Mobile IPv6 without Home Agent

Cui Bo

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Computer Engineering

Prince of Songkla University

2010

Thesis Title	Mobile IPv6 without Home Agent
Author	Miss Cui Bo
Major Program	Computer Engineering

Major Advisor:

……..…….....
(Mr. Kevin Robert Elz)

Co-advisor:

.....
(Assoc. Prof. Dr. Sinchai Kamolphiwong)

Examining Committee:

..…….……Chairperson
(Asst. Prof. Dr. Nittida Elz)

……..……
(Mr. Kevin Robert Elz)

……..……
(Assoc. Prof. Dr. Sinchai Kamolphiwong)

……..……
(Dr. Panita Pongpaibool)

The Graduate School, Prince of Songkla University, has approved this thesis as partial fulfillment of the requirements for the Master of Engineering Degree in Computer Engineering

……..……..
(Prof. Dr. Amornrat Phongdara)
Dean of Graduate School

Thesis Title	Mobile IPv6 without Home Agent
Author	Miss Cui Bo
Major Program	Computer Engineering
Academic Year	2010

ABSTRACT

In Mobile IPv6, the home agent (HA) is an essential component. Mobile nodes will not function properly if the HA cannot be reached. Enhanced Route Optimization uses a Cryptographically Generated Home Address in Mobile IPv6, to enhance the security and reduce the handoff delays. Unfortunately, a CGA cannot provide home prefix validation. The 'home test ' signal remains needed for handover purposes. When the correspondent node or the mobile node cannot connect to the home agent for any reason, a "return-to-home" flooding attack could occur if this requirement were ignored. This thesis proposes a solution to enhance the mechanism for Mobile IPv6 communications and avoid this problem, and thus allow Mobile IPv6 to continue operating when no home agent is available.

Keywords: Mobile IPv6, Enhanced Route Optimization, home agent, CGA, home keygen token, care of keygen token

ACKNOWLEDGEMENT

First and Foremost, I would like to express my deepest gratitude to Mr. Robert Elz, my supervisor, for his constant advice, suggestions, teaching, extraordinary patience, and great help with this thesis. He has walked me through all the stages of my master studying, and being a role model have been the most important factors in increasing my confidence and improving my skills.

I would also like to thank to Assoc. Prof. Dr. Sinchai Kamolphiwong, who provided valuable information and advice at individual discussions.

I would also sincerely thank committee members Asst. Prof. Dr. Nittida Elz, Assoc. Prof. Thossapom Kamolphiwong, and Dr. Panita Pongpaibool to review and give the comment this work to be better.

I thank all the lecturers and staff in the Department of Computer Engineering who helped me with my studies. In addition, I thank the Department of Computer Engineering, Faculty of Engineering, Graduate School, and Prince of Songkla University, who offered me help to finish my university work.

I would like to express my heartfelt gratitude to my family who have supported my study in Thailand. I also thank my friends for their help with my study and life.

Cui Bo

TABLE OF CONTENTS

Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

LIST OF ABBREVIATIONS AND SYMBOLS

BA	Binding Acknowledgement
BC	Binding Cache
BU	Binding Update
BUL	Binding Update List
CGA	Cryptographically Generated Address
CN	Correspondent Node
CoA	Care-of Address
CoT	Care-of Test
CoTI	Care-of Test Init
DHCP	Dynamic Host Configuration Protocol
DoS	Denial of Service
EBA	Early Binding Acknowledgement
EBU	Early Binding Update
ERO	Enhanced Route Optimization
ESP	Encapsulating Security Payload
HA	Home Agent
HoA	Home Address
HoT	Home Test
HoTI	Home Test Init
ICMPv4	Internet Control Message Protocol version 4
ICMPv6	Internet Control Message Protocol version 6
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6

KAME	KArigoME[1]
MH	Mobility Header
MIP	Mobile IP
MIPv4	Mobile IPv4
MIPv6	Mobile IPv6
MN	Mobile Node
ND	Neighbor Discovery
PHKT	Permanent Home Keygen Token
PKI	Public Key Infrastructure
RO	Route Optimization
RR	Return Routability
SEND	SEcure Neighbor Discovery
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

[1] KArigoME means “turtle” in Japanese

CHAPTER 1

INTRODUCTION

1.1 Motivation

Mobile Computing is becoming increasingly important due to the rise in the number of the many new types of mobile devices, such as mobile phones and mobile computers and the desire to have continuous network connectivity to the Internet irrespective of the physical location of the node. The Mobile Internet Protocol (Mobile IP) is an extension to the Internet Protocol proposed by the Internet Engineering Task Force (IETF), which enables users to maintain nonstop connectivity using their home IP address regardless of physical movement.

The basic Mobile IPv6 protocols provide a direct way for transmitting packets between the mobile node and correspondent node known as Route Optimization. The Return Routability procedure [3] builds proof to the correspondent node that the mobile node is in fact addressable at its claimed care-of address and home address. This is done by testing whether packets addressed to the two claimed addresses are routed to the mobile node. The mobile node can pass the test only if it is able to supply proof that it received certain data (the "keygen tokens") which the correspondent node (CN) sends to those addresses. Using Return Routability there can be a long delay before a binding update (BU) to the CN is accepted.

There is a new protocol – Enhanced Route Optimization [17], which by using Cryptographically Generated Addresses (CGA) [15] in its authentication method can avoid the BU long delay problem. Using this mechanism, we could reduce the handoff latency, securely authenticate a mobile node (MN) without preconfigured credentials or a public-key infrastructure, and reduce the level of signaling overhead compared to a base mobile IPv6 correspondent registration, especially when the mobile node does not move frequently. The Home address Test is still needed although the CGA uses public key signatures for authenticating binding messages which is much more difficult to attack. It is possible for the network to have a problem between

the correspondent network and the home network. Then the correspondent node cannot talk to the home agent to check the reachability of mobile node's claimed home address. If the CN accepts a BU without validating the home network prefix, an attacking MN may be able to trick a CN into sending much data to the victim home network. Thus the CN must refuse BUs until the home address prefix has been validated.

The effect of this is that although with ERO the home agent is not required as part of the validation that the mobile node is the node it claims to be, it is still required to validate that the mobile node is entitled to the identity it claims.

This means that if the home agent is unavailable or unreachable, Mobile IP, even Mobile IPv6 with ERO is useless.

A rational reason for a mobile node to move is to leave a broken, unreachable, network and join a working link. Unfortunately, as the home agent cannot move, this limitation of Mobile IP means that it is ineffective in this situation.

This thesis aims to correct this defect, and allow Mobile IPv6, with ERO, to function even when the mobile node's home agent is unreachable. This is to be accomplished without introducing any new security vulnerabilities.

1.2 Objective

1) To investigate a solution which solves the problem that node can't keep communicating if it moves away from a failed home link.

2) To design and implement a prototype of this solution to verify that the design is effective and that no security problems occur, and test the implementation using a small private test network.

1.3 Advantages

1) It could make the node work normally when the network between mobile node and home agent or between Home Agent and correspondent node is down. That allows mobile nodes to communicate correctly without a home agent.

2) Avoiding the home test may allow the delay in achieving a binding between a mobile node and its correspondents to be reduced.

3) The accomplishment and result could be proposed as an alternative solution, used in real world and published.

1.4 Scope of work

Study and investigate Mobile IPv6 technique and Enhanced Route Optimization for mobile IPv6. Find the purpose of my project.

1) Find out a possible solution to make the nodes work normally while network link broken for a short time and avoid the possible corresponding “return-to-home” flooding.

2) Only a prototype implementation is expected. This implementation will be done on a UNIX system.

3) Testing to require only small experimental network.

1.5 Work plan

1) Study the Mobile IPv6 technique and the Enhanced Route Optimization. Research the Enhanced Route Optimization authentication methods and security problem.

2) Find the purpose of this project and solution of the problem we want to be improved. Write proposal.

3) Design the experimental network topology in order to test the Enhanced Route Optimization in normal mobile IPv6 network and our proposed solution.

4) Design a possible approach to achieve our requirements and implement it under our experimental network.

5) Test the new approach and evaluate the implemented solution.

6) Analyze the result and form conclusion.

7) Write the final report.

1.6 Outline

This document is organized in 7 chapters as follows:

Chapter 1, Introduction, this chapter, provides the motivation, objective and scope of this work. In addition, it also presents the work plan of this study, and a brief summary of the remainder of the thesis.

Chapter 2, introduces the underlying concepts necessary to understand the work of this thesis. It provides an introduction to Mobile IP, and Mobile IPv6 in particular, and explains how Route Optimization allows mobile nodes to communicate directly with correspondent nodes. It then introduces Cryptographically Generated Addresses and shows how these are used to allow Enhanced Route Optimization with lower delays and better security.

Chapter 3, explains the defect we observe in Mobile IP where at least one plausible reason for which a node may become mobile fails to be handled. It provides the problem statement for this thesis. This chapter concludes with the outline of a potential solution to the identified defect.

Chapter 4, takes the solution outlined in chapter 3 and expands that into a fully designed solution. It explains the new procedures required and the operational changes needed to allow the solution to avoid introducing new security problems.

Chapter 5, introduces our prototype implementation of the solution from the previous chapter. We show how the various functions are implemented and mention the shortcuts taken that would need correcting in a production implementation.

Chapter 6, explains the testing carried out to verify that both the solution from chapter 4, and the implementation from chapter 5 operate as intended. It begins with a description

of our small (minimal) test network, explains the tests carried out, and shows the results of the testing.

Chapter 7, concludes this thesis. We summarise the results and discuss the limitations of the approach taken. Finally we make some suggestions for areas we feel would be benefited by future work to continue what has been accomplished here.

CHAPTER 2

BACKGROUND INFORMAITON

This chapter provides the background information upon which this thesis is based. Section 2.1 will introduce the technology of mobile IP. Section 2.2 provides an overview of IPv6. Some terminology, basic operations, security issues and the Return Routability procedure of Mobile IPv6 will be introduced in section 2.3. Cryptographically Generated Addresses will be introduced in the section 2.4 then used in section 2.5, which will introduce Enhanced Route Optimization.

2.1 Introduction

The Internet Protocol (IP) uses addresses for two purposes. They identify end-points of communications (hosts, or nodes), and they provide locator information – they indicate where in the Internet a node is connected. This dual role is not an issue for stationary hosts, it does no more than slightly complicate the assignment process. However for a node that moves there is a problem. To correctly locate the node the address must change to reflect its new attachment point to the Internet. But if its address changes, so does its identity, it no longer seems to be the same node, and so communication that were in progress when the node moved all fail.

To deal with this problem, the Mobile IP [1] protocol was created.

Mobile IP is a standard communications protocol, which provides an efficient, scalable mechanism for roaming within the Internet. Users can use their local IP addresses permanently regardless of having a constant link-layer point of attachment. Each mobile node is always identified by a fixed address, known as its home address, no matter what its current point of attachment to the Internet, allowing for transparent mobility with respect to the network and all other devices. The only devices which need to be aware of the movement of this node are the mobile device and an agent serving the user's home network.

Other nodes, known in Mobile IP as Correspondent Nodes, continue communicating with the mobile node's home address. The agent on the mobile node's home network, called the home agent, collects packets destined to the mobile node and forwards them to the mobile node. The mobile node merely needs to keep the home agent aware of its current location, and the address, known as a care-of address, at which it can be temporarily reached. The home agent and mobile node are well known to each other, they are configured to be aware of each other, and exchange data using encrypted packets so each is sure that only the other can receive or transmit those packets and authentication is assured.

Internet Protocol version 6 (IPv6 [2]) has been designed by the IETF, in order to replace the current version of the IP protocol used in the Internet (IPv4). There is a specified protocol for IPv6, known as Mobile IPv6 (MIPv6 [3]), which allows nodes to remain reachable while arbitrarily moving around in the IPv6 network. MIPv6 has the same basic design as Mobile IPv4 (MIPv4 [19]), but some details vary, such as packet formats, movement detection, flexible options and so on. MIPv6 also adds Route Optimization, to be described in section 2.3.2.2. We will describe MIPv6 in more detail later, but except where noted, the concepts apply to MIPv4 as well.

2.2 Overview of Internet Protocol version 6 (IPv6)

As technology continues to soar in usage across the globe, the limitation of the Internet Protocol Version 4 (IPv4), which has been the dominant Internet Protocol technology for twenty years, is that we are rapidly reaching a point where available network address space is running out. IPv4 allows for about 2^{32} or 4,294,967,296 addresses which have already been effectively completely allocated and therefore, no room remains for growth. Thus, causing a new version of the Internet Protocol, called IPv6, to be designed.

IPv6 represents a significant update to IP, but its modifications and additions are made without changing the core nature of how IP works. Addressing is the place where most of the differences between IPv4 and IPv6 are seen, but the changes are mostly in how addresses are implemented and used. Here are some general characteristics of the IPv6 addressing:

1. The two main functions of addressing are still network interface identification and routing. Routing is facilitated through the structure of addresses on the internetwork.
2. IPv6 addresses are still the ones associated with the network layer in TCP/IP networks and are distinct from data link layer addresses.
3. Addresses are still assigned to network interfaces, so a regular host like a PC will usually have one (unicast) address, and routers will have more than one, one for each of the physical networks to which it connects.
4. IPv6 addresses are like classless IPv4 addresses in that they are interpreted as having a network identifier part and a host interface identifier part (a network ID and a host ID), but that the representation is not explicitly encoded into the address itself.

One of the key advantages of IPv6 is its use of much larger addresses. The size of an address in IPv6 is 128 bits, which is four times the larger than an IPv4 address and offers a more permanent solution, allowing for about 2^{128} addresses. The conventional form for representing IPv6 is:

X: X: X: X: X: X: X: X, where Xs are the hexadecimal values of the eight 16-bit pieces of the 128-bit address. Example: fec0:1:0:1:0:0:0:1234

This example IPv6 address contains a long strings of zero bits. In order to make it easier to represent such addresses, the use of "::", called double colon, indicates that there are multiple omitted groups of 16-bits of zeros. The "::" can appear only once in an address and it can be used to compress the leading, trailing or contiguous sixteen-bit groups of zeros in an address. The example can be represented as: fec0:1:0:1::1234 .

The prefix is the part of the address that indicates the bits that are the bits of the subnet prefix, which is represented by the notation: IPv6-address/prefix-length. The "prefix-length" part is a decimal value representing how many of the leftmost contiguous bits of the address comprise the prefix. For the example address the representation might be: fec0:1:0:1::1234/64.

The prefix identifies the network to which this address belongs and the remaining bits are used to select one interface (and its host) connected to that network.

2.3 Mobile IPv6

2.3.1 Mobile IPv6 components and terminologies

Mobile IPv6 is intended to enable IPv6 nodes to move from one IP subnet to another. Before talking about the operation technique, we need to learn about the basic components of Mobile IPv6 network and the respective terminologies.

Mobile IPv6 introduces three new types of network entities, as figure 2.1 shows:

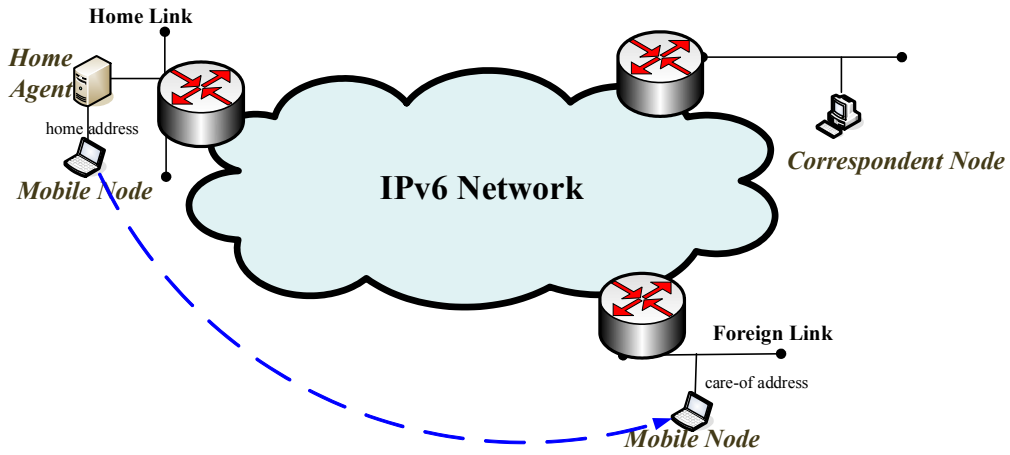

Figure 2.1 Mobile IPv6 components

- **Mobile node (MN)**: a IPv6 node that can change its location from one link to another within the Internet topology, while still being addressable via its home address.
- **Home agent (HA):** a node that is often a router which resides on the home link and maintains registration information of the mobile node while it moves away from its home link. The mobile node needs to tell home agent its new location when it moves to another link.
- **Correspondent node (CN):** a stationary or mobile node, with which a mobile node is communicating.

We should present some terms before we talk about Mobile IPv6, which will appear often in the later information or have been mentioned in the previous description [3] [4]:

- **Home link:** the link where MN is connected when at home, and on which mobile node's home subnet prefix is assigned.
- **Home address:** a stable 128-bit unicast routable address assigned to the mobile node when it attached to the home link. Like all IPv6 addresses, the home address is usually based on the prefix assigned to the home link combined with the mobile node's interface identifier.
- **Foreign link:** a link which is not mobile node's home link.
- **Care-of address:** the mobile node will get a new address, called its care-of address when it is attached to a foreign link. For the stateless configuration, this address is a combination of the foreign subnet prefix and an interface identifier determined by the mobile node. A mobile node can be assigned multiple care-of addresses. The one which is registered to the home agent with the mobile node's given home address is called its "primary" care-of address.
- **Binding:** an association of a mobile node's home address with a care-of address. This allows the home agent to forward packets to the mobile node's current location. The binding has lifetime, and needs to be refreshed if the lifetime expires or updated if the mobile node gets a new address because it moved to a new link.
- **Binding cache (BC):** a cache storing a number of binding information for one or more mobile nodes in volatile memory, which is maintained by either the home agent or the correspondent node.
- **Binding update list (BUL):** a list containing all bindings that were registered with the mobile node's home agent or a correspondent node. This list is managed by the mobile node, and provides the information for mobile node to take appropriate actions.

2.3.2 Basic operations of Mobile IPv6

The standard IP routing mechanisms will deliver packets destined to a mobile node's home address to its home link while mobile node remains within the home link. When the mobile node moves to another link, the home agent will be a proxy for the mobile node on the home link.

2.3.2.1 Triangle Routing

When a mobile node has moved away from home and attached to a foreign link, conventional IPv6 mechanisms, such as stateless address autoconfiguration [6] or stateful address autoconfiguration such as DHCPv6 [7] or PPPv6 [8], allow the mobile node to obtain its care-of address based on the prefix of the foreign link. The decision about which manner of automatic address configuration to use is made according to the methods of IPv6 Neighbor Discovery [9]. A mobile node may have more than one care-of address at a time. Following address configuration, the mobile node informs its home agent of its movement by sending a binding update (BU) message. The binding update message contains the mobile node's home address and its care-of address.

The purpose of the binding update is inform the home agent of the mobile node's current location (current care-of address). The home agent is responsible for storing this information in order to forward the packets to the mobile node's care-of address. The home agent contains a binding cache, which has all bindings from mobile nodes. Each entry in the binding cache stores a binding for one home address. When the home agent receives the binding update message, it will validate the message. If the binding update message is accepted, the home agent will search its binding cache to determine whether an earlier entry exists there for the mobile node's home address. If an entry is found, the home agent updates that entry with the new information received in the binding update. Otherwise, if no entry is found, it will create a new one.

The home agent intercepts any IPv6 packets addressed to the mobile node's home address on the home network, and tunnels each intercepted packet to the mobile node's current care-of address indicated in this Binding Cache entry. To tunnel the packet, the home agent encapsulates it using IPv6 encapsulation [12].

The binding update between the mobile node and the home agent needs to be authenticated. Otherwise, a malicious node could pretend to be the mobile node, the one which is communicating with the correspondent. That way, the malicious node could hijack the traffic intended for a node by causing it to be redirected elsewhere than to the real location of the node. The solution to secure the bindings between the mobile node and its home agent will be mentioned later and the details of attacks will be introduced in section 2.3.3.

The basic routing mechanism in Mobile IPv6 is known as triangle routing, which is an indirect way for transmitting the packet between the mobile node and the correspondent node. The correspondent node has no idea that the mobile node has changed its location. As figure 2.2 shows, in triangle routing, all packets sent to a mobile node must be forwarded to and from the mobile node at its current location by its home agent.

The home agent proxies the mobile node at its home address and mainly serves as a relay for packets exchanged with the correspondent node.

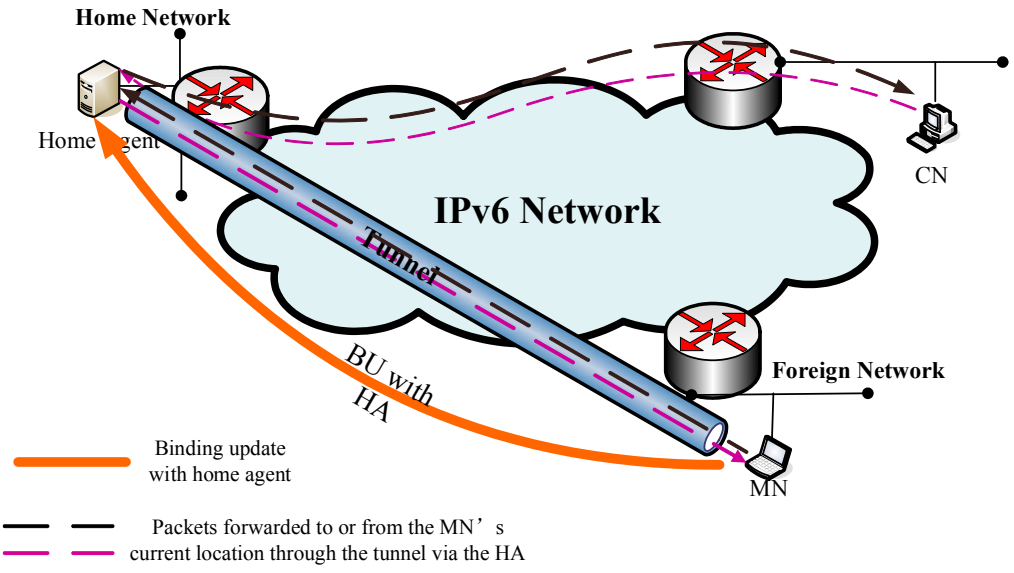

Figure 2.2 Triangle Routing

Triangle routing, because of its poor route selection, has some problems as follows [11]:

1) increased impact of possible network partitions
2) increased load on the network
3) increased delay in delivering packets

Messages exchanged between the mobile node (MN) and the home agent (HA) are protected using IPsec [10] and no new security mechanism exists for this purpose. IPsec can be used to authenticate and encrypt packets at the IP level. It is a suite of protocols "designed to provide interoperable, high quality, cryptographically-based security for IPv4 and IPv6" [22]. The use of the IPSec Authentication Header (AH) [23] and the Encapsulating Security Payload (ESP) [21] and a key management mechanism help to ensure the integrity of the binding update messages between the MN and the HA. To prevent the MN from sending a binding update for another mobile node using its association, the HA must verify the binding update message contains the correct home address, either as the source of the packet or in an optional field at end of the packet, and the correct security association.

The binding update and the following binding acknowledgement are authenticated using a preconfigured IPsec security association between the mobile and the home agent. This is the first point where we depend on the long-term trust relationship between the mobile node and its home network.

The approach IPsec uses to address security is by managing two concepts: privacy and authentication. These concepts involve techniques such as encryption, secret key negotiation and digital signatures. Moreover, the effectiveness of a true secure communication relies on a secret shared by the parties involved that no one can guess. The biggest problem with the IPsec method is the key distribution. Key distribution of the IPsec, which is called Internet Key Exchange (IKE) [20], uses either preshared secrets or public keys in the key exchange.

When authentication is needed between a mobile node and a home agent, which must have some relationship in advance, because the mobile node uses services of the home agent, the needed secrets might be exchanged beforehand or some private public key distribution can be utilized. The mobile node and home agent know each other, and thus can have a pre-established strong security association between them. Here, we assume the bidirectional IP

tunnel using the IPsec security protocol can provide enough security between mobile node and its home agent.

2.3.2.2 Route Optimization

Packets between the mobile node and its correspondent node have to travel via the home network, which may be far away. The process to achieve the other, direct, way for transmitting between the mobile node and correspondent node is known as Route Optimization (RO). Given that IPv4 was not built with mobility in mind, Mobile IPv4 was designed as an extension to the base IPv4 protocol to support mobility. The most significant difference between MIPv4 and MIPv6 is that MIPv6 is integrated into the base IPv6 protocol and not an add-on feature. While the Route Optimization capability for nodes is optional in IPv4, all Mobile IPv6 nodes should be designed with this capability.

When a mobile node receives a packet tunneled from the home agent, route optimization can be started. The mobile node informs the correspondent node of its current location also through using a binding update message. The binding update contains the mobile's home address and its current care-of address. The correspondent node maintains a binding cache similar to the one maintained by home agent. As figure 2.3 shows the mobile node sends a binding update message to the correspondent node, to notify the correspondent node of its new location. The correspondent node adds this binding to its binding cache, which is effectively a routing table: it tells that packets addressed to the HoA should instead be sent to the CoA. This binding needs to be refreshed every few minutes if the mobile stays at the same CoA through sending a new binding update. Finally, the correspondent node acknowledges this message according to this binding. If the binding expires or if it is explicitly deleted by the mobile node (by sending a BU with a lifetime of zero), the correspondent will remove this binding from its cache and then revert sending future packets to the mobile node's home address.

As figure 2.4 shows, when sending a packet to any IPv6 destination address, and if a cached binding for this address is found, the node routes the packets directly to the mobile node at the care-of address indicated in this binding.

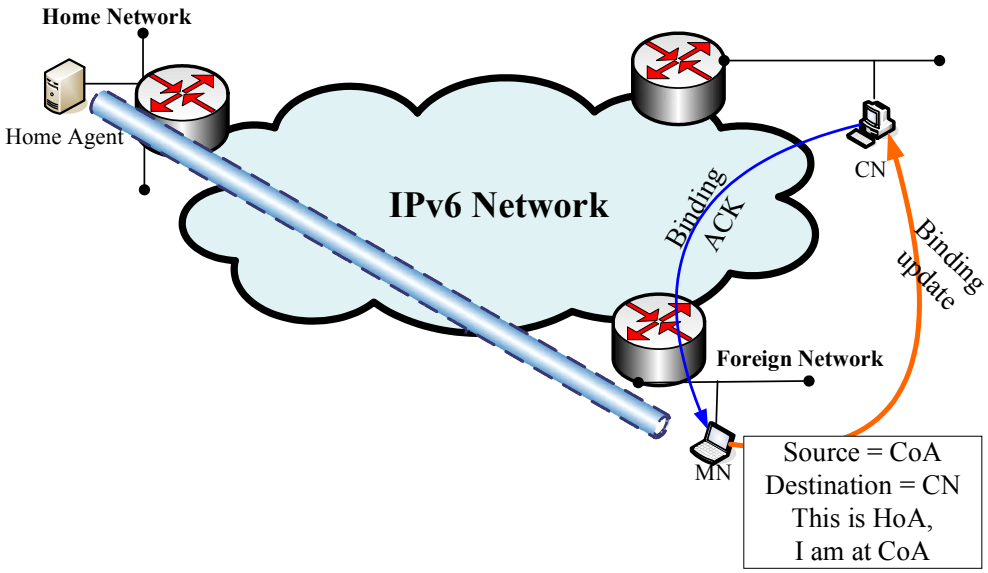

Figure 2.3 Binding update and binding ACK messages exchanged between the MN and CN

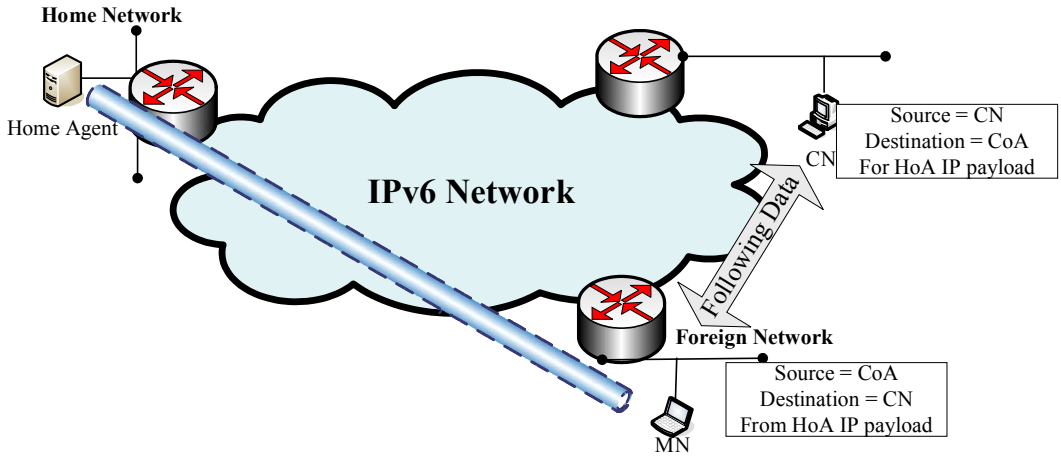

Figure 2.4 IPv6 packets transmitted directly between the MN and CN after Binding with CN

The packets direct to the correspondent node from the mobile node have a header field called the home-address destination option (HAO), which contains the mobile node's HoA. The packets from the correspondent to the mobile contain this HoA in a type-2 routing header (RH). When a correspondent node is sending a packet, it compares the destination address against the home address in its binding cache. If a binding exists, it sets the destination IP address to the CoA and inserts the RH after the IP header. The mobile node receives the packet, copies this HoA from the RH back into the destination address field and removes the RH, thus

re-creating the original packet. Similarly, when the mobile node is preparing to send a packet to a correspondent node, it uses the CoA as the source IP address and inserts the HAO. When the correspondent obtains this packet, it overwrites the source address with the HoA from the HAO, again re-creating the original packet. This way, the mobility is transparent to the upper layer protocols, including IPsec and the transport layer. The only address they can see for this mobile node is its home address. [24]

2.3.3 Security Issues

The security of Mobile IPv6 has been the primary issue to solve to allow for standardization of Mobile IPv6. In the area the data security, the basic objective during the development of Mobile IPv6 has been that it must be at least as secure as basic IPv6 or IPv4 and it should not introduce any new security threats to the network. The biggest vulnerability is the authorization of Binding Updates. As discussed, Route Optimization of MIPv6 is built into the IPv6 protocol and it greatly improves the efficiency of routing by eliminating triangle routing. However, RO also adds binding updates sent by a MN to its CNs, and so, it greatly increases the security risk of MIPv6. Unauthenticated or malicious BUs open the door for many types of attacks, a few of which will be discussed below.

2.3.3.1 False binding update attacks

A spoofed binding update could be sent to either a home agent or a correspondent node. By spoofing binding updates, a malicious node can redirect packets to itself or another node, or prevent the original node from receiving the packets intended for it. For example, as figure 2.5 shows, both nodes A and B are communicating with each other. Now, an attacker, C, is sending a false binding to node B, claiming that it is the node A, and has moved to a new care-of address. If node B trusts this binding it will create a false binding for node A's care-of address, then subsequently redirect future packets to node C. That way, the node A would

not receive the data which it desired, and if the data in the packets is not protected cryptographically, node C will be able to see all of node A's sensitive information.

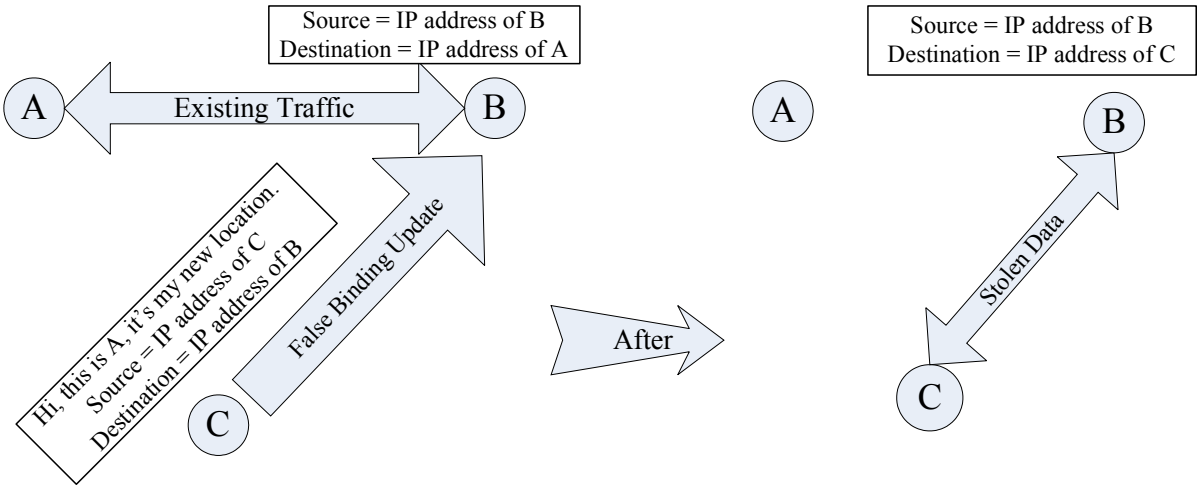

Figure 2.5 False Binding Update attack

2.3.3.2 Man-in-the-Middle attack

An attacker node can also send false BUs to both mobile node and correspondent node together, in order to set itself as a Man-in-the-Middle between them. For example, as figure 2.6 shows, nodes A and B are communicating with each other. We introduce a node C, which sends two false BUs, one to node A and the other to node B. The node C tricks node A and claims itself as node B which has moved to new location C. On the other side, the node C sends a false BU to B to claim itself as A which has moved to care-of address C. This would cause both nodes A and B to send all packets to node C rather than to each other. C can then forward, possibly altered, packets from A to B, and vice versa, which can lead to both A and B believing that nothing is wrong. That way, it's easier for attacker to capture or even alter the sensitive information of any nodes if the packets are not cryptographically protected.

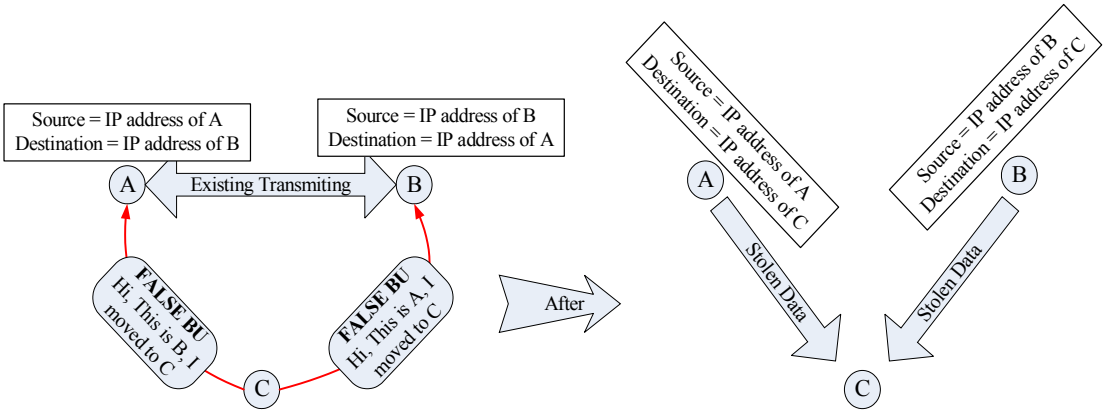

Figure 2.6 Man-in-the-Middle attack

2.3.3.3 Denial-of-Service attack

By sending spoofed BUs, an attacker also can send huge unwanted packets to attack a node or a network. For example, as figure 2.7 shows, node A is communicating with node B, it requests the node B to send some huge videos to it. After the node A has received the first packet, it sends a false BU to the correspondent B, saying to redirect subsequent data traffic to the new location C, which could be an arbitrary node. That way, the node C will then be bombed with the huge unwanted packets. Similarly, the attacker could also, through this attacking approach, using the spoofed BUs to redirect several streams of data to any addresses with the network prefix of a particular target network, thereby congesting an entire network with this unwanted data.

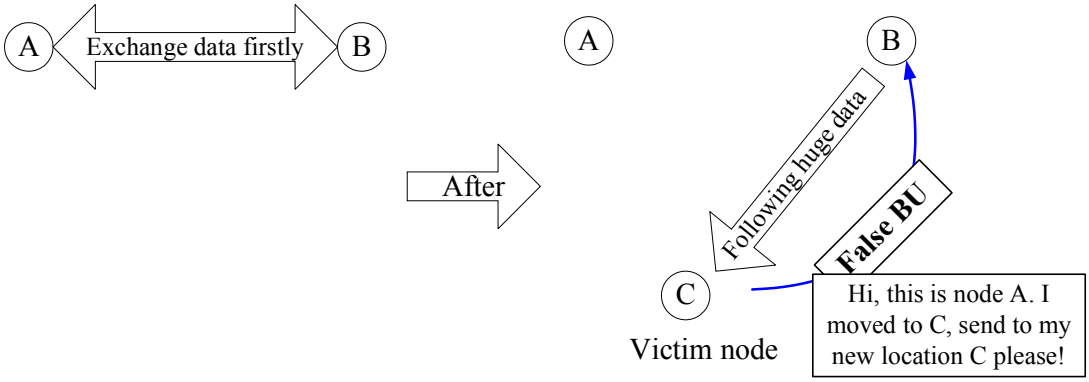

Figure 2.7 Denial-of-Service Attack

All these threats are operated by unauthenticated or malicious BUs. We have seen that unauthenticated location information makes it possible for an attacker to misinform correspondents about the mobile's location, and thus, to redirect packets for the mobile to a wrong destination. This can lead to the compromise of secrecy and integrity as well as to denial-of-service because the target modes are unable to communicate. The obvious solution to the BU spoofing is to authenticate the binding exchanges between HA and MN, and MN and CNs. A typical authentication system would use a suite of strong cryptographic authentication protocols and a certification infrastructure, such as IPsec and IKE which can provide protection between the HA and MN as has been mentioned in the section 2.3.2.1.

However, IPsec depends on the public key infrastructure (PKI), which is not widely deployed. A correspondent node can be any node in the network, so the mobile node and the correspondent node will most probably have no relationship in advance. There does not currently exist any infrastructure that could be used to authenticate all IPv6 nodes. We should consider unconventional authentication methods that work without special security infrastructure. We need an alternative to ensure binding message validity using the Mobile IPv6 specific network architecture. This security method will be introduced in the next section.

2.3.4 Return Routability Procedure

When considering authentication of the binding messages between a mobile node and some unknown correspondent node, no preshared secret can be used. There also doesn't exist a global public key infrastructure that could be utilized, so at least some other key distribution system than IKE would be needed. Because of that, IPsec as a whole isn't very usable for authentication between the mobile node and the correspondent node.

The Return Routability (RR) method was developed to provide an infrastructureless and adequate authentication method between a MN and a CN. First, it ensures that the MN is able to receive messages sent to both its HoA and CoA, which relies upon correct operation of the routing system and HA, and authentication of MN at HA. Then RR concludes that both the HoA and CoA must represent the same system. After that it protects the binding

messages between the MN and the CN using keys returned to the MN during the HoA and CoA tests. The MN can receive messages with the HoA only if the MN has created a valid binding to the HA in advance.

Before discussing the details of the RR procedure, we introduce some terms which will be used in later sections:

A node key, or Kcn, is a 20-octet secret, random number held by every correspondent node that helps to identify itself and the keygen tokens that it generates.

A nonce is a number, normally 64 random bits, held by each correspondent node and updated at regular intervals.

A nonce index is associated with each nonce to help the CN identify which nonce, the current one or one of a previous few, was used with a particular message.

The binding management key, denoted kbm, is used to key the hash algorithm, and is established using data exchanged during the return routability procedure.

The function used to compute hash values is SHA1 [33]. Message Authentication Codes (MACs) are computed using HMAC_SHA1[34, 33]. HMAC_SHA1 (k,m) denotes such a MAC computed on message m with key k.

The Return Routability procedure builds proof to the correspondent node that the mobile node is in fact addressable at its claimed care-of address and home address. This is done by testing whether packets addressed to the two claimed addresses are routed to the mobile node. The mobile node can pass the test only if it is able to supply proof that it received certain data (the "keygen tokens") which the correspondent node sends to those addresses. These data are combined by the mobile node into a binding management key. In following procedure details, we will show how RR provides protection between the MN and CN.

The RR procedure consists of two checks, a home address and a care-of address check. We must assure the two addresses are addressable. The home address must be verified to prevent spoofing of binding updates. The care-of address must be verified to protect against denial-of-service attacks in which the correspondent node is tricked to flood a false care-of address with packets. The real return routability checks are the message pairs (Home Test, BU) and (Care-of Test, BU). The home test init (HoTI) and care-of test init (CoTI) packets are only

needed to trigger the test packets, and the BU message represents a combined routability response to both of the tests. Figure 2.8 shows the whole procedures of RR.

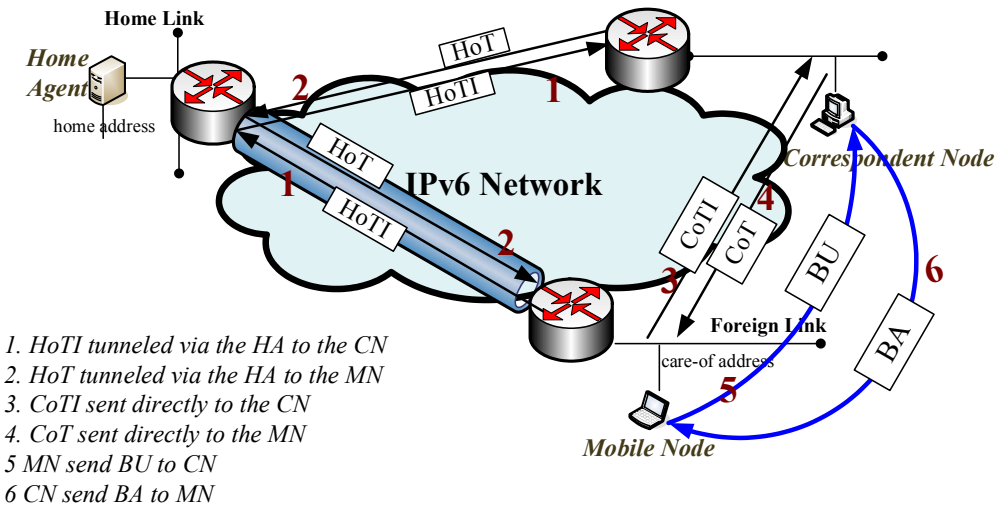

Figure 2.8 Return Routability Procedure

1) Home Address check: The Home Address check consists of a Home Test (HoT) and a subsequent binding update. The HoT is assumed to be tunneled by the home agent to the mobile node. The HoT contains a cryptographically generated token, home keygen token, which is formed by calculating a hash function over the concatenation of a secrete key Kcn known only by the CN, the source address of the HoTI packet, and a nonce. The index to the nonce is also included in the HoT packet, allowing the correspondent node to more easily find the appropriate nonce. (Steps 1 and 2 as shown in figure 2.8.)

In most cases the HoT packet is forwarded over two different segments of the Internet. It first traverses from the correspondent node to the Home Agent. On this trip, it is not protected and any eavesdropper on the path can learn its contents. The HA then forwards the packet to the mobile node. This path is taken inside the IPsec ESP protected tunnel, making it impossible for the outsiders to learn the contents of the packets.

2) Care-of address check: The care-of address check is sent directly from the CN to the MN's care-of address. The token is created in a slightly different manner in order to make it impossible to use home tokens for care-of tokens or vice versa. (Steps 3 and 4 as shown in figure 2.8.)

3) When the mobile node has received both the HoT and the CoT messages, it creates a binding key Kbm by taking a hash function over the concatenation of the tokens received. This key is used to protect the first and the subsequent binding updates, as long as the key remains valid.

Kbm = hash (home token | care-of token)

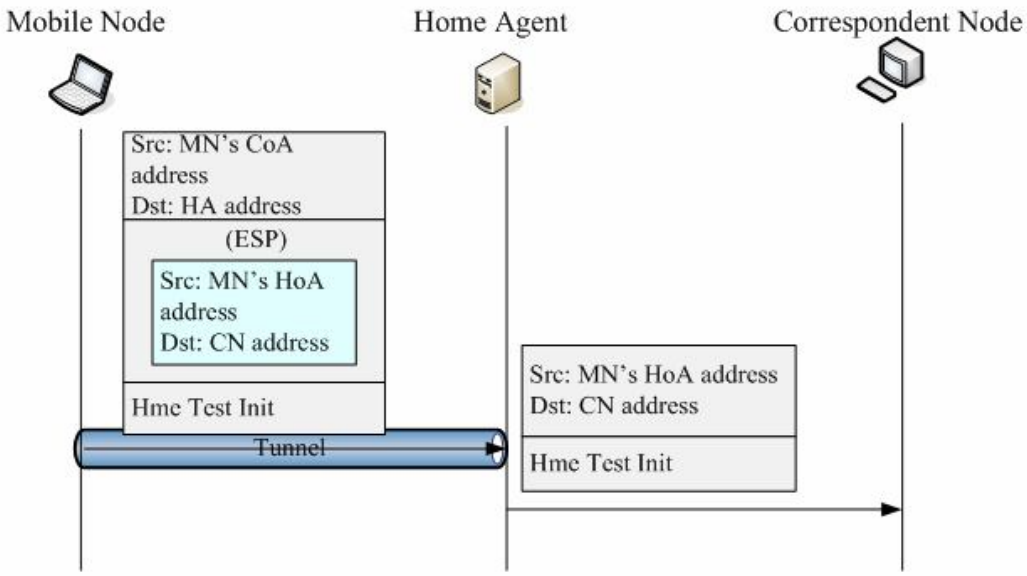

Figure 2.9 Home Test Init message

Now let us see the details of each step:

1. The mobile node sends a Home Test Init (HoTI) message with the home address as source to the correspondent node address. This packet is sent from MN's CoA in a tunnel and sent to the HA, then the HA forwards this packet from MN's HoA to the CN. Figure 2.9 shows the essence of this packet.
2. The mobile node sends a Care-of Test Init (CoTI) message from the care-of address directly to the correspondent node. The mobile node conveys its care-of address and

care-of init cookie in the care-of test init message to correspondent node. The care-of init cookie is to be returned later. Figure 2.10 illustrates this packet.

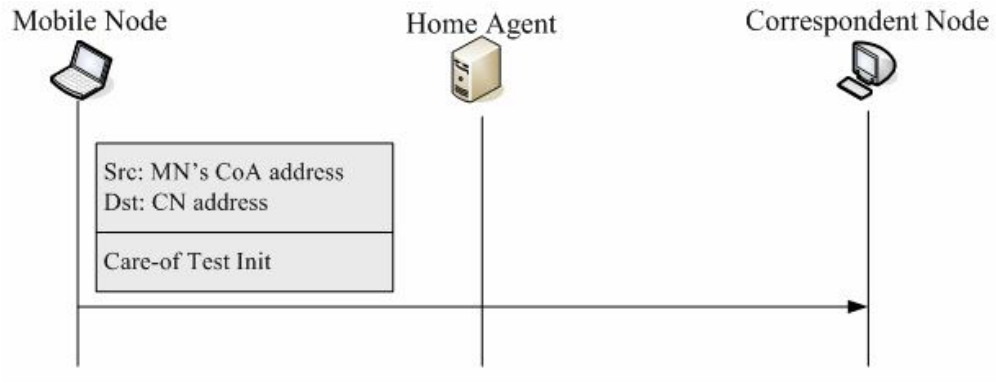

Figure 2.10 Care-of Test Init message

The previous two steps can be performed at the same time by the mobile node, to request keygen tokens from correspondent node. Each message also has an init cookie, which is a 64-bit random value and must be returned by the CN in the next step to very the identity of the correspondent node.

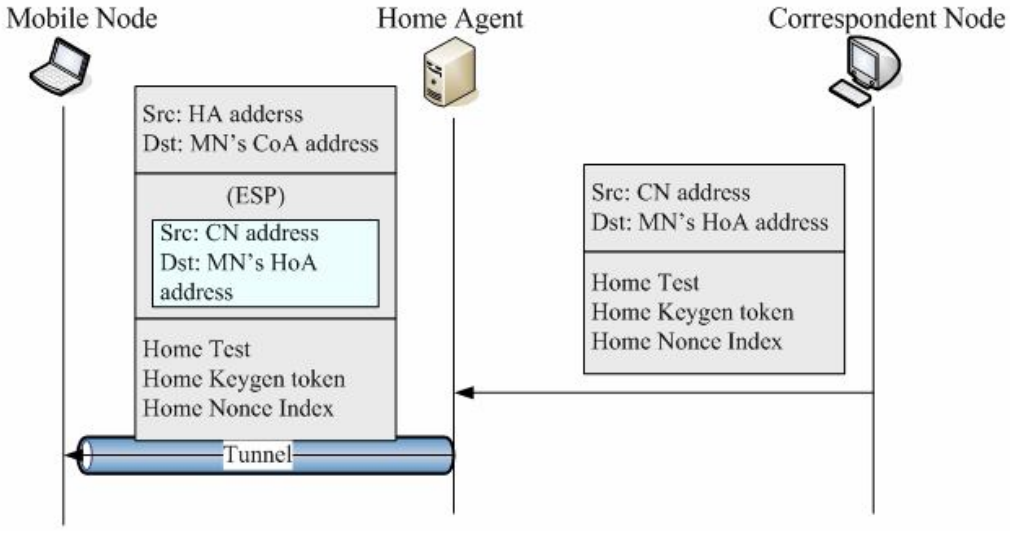

Figure 2.11 Home Test Message

3. The correspondent node sends a Home Test (HoT) message in response to the HoTI message (sent to the MN's HoA and relayed to the MN by the HA). This message contains the home keygen token, which will later be used by the mobile node to

prove its home address to correspondent node. The home init cookie from the mobile node is returned in the Home Test message, to ensure that the message comes from a node on the route between the home agent and the correspondent node. The home nonce index allows the correspondent node to efficiently find the home nonce value that it used in creating the home keygen token. This packet is shown in Figure 2.11.

4. The correspondent node sends a Care-of Test (CoT) message in response to the CoTI message (sent directly to the mobile node). In this message, mobile node can obtain a care-of keygen token, which is used to prove the reachability of its current care-of address. The care-of init cookie is returned to the mobile node to ensure that the message comes from a node on the route to the correspondent node. The care-of nonce index is provided to identify the nonce, which is used for calculating the care-of keygen token similar to the home nonce used in creating home keygen token. Figure 2.12 shows this packet.

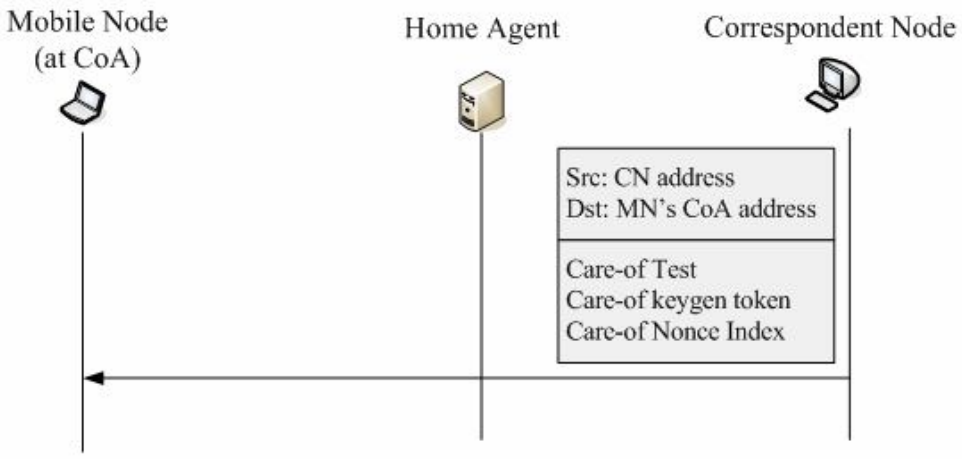

Figure 2.12 Care-of Test message

Steps 3 and 4, use the information from the received init messages from the MN, from steps 1 and 2. The CN generates a home keygen token and a care-of keygen token from a hash function using the first 64 bits of the MAC, Kcn, home address and nonce. The generations of home keygen token and care-of keygen token as follows:

home keygen token: = First (64, HMAC_SHA1 (Kcn, (home address | nonce | 0)))

Where | denotes string concatenation, the secret value is only kept in CN, HMAC_SHA1( ) denotes a keyed hashing MAC scheme using hash function SHA1, and First (n, M) denotes the first n bits of message M.

Similarly, the care-of keygen token is calculated as below:

care-of keygen token : = First (64, HMAC_SHA1 (Kcn, (care-of address | nonce | 1)))

5. When the home and care-of keygen tokens are both received by the MN, it creates a binding key, denoted by Kbm, generated from *SHA1 (home keygen token | care_of keygen token)*. Kbm becomes the shared secret key between MN and CN via the RR procedure. Soon after, the MN sends a BU message to the CN. The binding update message contains MN's home address, home and care-of nonce indexes, sequence number, and the MAC. The MAC is a new value, which is calculated as:

   MAC = First(96, HMAC_SHA1 (Kbm, (CoA | CN's address | BU))),

   where BU indicates the binding update message itself.
6. While CN receives the BU with message authentication node using Kbm as MAC key, it can rebuild Kbm dynamically and verify the validity with the help of home and care-of nonce indexes. If it is legal, then CN optionally sends back an acknowledgement with MAC = First (96 (HMAC_SHA1 (Kbm, (CoA | CN's address | BA)))) if the MN requested that.

The RR procedure partly consists of sending home test init message to the home agent to be forwarded from the mobile node's home address to the correspondent node, and receiving the home test message which is sent by the correspondent node to mobile node's home address intercepted there by the home agent and forwarded to the mobile node. The mobile node gets a home keygen token, from the home test packet. The mobile node can use this token to demonstrate its home address actually belongs to it. The care of test is also included in this procedure, which can demonstrate the mobile node's current care-of address really belongs to it.

The RR procedure precludes impersonation, denial of service, and redirection-based flooding by attackers which are not on the path from the correspondent node to the victim mobile node, and it is sufficiently lightweight not to expose expensive operations. But

the return routability procedure fails to protect against attackers that are located on the critical path (on the path from the correspondent node to a stationary victim or from the correspondent node to the victim's home agent), it can eavesdrop on the returning HoT message and learn the information that is necessary for spoofing the BU, leading to its breaking of the protocol.

This is within design parameters as attacks by on-path nodes can be carried out upon non-mobile nodes. Thus Mobile IP does not reduce the security level because of this. Also, Return Routability requires regular repetition of the entire process, so an attacker would need to remain in place for a lengthy period to undertake a meaningful attack, merely snooping one packet and vanishing is insufficient. In addition to that, there are some problems with Return Routability:

Firstly, it still relies upon routing system integrity, it can be defeated using a pair of attackers, one on the home link, another one on the foreign link, so the message to the HoA can be observed before (at the same time) as it gets to the HA (after that it is encrypted and so safe), and the message to the CoA can be observed on the foreign link, the two attackers can also send the HoTI and CoTI messages.

Secondly, it requires the HA being operating and reachable from both correspondent node and mobile node.

2.4 Cryptographically Generated Addresses (CGA)

Mobile IP [1] is not the only Internet related protocol that faces the problem of needing to be able to assure a correspondent (or peer) that the address it claims to own is legitimately allocated to the node claiming it.

Neighbor Discovery [9], the set of protocols that implement the network to link layer address mapping mechanism, is another with the same issue: "Is the node claiming address A the actual owner of that address".

The Secure Neighbor Discovery (SEND) [16] working group of the IETF developed a solution to this problem, using a new form of address. This is relevant to our work as this same new address form has also been used for the similar purpose in Mobile IPv6.

In this section we introduce this new address type and explain its operation and properties, then in the following section we will explain how these addresses are used to enable Enhanced Route Optimization [17] for Mobile IPv6.

IPv6 provides a large address space as well as simple header structure, flexible options, powerful QoS functionalities, security, mobility, and so on. Among these various services provided by IPv6, the Neighbor Discovery (ND) protocol [26] uses 5 different message types, and provides the following features: router detection, prefix detection, parameter detection, address auto-configuration, address translation, next hop determination, neighbor unreachability detection, duplicate address detection, redirection, and so on. Even though the extended header of IPv6 supports IPsec, the ND protocol implemented on top of ICMPv6 does not support its own security mechanism. So, the ND protocol is vulnerable to sniffing, spoofing, modification and fabrication.

The solution to this problem developed by the SEND working group depends upon a new address type: the Cryptographically Generated Address (CGA) [15]. Cryptographically Generated Addresses (CGAs) are IPv6 addresses, which are used to secure the association of an IPv6 address, with a public key. It allows proving that the sender of SEND messages is the real owner of its CGA address. As stated in [32], "*The CGA defines a decentralized mechanism to bind the public key to its owner, and is radically different from the legacy approach which centralizes the binding through an electronic certificate being generated by a centralized unique entity. SEND makes this decentralization possible by the compulsory use of a pair of self-generated RSA keys and an RSA signature option. The CGA also makes use of some electronic certificates but these certificates are only attached to the routers and serve to prove that a router is authorized to announce itself as a router with declared subnet prefix on the local link*".

2.4.1 Overview of CGA principle

The interface identifier of a CGA is generated by computing a cryptographic one-way hash function from a public key and auxiliary parameters. Securely associating a

cryptographic public key with an IPv6 address can be verified by re-computing the hash value and by comparing the hash with the interface identifier. Messages sent from an IPv6 address can be protected by signing the message with the corresponding private key. So, the verification of an address ownership works without a certification authority or any security infrastructure.

2.4.2 CGA Format and Parameters

In principle, CGAs are generated like IPv6 addresses by concatenating a 64 bit long subnet prefix with a 64 bit long identifier. However, in CGAs the identifier is a hash value formed from a CGA parameter set, including among others, a public key. Knowing these CGA parameters, any receiver of IPv6 packets with a CGA as source address can re-calculate the hash value, and verify if it matches the one contained in the 64 bit identifier of the packet's source address. Figure 2.13 shows the structure of a CGA. The leftmost 64 bits of the CGA is the network prefix and the rightmost 64 bits of the address is a cryptographic hash value computed with sender's public key and auxiliary parameters. The security parameter (Sec), is an unsigned three-bit integer (bits 0-2), which exists to increase the strength of the CGA to protect against brute-force attacks. The "u" and "g" bits are from the standard IPv6 address architecture format of interface identifiers [25].

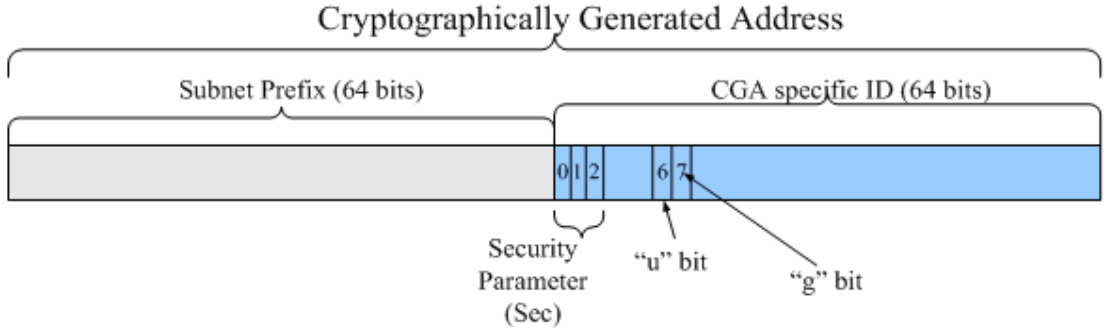

Figure 2.13 Structures of CGA

The CGA parameters mentioned above, which are used for calculating the CGA, are formed into a string as presented as figure 2.14. Table 2.1 describes each parameter.

modifier	subnet prefix	collision count	public key	extension fields

Figure 2.14 CGA Parameters data structure

Table 2.1 Description of CGA Parameter fields

Parameter	Description
Modifier	128 bit random number. Implement the hash extension and to enhance privacy by adding randomness of the address.
Subnet Prefix	Subnet prefix of the CGA, 64 bits.
Collision Count	8 bits long, and incremented during CGA generation to recover from an address collision detected by duplicate address detection.
Public Key	The public key of the address owner, variable length.
Extension Fields	Not used in the current specification.

2.4.3 CGA Generation and Verification

Using the CGA parameters, the identifier part of the CGA can be calculated by the following steps:

1. Generate a public/private key pair. This may be done well in advance.
2. Choose any arbitrary value for the 128 bits modifier.
3. Choose appropriate value between 0 and 7 as security parameter Sec. The higher the value selected for Sec, the more difficult it will be to break a generated CGA address with brute-force attacks, but also longer it will take to generate the CGA itself.
4. Concatenate the selected modifier, the subnet prefix and the collision count (both set to zero), and the public key value, and calculate from this concatenation a 160bit hash value using the SHA-1 algorithm. The hash value Hash2 will be the first 112 bits.
5. Compare the (16*Sec) first bits of Hash2 with zero. If they don't mach, increase the modifier by 1 and calculate the next hash value. This will be repeated until the (16*Sec) first bits of Hash 2 are all zero. Note that if Sec is zero this step does nothing.

6. Concatenate the final value for the modifier, the real subnet prefix, the collision count set to zero, and the public key, and calculate from this concatenation a 160 bit hash value using the SHA-1 algorithm. The hash value Hash 1 will be the first 64 bits of the result.

7. The CGA specific interface identifier will be Hash1, with the first 3 bits replaced by the Sec parameter, and bits 6 and 7, the “u” and “g” bits, set to zero.

8. Optionally, one could now perform collision detection in order to check if someone else on the subnet is using the same IPv6 address. If so, the collision counter should be increased by 1, and a new Hash 1 value should be generated with its modified CGA parameter. In order to protect against DoS attacks, this process is stopped after three collisions.

In order to allow the receiver to verify a CGA, it needs to have the CGA parameters as well as the CGA itself. When a node receives a packet form one of its neighbors which used a CGA address, it can execute the CGA verification algorithm. The received packet should contain an ICMPv6 CGA option carrying the final CGA parameters Data Structure.

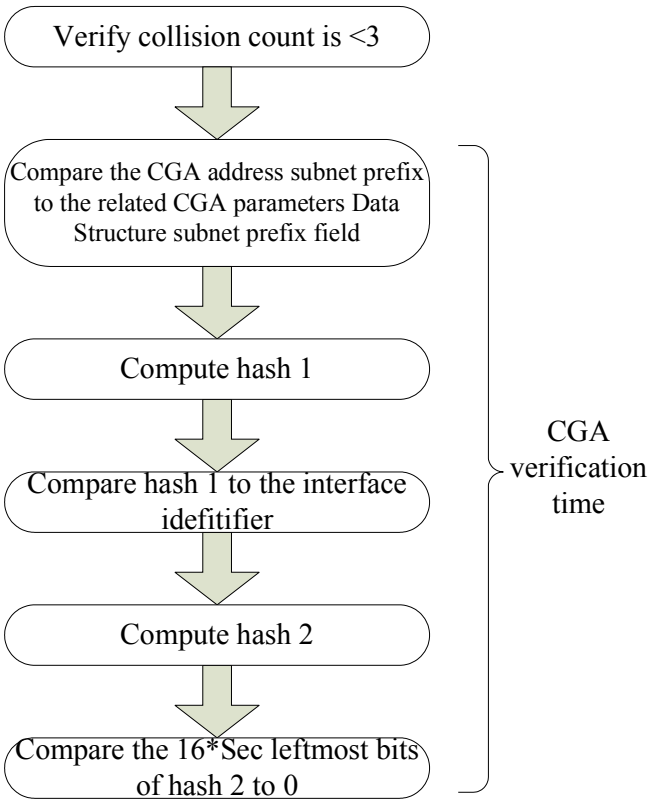

Figure 2.15 CGA verification algorithm

The verification algorithm presented in figure 2.15, starts by checking that the collision count value is less than 3. The subnet prefix from the sender's IPv6 source address (obtained from the packet header) is then compared to the subnet prefix contained in the CGA parameters data structure. Next, hash1 is computed and compared to the interface identifier, omitting bits 0 to 2, bit 6 and bit 7 (encoding respectively the SEC value, u and g). Finally, Sec is extracted from the interface identifier (the 3 leftmost bits) and hash2 is computed. Its (16*Sec) leftmost bits are checked to be equal to 0. If any of the previous tests fails, the CGA address is considered as unsecured.

Additionally, the sender signs the associated message with an RSA signature by using the complementary private key that was used to generate the CGA. In cryptography, RSA (which stands for Rivest, Shamir and Adleman who first publicly described it) is an algorithm for public-key cryptography. It is the first algorithm known to be suitable for signing as well as encryption, and was one of the first great advances in public key cryptography. The RSA signature option is shown in figure 2.16, it has a key hash field that contains the most significant 128 bits of the SHA-1 hash of the public key that was used to generate the signature. This value should correspond to the public key that is contained in the CGA option of the same message. The Digital Signature field contains the signature value itself that is generated with the RSASSA-PKCSI-v1_5 algorithm and SHA-1 hash as defined in [41]. The receiver of the protected message can then verify the message's authenticity primarily based on the RSA signature, while also using the CGA data structure.

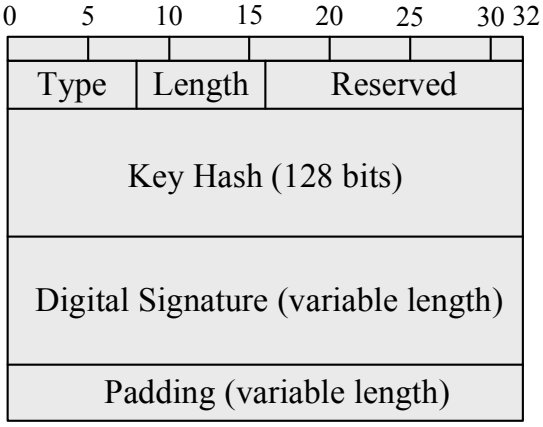

Figure 2.16 RSA Signature Option

After a successful verification, the receiver can securely assign a certain public key to an IPv6 address, information which is usually provided by certificates. With this information the owner of a CGA uses its private key in order to sign messages, knowing that the receiver will be able to obtain the appropriate public key and use this for the verification of the message signature. When the signature is verified, receiver knows that message was generated by holder of private key associated with public key that generated the CGA. That is, this message was generated by the owner of the public key, which is the owner of the CGA.

To summarise, the CGA is generated by a procedure that binds it to a particular public key. A message sent from the CGA can be signed by the associated private key demonstrating that the owner of the CGA sent the message received.

2.5 Enhanced Route Optimization

Section 2.3.4 indicated areas where an improved Binding Update security mechanism could be of benefit. Using Return Routability there can be a long delay before a BU to the CN is accepted, and there are the remaining security issues. If we instead use CGA method for Mobile IPv6 we will reduce this problem. Using CGA we can just send binding update information, just one message can authenticate the MN, avoiding the BU long delay problem. Finally CGA uses public key signatures for binding message authentication; it is much more difficult to attack.

There is a new protocol – Enhanced Route Optimization [17], one optimization of which proposes use of an initial CGA-based authentication to securely exchange a secret Permanent Home keygen token (PHKT) that replaces the home keygen token produced during the previous HoT procedures between the mobile node and correspondent node. It can ensure packets only be directed to the legitimate recipient without repeating HoT in future. The purpose of HoT is ensuring packets can only be redirected by the legitimate recipient. The legitimate recipient is identified by the home address, and only the legitimate recipient is expected to receive the home keygen token sent to the home address. Base Mobile IPv6 requires mobile node to renew a correspondent registration at least every 7 minutes. A compete correspondent registration

involves HoT and CoT also. So the HoT needs to repeat and repeat for a correspondent registration until the mobile node wants to delete this registration with the CN.

If the mobile node does not use a CGA as its home address then the Return Routability procedure is the only current applicable authentication method.

If the MN is using a CGA, then there are two additional authentication procedures.

First, if this is the first BU for a particular CN, so the MN has no permanent keygen token, then the MN uses the properties of its CGA and sends a signed BU to authenticate itself. In return, it receives a permanent home keygen token it can later use to authenticate itself. This token is returned encrypted using the MN's public key so only the MN can decode it.

A node that uses a CGA at a certain time can prove at a later time that it is still the same node when it uses this CGA again. Instead of relying on the routing property, as with the HoT, this proof can be drawn from CGA's special interface identifier. The owner of the CGA signs important packets with its private key and includes its public key along with the auxiliary data in these packets. Since it is computationally hard to produce another public/private key pair that hashes to the same CGA, the recipient of the signed message can verify, by computing the hash and comparing it with the CGA's interface identifier, that the sender must be legitimate owner of this CGA. Using this mechanism, we could avoid repeating the HoT procedure in future binding procedures, reduce the handoff latency, securely authenticate the mobile node without preconfigured credentials or a public-key infrastructure (even with an attacker present on the path from the correspondent node to the mobile node), and degrade the level of signaling overhead compared to a base mobile IPv6 correspondent registration especially when the mobile node does not move frequently.

This procedure still requires a preceding Home Address reachability check (HoTI/ HoT) for the BU message to verify that the MN is the legitimate owner of the home address. The reason of doing home address reachability check is the CGA property of home address can't assure the owner's reachability at the home address. While CGA generation function cryptographically ties the interface identifier of a home address to the subnet prefix of the home address, the function accepts any subnet prefix and so it does not prevent a node from cryptographically generating a CGA with a spoofed subnet prefix. So anyone could

cryptographically generate home address by using its own public key with the victim's subnet prefix and then claim this address as his home address. Because the CGA identifier is the output of a cryptographic has function, it is effectively a random number. There is no known way to select a particular desired value, so it is not possible to spoof a particular victim IPv6 address this way.

If we ignore the step of home address test, a malicious node could generate this address and perform the correspondent registration to a correspondent node as normal. Then the attacker tricks the correspondent node into sending flooding packets to its care-of address and subsequently deregisters the binding or lets it expire. That results in the correspondent node redirecting these flooding packets to the victim's home network. This attack is known as "return-to-home flooding attack". Because of using the CGA as home address, with the attacker stealing someone's subnet prefix as its home address's prefix, it can't pick upon a particular node on the network to attack. This Denial-of-service attack is just bombing on the network.

2.5.1 Procedure of Enhanced Route Optimization

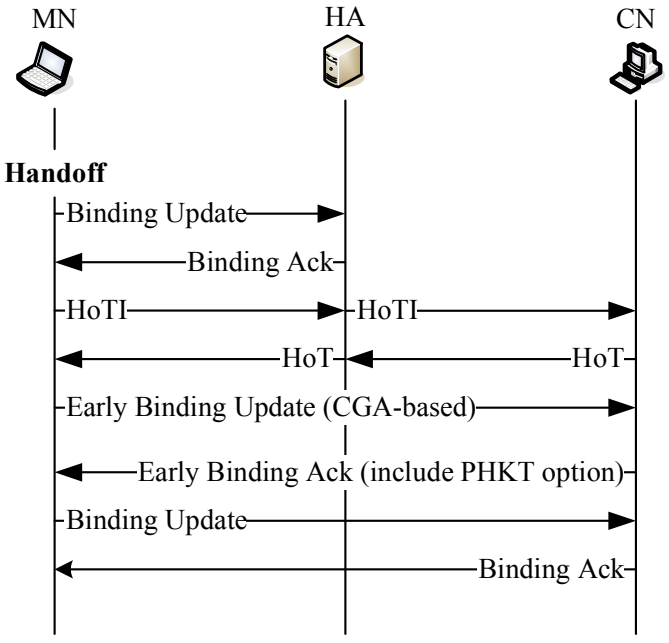

Figure 2.17 The procedure of correspondent registration based on CGA authentication

Figure 2.17 shows the basic sequence of operations of a node using ERO to perform a binding update with a correspondent node.

As an additional enhancement, ERO allows a correspondent node to send payload packet to a mobile node's new care-of address before the mobile node has been found to be reachable at the care-of address. When the MN moves to a new location, it first updates its binding at the CN to notify it of the new care-of address without providing a proof of reachability. The proof of home address ownership can avoid the threat of faking home address, but an attacker may still bind a correct home address to a false care-of address and thereby trick a correspondent node into redirecting packets, which would otherwise be delivered to the attacker itself, to a third party. Ignoring to verify the reachability of a mobile node's claimed care-of address could therefore cause one or multiple correspondent nodes to unknowingly contribute to a redirection-based flooding attack against a victim mobile node chosen by the attacker. So the care-of test is as important as the home test remained in Enhanced Route Optimization (ERO), though the ERO method allows this test to be slightly deferred, using a credit based authorization scheme for protection, to further speed BU processing. This method is not important to this work.

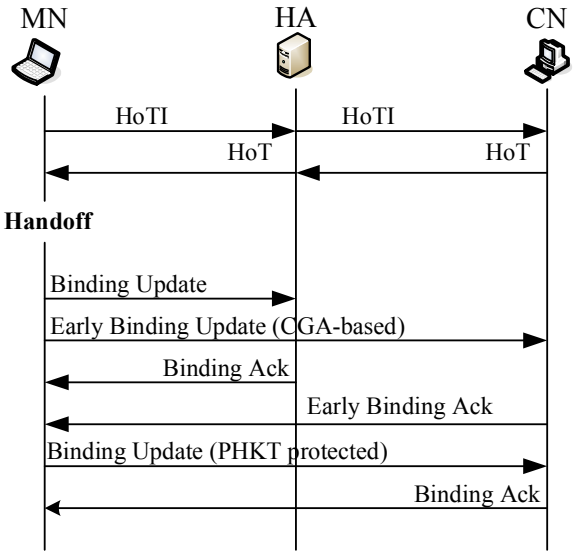

Figure 2.18 HoT is processed before MN handoff

ERO uses a CGA to validate the mobile node's home address and request a concurrent care-of address test for increasing handoff efficiency. As figure 2.17 shows, after the

MN changes IP connection, it performs HoTI/HoT. Sometimes, the HoTI/HoT maybe performed before the mobile node changes location, as shown in figure 2.18. In order to remove the delay source, ERO permits deferred CoTI/CoT. Actually, the CoTI is performed as an option in the first BU, which is called an Early Binding Update (EBU), the CoT that is combined with the Early Binding Acknowledgement (EBA). The BA or EBA that follows the CGA protected BU carries the PHKT, which is used to authenticate future BUs from the MN to the same CN.

Once the mobile node has authenticated itself using its CGA, and has received a permanent keygen token, it can use that token, replacing the temporary token the HoTI/HoT procedure provides, to authenticate all future BU messages to this CN. Thus no future HoTI/HoT exchanges are required and no further public key encryption operations, as figure 2.19 shows.

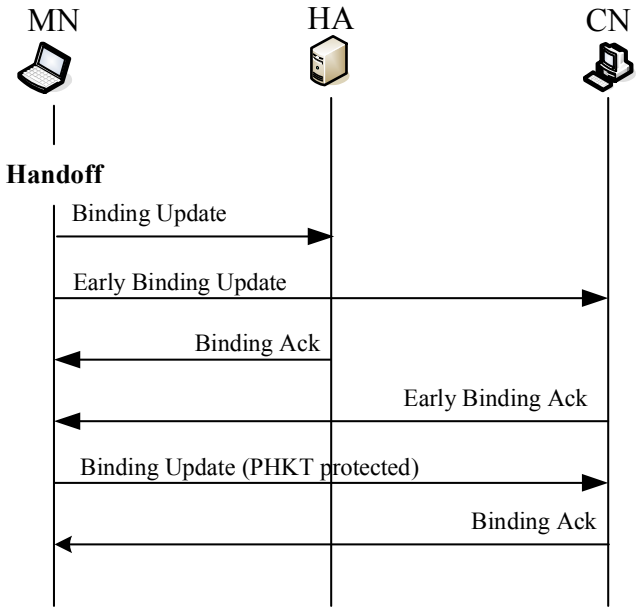

Figure 2.19 The correspondent registration with authentication through the MN's permanent home keygen token

Kuang Shilei[39] has provided previous work for us, which focused on implementing, investigating and evaluating the application of CGAs to Mobile IPv6. From that work, we found a problem with applying CGA in general is that they involve computationally expensive algorithms. Because of this, ERO [17] requires use of Semi-Permanent Security Associations to increase efficiency. But the first BU CGA-based will still be a problem for a small device with low processing power. Similarly, any CN that communicates simultaneously

with many MNs may find its resources pressured by the costs of the CGA algorithms. In addition, the CGA method depends on the public-key algorithms that allow a DoS attack by requesting CGA verification in which an attacker can use a long key length that will consume target CPU before determining that the packet is faked.[35]

2.6 Summary

In this chapter, we have shown the basic mechanism of Mobile IP and the issue it was required to solve. For IPv6 we show Route Optimization, practical as with IPv6 we can assume nodes understand the Mobile IP protocols, which was not possible with IPv4.

We then showed the extension of Route Optimization known as Enhanced Route Optimization developed to provide better security and reduced delays.

This provides the background knowledge necessary to understand the remainder of this thesis. In the following chapter we will demonstrate a problem that Mobile IP does not handle currently. The remainder of the thesis will be devoted to the description of a solution to this problem.

CHAPTER 3

PROBLEM STATEMENT

In normal mobile IP, the home agent (HA) is an essential component. Mobile nodes will not function properly if the HA cannot be reached. In this chapter, we introduce some cases where communication fails between the mobile node and the correspondent node because the mobile node or correspondent node cannot connect to the home agent for some reason.

The cases of node communication failures and the possible solutions under present mechanisms and authentication methods are presented in sections 3.1 and 3.2 respectively. In section 3.3, a more reasonable solution is proposed for solving the problem.

3.1 Case That Fails

Mobile IP can satisfy the demand that users are no longer required to work in their company's offices and they expect to be connected to their company's home network while they may be moving from place to place. One perfectly rational example for node moving is if the user may be traveling from city to city (or from a country to another country). He commences work before leaving and he wants to continue this work during his travels. Mobile IP can solve this problem. The traveler needs to register his current care-of address every time he moves with its home agent after changing his location, in order to ensure the home agent knows his current location. After the first packet is forwarded between the correspondent node and the traveler, a binding can be established between them so the correspondent node has the knowledge of his current care-of address and stores it in its cache. Later communications can be done directly between the traveler and the correspondent node.

As well as physically moving from one place to another place, we have another scenario where Mobile IP may assist. A node may want to get more bandwidth or other advantage, and it has some existing connections. For example, a node connects to a hundred megabit network

now, and is using it to transmit packets with another node. Then, it wants to connect to a gigabit network. We assume this node is able to connect to the gigabit network. If it did that, it will get a new address. Then it can use mobile IP to allow the existing connection to remain operational.

In both of the above cases, mobile IP can help the node keep existing connections. But if the home agent is unavailable or unreachable, perhaps the links between the home agent and the correspondent node or between the home agent and the mobile node are broken, the packets sent to or from the correspondent node via the home agent never reach their destination. In this case, what will happen?

Here is a third scenario causing the need for a node to move. Consider a node A at its home network, not intending to be mobile. This node is engaged upon important work with a correspondent node. Then the network link between the correspondent node and the node's home network breaks (as figure3.1 shows).

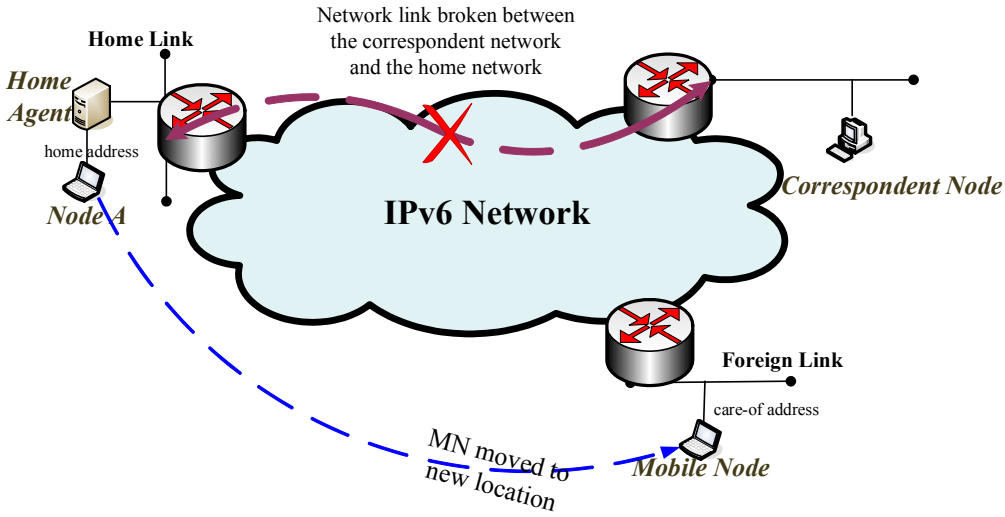

Figure 3.1 Network link between the CN and HA fails

The correspondent can't talk to the home network and so the packets can't be transmitted between the node A and the correspondent node. In this scenario, the node A decides to become a mobile node and wants to move to some place where it can talk to the correspondent. An intelligent place for the node to move might be to the correspondent network. Where the node A moves to it is able to communicate with the correspondent node. But for this movement

everything is too late. We can't do triangle routing, because the packets can't be forwarded between the correspondent node and the mobile node via the home agent. Route Optimization is provided by Mobile IP to allow transmitting packets directly without relaying by the mobile node's home agent. Perhaps we can use the RO to direct packets between node A and the correspondent node? Consider the security problem, which requires the mobile node to have obtained a home keygen token from the correspondent node before the link failed and use it to prove the ownership of the home address. But before the link broke, this node was intending to stay at home forever, so it didn't prepare to get a home keygen token to prove itself.

In this case, the mobile node wants to keep its connection and moves to another network where it can talk to the correspondent node. The mobile node can use CGA authentication to identify itself. If the mobile node has obtained a home keygen token, it can prove the relationship between the mobile node and its home subnet prefix without doing another home test. In order to be using home keygen token to prove itself, the node needs to have a plan on moving before the network link has a problem and have obtained a home keygen token already. But here we have assumed the node never planned to be mobile and so no home keygen token has been obtained for the mobile node and now we can't do the home test. The CGA can't prove the prefix of mobile node's home address, it can only prove this home address was generated by this mobile node, it can't prove that it is entitled to use this home address. So we also need the home test to help the mobile node to validate the prefix of the home address. When the mobile node does a binding update to the correspondent node, the correspondent node should not trust the mobile node and must refuse the binding update. If it did not, then from the security perspective the correspondent node might later send packets to the home address. In that case, a "return-to-home flooding" attack would be possible.

For this situation, using basic mobile IP, without route optimization, also can't help us solve the problem. What we trying to do is to make the node can work normally, when using home agent as a relay between the mobile node and the correspondent node but it fails and the link between the home network to the correspondent node breaks or the link between the mobile node to the home agent breaks.

3.2 Possible Solutions

In order to avoid the flooding attack happening, we must have the home keygen token from home test message and use it to prove home address's property. Here are some possible solutions for that:

First, if the mobile node is at its home network and can get a home keygen token before the link fails, then if it can move quickly enough after its home network link goes down and it is able to do a binding update before the token expires, then the binding will be accepted. The mobile node can't repeat the home test. The probability of this method succeeding is limited in that the return routability procedure must be repeated in intervals of at most 7 minutes. That means the mobile node needs to detect that the network link has failed and determine that it won't come back in a short time, then decide to move and arrive at another network and perform the binding update, all in a very short time, within a period with an upper bound of seven minutes. It's not very practical.

Another possible method for avoiding this problem, is to make the mobile node always do a binding update and obtain a permanent home keygen token before the mobile node moves. This however makes the mobile node and correspondent node do a lot of computation, just in case the network happens to break. In order to avoid losing connections, the mobile node has to obtain a permanent keygen token from all correspondent nodes, and requires all nodes to perform home tests with all CN's just in case this might be needed, not only the ones which happen to be active when the link failed. This is necessary so the node is prepared to keep an existing connection if the link breaks while communicating with this CN. This method also uses the mobile IP functions to handle the problem. It undertakes the cost of doing CGA work and doing the home test work to every correspondent node although usually the mobile will not want to move in practice. We know the cost of doing CGA is significant [15]. Not only is the mobile node doing lots of work, but correspondent node also has to participate. It has to maintain the binding cache to remember the tokens even though the mobile node doesn't move at all. It is unlikely that nodes can be convinced to perform all these expensive operations, just for insurance.

3.3 Proposed Solution

We have seen that the two current available solutions are not practical. We need to find another possible solution, to solve the problem if the network link fails, and no home keygen token was already available for the mobile node. We want the mobile node to be able to move to another place where it can talk to the correspondent node. In order to avoid the flooding attack happening, we must prevent the correspondent node redirecting packets to the home address.

If some one wants to fake an address and attack using Mobile IP, it only can fake a home address or care-of address. If one or both of those addresses are successfully faked, the attacker can initiate a flooding attack. The attacker might trick a correspondent node into sending lots of packets to one of those two networks. We must prevent this.

In the situation we are attempting to handle, the network link fails and the mobile node has no home keygen token to prove its home address. We can't do the home test to prove the home address really belongs to the mobile node. We assume that the mobile node has moved away from the network with the problems. That means the mobile node can use its care-of address to talk to the correspondent node. The mobile node can do the care-of test to prove the care-of address belongs to it, as in basic mobile IP, which solves the problem of flooding the care-of. The mobile node can talk to the correspondent node, send packets and get replies.

In our situation, if some one lets the correspondent node forward packets to the home address, that would allow an attack by using faked home address. We can't validate the home address. We can't communicate with the home network. There is no way for anybody on that network to say this is OK, because that network is disconnected. So we can't possibly validate that the home address really belongs to the mobile node. So we can't trust any procedure that allows the correspondent node to send packets to the home address. Because that way a flooding attack would be possible. The only case the correspondent node should send packets to the home address is after the mobile node has validated its home address to the correspondent node. This is the problem we're trying to solve. We will examine the issue from the correspondent node's viewpoint in two cases:

If a mobile node is an attacker, the only right thing for the correspondent node to do is just not send packets to the home address.

Alternatively, the mobile node is not an attacker, but is a real mobile node.

In the normal scenario, we want the correspondent node to send the packets to the home address, so the mobile node can receive them, perhaps forwarded by the Home Agent.

But in our scenario, we're assuming the packets can't get to the home address anyway because the network link has some problem.

If the correspondent node is sending to the mobile node at its care-of address, and the mobile node stops sending binding updates, the normal procedure would be for the correspondent node to revert to send to the home address. Because, here, that address has not been verified that would open the flooding attack possibility. Here we know the mobile node is not an attacker, but the correspondent node cannot know, or assume, that.

If the network link to the mobile node's home comes back, the mobile node can do a home test and prove its home address. Of course, after the mobile node's home link comes back and if the mobile node returns home, the correspondent node can send packets to the mobile node's home address. If the mobile node doesn't return home, it can still do the home test to validate its home address to correspondent node. After the home test has been completed, everything can work correctly as normal.

In this case we also cannot return a permanent home keygen token, as that signifies verification of the home address. Without that, the repeated required BU packets from the mobile node to the correspondent node would all need to be authenticated using the CGA and signature, with the related computational costs.

We start with an examination of the operation of a correspondent node in normal Mobile IP. Before the mobile node starts send binding update, the correspondent node sends packets to the mobile node's home address. Before we doing any kind of route optimization, the triangle routing works. The correspondent node just keeps sending packets to the same place, where it always was sending. If we are doing route optimization, after the correspondent node receives the binding update from the mobile node, it sends packets to the mobile node's current care-of address. The binding gets renewed and renewed and if the mobile node moves again, we change the binding. That is the current mobile IP specification. When the mobile nodes are

known to be moving around, we have binding cache entry in the correspondent node, so we don't worry about that when we want to do route optimization.

Eventually when the binding cache entry goes away, packets revert to the home network. That's what we must change. The point of our solution is to prevent the correspondent nodes from sending packets to the mobile node's home address if there is no authentication of mobile node's home address but still accept the binding update in this case if it is CGA verified. Because we don't let the correspondent node send packets to the home address without validating the mobile node's home address, we can avoid the "return-to-home flooding attack". The home test is one way that we can use it to verify the home address really belongs to the mobile node. It is possible there may be other methods. For example, we may be able to obtain information from the transport layer (eg: TCP) that indicates that it has been in contact earlier with the MN at its home address, with no binding cache entry redirecting packets. Whether this kind of validation is practical, and safe, is for further study and is not part of this work.

3.4 Summary

We have shown a plausible scenario where a node might want to become mobile and use the services of Mobile IP to allow it to retain existing connections, but where mobile IP is unable to help as currently designed. It appears that by taking advantage of Enhanced Route Optimization and making a simple change to the procedures this defect can be cured.

In the next chapter we will present a detailed design for the solution proposed.

CHAPTER 4

DESIGN

In the previous chapter, a limitation of existing Mobile IP was revealed and a proposed solution was mentioned. This chapter will give the design details of this proposed solution. The key points of this solution are: make the correspondent node receive a binding from mobile node although the MN cannot provide the authentication of its claimed home address, and avoid correspondent node reverting packets to the MN's claimed home network when the mobile node requests to delete or expire a particular binding if the mobile node has not validated its HoA prefix to the correspondent node.

A design overview is given in section 4.1. The altered process of binding at the correspondent node is described in section 4.2. In section 4.3 a new extra flag bit and its operations are introduced. The solution to avoid reverting packets back home is given in section 4.4. The limitations of this design will be given in section 4.5.

4.1 Design Overview

The solution should solve the communication problem between the mobile node and correspondent node when the network link fails so that the home agent cannot assist with the security for the mobile node, but must not add a new security problem (it must avoid the "return-to-home" flooding attack mentioned in the previous chapter). The solution will be based on the current transmitting mechanisms (we assume the nodes support Enhanced Route Optimization). We do not modify the infrastructure and basic operations of the Mobile IPv6. The binding authentication method for the correspondent node must be modified to permit our solution to operate.

When a MN communicates with a correspondent node from its current care-of address, it must update its binding with its home agent and the correspondent node. Before the

correspondent node accepts a binding, it must require verification of the MN's home address. In our solution, this rule will be changed because the HoT can not be accomplished since we want to function when the home network link cannot be reached. We require less authentication in the procedure the CN uses to validate an incoming BU, and allow the CN to accept a "special" BU, which will be introduced in section 4.2.

With this modification, we obviously need to be concerned with the security of the result. Our solution must ensure that packets can be passed safely between the communicators. In order to prevent the "return-to-home" flooding happening, a new flag will be added in our design, and through it, managing the correspondent binding cache to avoid this attack.

Our solution is based on the current Enhanced Route Optimization [17]. We assume there is no permanent home keygen token obtained for mobile node to prove itself and that every node supports Route Optimization.

4.2 Binding Update with Correspondent Node

For route optimization to work, the relationship between the mobile node and the correspondent node is bound through sending a binding to the CN. The BU message must be verified by the CN before it is accepted. When a correspondent node receives a BU message, it first checks the existence of a Binding Authorization Data option and a Nonce Index Option. The message will be dropped if these options do not exist. The Nonce Index Option contains the values of home nonce index and care-of nonce index, which are used for generating a home keygen token and a care-of keygen token by the correspondent node. From the tokens, the CN can generate the shared secret information which was used by the MN when it created the BU message. The way of getting the values of home nonce index and care-of index has been described previously.

In our environment, the network link between the mobile node and its home network or between the correspondent node and the MN's home network is assumed broken. So the home nonce index value cannot be obtained through exchanging the HoTI and HoT messages via the home agent. Only the care-of nonce index can be obtained through exchanging the CoTI

and CoT messages directly between the MN and the CN, as the figure 4.1 shows in which the broken line means the HoT procedures cannot be done whereas the solid line denotes that the COT procedures work well. ERO, in order to reduce the transmitting delay, permits the CN to receive an EBU first if the home address of this EBU can be authenticated, and then keeps checking the sender's care-of address through checking the following complete binding update message, in which MN should calculate the authenticator using new care-of keygen token that is obtained from the returned Early Binding Acknowledgement (EBA) message from CN. After the CN finishes checking the MN's care-of address, the binding procedure is complete and future packets can be transmitted safely between the MN and the CN. To permit this enhancement the care-of nonce index may be omitted from the first BU.

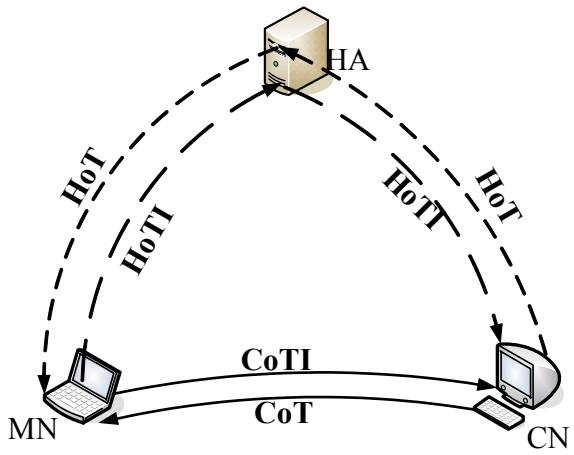

Figure 4.1 Get the nonce index values while the network link has problem

The shared secret key ***Kbm***, is important information for CN to verify a binding. But here, there is no way to obtain it because we can not generate a home keygen token before generating this key. That way, the CN cannot check the reachability of the MN's claimed home address. Then, even if the MN sends the BU to the CN, it will be rejected.

ERO provides three methods for a correspondent node to authenticate a binding. Except for the method that is based on the knowledge of the MN's permanent home keygen token (this requires the MN to have obtained a permanent home keygen token already), the others need the services of the HA to provide security service. Figures 4.2, 4.3 and 4.4 show the essentials of the three authentication methods in ERO:

Home nonce index *Non-null*	CGA Parameters option	Signature option

Figure 4.2 Authenticate based on the CGA property of MN's home address

Home nonce index *Zero*	Permanent home keygen token option

Figure 4.3 Authenticate based on the MN's permanent home keygen token

Home nonce index *Non-null*

Figure 4.4 Authenticate based on the proof of reachability at MN's home address

Figures 4.2 and 4.4 show the authentication methods cannot function because no home nonce index value can be obtained in our situation. We also assumed the MN has not obtained a permanent home keygen token in preparation. So the method shown in figure 4.3 also cannot function.

In order to allow the CN to accept the BU, we must satisfy the base requirements of validating the binding rules. So, here, a home nonce index value is needed to be used to allow the CN to generate a home keygen token. For this requirement, we need to build a new type of BU if the MN cannot accomplish the home testing. For this special BU, we set the home nonce index value to be zero. The essential requirements of this special BU are as figure 4.5 shows:

Home nonce index *zero*	CGA Parameters option	Signature option

Figure 4.5 Essential components of the special BU

Mobility Header	**Other Mobility Option**	**CGA Parameters Option**	**Signature Option**	**Nonce Indices Option** *Home nonce index = 0*	**Binding Authorization Data**

Figure 4.6 Format of special BU

The new special BU message is built as figure 4.6 shows. But this BU cannot be accepted by the CN because the reachability of the home address of this BU sender cannot be checked. We have mentioned the ability of the CGA and what it can prove. The case of figure 4.2 shows that a binding message that includes CGA option followed by a signature option, still needs to check the sender of this binding's claimed home address, which is done through doing the HoT procedures to get a home keygen token which is then used to generate a shared secrete key. Figure 4.3 shows the case where a binding update's sender has obtained a permanent home keygen token, it can use this token to authenticate itself in future binding procedures without doing any more HoT. The situation of figure 4.4 shows normal RR procedure case. Here, in our designed case, figure 4.5, we have the CGA information, but we cannot do HoT to verify the sender's ownership of this claimed home address. If we cannot finish HoT procedure, the home keygen token cannot be obtained and the value of home nonce index will be unavailable. The basic Mobile IPv6 [3] also requires a binding update message that the nonce indices option cannot be expired. So, we define the home nonce index to be zero because this way is more compatible with the older specification and that makes the implementation simpler. Because the MN cannot obtain proof of entitlement to its home prefix, the verification of this binding also cannot be done for CN. In order to achieve our goal of keeping existing communications between MN and CN when the home network is unreachable, we need to alter the normal rules for CN to validate the special BU. Here, we make the CN accept this special BU if the CGA parameters option followed by the signature option exists and the value of home nonce index is zero. If the care-of testing is not yet complete, this special BU will be a special "Early Binding Update message" (EBU) and the MN should send this BU again to make CN accomplish checking its claimed care-of address. As is normal with ERO [17], if the CN received a binding as the special one we built, the status code 151 will be sent with the BA message to MN. This status code 151 is used to indicate to the MN to keep trying home testing to get a home keygen token.

The sequence of operations of the binding procedure are shown in figure 4.7. The broken line means the home agent is removed from the communication between the MN and CN. The MN moves to a foreign link, it still tries to send a BU message to HA and do HoT for CN. But these cannot be finished because home agent does not exist. So, MN tries to send a special "EBU" to the CN. After the CN verifies this special EBU message, it processes it and

adds the information of this special binding to its binding cache, and then returns a special Early Binding Acknowledgement (EBA) with status code 151 to the MN. The CN uses this special EBA message to inform the MN to keep trying the HoT. After this special EBA reaches the MN, it will send a BU to the HA and try the HoT again. If those fail, a repeated special BU will be sent to the CN later.

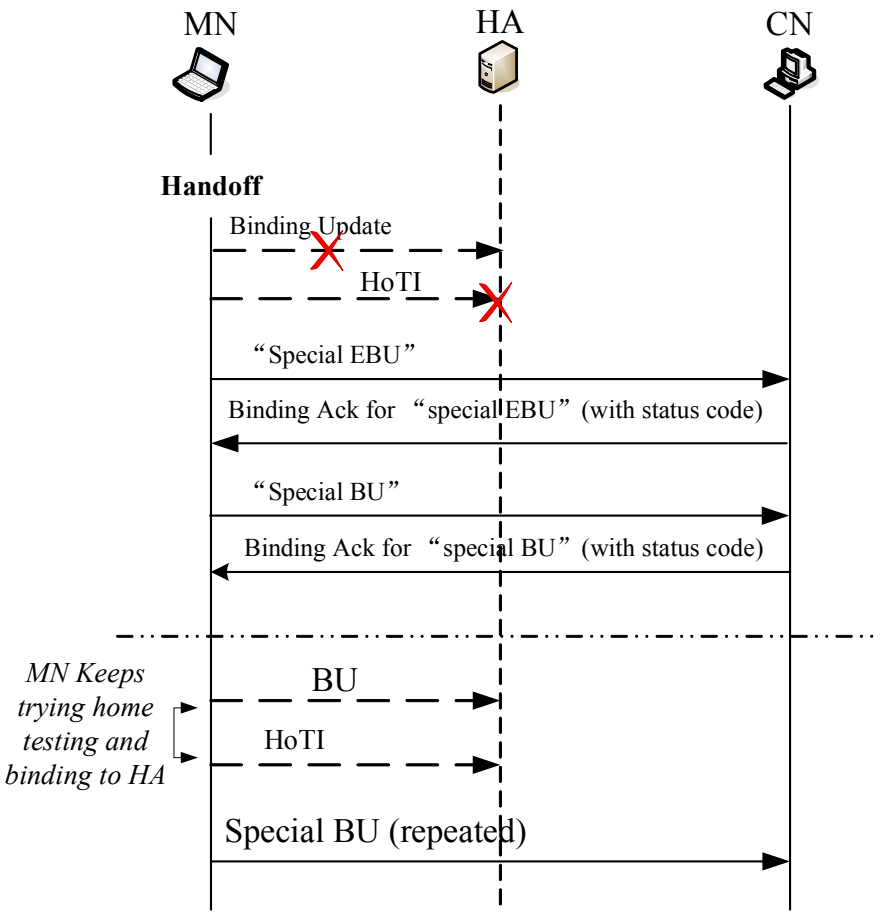

Figure 4.7 Binding Procedure of the Special BU

4.3 PU Flag and Its Operations

4.3.1 PU Flag Introduce

We have made the CN accept a special BU as described in the previous section. But the evidence of the prefix of this special BU sender's home address is not proved. In order to allow the CN to know whether the proof of MN's home address prefix is available or not, an extra Proved Usable bit, which we call the PU bit, is added in our design [18].

The new PU bit is initially reset in the binding cache at the correspondent node. When the CN obtains the proof of the prefix of the MN's claimed home address, the PU bit is set to indicate the authentication of MN's home address has been accomplished. Otherwise, the reset PU bit flags the mobile's claimed home address as unauthenticated.

4.3.2 PU Operation

We add the new PU bit to the CN's binding cache. The value of the PU bit is added along with the BU when that is accepted by the CN, with the binding information is added to the binding cache.

If the MN can prove the ownership of its home prefix, the PU bit is set to one. Otherwise, the value of PU bit is set to zero. The operational details of each case are illustrated as shown in figures 4.8 and 4.9. When the correspondent node receives an early binding, after verifying this special EBU, the CN finds that the sender's claimed home prefix is not authenticated and the HoT is not finished, then the CN sets PU to zero for this special EBU and acknowledges this message to tell the MN keep trying the home test procedure (using the status code 151) and do a complete special binding (using the care-of keygen token contained in the CoT option within the acknowledgement message). After the correspondent node processes this binding, packets can be transmitted between the correspondent node and the mobile node at its care-of address. If the network link is repaired, the home agent will be reachable and the mobile

node can validate itself through home testing. After the mobile node finishes authenticating its claimed home address, the correspondent node will update the PU value to be one for this mobile node. Then the future operation is normal.

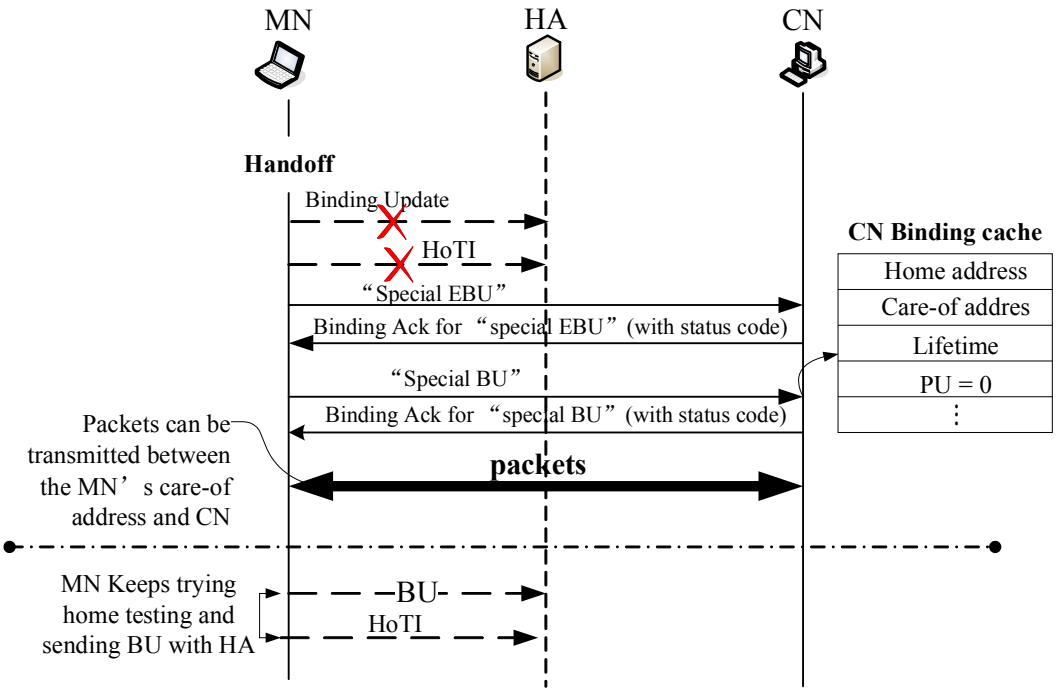

Figure 4.8 The PU operation when the MN's home address is unauthenticated

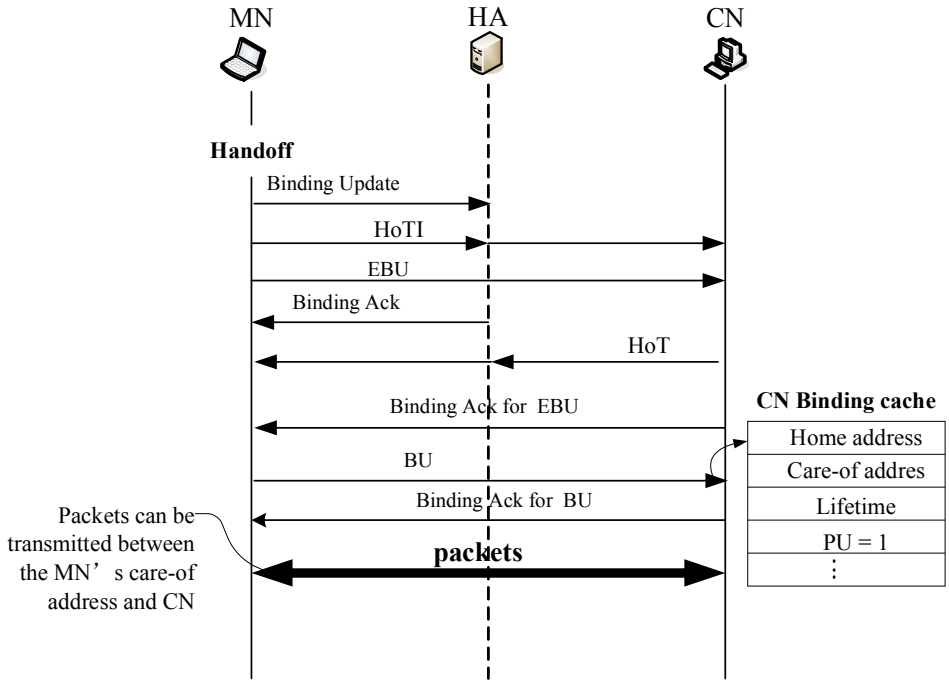

Figure 4.9 One case PU operation when the MN's home address is authenticated

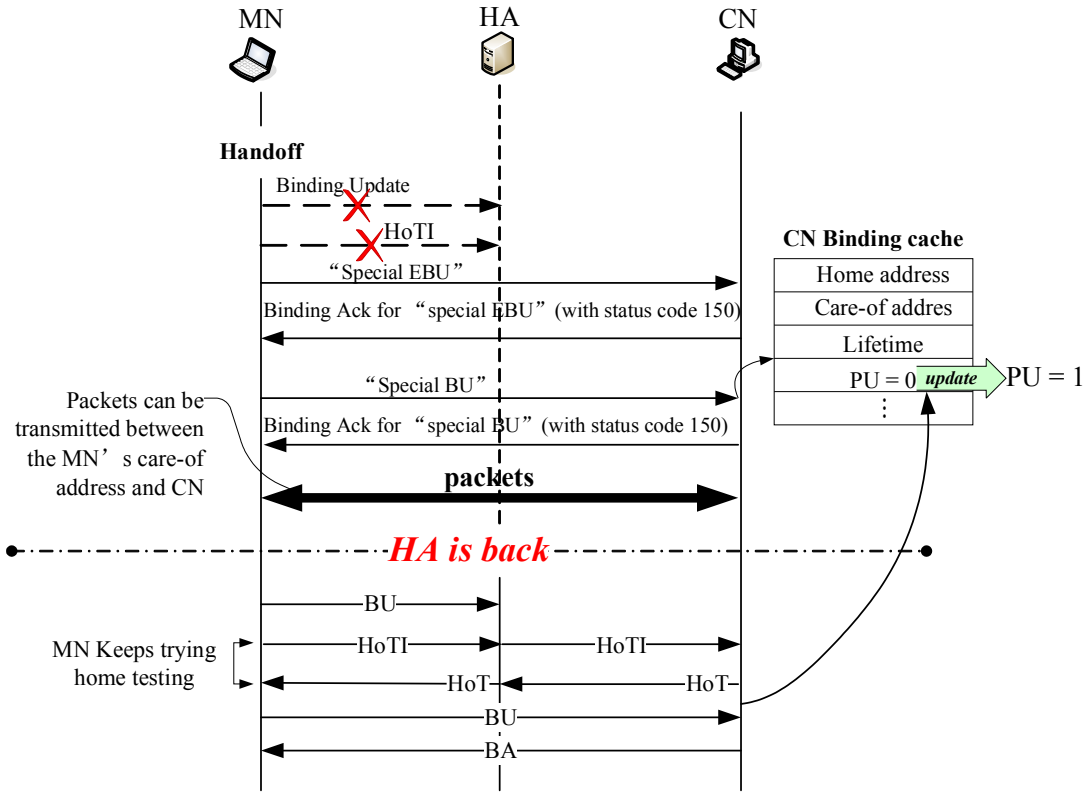

Figure 4.10 Case where value of PU is updated after HoT procedure is accomplished.

Figure 4.10 shows the case where a special BU is received by the CN first, and adds its information in its binding cache and sets PU to zero. After a while, the home agent recovers, the MN can finish HoT and binding with HA, and then resends a BU to the CN. After the CN processes this completely authenticated BU, the CN updates the information for this particular node in its binding cache and sets the PU value to one.

4.4 Avoid Reverting Packets Back Home

We permit the CN accept a BU although the proof of sender's entitlement to its home prefix has not been obtained. A malicious node might use this point to trigger a flooding attack as was illustrated in section 2.5.

In order to solve this security problem, also, an easier way is through operating this PU bit to manage the binding cache at correspondent side, refusing to delete a binding which PU shows "unauthenticated" status to indicate a binding where the MN's home address is not

proved. Because the CN doesn't know whether the sender of this binding is a good guy or a malicious node, it only knows whether this home prefix is authorized or not, as indicated by the state of the PU bit. Mobile IP requires the CN to check its binding cache before it directs packets. As long as a binding exists, packets will be directed to the nominated CoA instead of the apparent destination (the HoA). Thus, the CN finds in this particular binding that PU status is zero, it refuses this request of deleting this binding. Packets will not revert to the sender's claimed home address, but continue to use the current validated care-of address. That way, the "return-to-home" flooding attack cannot happen. If a mobile node wants to expire or delete a binding with CN, if the PU shows "unauthenticated", the CN must refuse to delete this binding. But it can be update the binding if the MN moves to a new location. Of course, there are other methods which also can avoid redirecting to applier's claimed home, such as make CN simply cease sending packets anywhere for this connection until it receives a later binding update. In order to avoid home flooding happening, CN cannot send packets back to MN's home, and the binding about this MN and its care-of address is deleted. A reasonable choice for the CN is to stop sending any more packets for this MN, until the updated binding information or a new binding arrives. In our implementation we adopted the former approach, to make the CN keep the binding in its binding cache and send traffic to the last validated care-of address. This is easier for our implementation and for testing our design.

4.5 Limitations of Design

Our solution is designed to cure one particular defect in Mobile IPv6, it is not a perfect solution. Some limitations:

First, this solution requires mobile nodes to be pro-active in establishing optimized routes, they cannot simply react to having received data relayed via the home agent as a signal that route optimization is required. This requires the Mobile IP implementation to be aware, or able to obtain knowledge of, all existing peer correspondent nodes.

Second, no incoming connections can be received from unpredicted nodes. In normal Mobile IP, the packets would be relayed by the home agent for the connection request

from an unknown node. When the mobile node receives that relayed request packet it can use the RO if it wants. Here, CN is known only by its HoA (from the DNS usually) and we don't have the service of the HA, so if an unknown node wants to initiate a connection, it cannot succeed and our method cannot assist in this case.

Third, our proposed solution requires both the mobile node and the correspondent node to understand this new protocol [17] and it could not work in other cases.

But our solution is better than nothing working at all.

4.6 Summary

In this chapter we have shown how by adding a new bit (the PU bit) to the binding cache we can allow route optimization to succeed even when no home agent can be accessed to assist with node address verification.

In the following chapter we will describe the prototype implementation built to test our hypothesis that the solution designed is both sufficient and correct, then the succeeding chapter will provide the results from testing using the prototype implementation.

CHAPTER 5

IMPLEMENTATION

In this chapter, we describe the prototype implementation built to test our design to allow communication between the mobile node and the correspondent node to continue while the home network link is broken. The key element of this solution is: to change the authentication rules of the correspondent node and manage the binding cache at correspondent node side. The mobile node must then send a Binding Update to the CN in the new form.

An overview of this implementation is presented in section 5.1. Section 5.2 describes the procedures for binding with the correspondent node and the procedure for processing Binding Acknowledgement messages. The PU operation is introduced in section 5.3 and the method to avoid reverting packets back home is given in section 5.4. Section 5.5 provides some details of limitations of the prototype implementation.

5.1 Overview of the implementation

To achieve the purpose of keeping communication between the mobile node and the correspondent node while the home agent is unreachable, one of the key requirements is to alter the binding rules for the correspondent node. When the correspondent node detects a mobile node triggering route optimization, the binding between the two must be made before data packets can be transmitted. Our implementation started with a working implementation of Enhanced Route Optimization [39], but the authentication methods of it need some changes. When a mobile node cannot provide the proof of entitlement to use its home address's prefix in a binding to the correspondent, we still make the CN accept this binding. In order to distinguish this partly unauthenticated kind of BU, a new "PU" bit is inserted in the binding cache at the correspondent node.

The functions required by the implementation are modifications to the existing SHISA mobility stack [27] on FreeBSD 5.4 [28]. SHISA is implemented on top of the KAME IPv6 stack [29] that contains all of IPv6 implementation including the SHISA project. The Mobile IPv6 implementation from KAME-SHISA does not support ERO, but Kuang Shilei has implemented and tested the ERO under the KAME-SHISA [39]. Our implementation is a continuation of that work, which has implemented the use of Cryptographically Generated Addresses in Mobile IPv6 to improve security and reduce handover delays, accomplish the functions of generating and verifying CGAs for Mobile IPv6 ERO, and using RSA [30] algorithms to sign and verify Binding Update and encrypt and decrypt the permanent home keygen token. In his implementation, the concurrent care-of address test is included but the Credit-Based Authorization is not yet supported. Here, we also simply assume there is no attempt is being made to attack the CoA in our tests, as further testing of basic ERO is not part of this thesis.

Table 5.1 SHISA programs

Program	Function
mnd	The mobile host functions
had	The HA functions (for MIPv6 and NEMO BS)
cnd	The CN functions and to provide route optimization of CN
babymdd	A simple movement detector of MN and MR
mrd	The MR functions
nemonetd	The tunnel setup functions for NEMO BS

SHISA consists of several user space programs and a modified kernel. Table 5.1 shows the programs of SHISA stack [31]. The programs of **mrd** and **nemonetd** (for NEMO network mobility [42] function) are not relevant to our work. The remaining programs: **mnd**, **had**, **cnd** and **babymdd** handle, respectively, the MN signaling messages, the HA signaling messages, the RO responder (correspondent) signaling messages and the movement detection procedure. The binding database that links the HoA and CoA of an MN is maintained by **mnd** and **mrd** on the

MN/MR side, and by **cnd** and **had** on the CN/HA side. In our implementation, the binding cache at the correspondent node is of concern. It is maintained by the **cnd**. The subset of the information of the databases that is necessary for the packet input and output processes in the kernel is injected by these programs using the mobility socket, which is a newly designed socket of the SHISA implementation, to provide the communication interface used between the kernel and the user space programs, and between the user space programs. It can also be used as a notification mechanism from the kernel to user space programs.

To get better performance, the SHISA stack separates operations into two layers: the packet processing of normal traffic should be done in kernel space, while the signal processing for Mobile IP operations should be done in user space. Our implementation requires work to be done in both spaces.

5.2 Binding Update with Correspondent Node

5.2.1 Trigger an Binding Update message

ERO [17] permits the temporary home keygen token to be prepared before the MN hands off. This can be used to provide the proof of the MN's home prefix ownership. That way saves a possibly long round trip through the home agent during the critical handoff period. This is primarily of benefit to mobile nodes that expect imminent movement, either signaled by link layer monitoring, or deduced from a pattern of past movement events. For this project, we assume there is no temporary home keygen token for preparation, which means no proactive home test has been done before the mobile node moved.

In the SHISA stack [27], the **babymdd** daemon is used to provide a simple movement detection function. After the mobile node moves to foreign link, **babymdd** sends a message to the **mnd** daemon. Then **mnd** could obtain a list of all existing conncetions from the kernel, and commence EBU on each different correspondent node included in that list. In a normal IPv6 mobile node, RO can be triggered when the mobile node receives a tunneled packet. In this project, the home agent does not exist so it can not function for tunneling packets between

the mobile node and correspondent node, the mobile node cannot receive hint information to start RO. For our solution, we want the mobile node to send an Early Binding Update message directly to the correspondent node after it moves to the foreign link. This should involve determining the CN address(es) from the kernel's list of connections, triggered by the movement event. For simplicity our implementation knows the CN address, and we give external advice which uses a UNIX interrupt signal and operates on the **mnd** daemon to start sending an EBU or special EBU.

If the MN's home address is unauthenticated, our new special Early Binding Update message will be used. In this special early binding message, the Care-of Test Init option is included, which is used for concurrent care-of testing. The Care-of keygen token will be returned within the Care-of Test option contained in the EBA or special EBA message. Through exchanging the EBU/EBA or special EBU/special EBA pairs, the concurrent CoT procedures can be accomplished.

We have given the design of the essential parts of the special BU/EBU message in the previous chapter. The architecture of an EBU message is as figure 5.1 shows.

Mobility Header	Other Mobility Option	CGA Parameters Option	Signature Option	Care-of Test Init Option	Nonce Indices Option	Binding Authorization Data

Figure 5.1 Architecture of EBU for CN

A mobile node sends a Binding Update message with a Binding Authorization Data option, which includes the Authenticator value computed by the procedure described in the previous chapter, and a Nonce Index option that contains the home nonce index and the care-of nonce index, which have been used when generating a shared secret to compute the authenticator.

In our designed special binding update message, the home nonce index value is set to be 0 indicating that no Home Test has yet been performed, and that consequently no home keygen token is available. Also, when the mobile node builds this binding, it adds the authorization data into the authenticator in the Binding Authorization Data option for CN to use later in verifying this binding, in a different way than the normal case. We have described the authenticator calculation method in section 2.3.4, the home keygen token value is used in this calculation algorithm as follow shows:

Authenticator = First (96, HMAC_SHA1 (kbm, (care-of address | CN address | BU)))

While, the shared secret key *kbm*, is calculated as:

$$Kbm = SHA1\ (home\ keygen\ token \mid care\text{-}of\ keygen\ token)$$

(if MN is at a foreign network)

Or

$$Kbm = SHA1\ (home\ keygen\ token)$$ (if MN is at home)

From the above calculating functions, we can see that the home keygen token is an important parameter that used for verifying the HoA of mobile node. In our environment, there is no verification of the home address, and if the care-of test is deferred, no care-of token either, so for our special EBU we use:

$$Kbm = 0$$

(if MN is at foreign network while the home network is unapproachable, and CoT is incomplete)

If the MN has obtained a care-of keygen token, it can send a special BU instead the above special EBU. That way, the Kbm is caculated as:

$$Kbm = SHA1\ (care\text{-}of\ keygen\ token)$$

(if MN is at foreign network while the home network is unreachable, and has done the CoT)

The case where $Kbm = 0$ gives no authenticator of course, there is nothing to authenticate, and an alternative would have been to omit the authenticator. We chose to retain it to retain more of the normal packet processing, and because after the first iteration, the care-of keygen token will be available, and the authenticator needed to complete the CoT.

The *send_bu ()* function is called when the mobile node sends a binding update message to the correspondent node. The flowchart in figure 5.2 shows the important parts of the algorithm for building a binding sent for each of the variant authentication requirements, which includes the special case that the home keygen token is not obtained. For the special case, we send the designed early special binding with a home nonce index that is filled with 0 and the shared secret kbm is created as 0 (the case of the left second column in figure 5.2) or a complete special binding which kbm is calculate using care-of keygen token only if MN has finished the CoT procedure.

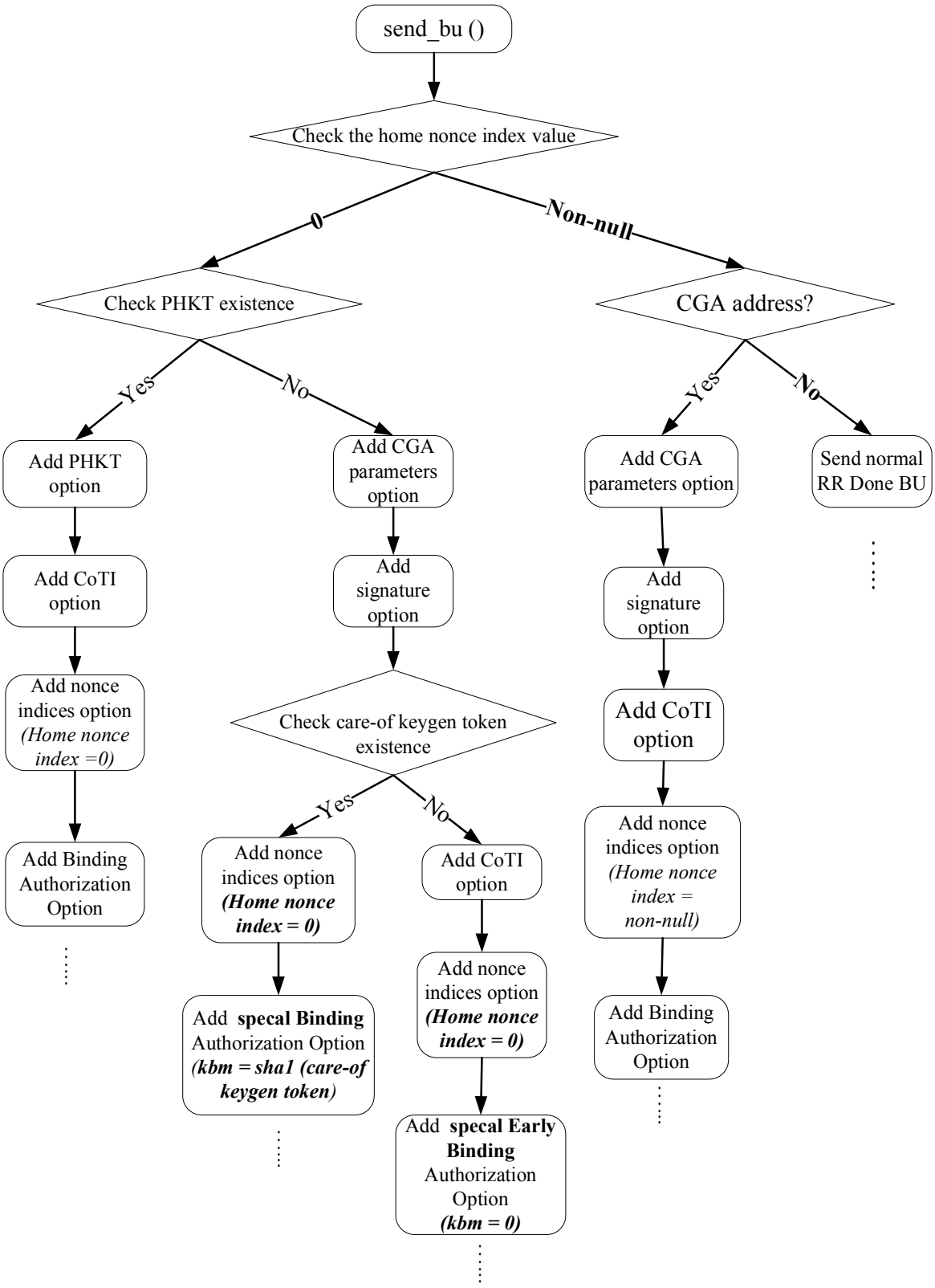

Figure 5.2 Send Binding Update for Correspondent Node

5.2.2 Binding Validation for Correspondent Node

BU validation processing will be done by the **cnd** daemon. When a BU message is received by a CN, it will be cryptographically protected by one of the ERO procedures depending upon the authentication method that the sender has chosen. The *dest6_mip6_hao( )* function checks the next header to be processed, and if the next header is a Mobility Header and the message type is BU, the function swaps the HoA in the HAO option and the source address of the IPv6 header. The BU message will be validated in the processing routine that calls the *receive_bu( )* function. If the BU message has been sent as a result of one the proper ERO procedures, the message will pass the cryptographic verification, otherwise it is dropped.

Following the BU processing rules of ERO[17], when a CN receives a BU message, it must first verify the sending mobile node is the legitimate owner of the HoA specified in the message. The authentication method chosen based on the home nonce index contained in the Nonce Indices option within the BU message, and the existence of CGA Parameters and Signature options in this BU. In order to allow our special binding to pass the verification by the correspondent node, we need to modify the authentication methods of the CN. A BU that is validated still also needs to satisfy the requirements of the basic Mobile IPv6 [3]. The overall process of the verification procedure is as shown in figure 5.3. When the correspondent node receives a binding, it should check the existence of the authorization and nonce indices options. If they exist, it will continue verifying this binding. In this processing procedure, if the nonce index has expired, the CN cannot accept this binding and it should return a BA message to inform the MN to obtain a fresh nonce.

In this procedure, the new designed bit “PU” is also added when the correspondent node processes the binding. When the correspondent node compares the authenticator with the value calculated from the information of home nonce index and care of index in the binding, for the special case, the shared secret kbm is created using the care-of keygen token only if the ownership of mobile node’s home address cannot be proved but a care-of keygen token has been obtained, or setting to zero if it is an early special binding. After the binding is validated, the function *mip6_bc_add()* will be called, which causes the binding information to be added to the binding cache in the correspondent node.

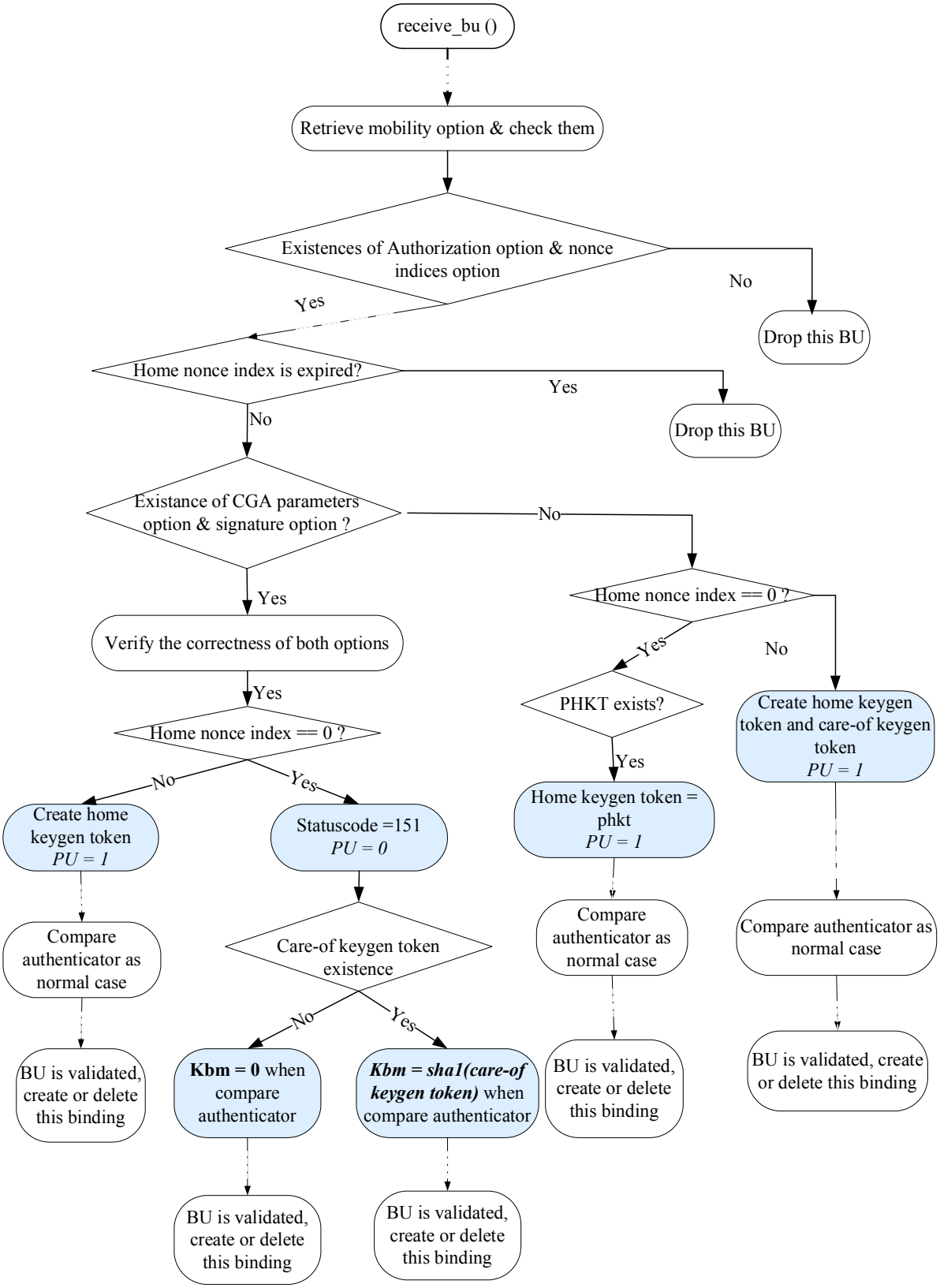

Figure 5.3 Correspondent validates BU procedures

5.2.3 Binding Acknowledgement for Mobile Node

If the binding can be validated under ERO [17] using one of the three defined authentication methods, the binding will be fully authenticated. That way, the status of this binding is accepted. If the binding was sent as our designed special case, we also make the correspondent node accept this binding through making a small change in the verifying binding procedures, but a Binding Acknowledgement (BA) message also needs to be sent to the mobile node with status code 151 to provide notification to the MN that its binding has been accepted, but that it should keep trying the Home Test.

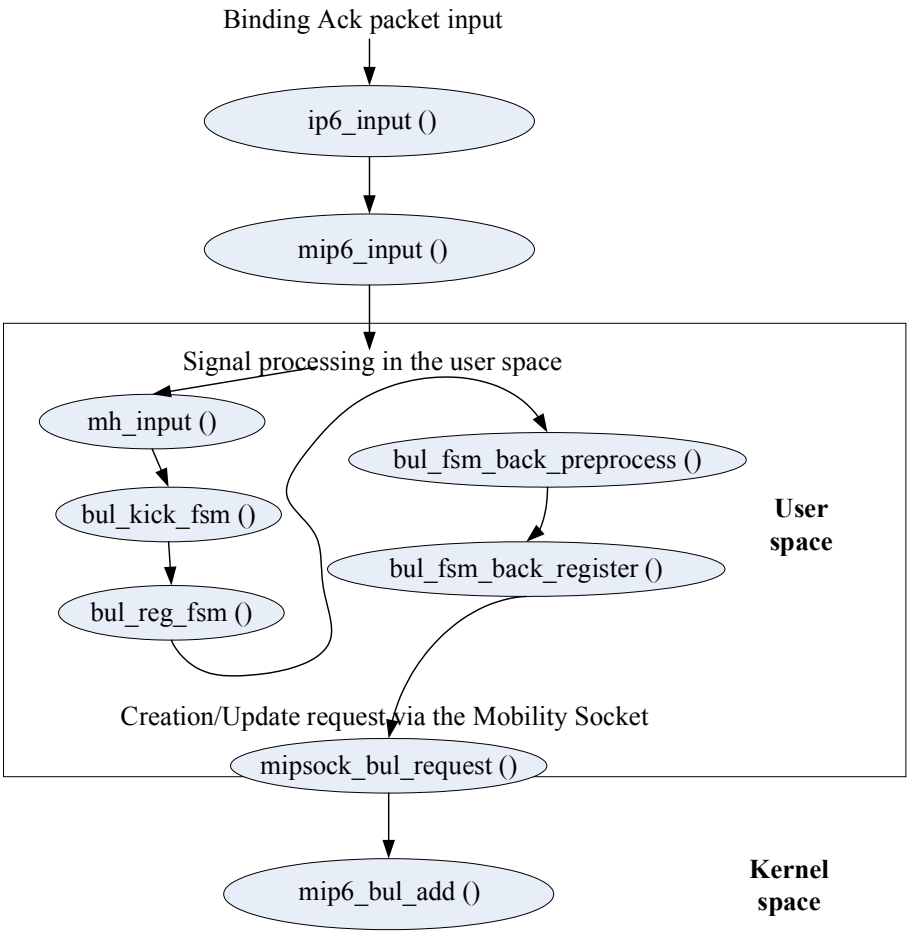

Figure 5.4 Binding Ack packet processing

In the SHISA stack, the BA message is received by the general signal processing function *mip6_input()* and passed to the IPv6 raw socket mechanism to deliver the message to the user space. Once the packet is processed, the SHISA user space program updates the related binding information using the Mobility Socket mechanism. Figure 5.4 shows the processing of a BA packet. In original design of ERO [17], if the sender of a binding cannot provide the proof of its home prefix ownership, the CN must refuse this binding. The MN also will drop the binding and go to try HoT. In our solution, in order to keep the communication between the mobile node and correspondent node when the home agent cannot be reached, we make the CN receive our new special BU, so the binding information also cannot be dropped by the mobile node. The function *bul_fsm_back_preprocess()* is called for the MN to process BA messages. After the mobile node has processed the BA message, the information of this binding will be passed to the kernel space via the Mobility Socket. If there is no binding update list existing for this binding information, the mobile node will add this binding update list in its binding update database. The procedure of creating a binding update list for a mobile node is as figure 5.5 shows.

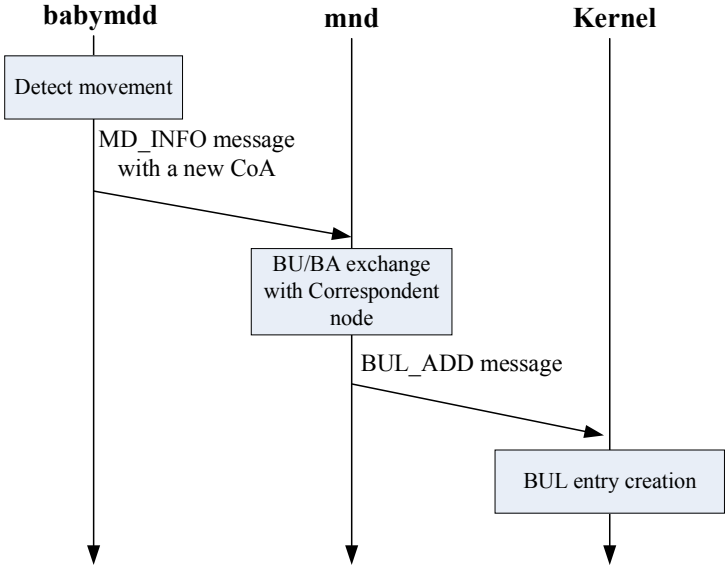

Figure 5.5 Creating a BUL for MN

5.3 PU Operation

In section 5.2.2, we have mentioned a new designed bit "PU" that is added with other information together in the binding cache at the correspondent node side when a binding is accepted by the CN. It is one bit, so two values is all that is possible. This bit indicates whether the ownership of binding sender's home prefix is authenticated or not. The possible values for three important variants of a binding are shown in the table 5.2 and the values of "PU" bit setting for each different binding is shown in the table 5.3.

Table 5.2 All different status of CGAs, home nonces and care-of nonces

Options	**Values**		
CGAs	**yes**		**no**
Home nonces	Valid	Expired	0
Care-of nonces	Valid	Expired	0

In table 5.2, for the home nonce and care-of nonce, the values could be one of "Valid", "Expired" and "0" status. The "0" indicates the HoT or CoT is not finished. We use "yes" and "no" to indicate whether the node's home address is a CGA or not.

Table 5.3 PU values setting for all different bindings

	CGAS	**Home Nonce**	**Care-of nonce**	**Status of BU**	**Values of PU**
1	Yes	0	Valid /0	accepted	0
2	Yes	0	Expired	Not accepted	
3	Yes	Valid	Valid /0	accepted	1
4	Yes/No	Expired	Valid / Expired /0	Not accepted	
5	No	Valid	Valid	accepted	1
6	No	PHKT	Valid /0	accepted	1

In table 5.3, cases 3, 5 and 6 correspond to the ERO [17] defined requirements for binding authentication cases, the "PU" is set to 1 as the mobile node can provide the proof about its home prefix ownership. Case 1 is the our designed special BU/EBU (if the care-of nonce index is 0, it will be an EBU), using "0" for "PU" bit to indicate the sender's home address of this binding is not validated. Case 2 is different from our designed special BU as it has a care-of nonce index that is expired. It cannot satisfy the basic Mobile IPv6 [3] defined requirements, so it will be refused by CN. The reason of dropping case 4 is the same as case 2, the home nonce index also cannot be expired.

5.4 Avoid Reverting Packets Back Home

If a mobile node wants to delete a binding with a correspondent node, a binding update message is required where the binding lifetime is 0 or the care-of address is set to the home address. After the correspondent node receives a binding like this, it will see if a particular binding for this mobile node exists in its binding cache first. Also the *receive_bu()* function will be called to process this binding. After this binding is successfully processed, the *mip6_bc_delete()* function is called. *mip6_bc_lookup()* function is used to find this binding of this mobile node. If this binding exists and its binding state is "VALID", then it will pass this requirement that mobile node moves back home to kernel through mobility socket, *mipsock_bc_request(bc, MIPM_BC_REMOVE)*. In previous section, we mentioned a new flag "PU" is designed to indicate whether the mobile node's home address is authenticated or not. After this request to delete the binding is passed to the kernel, the correspondent node checks this binding entry information in its binding cache. If the PU bit is equal to 0, the correspondent will retain this binding entry in its binding cache. That way, later packets will continue to be sent to the current proved care-of address of this MN rather than to the unverified home address.

5.5 Implementation Limitations

Our implementation does not seek to learn correspondent node address for the mobile node to use to send binding updates. Rather for our prototype we simply “know” the appropriate CN address to use.

Similarly our implementation does not send the special Binding Update upon detecting that the mobile node has moved, it must be instructed by an explicit signal from another process to commence the new procedures.

The correspondent node correctly avoids transmitting packets to an unverified home address by retaining the last verified binding cache information, but in our implementation that information is never discarded, unless the mobile node verifies its home address. This could lead to a denial of service attack by overflowing the capacity of the CN’s binding cache with old bindings of unverified home addresses that are not removed even though the CN has no packets to send to that address, and which thus could not be the subject of a flooding attack.

All of these limitations could be corrected by additional programming, but none detract from our purpose of showing that the design in the previous chapter achieves its aim, so that additional work has been deferred to some later production implementation.

5.6 Summary

This chapter has given details of the implementation undertaken to demonstrate the effectiveness of the solution designed in the previous chapter.

The next chapter will explain the tests undertaken and results obtained using the implementation from this chapter.

CHAPTER 6

TESTING

This chapter describes the testing of our prototype implementation of the solution design. Using a simple test network we test the ability of a mobile node to move and continue communicating without the home agent assistance, and verify that this does not permit flooding attacks of the claimed home network.

The equipment and software along with the network topologies designed and deployed are presented in section 6.1. The testing scenario and results are described in section 6.2, with a summary of results in section 6.3.

6.1 Testbed Deployment

6.1.1 Equipment and Software

For the requirements of this project, the experimental testbed is the minimum possible Mobile IPv6 network. This testbed consists of three machines which play roles of two IPv6 nodes as the mobile node and a correspondent node respectively and an IPv6 router that also servers as home agent. All of them run the FreeBSD operating system. We chose the most recent (final) version of Mobile IPv6 from the KAME project, which is kame-20061106-FreeBSD. The details of equipment and software are shown in table 6.1.

SHISA consists of several user space programs and the modified kernel functions [31]. Based on the node type, one or several SHISA programs run on a node. In addition, a user can choose to drop or replace functions by stopping or changing programs. In other words, the modified function can run on the same node type. The table 6.2 gives the category of the node type and the SHISA programs each node type runs.

Table 6.1 Equipment and Software

Testbed Compodents		Software
Mobile Node	CPU: Pentium 4, 2.4G HZ 1G RAM 80G hard disc	Freebsd 5.4-RELEASE and kame-20061106-freebsd54
Correspondent Node	CPU: Pentium 4 1.8G HZ 512MB RAM 200G hard disc	Freebsd 5.4-RELEASE and kame-20061106-freebsd54
IPv6 Router	CPU: Pentium 2 400M HZ 256 MB RAM 40G hard disc	Freebsd 5.4-RELEASE and kame-20061106-freebsd54

Table 6.2 SHISA Programs categorized by the node type

Node Type	Mobile IP daemons
Mobile Node	mnd babymdd (cnd)
Correspondent Node	cnd
IPv6 Router/ Home Agent	had

The Home Agent Daemon **had** is used to configure the kernel to function as a home agent. It determines the home link by looking at the address of the interface it was called with and passes this information to the kernel. By connecting to port **7778** on localhost, it is possible to view statistics and the binding cache as well as to clear the cache.

The Mobile Node Daemon **mnd** is used to configure the kernel to function as a mobile node. It determines the Home Address of the MN by looking at the home address assigned to the interface it was called with. This information is passed to the kernel. By connecting to port **7778** on localhost, it is possible to view the binding cache, the list of home agents and a list of hosts for which no route optimization should be done. It is also possible to view statistics, as well as to clear the lists and cache.

The correspondent Node Daemon **cnd** is used to configure the kernel to function as a correspondent node supporting route optimization. By connecting to port **7777** on localhost, it is possible to view statistics and to clear the binding cache of the CN.

The baby Movement Detection Daemon **babymdd** runs on a mobile node. It is responsible for determining the current network location of the node, as well as notifying the kernel about a change of location. This is done by polling the status of the given interface at regular intervals, as well as by watching a routing socket for changes in routing information. [40]

6.1.2 Network Design

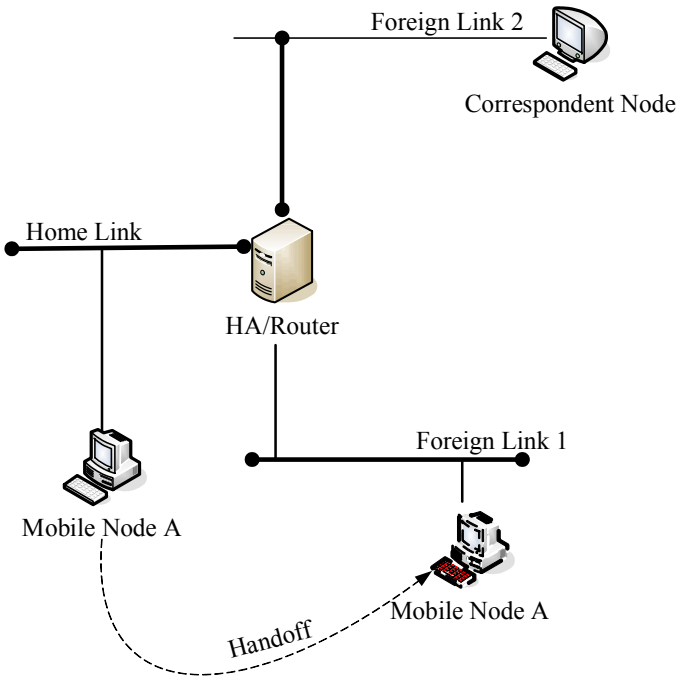

Figure 6.1 Experiment Testbed Architecture

The network design must satisfy our research and future work requirements. We have mentioned this testbed is a minimal Mobile IPv6 network. Figure 6.1 shows the testbed architecture. This testbed consists of three machines, the Mobile Node, Correspondent Node and IPv6 Router respectively. The Mobile Node needs to move to a foreign link, so this architecture must provide two or more different networks at least. We use the PC-based IPv6 router instead of the commercial IPv6 router to allow the latest IPv6 features to be easily deployed.

The experimental network just requires three computers, two additional Ethernet cards and several cables. We merge the home agent with IPv6 Router function on one machine. The home agent is required for initially assigning the home network link to the mobile node in our testing. We use two network cards to simulate foreign link 1 and foreign link 2. The former one is the network that mobile node will move to (care-of address) and the later one is assigned as correspondent network.

6.1.3 Network Configuration

The Testbed network is configured to use IPv6 exclusively. The configuration of our testbed is as follows:

1. The Home Link is assigned the network prefix 2008:ffff:ffff:ffff::/64.
2. The Foreign Link 1 is assigned the network prefix 2008:ffff:ffff:face::/64.
3. The Foreign Link 2 is assigned the network prefix 2008:ffff:ffff:feed::/64.
4. The Mobile Node's home address is 2008:ffff:ffff:ffff:34a2:7904:ef45:1b0e.
5. The Home Agent's home interface address is 2008:ffff:ffff:ffff::1.
6. The Correspondent Node's address is 2008:ffff:ffff:feed:204:5aff:fe4a:fb7c.

6.1.4 Host Configuration (Mobile Node, Correspondent Node, Home Agent/Router)

1. Mobile Node

The configuration of the mobile node is different from other nodes. When the mobile node is at home, it can use its home address to receive packets from others. When the mobile node moves to a foreign link, it will get a care-of address and the packets will be delivered to the care-of address then transported to the home address. So we configure mip0 and r10 as two interfaces to the mobile node. The interface mip0 is configured as a pseudo interface and it holds the MN's home address when the mobile node has moved to foreign link. Figure 6.2 gives the description. When the mobile node remains at home, the packets will be directed to the mobile

node's physical address interface rl0 directly, where the home address will be assigned. When the mobile node moves to foreign link, the packets will be sent to the physical address interface rl0 first, here the current care-of address of mobile node remains. Then the packets delivered to the virtual interface mip0, where the home address of mobile node is held.

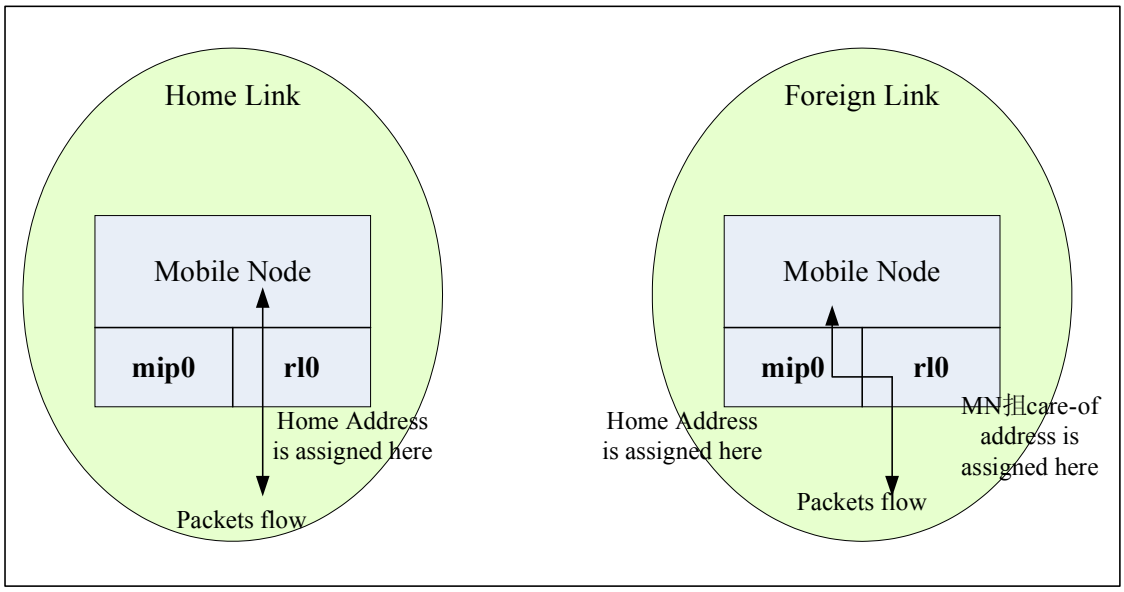

Figure 6.2 Interfaces Assigned on Mobile Node

We configure the mobile node using the /etc/rc.conf file, containing the information shown below:

ipv6_enable = "YES"

ipv6_gateway_enable = "NO"

ipv6_network_interfaces = "r10 mip0" *(mip0 is the Pseudo interface)*

ipv6_ifconfigure_r10 = "up" *(r10 is the Ethernet card interface)*

ipv6_ifconfiture_mip0 = "2008:ffff:ffff:ffff:34a2:7904:ef45:1b0e **home**"

ipv6_mobile_enable = "YES"

ipv6_mobile_nodetype = "mobile_node"

ipv6_mobile_security_enable = "NO" *(Configure IPSec)*

2. Correspondent Node

In our experimental network, only one correspondent node is communicating with the mobile node. For our purposes, Route Optimization must be supported by this correspondent node.

The correspondent node is configured using the following extract from /etc/rc.conf:

ipv6_enable = “YES”

ipv6_gateway_enable = “NO”

ipv6_network_interfaces = “dc0”

ipv6_ifconfig_dc0 = “2008:ffff:ffff:feed:204:5aff:fe4a:fb7c”

ipv6_mobile_enable = “YES”

ipv6_mobile_nodetype = “correspondent_node”

3. Home Agent/Router

We configure another machine as a static router for this testbed. There are three Ethernet cards in it. The interfaces x10, r10, pcn0 are configured to the foreign link 1, foreign link 2 and home link respectively. In order to assign the mobile node home link prefix, the home function exists in this configuration initially. After the mobile node has its home address, we remove the had daemon using the “kill” command. We disable the IPv6 mobile security function here, because home agent will not actually be used, configuring security for it is just wasted effort.

The details of Home Agent/Router configuration in /etc/rc.conf file are as follows:

ipv6_enable = “YES”

ipv6_gateway_enable = “YES”

ipv6_router_enable = “YES”

ipv6_router = “/usr/local/v6/sbin/route6d”

rtadvd_enable = “YES”

rtadvd_interfaces = “x10 r10”

ipv6_network_interfaces = “pcn0 x10 r10”

ipv6_ifconfig_pcn0 = “2008:ffff:ffff:ffff::1 prefixlen 64” *(home link interface)*

ipv6_ifconfig_x10 = “2008:ffff:ffff:face::1 prefixlen 64” *(foreign link 1 interface)*

ipv6_ifconfig_r10 = “2008:ffff:ffff:feed::1 prefixlen 64” *(foreign link 2 interface)*

ipv6_mobile_enable = “YES”

ipv6_mobile_nodetype = “home_agent”

(home agent function exists initially, we remove it later using “kill” command)

ipv6_mobile_home_interface = “pcn0”

ipv6_mobile_security_enable = “NO”

6.2 Testing and Result

6.2.1 Testing Scenario

This section presents our testing scenario, which simulates a broken home network but still allows the mobile node to continue communicating with correspondent node. We begin by having the mobile node (A) establish a connection to the correspondent node from its home link, as figure 6.3 shows. After the mobile node has its home address, we remove the home agent. Then move A to foreign link 1, and verify that it can keep communicating with the correspondent node, as shown in figure 6.4.

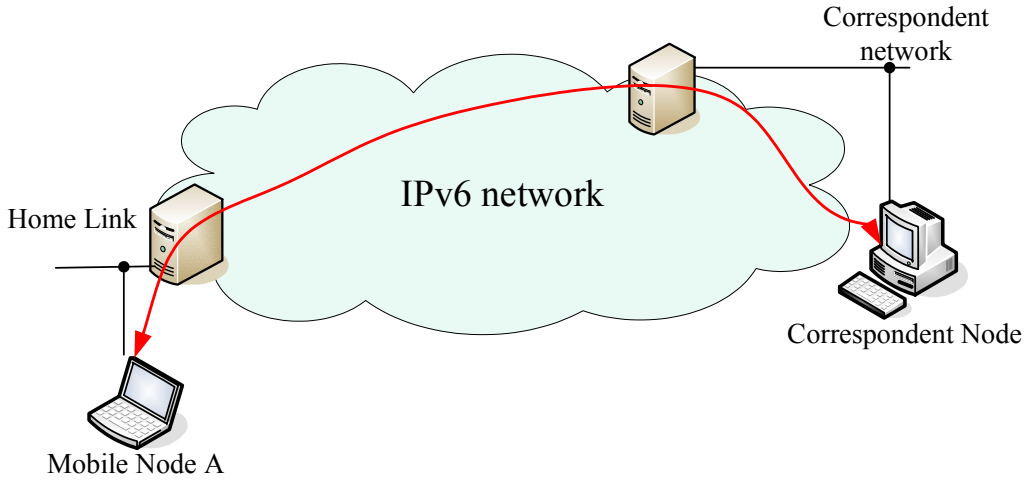

Figure 6.3 Mobile Node communicate with correspondent node at home first

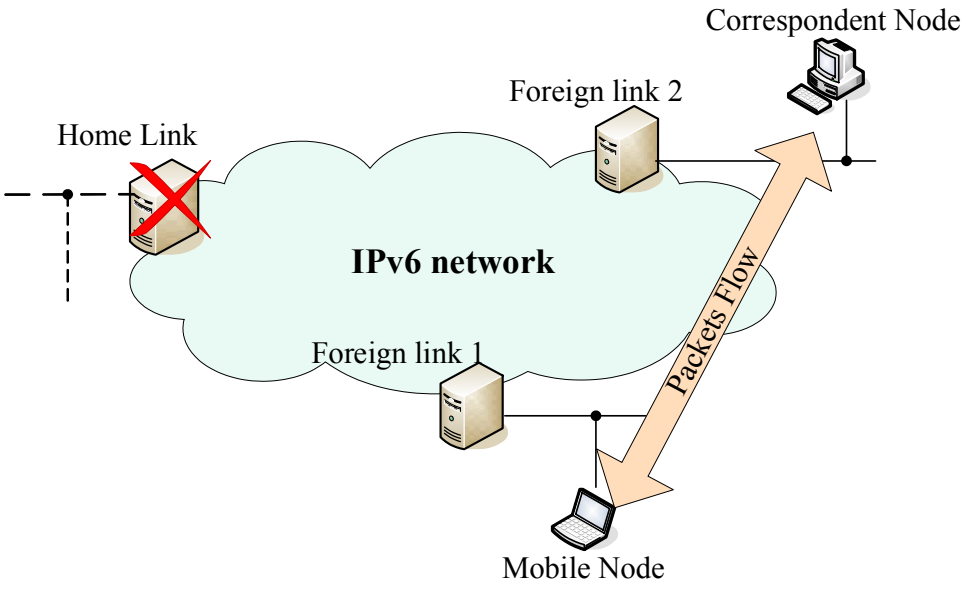

Figure 6.4 Mobile Node communicates with Correspondent Node without Home Agent at foreign link

6.2.2 Testing Result of Scenario

In the testing, we use the SSH program [37] on the mobile node to make a TCP connection with the correspondent node and use the **Wireshark** program [36] or **tcpdump** [38] to monitor the testing process at correspondent node side. The Mobile node's home address must be a CGA, which is manually configured in the */etc/rc.conf* file.

The packet flow of our testing scenario is shown in figure 6.5. We use **SSH** to cause the mobile node to establish a connection with the correspondent node from its home link. Then we use the **kill** command to remove the **had** daemon from our experiment to simulate the case of no home agent, and move the mobile node to the foreign link. We use the UNIX interrupt signal to generate a trigger event code to cause the mobile node to send the new Special Early Binding Update message to the correspondent node directly from its current care-of address. After the correspondent node receives this special EBU, it will process, validate and add this binding to its binding cache. Then it replies with a special EBA with the status code 151 to inform the mobile node that the binding was accepted but to keep trying the HoT procedure.

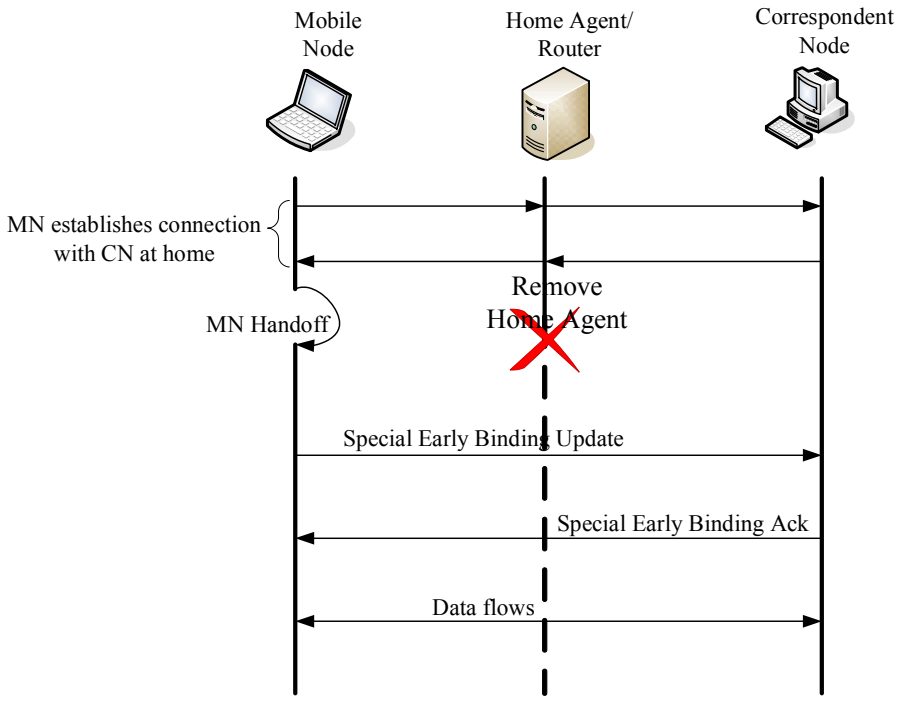

Figure 6.5 Packets flow of testing scenario

Now we will see the operation details of this testing procedure:

1. Remove **had** daemon from the Home Agent/IPv6 Router machine using *kill* command.

2. Establish connection between the mobile node and correspondent node via SSH when the mobile node stays at home.

3. Move mobile node to the foreign link (its care-of link).

4. Send a special Early Binding Update message from mobile node to the correspondent node.

Figures 6.6 to 6.12 are the results of this testing. Packets in these figures show the details of their being transmitted before and after the home agent is removed. In order to observe packets detail information more easily, we choose the wireshark to capture packets flows on correspondent node side. At same time, we run **tcpdump** on Home Agent/IPv6 Router machine to monitor the three interfaces which are pcn0 (interface of MN's home address), xl0 (interface of MN's care-of link) and rl0 (interface of correspondent link). We also can check the nodes status through system log (**syslog** [28]) files contained by every node, which records the Mobile IPv6 daemon information and status.

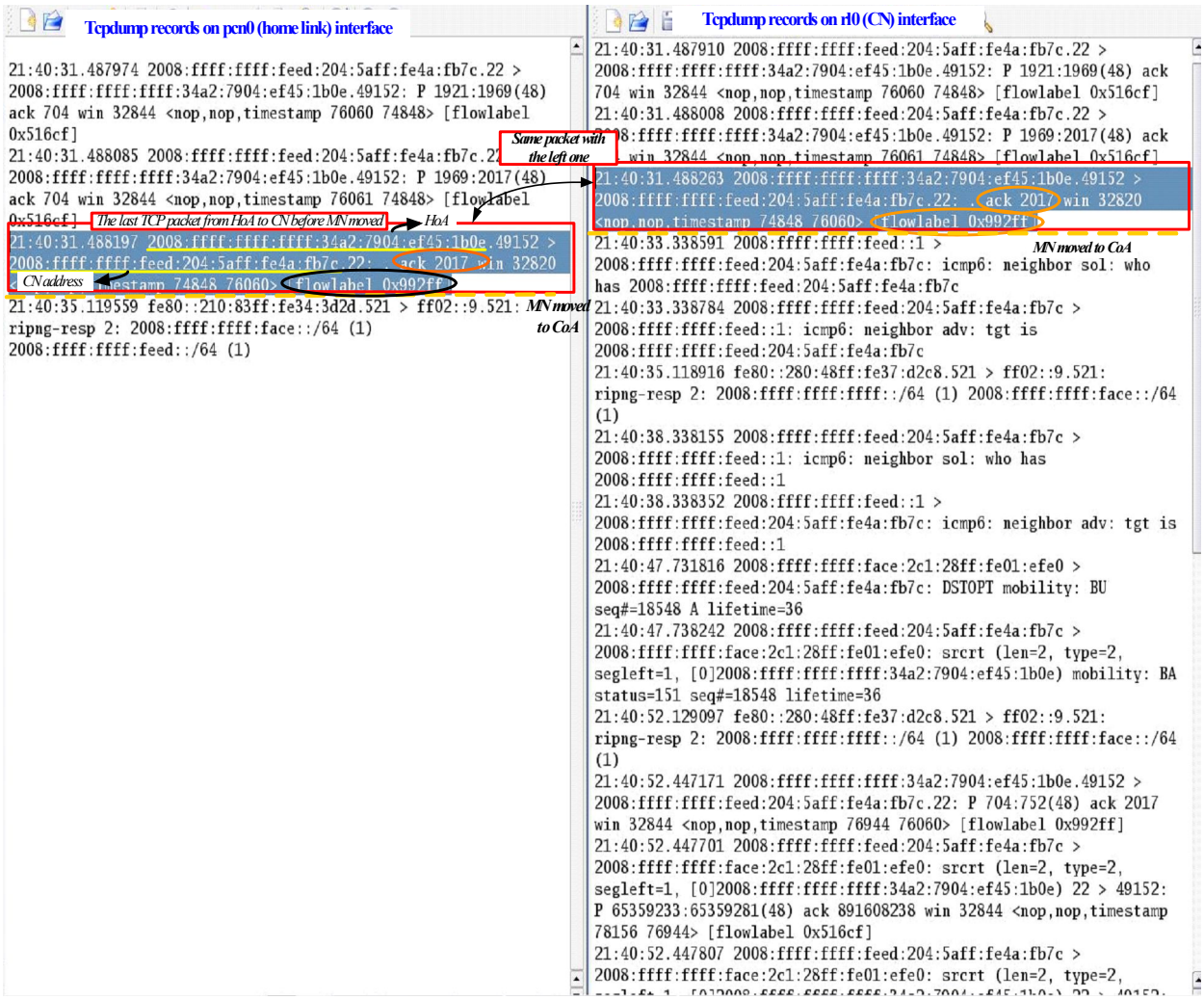

Figure 6.6 Last TCP packet from MN's HoA to CN before MN moved to CoA

Figure 6.6 shows the **tcpdump** records of both the pcn0 and rl0 interfaces. The information of interest has been highlighted in this figure. The marked packet is the last TCP packet from MN's home address to the correspondent address before the mobile node moved to the CoA. The left window in this figure is the records on pcn0 (MN's HoA) interface and the right window records testing results on rl0 (CN) interface. We can see that there are no TCP reply packets from CN to MN's HoA after the MN sent its last TCP packet.

After this packet was sent, we moved mobile node to the foreign link (its care-of link). Figure 6.7 shows results of packets transmitted after the mobile node moved. The left window in this figure is the **tcpdump** results on xl0 (MN's CoA) interface and the right window is recorded on rl0 (CN) interface. In the information highlighted in the left window, we can see that

MN attempts home registration after it moved to the foreign link, but no binding acknowledgement packets are returned from the HA. Then we cause the MN to send a BU to the CN, this is the special EBU. After the CN verified and accepted this special EBU, it returns a special EBA to MN, with status code 151, used for notifying the MN to retry the home test. We can see details of both messages in figures 6.8 and 6.9.

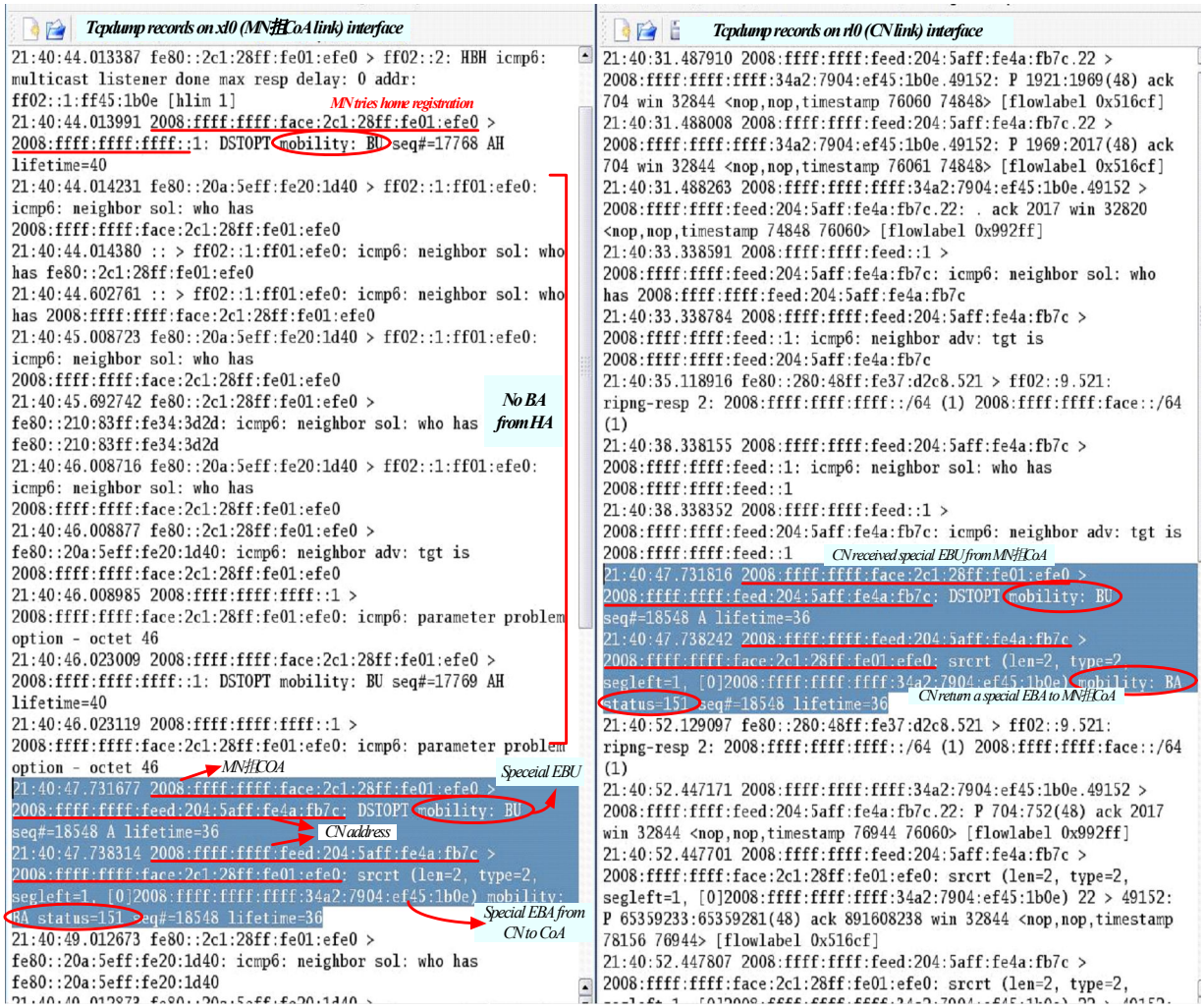

Figure 6.7 Packet transmitted after mobile node moved to foreign link

Figure 6.8 shows the details of our new special EBU that is sent from the current care-of address of Mobile node to the address of Correspondent node directly. We can see that the home nonce index contained in this message is the intended 0, and it includes the CGA option (0x0c), signature option (0x0d) and CoTI option (0x0f) from ERO. The important information we need to note has been marked by underlining.

Note that wireshark has not yet been taught the new mobility option codes for ERO [17] and so displays them as “unknown option”. The text in the figure in italics has been added to assist the reader.

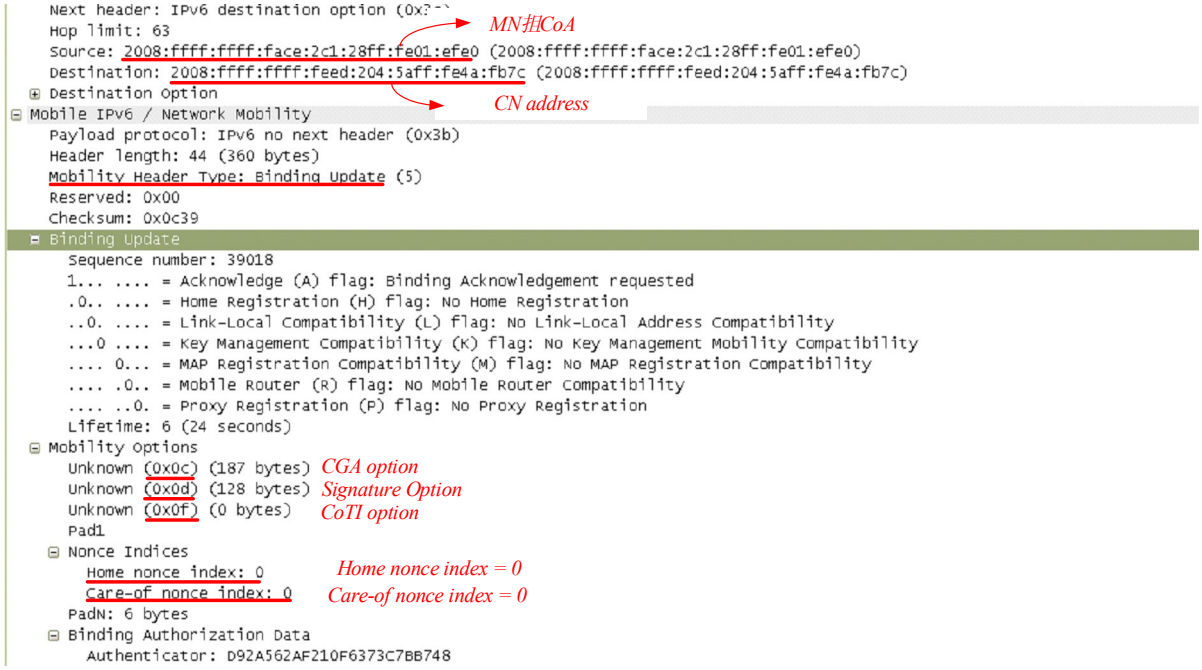

Figure 6.8 Detailed information of Special Early Binding Update message

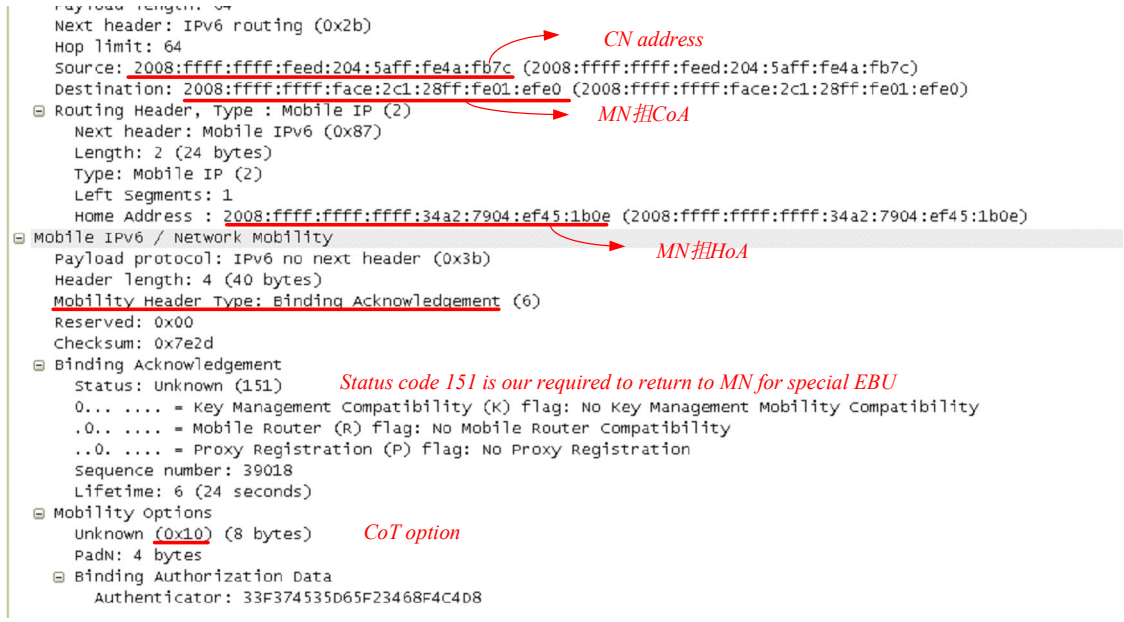

Figure 6.9 Detailed of Early Binding Acknowledgement message

Figure 6.9 shows the special Early Binding Acknowledgement (EBA) message which is sent from the correspondent node to the mobile node at its care-of address. The important information has also been highlighted in this figure. The status code 151 is returned in this special EBA message to inform the mobile node keep trying HoT. The CoT option (0x10) corresponds to the CoTI option in the special EBU message. After the CoT option is returned to the mobile node, the concurrent care-of address test designed as part of ERO is finished.

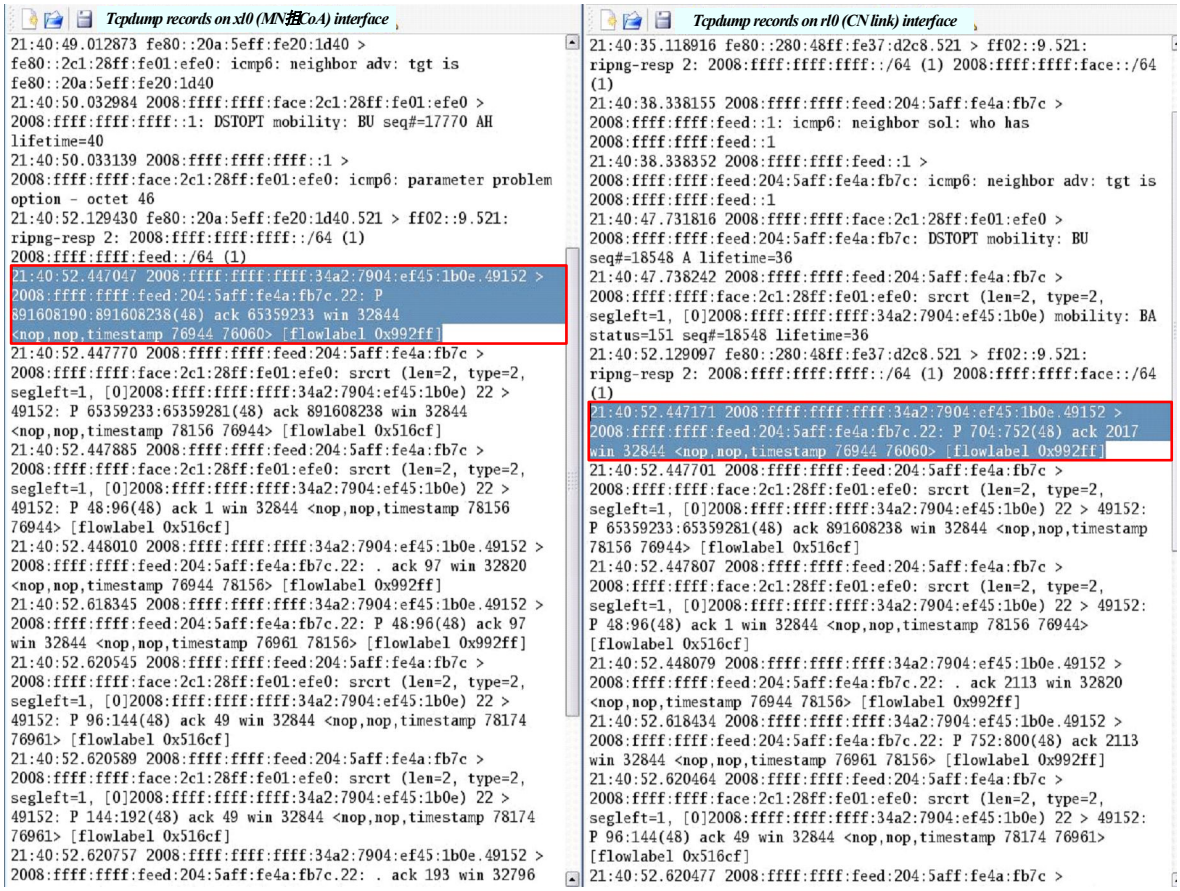

Figure 6.10 Packets transmitting after communications recovered

After exchanging the special EBU and EBA messages, the communication is recovered between the mobile node at its care-of address and correspondent node. Figure 6.10 shows results after the communication has recovered between them. We can see that the packets lost after the mobile node left its home link will be retransmitted after recovering communication between MN and CN. This is standard TCP behavior. The packet marked using a rectangle is a retransmit of the last TCP packet sent from MN's HoA to CN before the MN moved, as shown in figure 6.6.

We can observe some information from the **cnd** daemon in the log file, shown annotated in figure 6.11. The correspondent node processes a binding message after receiving the binding from the mobile node. In this processing, we find this message is our special EBU message, which includes the CGA option, signature option, CoTI option, where the home nonce index and care-of nonce index are both 0 in the nonce indices option. Then the correspondent node will add this binding in its binding cache and pass the binding information to the kernel through "mipsock". In the binding information (BC info), our new "PU" (proved_usable) flag bit, is included, which has value 0 here, to indicate that the sender's home prefix of this binding is not authenticated. So, the status code 151 will be included in the BA message to inform the mobile node keep trying the home test.

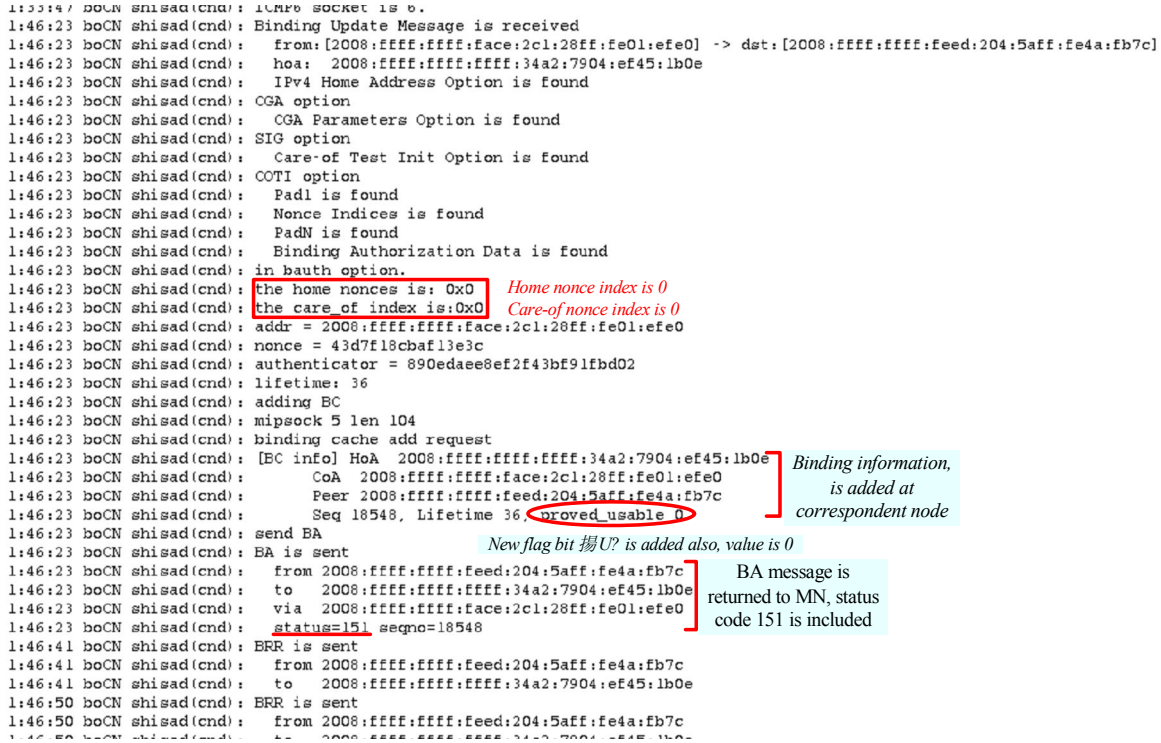

Figure 6.11 Details of Log file of cnd daemon

In this test, MN's home address was not validated, so no packets should be directed to mobile node's home address when the binding at the CN expires. In order to test this case, we unplug mobile node from its CoA link, to simulate the mobile node moving to another place or crashing or this link becoming unreachable. Then we observe the packets transmitted

after this operation. Before we unplug the MN, we use **ping** [14] to serve as time marker. Figure 6.12 shows **tcpdump** records from the IPv6 router, which monitors both interfaces xl0 and rl0. We can see that MN is unplugged after receiving "echo reply seq 3" from CN. Then CN tries to send some binding refresh messages to MN but cannot get a reply. After some time, the CN stops sending packets to MN, neither MN's CoA nor MN's HoA. We can see in figure 6.6, in left window, which records interface pcn0 (MN's HoA) interface at same time, the time of last packet is transmitted in pcn0 is "21:40:35.119559".

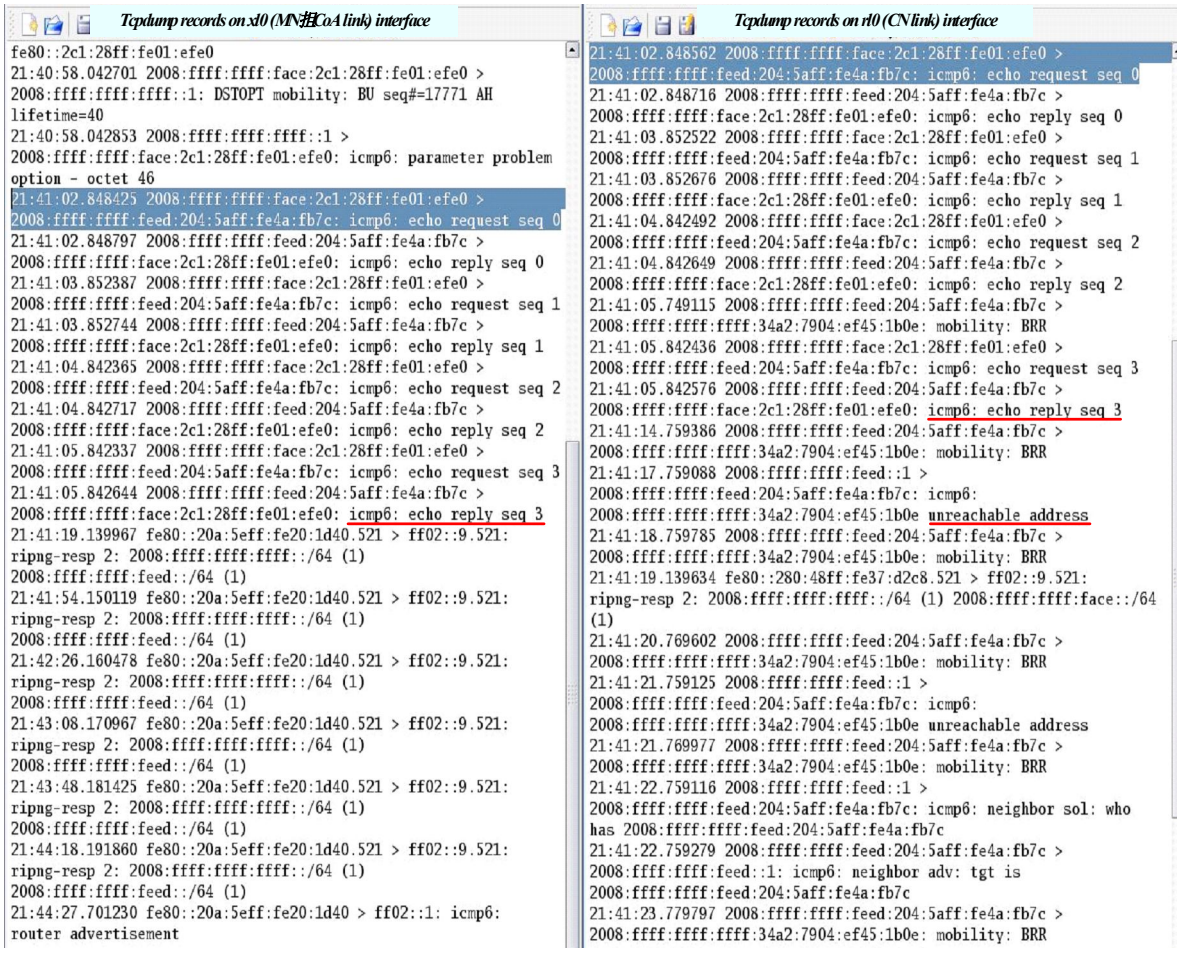

Figure 6.12 Tcpdump records after unplug MN

6.3 Summary

We have described our experimental environment and testbed design in the first section of this chapter. The following section revealed the tests performed and results obtained. These demonstrate both that with our design it is possible for a mobile node to move and continue communicating without home agent assistance, and that no “return-to-home” flooding attack is possible, thus satisfying the objective of our project.

In next chapter, we will conclude our work, and discuss the advantages and limitations of this project.

CHAPTER 7

CONCLUSION AND DISCUSSION

This chapter concludes and summarizes our work, mentions the advantages it offers, and discusses some limitations of the solution designed, and of our prototype implementation of that solution.

We end with some suggestions for future work that could follow from that offered in this thesis.

7.1 Conclusion

Enhanced Route Optimization for Mobile IPv6 provides lower handoff delays, increased security, and reduced signaling overhead. It still uses the security service of home agent to a small degree, to provide the home prefix authentication through doing home test if no a prepared permanent home keygen token exists. In some situations the home agent may be not reachable or its network link might be down. A node that moves, perhaps to avoid the broken network link, will be unable to use Mobile IP as previously defined, and thus its existing connections will all be lost. This thesis designed a solution for such nodes to allow them to keep the existing connection in this case and not rely on the home agent help to validate the home prefix of the mobile node.

Our solution focuses upon the operation of the correspondent node. If the home prefix of mobile node can be authenticated, the correspondent node will process the binding from this mobile node. If the home prefix of mobile node is not validated, the correspondent node will verify this binding according to our designed special binding update message requirements: if the sender's home address of this binding is a CGA and the CGA option and signature options are contained in this binding, but the home nonce value cannot be obtained through exchanging HoTI and HoT messages, we give the value 0 to home nonce index and permit the correspondent node

to receive this designed special BU or special EBU message. In order to distinguish whether the sender's home prefix is validated or not, a new designed flag bit "PU" is added to the binding cache at correspondent node. For the received special BU or special EBU, the value of "PU" is set to 0 which means that the mobile node still needs to keep trying the HoT. This value is also passed into the kernel together with other binding information after the correspondent node has validated the binding. If a binding expires, but its home prefix is not authenticated, the PU bit will prevent the binding being removed. That way, the "return-to-home" flooding attack can be avoided.

7.2 Discussion

The result of this thesis has several benefits. However, some limitations also exist. We will summarise there benefits and limitations in this section.

Advantages

1) The results achieve the purpose we set out to achieve, in that the nodes keep communicating after moving, even though the home agent is unreachable.
2) Our implementation is derived from previous work [39] that implemented Enhanced Route Optimization. It improves Enhanced Route Optimization, so it need not rely on the home agent help the mobile node do a home test to avoid the security issue known as flooding the home. We don't need to prepare anything in advance to avoid the flooding attack.
3) Operations focus on the correspondent node. The small modification can provide a big benefit to Mobile IP.

Limitations

1) The mobile node's home address is manually configured as a CGA in our implementation.
2) The solution is not intended for communicating without home agent for a long time. We are not seeking to delete the home agent from Mobile IPv6.

3) The correspondent node, which could be any internet node, needs to be upgraded to understand this modified protocol for any benefit to be obtained.
4) Incoming connections from unknown third party nodes are not handled by this procedure – assistance from the home agent remains required to enable success for that kind of communication. This is because the mobile node cannot initiate a binding update to a correspondent node it is unaware of.
5) The implementation is not suitable for general use, it is of only prototype quality, and exists only for the mobility stack in what is now quite an old version of FreeBSD.

7.3 Performance Issues

This developed work could be used for further enhance the performance of ERO by allowing connections for which no keygen token has been obtained in advance to get the benefits that ERO would allow if one had. That is, when the home agent is still available, a MN can still use the special EBU to a CN while it is completing its home test. This should allow the MN to avoid the costs of obtaining (and maintaining) keygen tokens prior to movement – including for those connections that complete before the node moves, for which obtaining keygen tokens is just wasted effort.

In our work we don't test this possible benefit, it is left for future study.

7.4 Future Work

A proposed solution that keeps the existing communication between mobile node and correspondent node while the home agent is unreachable is given in this work. We have implemented a prototype and tested its result under our experimental testbed. More operational experience is required to detect any practical problems with the solution. The extension in the future should be around the following issues:

1) We could put some functions to auto configure a CGA for the mobile node's home address, which could reduce the burden of manually configuring CGAs if there are many mobile nodes.
2) This solution is only for a temporary missing home agent function to keep communication between nodes, because we are not seeking in delete the home agent function from mobile IP network. If the home agent cannot be reachable for a long time, the mobile node can choose another home agent instead of the dead one.
3) Our solution leaves unauthenticated bindings in the kernel binding cache to prevent the owner of the unauthenticated home prefix being subject to flooding attacks. However we have no current mechanism to remove these bindings without the mobile node successfully validating its home prefix. One of two mechanisms might be suitable to handle this issue – either we can detect that there is no flooding attack, as this CN is making no attempt to send anything to the MN (or no longer is) and so the binding is no longer necessary. Or, we could move the decision on deleting, or not deleting a binding (and hence the PU bit) out of the kernel and handle it entirely in the user space daemon, allowing more flexibility in management of the cache than is possible inside the kernel.
4) It might be possible to allow incoming connections when no home agent is available by detecting this situation and dynamically updating the mobile node's Domain Name System data (its name to address mappings). When, and indeed whether, this would be successful, or appropriate, given the DNS update delays inherent in its cache friendly design requires study.

REFERENCES

[1] Charles Perkins, "IP Mobility Support", RFC 2002, Oct. 1996.

[2] Stephen E. Deering and Robert M. Hinden, "Internet Protocol, Version 6 (IPv6) Architecture", RFC 2460, Dec. 1998.

[3] David B. Johnson, Charles E. Perkins, Jari Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[4] Microsoft Corporation, "Understanding Mobile IPv6", Apr. 2004, Server 2003 White Paper.

[5] Qing Li, Tatuya Jinmei, Keiichi Shima, "IPv6 Advanced Protocols Implementation", May 2007, Morgan Kaufmann, ISBN-10: 0123704790, ISBN-13: 978-0123704795.

[6] Susan Thomson, Thomas Narten, "IPv6 stateless Address Autoconfiguration", RFC 2462, December 1998.

[7] R. Droms, Ed., Jim Bound, Bernie Volz, Ted Lemon, Charles E. Perkins, and Mike Carney, "Dynamic Host Configuration Protocol for IPv6", RFC 3315, July 2003.

[8] S. Varada, Ed., Transwitch, D. Haskins, and E. Allen, "IP Version 6 over PPP", RFC 5072, September, 2007.

[9] Thomas Narten, Erik Nordmark, and William Allen Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[10] Jari Arkko, Vijay Devarapalli, and Francis Dupont, "IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", RFC 3776, June 2004.

[11] Charles E. Perkins, and David B. Johnson, "Mobility Support in IPv6", 1996, Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking, Rye, New York, United States.

[12] Alex Conta, and Stephen Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.

[13] Timo Koskiahde, "Security in Mobile IPv6", Tampere University of Technology, April 2002, 8306500 Security Protocols.

[14] Ping [Online], http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/ping.mspx?mfr=true (last accessd, December, 2010)

[15] Tuomas Aura: “Cryptographically Generated Addresses (CGA)”, RFC 3972, March 2005.

[16] Jari Arkko, Jari Arkko, Brian Zill, and Pekka Nikander, “Secure Neighbor Discovery (SEND)”, RFC 3971, March 2005.

[17] Jari Arkko, Christian Vogt and Wassim Haddad, “Enhanced Route Optimization for Mobile IPv6”, RFC 4866, May 2007.

[18] Robert Elz, Sinchai Kamolphiwong, “Towards Universal Mobile-IP”, 2008. Proceedings of the International Conference on Mobile Technology, Applications, and Systems, Yilan, Taiwan.

[19] Charles E. Perkins, “IP Mobility Support for IPv4”, RFC 3220, January 2002.

[20] Dan Harkins and Dave Carrel, “The Internet Key Exchange (IKE)”, RFC2409, IETF, November 1998.

[21] Stephen Kent and Randall Atkinson, “IP Encapsulating Security Payload (ESP)”, RFC 2406, November 1998.

[22] Stephen Kent and Randall Atkinson, “Security Architecture for the Internet Protocol”, RFC 2401, November 1998.

[23] Stephen Kent and Randall Atkinson, “IP Authentication Header”, RFC 2402, November 1998.

[24] Tuomas Aura and Michael Roe, “Designing the Mobile IPv6 Security Protocol”, Technical Report, Mircrosoft Research, Inproceedings, MSR-TR-2006-42, April 2006.

[25] Robert M. Hinden and Stephen E. Deering, “IP Version 6 Addressing Architecture”, RFC 4291, February 2006.

[26] Thomas Narten, Erik Nordmark and William Allen Simpson, “Neighbor Discovery for IP Version 6 (IPv6)”, RFC 2461, December 1998.

[27] SHISA mobility stack [Online], http://www.kame.net/newsletter/20041211/, December 2005

[28] The FreeBSD Project [Online], http://www.freebsd.org/releases/5.4R/announce.html (last accessed November, 2010)

[29] The KAME project [online], http://www.kame.net (last accessed November, 2010)

[30] Menezes, Alfred, van Oorschot, Paul C and Vanstone, A. Scott, “Handbook of Applied Cryptography”. CRC press, Oct 1996. ISBN 0-8493-8523-7

[31] Keiichi Shima, Koshiro Mitsuya and Ryuji Wakikawa. “SHISA: The MIPv6/NEMO BS Stack Implementation Current Status”, Asia BSD conference, March 2007.

[32] Tony Cheneau, Aymen Boudguiga and Maryline Laurent, “Significantly improved performances of the cryptographically generated addresses thanks to ECC and GPGPU”, Computer & Security, Volume 29, Issue 4, pages 419-431, June 2010.

[33] National Institute of Standards and Technology, “Secure Hash Standard”, FIPS PUB 180-1, April 1995, http:// www.itl.nist.gov/fipspubs/fip180-1.htm

[34] Hugo Krawczyk, Mihir Bellare, Ran Canetti, “HMAC: Keyed-Hashing for Message Authentication”, RFC 2104, February 1997.

[35] Kuang Shilei, Robert Elz, and Sinchai Kamolphiwong, “Investigating Enhanced Route Optimization for Mobile IPv6”, The Thirteenth IEEE Asia-Pacific Computer Systems Architecture Conference (ACSAC 2008), Aug 2008.

[36] Wireshark [Online], http://wireshark.org (last accessed Oct. 2009)

[37] Tatu Ylonen, Chris Lonvick, “The Secure Shell (SSH) Protocol Architecture”, RFC 4251, January 2006.

[38] Joseph D. Sloan, “Network Troubleshooting Tools”, O’Reilly Media, August, 2001. ISBN 0-596-00186-X.

[39] Kuang Shilei, “Investigating Crytographically Generated Addresses in Mobile IPv6”, Master of Computer Engineering Thesis, Prince of Songkla University, 2008.

[40] Folkert Saathoff, “A short overview of the SHISA MIPv6 stack”, Dec 2005. http://www.feedface.com/howto/SHISA.Overview.pdf

[41] RSA Laboratories, “RSA Encryption Standard, Version 2.1”, PKCS #1, 14, June, 2002.

[42] Vijay Devarapalli, Ryuji Wakikawa, Alexandru Petrescu and Pascal Thubert, “Network Mobility (NEMO) Basic Support Protocol,” RFC 3963, January 2005.

VITAE

Name Miss Cui Bo

Student ID 5010120147

Educational Attainment

Degree	Name of Institution	Year of Graduation
Bachelor of Engineering	JiangXi University of Science and Technology	2006

List of Publication and Proceedings

[1] Cui Bo, Robert Elz and Sinchai Kamolphiwong, “Mobile IPv6 without Home Agent”, in proceeding of the seventh annual international conference of ECTI-CON 2010 , Chiang Mai, Thailand, May, 2010.