minimal and accessible, while mobile nodes are numerous and have limited accessibility.

Table of Contents

1. Introduction

Several protocols exist, either as proposed standards or works in progress, that "allows nodes to remain reachable while moving around in the IPv6 Internet." [2] This document identifies a common problem among these protocols, and proposes a new protocol to overcome this obstacle.

1.1 Problem Statement

One constant between each IP mobility protocol is the concept of the care-of address, and that these care-of addresses must be registered with a mobility agent. Each protocol differs in the nature of the care-of address and how it is registered, but every protocol treats registration as a single process. In effect, registration is non-layered.

This means that a new protocol will have an impact upon every mobility agent. Even if the protocol had minimal changes in terms of code, it would need to be distributed to every node involved in the registration. This includes mobile nodes that, in a worst case scenario, could be cell phones that can not be upgraded. At best it would be many laptops that have limited accessibility compared to the router infrastructure.

1.2 Layered Registration

This document proposes a framework which layers the registration process. This is similar to the TCP/IP stack where similar functions are grouped into a single layer. When better code for those functions are available, only they need to be upgraded. When a new protocol is designed, only those functions need to be replaced.

Note that the framework manages IP mobility protocols, and thus it operates solely within the IP layer. It does not dictate routing procedures, and as such does not require changes to IP. In terms of mobility the framework only directs the registration of care-of addresses; how each protocol uses that care-of address for routing is of no concern to the framework. It is a binding process only.

A mobility layer architecture must be designed such that at least the mobile node remains isolated. The details of how the layers inter-operate is defined in later sections. This section is dedicated to discussing the design of the layer stack, as shown in the following diagram.

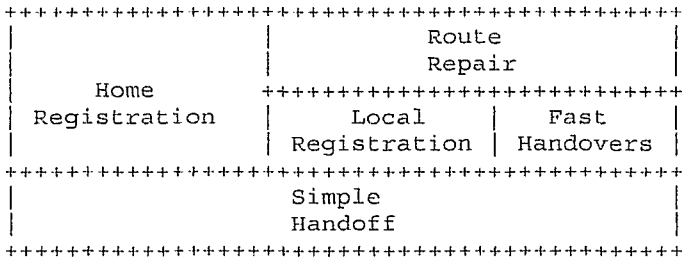


Figure 1

Before proceeding the authors wish to point out that L3-Repair (Route Repair) should extend across the stack but, due to implementation reasons, Home Registration is simplified into a single layer. Perhaps future versions of the framework will provide for a division in functionality (and at no cost to the mobile node).

Careful readers may observe that the upper layers directly correspond to routers. Based upon this observation the reader may determine that Simple Handoff layer reflects the mobile node. As an example, Simple Handoff exchanges messages (BU/BACK) with the Registration layer. How Registration processes these messages is specific to that layer only, and a change in implementation has no deployment issues with respect to other layers/protocols.

Yet why bother to make a distinction since in un-layered registration this functionality already resides in separate nodes? Obviously the internal workings of the home agent has no impact upon that of the mobile node, such that a layer for route repair here would be purely abstract.

The advantage becomes apparent when one considers how registration is handled within the Access Network. If an AN was upgraded from MIPv6 to HMIPv6, to gain the benefits mobile nodes would need to understand the MAP option [12]. This requires a corresponding upgrade in mobile nodes. Though MIPv6 mobile nodes would not gain any benefits, these two protocols may inter-operate. However, other upgrades may prevent route establishment altogether.

The goal then is provide a constant interface to the mobile node, making the inner workings of the AN transparent. The natural interface is the Access Router.

Home Registration is primarily hosted by the home agent, while Local Registration and Fast Handoff are primarily hosted by access routers. Due to this relationship these protocols reside within a single layer in parallel. This is for security reasons; if control had to pass through an access router to a home agent then a security relationship must exist between these two nodes.

Providing this security takes a long time to establish, and would likely exist for a short period of time. To avoid this registration begins with one protocol and finishes with the other, requiring only security relationships between the mobile node and its correspondents.

Finally, the reader should consider that this is a work in progress in its early stages. There are many issues that remain to be addressed; the current goal of this draft is to establish a conceptual foundation towards a new approach to IP mobility through the design of a basic framework.

This framework is meant to replace MIPv6, least its design becomes flawed by inter-operability issues. However this document attempts to reuse as many terms, concepts and message formats as possible to facilitate a conceptual transfer between protocols. It also helps for code reuse. If some compatibility for MIPv6 mobile nodes is possible, all the better, but this is not a design objective.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119.

This document defers to the Mobility Related Terminology draft for definitions. Further terminology is referred to in MIPv6[2], LMM[4], and FMIPv6[6] documents, with exceptions as noted.

rCoA - Regional Care-of Address

A care-of address that is globally routable. This address may reside directly with the mobile node, or may be used in some intermediary manner to reduce the effects of micro-mobility.

lCoA - Local Care-of Address

An address assigned to the mobile node used by the local domain to direct packets within the access network.

mCoA - Mobile Care-of Address

A generic term for the care-of address that terminates at the mobile node itself. Thus it can be either regional or local.

DA - Domain Advertisement

A modified router advertisement specifying the mobility properties of the access network.

FA - Foreign Agent

An access router that participates in route establishment on behalf of the mobile node. The access router is mobile aware.

TA - Tunneling Agent

A forwarding node that can (de)tunnel packets bound for the mobile node. This acronym may be prefixed by the type of router: F (foreign), H (home) or L (local).

BA - Binding Agent

A node that participates in route establishment for mobile nodes. This could a Home Agent (HBA), Foreign Agent (FBA), etc. The BA may also be a TA.

L1-Reg

This layer processes Domain Advertisements and manages the Binding Update List.

L2-Reg

This layer acquires and registers care-of addresses upon request. It does not activate registration, nor does it garuantee successful registration.

L3-Repair

This layer maintains routes for care-of addresses. While it does not initiate repair, it may monitor status and issue warnings. It does not garuantee the existence of a route.

MMF - Mobility Management Framework

A layered process of registering care-of addresses. The framework DOES NOT dictate how those addresses are to be used.

Base Handover

Used in context of, and refers to, Fast Handovers [6].

Local Handover

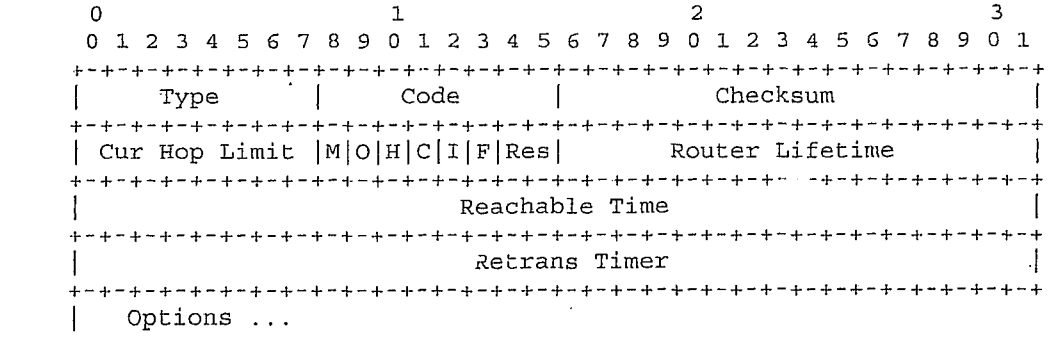
A fast handover that provides transient route repair [7].

3. Domain Discovery

This section introduces the concept of a Mobility Domain Advertisement (DA), which is simply a modified version of the Router Advertisement described in [2] [9]. Each FA must periodically transmit domain advertisements.

3.1 Domain Advertisement Message Format

The DA is a Router Advertisement [2] [9], with an additional two bits defined. A third bit for fast handovers is also provided.



Care-of Address (C)

This domain requires the MN to change its COA.

Local Registration (I)

This domain supports local mobility management.

Fast Handover (F)

This router supports fast handovers as described in this document.

Reserved (Res)

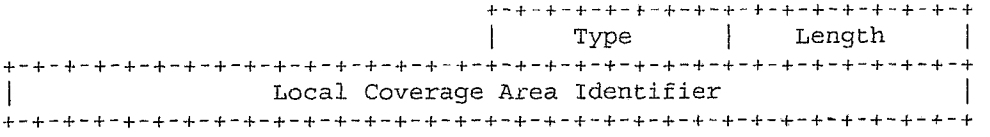
Reduced from a 5-bit filed to a 2 bit field to account for the additional bits.

In IPv6 care-of addresses are co-located. Thus the FA does not need to act as a TA, and in turn does not need to advertise a care-of address.

Furthermore the DA does not contain rCoA information, for example the MAP option. Currently this is considered protocol dependent - the mobile node acquires a rCoA through registration. Future work may consider how to propagate rCoA prefix for autoconfiguation.

3.2 Local Coverage Area Option Format

This extension helps the MN identify when to register a new rCoA with the HA. If the domain supports local registration it MUST include the LCA extension in its DA.



Type 10

Length The length of this extension in bytes, which is always 6.

Identifier A semi-unique binary identifier that indicates which registration LCA this FA belongs to.

The 32-bit identifier provides a unique identity within a geographical area, with values being recycled between areas. Note that this document does not require each FA to be uniquely identified, only the LCA that they belong to. The use of the identifier is described in detail in Section 6.

3.3 Additional Options

Other options, as specified in Section 7 of MIPv6[2], MAY be included in the DA as appropriate.

4. Generic Registration: An Overview of the Framework

Domain solutions are not entirely compatible with each other. Some support LMM, others allow a non-changing co-located COA [13]. A MN aware of only one domain-type will typically not be able to register when roaming to a different domain-type.

How a particular protocol in the framework delivers packets to the mobile node is irrelevant. What is important is that the mobile node configures the appropriate types of care-of addresses sponsored by the access network and registers them. The framework is a binding process only.

Thus the Domain Advertisement is more than a means of movement detection. It also states the properties of the access network in the form of the C, I and F flags. Generic Registration uses this to determine which one, some or all, of the upper layer protocols to activate: home, local or fast registration.

4.1 Domain Transparency

The mobile node doesn't need to understand what a MAP is, only that a regional care-of address and local care-of address needs to be configured. This way another protocol can provide the same service without requiring upgrades to the mobile node.

4.1.1 Registration Scenarios

To accomplish this all of the various "services" must be categorized. The following figure shows a list of possible topologies, along with the care-of addresses used to provide mobility.

A. MN-HA	changing co-located rCoA	C=1,I=0
B. MN-FTA-HA		C=1,I=0
C. MN-(LMM)-HA	rCoA + changing co-located lCoA	C=1,I=1
D. MN-FTA-(LMM)-HA		C=1,I=1
E. MN-HA	non-changing co-located rCoA	C=0,I=0
F. MN-(LMM)-HA	rCoA + non-changing co-located lCoA	C=0,I=1

Figure 2

Every scenario has a mobile node, home agent and foreign binding agent (the last of which is now shown for simplicity). More advanced protocols may introduce intermediary steps, such as local mobility management (LMM). Note that FTAs do not exist in IPv6; scenarios B and D are shown here only for design issues and are not further discussed in following sections.

The middle column indicates the types of care-of addresses made available by that scenario. Careful readers will observe similarities between these addresses, which are translated into the C and I flags shown in the third column of Figure 2.

4.1.1 The C Flag

When the MN receives a DA with the C bit set it MUST change its mCoA. This would occur when the access network uses prefix-based routing as part of the mobility solution. Otherwise, if the protocol allows the mobile node to keep its current care-of address, then the C bit MUST be set to 0.

The significance of C=1 is that the mobile node must register its care-of address with either a home agent or lmm domain after every L3 handover. This may be either, or both, home and local registration depending upon the I flag.

Aside from initial setup, C=0 never needs to update its home agent after each L3 handover. However route repair is still required, and this is considered a form of local registration.

4.1.2 The I Flag

The I bit indicates the presence of local mobility management. When the flag is set LMM is active, and when '0' LMM is absent.

"LMM schemes allow the Mobile Node to continue receiving traffic on the new subnet without any change in the Home Agent or Correspondent Node binding." [4]

When absent registration is fairly straight forward, simply update your correspondents when your rCoA changes. However things change when the rCoA remains relatively constant.

For LMM domains the framework mandates the existence of a rCoA and lCoA. The mobile node always registers the rCoA with its correspondents, but in the case of LMM the rCoA directs traffic to the Access Network instead of the mobile node. The AN then uses the lCoA to direct traffic to mobile node within its domain.

4.2 Initiating Registration

4.2.1 Registration Events

L1-Reg can be described as a state machine, though it need not necessarily be implemented that way. Visualize four states with each representing one of the topology scenarios described above. Technically other states are possible but they do not need to be discussed at this time.

Figure 3 shows the proposed state machine. A transition occurs when a Domain Advertisement arrives at the mobile node, and this produces a Registration Event.

Note that the MN can start in any state, as well as directly transfer to any other possible state. The event for "initial state" and any outside transition to that state are identical.

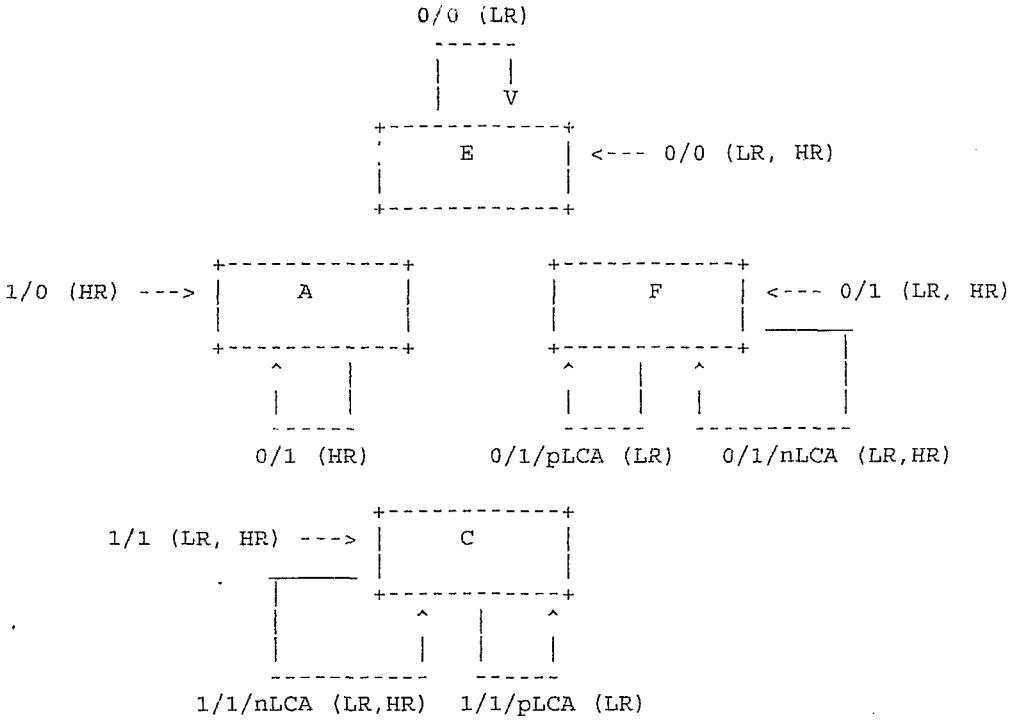


Figure 3

The legend associated with this table is as follows:

#/#	C/I pair	input
pLCA | nLCA	previous | new LCA	input
LR	Local Registration	(output)
HR	Home Registration	(output)

As an example we'll consider the most complex transition. The mobile node rests in State (scenario) A. It receives a DA with the C/I pair of 0/1. The DA must also advertise a LCA Identifier, that in this case is new since the previous state had none.

As a result the mobile node transfers to State F and as it does so it initiates Local Registration (L2-Local), followed by Home Registration (L2-Home).

C	I	scenario	local registration	home registration
0	0	E	every time	first time only
0	1	F	every time	new LCA
1	0	A	never	every time
1	1	C	every time	new LCA

Figure 4

Figure 4 shows an alternative way of viewing registration events. Note that L2-Home refers to consecutive advertisings of the same C/I pair. If 0/0 was advertised followed by 1/0, returning to 0/0 would again require L2-Home for the "first time". Conversely, a "new LCA" does not mean a never previously discovered LCA, but rather a LCA that is different from the one advertised by the previous Foreign Agent. It is a (n-1) relationship only.

4.2.2 Flow of Control

When the state machine "outputs" only one of the L2-Reg protocols, flow of control is simple. The mobile node directs a Binding update to that protocol and waits for a reply. If it was a success then the process is complete.

Yet when both L2-Local and L2-Home is required to maintain a valid binding, then the mobile node MUST first initiate L2-Local. Only after this process has been completely determined as either a success or failure may the MN initiate L2-Home.

The purpose of this arrangement is as follows:

a) In L2-Home the MN registers rCoA, which it acquires during local registration (L2-Local).
b) If the HA registration fails, the MN can still converse with correspondent nodes (via rCoA). The MN may also converse within the domain using lCoA.
c) It is a "two-shot" process since a single pass would require additional security resources.

1 L2-Local fails then the MN can still attempt the "default registration", where the mobile node partially pretends that it is in Scenario A. The difference is that the BU must contain a Type 0 Routing Header for a stopover at the foreign agent. This allows the foreign agent to inspect the BU to make sure it does not violate any properties of the domain, such as private space.

5. Framework Message Types

The framework uses message formats defined in MIPv6[2] for reasons described above. Some have been modified to adapt to the needs of the framework, while in other cases new options are defined. This section describes the message formats used by the framework.

5.1 Mobility Header

The Mobility Header is an extension header used by all nodes which support mobility to create and manage bindings. All framework options are contained within this header. The basic format of the header is unchanged from MIPv6[2].

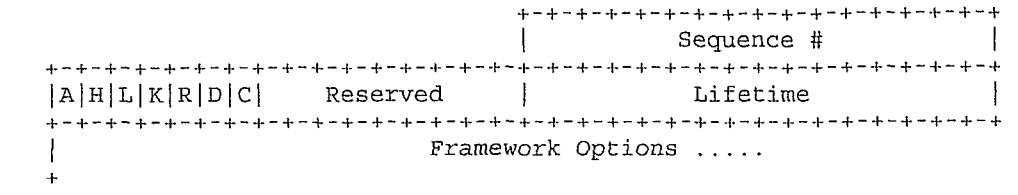
5.1.1 Unmodified Mobility Headers

This version of the framework does not discuss, in detail, binding management with correspondent nodes. Thus the following message types are not mentioned: HoTI, CoTI, HoT, CoT, the Binding Refresh Request or the Binding Error.

There is no change required for the Binding Acknowledgement message within the framework.

5.1.2 Binding Update Message

The mobile node uses this message to notify other nodes of a new rCoA. Three bits have been added to this message to support binding proxy by the foreign agent. How this message is used is described in Section 6.1.



Local Route Repair (R)

The mobile node requests route repair within the domain by setting this bit.

Regional Care-of Address Request (D)

This bit must be set to request a new rCoA within a LMM domain. Otherwise it must be 0.

Care-of Address Request (C)

Used to request an acceptable CoA from a domain which allows the CN to retain its CoA between handoffs.

Reserved

Adjusted to an eight-bit field from a 12-bit field to account for the four flags defined above.

These bits are set by the mobile node to direct the foreign agent on the type of action to take on its behalf.

5.2 Framework Options

No new options are defined, but a new option may be specified in subsequent documents.

6. Mobile Node Operation

The framework describes the process of registration between the MN and AR. Section 4 described when to perform home or local registration. This section describes what the MN should do when the required registration type has been determined.

6.1 The Binding Update

The MN must add the BU mobility header immediately after the IP header. The values set in the BU and extensions added vary between scenarios, so each will be discussed separately.

In each case, the MN requires an acknowledgement and as such MUST set the A bit. Unless stated otherwise, each flag in the BU must be set to 0.

6.1.1 Scenario.A

After receiving the DA, the MN MUST acquire a new CoA. This scenario does not require local registration, thus the MN sends a BU only to the HA (dst. address). The source address is the MN's home address. The H bit MUST be set to 1.

The BU must contain an Alternate-CoA option holding the MN's new CoA.

Note that this basic case is MIPv6 [2].

6.1.2 Scenario.C

After receiving the DA, the MN MUST acquire a new CoA, which it can do using stateless autoconfiguration. Ensuing registration is determined by the DA identifier.

Home Registration:

Each time the MN encounters a new identifier it must record the identifier and create a BU for the foreign agent. The source address is the mobile node's home address. The BU must include an Alternate-CoA option containing its (new) lCoA.

The R and D bits MUST be 1. This requests that the FA dynamically acquires a new rCoA from the domain, and bind the lCoA to this address.

The MN will receive an acknowledgement from the FA with an option containing the rCoA. The mobile node must cache this address, but the MN does not use it for routing. The MN then sends a BU to the HA, with the home address as source. The BU must set the H bit to 1, and it must contain an Alternate-CoA option holding the mobile node's rCoA.

In the packet the mobile node MUST include a type 0 Routing Header (next header=43). This header must contain one segment representing the foreign agent. This allows the foreign agent to inspect the BU and verify that the rCoA is not site-local.

If the BACK from the FA was negative, the mobile node may replace rCoA in the binding update with its stateless-configured CoA. In this case the mobile node must delete the current LCA identifier and consider itself as existing in scenario.A.

Local Registration:

When the MN encounters the same LCA identifier, it must perform local registration with the foreign agent. The source address is the mobile node's home address. The binding update MUST contain an Alternate-CoA option holding the MN's rCoA. The mobile node must also include a second Alternate-CoA option containing its stateless-configured lCoA. This option MUST appear after the option containing the rCoA.

To signal the FA, the BU must set the R bit to 1. The mobile node SHOULD receive one acknowledgement from the FA with no options included.

6.1.3 Scenario.E

After receiving the DA, the MN MUST keep its current CoA. If this is the first time in the domain then the MN must statefully acquire its CoA via local registration.

The mobile node constructs a binding update for the foreign agent, with the source address the MN's home address. The R and C bits must be set to 1, and no options are required.

The mobile node should receive an acknowledgement from the foreign agent, with an option containing its rCoA. The mobile must configure it's interface to this CoA, and then construct a binding update for the home agent. The source is the mobile node's home address, and the H bit must be set.

Thereafter, the mobile node only needs to initiate route repair. To do this the MN builds a binding update for the foreign agent. The source address should be the mobile node's careof-address. The R bit MUST be set. No options are required.

The mobile node may instead use it's home address as source, in which case it must include an Alternate-CoA option containing it's CoA.

6.1.4 Scenario.F

This scenario combines the actions of the previous two scenarios, as the domain supports lmm and also allows the mobile node to retain its lCoA within the LCA.

Home Registration:

Each time the MN encounters a new identifier it must record the identifier and create a BU for the foreign agent. The source address is the mobile node's home address.

The R, D and C bits MUST be 1. This requests that the FA dynamically acquires a new rCoA from the domain, as well as statefully acquire its lCoA. L3-Repair will also bind these two addresses together.

The MN will receive an acknowledgement from the FA containing two options. The first option has the mobile node's rCoA, which it must record, but MUST not use it for routing (i.e. tunneling) purposes. The second option contains the mobile node's lCoA, with it must configure its interface. Then MN then sends a binding update to the home agent, with the home address as source. The BU must set the H bit to 1, and it must contain an Alternate-CoA option holding the mobile node's rCoA.

In the packet, the mobile node MUST include a type 0 Routing Header (next header=43). This header must contain one segment representing the foreign agent. This allows the foreign agent to inspect the BU and verify that the rCoA is globally routable.

If the BACK from the FA was negative, the mobile node may perform stateless-autoconfiguration to acquire a rCoA. The MN sends a binding update to the home agent, with source as its home address. The BU must contain an Alternate-CoA option with its new rCoA. In this case the mobile node must delete the current LCA identifier and consider itself as existing in scenario.A.

Local Registration:

When the MN encounters the same LCA identifier, it only needs to initiate route repair. To do this the MN builds a binding update for the foreign agent. The source address should be the mobile node's careof-address. The R bit MUST be set. No options are required.

The mobile node may instead use it's home address as source, in which case it must include an Alternate-CoA option containing it's CoA.

6.2 The Careof-Address

The following table shows what CoA value to register when building the BU. Note that except in scenario.A, the FBA will most likely populate the rCoA in the BU while proxying to the HA.

C	I	scenario	home	local
0	0	E	co-located rCoA	current lCoA
0	1	F	LMA rCoA	current lCoA
1	0	A	co-located rCoA	N/A
1	1	C	LMA rCoA	co-located lCoA

Figure 5 (Outdated)

7. Foreign Agent Operation

7.1 Layer Issues

The FBA is the primary host of L2-Reg. The FA is the topological interface to the access network, and thus in direct contact with the mobile node through router advertisements. That is why FAs are also referred to as Access Routers.

Chosing a single node within the AN to host L2-Local creates a single point of failure. Increasing this number means an increase in bandwidth due to server discovery. However the FA is already active in neighbour discovery via DAs, and if the FA has outage then the mobile node can not link to the access network period.

Some functions related to L3-Repair may also be located on the Foreign Agent to reduce response time. While the operation of L2-Repair is beyond the scope of this document (See Appendix C), the exchange of messages between layers must be defined.

Since the layer interface resides within a "single" node, these primitives take the form of data structures. This is signifcant because all other service primitives, including those of peer layers and even operations within a layer, are transferred as IP datagrams.

Appendix B defines service access points used in the experimental prototype. However, since this is system dependent, it is expected that this section will be moved to another draft in the future.

7.2 L2-Local Request Primitives

7.2.1 Initiate Route Repair

```
local_request_repair {
      8 bits     type
     16 bits     sequence
    128 bits     lCoA
    128 bits     rCoA
}
```

```
type          '0'
sequence      used to match request with confirm.
lCoA          address to be updated.
rCoA          address to bind with (may be null).
```

7.2.2 Request rCoA

```
local_request_rcoa {
      8 bits     type
     16 bits     sequence
    128 bits     lCoA
}
```

```
type          '1'
sequence      used to match request with confirm.
lCoA          data for autoconfiguring rCoA (may be null).
```

7.2.3 Request lCoA

```
local_request_lcoa {
     8 bits     type
    16 bits     sequence
}
```

```
type          '2'
sequence      used to match request with confirm.
```

7.3 L3-Repair Confirm Primitives

7.3.1 Repair Acknowledgement

```
local_confirm_repair {
     8 bits     type
    16 bits     sequence
     8 bits     code
}
```

```
type          '1'
sequence      used to match request with confirm.
code          status of repair action.
```

7.3.2 Configure rCoA

```
local_confirm_rcoa {
      8 bits     type
     16 bits     sequence
    128 bits     rCoA
      8 bits     prefix
}

type          '2'
sequence      used to match request with confirm.
rCoA          unique rCoA.
prefix        prefix length of rCoA.
```

7.3.3 Configure lCoA

```
local_confirm_lcoa {
      8 bits     type
     16 bits     sequence
    128 bits     lCoA
      8 bits     prefix
}

type          '3'
sequence      used to match request with confirm.
lCoA          unique lCoA within the access network.
prefix        prefix length of lCoA.
```

7.4 Binding Update List

The FBA maintins a binding update list which contains an entry for every active request. Once the FBA has created a Binding Acknowledgement for a mobile node, it's corresponding entry is removed from the list.

Each entry on the list contains the following fields:

* The link local address of a mobile node from which a binding update was received.

* The flags remaining to be processed from the original BU.

* The sequence number expected by the mobile node in the Binding Acknowledgement when its BU has been confirmed.

* The sequence number used in the last request sent to L3-Repair. Each request must use a new sequence, equal to the maximum sequence number plus one (modulo 2**16).

* The regional care-of address provided by, or supplied to, the mobile node.

* The local care-of address provided by, or supplied to, the mobile node.

* Prefix lengths fields for both addresses.

* An integer value representing the success or failure of route repair within the access network.

7.5 Processing Bindings

The foreign binding agent will receive a binding update request from a mobile node. When this happens the FBA immediately creates a binding update list (BUL) entry for that mobile node.

The entry will contain the mobile node's link local address and the sequence number provided in the BU. The entry MAY also contain either or both a rCoA and lCoA depending upon the nature of the request. These addresses will be contained in Alternate CoA Options. If more than one is present, then the first is the rCoA and the second is the lCoA.

The entry will also initially contain the flags transmitted in the BU. If the flags H, L or K are toggled then the FBA must abort the creation of a BUL entry by sending a Binding Acknowledgment to the mobile node indicating that it does not understand the request. Furthermore, if the F bit is toggled it MUST be the only bit toggled. Further discussion of this type of request is delayed to later sections.

That leaves the A, D, C and R bits. The acknowledgement is mandatory, its presence is assumed and need not be stored. It is the remaining bits that drives the operation of L2-Local.

The following figure illustrates the order each bit must be processed in (if present), the L2-Local request primitive that must be sent to L3-Repair and the data this primitive must contain.

Flag	Type	lCoA	rCoA
C	2	NO	NO
D	1	SHOULD	NO
R	0	MUST	MUST

Figure 6

When a confirm primitive arrives, the FBA identifies the BUL entry with the matching sequence number and populates the appropriate entry field. It then sets the flag for that request/ confirm pair to 0, checking afterwards if any remaining flags need to be processed. Once all flags are equal to 0, a binding acknowledgement is sent to the mobile node.

7.6 Example: Processing a Binding Update

To demonstrate, consider a BU arriving at the FBA from mobile node A, and that it contains an alternate CoA option with A's autoconfigured lCoA. The flags D and R have been set.

The FBA creates an entry for A with a confirm sequence equal to N, acquired from the BU. The flags D and R in the BUL entry are set to 1 and the lCoA field is populated with the value in the Alternate CoA Option.

The FBA notices the first toggled flag is D. It processes this flag by creating L2-Local request primitive Type 1. The sequence field is set to M, and this value is stored in the BUL's request sequence entry. The lCoA field is equal to that of the BUL entry.

How L3-Repair processes this request is beyond the scope of this draft; however the lCoA value was provided to assist in forming a unique rCoA iff this method is applicable to the protocol currently deployed in the access network.

L3-Repair creates a confirm primitive of Type '2' with a rCoA and matching prefix length provided by the domain. The primitive contains the sequence value M; the FBA finds the matching BUL entry and updates the rCoA and prefix fields. If the rCoA was null then the FBA MUST create binding acknowledgment for the mobile node indicating failure and then delete the BUL entry.

The FBA sets the D bit to 0 and inspects the BUL entry for any further flags with a value of 1. It discovers the R flag and creates a request primtive Type 0. This primitive has the sequence M+1, which replaces the value stored in the request sequence field. The rCoA and lCoA primitive values are equal to those found in the BUL entry.

How L3-Repair directs packets from rCoA to lCoA is beyond the scope of this draft. However the Type 0 request primitive informs the protocol that it must bind rCoA with the new lCoA. When this is complete it generates the confirm primite Type 1 with a code indicating the success or failure.

The FBA populates the BUL entry with the matching sequence number, and set its flag R to 0. The FBA discovers that no further flags need to be processed; it generates a binding acknowledgement for the mobile node (based upon the populated fields of the binding acknowledgement) and then deletes the BUL entry.

8. Fast Registration

A handoff solution refers to a process which reduces or hides the latency introduced by a handoff. In this case, latency is the period during which the HA/LMA delivers packets to the old CoA.

In base handoff solutions, such as FMIP[5][6], control signals are restricted to foreign agents and mobile nodes. However some solutions, such as F-HMIP[7], extend into the access network.

This section describes the interface between simple handoff and fast handover. An overview is also provided regarding how fast handoff protocols may operate within the framework. Specific operation is not provided because, like route repair, multiple protocols may be available for deployment. It is up to the network adminstrator to determine which one to use.

Unless stated otherwise, message formats refer to those defined in Fast Mobile IP [6].

8.1 Neighbour Discovery

Neighbour Discovery operates precisely as specified by Fast Mobile IP. There is no requirement to advertise the C and I flags associated with NAR, nor does it need to advertise the local coverage area identifier.

8.2 Mobile Node Operation

There are only two changes to the operation of the mobile node as specified by Fast Mobile IP:

1) The mobile node should be allowed to include its regional care-of address in the fast binding update. If present, this address must be contained in alternate care-of address option which must follow the PCoA's home address option.

2) The mobile node should retain "PCoA", as well as continue to receive packets destined for this address until indicated otherwise during local registration at NAR.

8.3 Foreign Agent Operation

Since the fast handoff has a close association to local route repair, it also may have a wide varieity of operations. Thus it is impossible to provide a single specification on how to operate the foreign agent.

The framework defines a layer boundary between the mobile node and the foreign. This is an innate interface in Fast Mobile IP, whose primitives are the fast binding update, fast binding acknowledgement and fast neighbour advertisement.

The remainder of this section is a set of suggestions on how to design fast handoff protocol to operate within the framework.

8.3.1 Scenario A

Fast Mobile IP was designed for this precise scenario. Thus fast handoff SHOULD operate as specified by Fast Mobile IP without any modifications.

8.3.2 Scenario C

Fast handoff MAY operate as specified by Fast Mobile IP. However, as discussed in [7], it may be more efficient to configure a transient tunnel located at the node that sponsors the mobile node's regional care-of address.

The foreign agent should inspect the local coverage area of the next access router. If this value differs from the foreign agent's own identifier then the transient tunnel SHOULD be established as specified by Fast Mobile IP.

Otherwise the foreign MAY forward the FBU to the node sponsoring rCoA. This node will then exchange HI/HAck with NAR, after which it will send FBack to the foreign agent. The foreign agent will send a new FBack message to the mobile node whose source is its own.

8.3.3 Scenario E

Fast handoff MAY operate as specified by Fast Mobile IP. However, since the mobile node's care-of address is not changing, it may be more efficient to bicast packets at the nearest common router.

8.3.4 Scenario F

This scenario is a combination of the previous two. Thus, so long as PAR and NAR reside within the same local coverage area then the foreign agent may initiate bicasting. Otherwise fast handoff SHOULD operate as specified by Fast Mobile IP.

9. Security Considerations

The framework will utilize many of the security procedures discussed in MIPv6[2]. With the process of binding proxy, the framework creates additional security risk by reintroducing the foreign agent.

Thus extra measures must be made to authenticate actions such as local registration. The foreign may also drop binding updates whose source address is not link local.

However, since home and local registration are separate activities, the home agent does not need to be aware of security associations with foreign agents.

Future versions of this document will include more security issues in detail.

10. Acknowledgements

[1] C. Perkins. "IP Mobility Support for IPv4". Request for Comments (Proposed Standard) 3344. August 2002.

[2] D. Johnson, et al. "Mobility Support for IPv6" (work in progress). draft-ietf-mobileip-ipv6-24.txt, June 2003.

[3] E. Gustafsson, et al. "Mobile IPv4 Regional Registration" (work in progress). draft-ietf-mobileip-reg-tunnel-07.txt, October 2002.

[4] C. Williams, editor. "Localized Mobility Management Requirements" (work in progress). draft-ietf-mipshop-lmm-equirements-00.txt.txt, October 2003.

[5] K. El-Maki, editor. "Low-Latency Handoffs in Mobile IPv4" (work in progress). draft-ietf-mobileip-lowlatency-handoffs-4-07.txt, October 2003.

[6] R. Koodli, editor. "Fast Handovers for Mobile IPv6" (work in progress). draft-ietf-mipshop-fast-mipv6-00.txt, March 2003.

[7] H. Y. Jung, et al. "Fast Handover for Hierarchical MIPv6" (work in progress). draft-jung-mobileip-fastho-hmipv6-02.txt, August 2003.

[8] S. Deering, R. Hinden. "Internet Protocol, Version 6 (IPv6)". Request for Comments (Proposed Standard) 2460. December 1998.

[9] T. Narten, et al. "Neighbor Discovery for IP Version 6 (IPv6". Reqeust for Comments (Proposed Standard) 2461. December 1998.

[10] S. Thomson, T. Narten. "IPv6 Stateless Address Autoconfiguration". Request for Comments (Proposed Standard) 2462. December 1998.

[11] J. Manner, M. Kojo, ed. "Mobility Related Terminology" (work in progress). draft-ietf-seamoby-mobility-terminology-04.txt, April 2003.

[12] S. Hesham, et al. "Hierarchical Mobile IPv6 mobility management (HMIPv6)" (work in progress). draft-ietf-mipshop-hmipv6-00.txt, June 2003.

[13] A. Helmy, M. Jaseemuddin, and G. Bhaskera, "Efficient Micro-Mobility using Intra-domain Multicast-based Mechanisms (M&M)", 2002, ACM SIGCOMM Computer Communications Review , Vol. 32, No. 5, pp. 61-72, November 2002.

Author's Addresses

Bryan Hartwell
Ryerson University
Email: bhartwel@ee.ryerson.ca

Mohammad Jaseemuddin
Ryerson University
Email: jaseem@ee.ryerson.ca

Appendix A - Future Objectives/Issues

Topics are listed in order of importance.

*Review Section 6 (some misplaced terminology, hard to read, etc.)
*Remove terminology describing framework layers as L1,L2 & L3.
* Add Appendix B: Service Access Points
* Add Appendix C: Hierarchical Route Repair, A Brief Example (1 pg)
*Move all primitives to section 5 (i.e. those in section 7).
Rewrite section 5.
*add correspondent nodes
*tackle security in detail:
*security as discussed in MIPv6
*issues caused by reintroducing FA. Maybe something like Coti/Cot for FA?
*Consider applicability of summary-appendix, which reduces document to key actions (MUST, SHOULD, etc.)
*Why not send RtSolPr message as FBU ? Saves waiting on PrRtAdv... I think there is a reason, need to look it up, but it would reduce a lot of signaling in framework.

Bibliography

[1] Jassemuddin, M. *Private Communication*, May 2003.

[2] Moy, J., "OSPF Version 2", RFC 2178, April 1998.

[3] Perkins, C. "IP Mobility Support for IPv4", RFC 3344, August 2002.

[4] Johnson, D., Perkins, C., Arkko, J. "Mobility Support in IPv6", RFC 3775, June 2004.

[5] Droms, R., et al. "Dynamic Host Configuration Protocol for IPv6", RFC 3315, July 2003

[6] Deering, S. "ICMP Router Discovery Messages", RFC 1256, September 1991.

[7] Narten, T., Nordmark, E, Simpson, W. "Neighbor Discovery for IP version 6 (IPv6)", RFC 2461, December 1998.

[8] Williams, C., editor "Local Mobility Management Requirements", draft-ietf-mipshop-lmm-requirements-03.txt (work in progress), July 2004.

[9] Hesham, S., et al. "Hierarchical Mobile IPv6 mobility management (HMIPv6)", draft-ietf-mipshop-hmipv6-02.txt (work in progress), June 2004.

[10] A. Helmy, M. Jaseemuddin, and G. Bhaskera, "Efficient Micro-Mobility using Intra-domain Multicast-based Mechanisms (M&M)", 2002, ACM SIGCOMM Computer Communications Review, Vol 32, No. 5, pp. 61-72, November 2002.

[11] Manner, J., Kojo, M., editor "Mobility Related Terminology", RFC 3753, June 2004.

[12] Koodli, R., editor "Fast Handovers for Mobile IPv6", draft-ieft-mipshop-fast-mipv6-02.txt (work in progress), July 2004.

[13] H. Y. Jung, et al. "Fast Handover for Hierarchical MIPv6" (work in progress). draft-jung-mobileip-fastho-hmipv6-03.txt, February 2004.

[14] Ramjee, R., et al "HAWAII: A Domain-based Approach for Supporting Mobility in Wide Area Wireless Networks.", http://www.bell-labs.com/usr/ramjee/papers/hawaii.ps

[15] A. Valkó, "Cellular IP - A New Approach to Internet Host Mobility," ACM Computer Communication Review, Vol. 29, No. 1, January 1999, pp. 50-65.

[16] RADVD, http://v6web.litech.org/radvd, radvd-0.7.2.tar.gz

[17] Mobile IPv6 for Linux, http://www.mobile-ipv6.org/software/, mipv6-0.9.5.1-v2.4.20.tar.gz

[18] McCann, P. "Mobile IPv6 Fast Handovers for 802.11 Networks" (work in progress). draft-ietf-mipshop-80211fh-00.txt, February 2004.

[19] Manner, J., Kojo, R. "Mobility Related Terminology", draft-ietf-seamoby-mobility-terminology-04.txt (work in progress), April 2003.