Automated HA Configuration for MIPv6

Wuttipon Noopetch

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Computer Engineering
Prince of Songkla University
2019

Thesis Title	Automated HA Configuration for MIPv6
Author	Mr. Wuttipon Noopetch
Major Program	Computer Engineering

Major Advisor	**Examining Committee :**
....	....Chairperson
(Assoc. Prof. Dr.Sinchai Kamolphiwong)	(Assoc. Prof. Dr.Sangsuree Vasupongayya)
	....Committee
	(Assoc. Prof. Dr.Sinchai Kamolphiwong)
	....Committee
	(Asst. Prof. Dr.Nittida Elz)
	....Committee
	(Asst. Prof. Dr.Anan Phonphoem)

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.Damrongsak Faroongsarng)

Dean of Graduate School

This is to certify that the work here submitted is the result of the candidate’s own investigations. Due acknowledgement has been made of any assistance received.

....Signature
(Assoc. Prof. Dr.Sinchai Kamolphiwong)
Major Advisor

....Signature
(Mr. Wuttipon Noopetch)
Candidate

I hereby certify that this work has not been accepted in substance for any degree, and is not being currently submitted in candidature for any degree

....Signature
(Mr. Wuttipon Noopetch)
Candidate

ชื่อวิทยานิพนธ์	การตั้งค่าโฮมเอเจนท์แบบอัตโนมัติสำหรับอุปกรณ์เครือข่ายเคลื่อนที่บนอินเตอร์เน็ตโพรโตคอลรุ่นที่หก
ผู้เขียน	นายวุฒิภณ หนูเพชร
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2561

บทคัดย่อ

Mobile IPv6 คือส่วนขยายของโปรโตคอล IPv6 ซึ่งออกแบบมาสำหรับอุปกรณ์พกพาที่เคลื่อนย้ายไปยังเครือข่ายใหม่ที่รองรับโปรโตคอล IPv6 โดย Mobile IPv6 สามารถรักษาการเชื่อมต่อของอุปกรณ์พกพาผ่านเครือข่ายบ้านหลังจากอุปกรณ์เปลี่ยนแปลงเครือข่าย อย่างไรก็ตาม Mobile IPv6 สามารถรองรับกระบวนการติดต่อโดยตรงระหว่างอุปกรณ์พกพาและคอมพิวเตอร์บนอินเตอร์เน็ตและรักษาการเชื่อมต่อให้ไม่ขาดหายหลังจากอุปกรณ์พกพาเคลื่อนย้ายได้ แต่น่าเสียดายเมื่ออุปกรณ์พกพาย้ายไปยังเครือข่ายอื่น ที่ไม่ใช่เครือข่ายบ้าน ก่อให้เกิดค่าที่ต้องสูญเสีย (operation cost) เพื่อแลกกับการรักษาการเชื่อมต่อและการเชื่อมต่อโดยตรง ดังนั้นถ้าอุปกรณ์พกพาอยู่ที่เครือข่ายบ้านเสมอแล้วมันจะไม่เกิดค่าที่ต้องสูญเสีย ดังนั้นอุปกรณ์พกพาจะต้องมีหลายเครือข่ายบ้านและมันจะต้องมีหลาย Home Agent ด้วยเหตุผลนี้พวกเราจึงเสนอโปรโตคอลการตั้งค่าโฮมเอเจนท์แบบอัตโนมัติสำหรับอุปกรณ์เครือข่ายเคลื่อนที่บนอินเตอร์เน็ตโพรโตคอลรุ่นที่หกเพื่อให้รองรับและง่ายต่อการใช้งานหลายเครือข่ายบ้านของอุปกรณ์พกพา เพราะว่าโปรโตคอลนี้จะช่วยลดการตั้งค่า Home Agent ของอุปกรณ์พกพา

คำสัญคัญ: Mobile IPv6, Multiple Home Agents

Thesis Title	Automated HA configuration for MIPv6
Author	Mr. Wuttipon Noopetch
Major Program	Computer Engineering
Academic Year	2018

ABSTRACT

Mobility support in IPv6 is an extension of IPv6 protocol. This protocol provides support for a mobile IPv6 node which relocate to another network location. It keeps connections of the mobile node operating via its home network after that change of network location. It also provides a method to allow communication directly between a mobile node and other computers on the internet and retain those connections after a move. Unfortunately, when a mobile node is not at home there are additional networking costs to establish and retain either direct or indirect connections. However when a mobile node is at its home network there are no significant extra costs. For this reason we propose the Automatic Home Agent Configuration protocol to allow a mobile node to configure each network it visits as another home network, and operate the mobile node with multiple simultaneous home network. While Mobile IP's design never required only a single home, the lack of an automated method to configure home agents has caused that to be a practical requirement. This new protocol overcomes that problem, and allows the costs of using Mobile IP to be reduced.

Keywords: Mobile IPv6, Multiple Home Agents

ACKNOWLEDGEMENT

I would like to thank my advisors: Assoc. Prof. Dr.Sinchai Kamolphiwong, has helped and provided the opportunity for my work. And Mr. Robert Elz, has devoted a lot of time to me and has always been very patient with my questions.

I would like to thank the Graduate School of Prince of Songkla University, and the staff of the Computer Engineering Department for assistance and providing the environment for me.

I would like to thank my friends in CNR and Graduate student room for their support and assistances.

Finally, I would like to thank my parents for their love and everything.

Wuttipon Noopetch

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (CONTINUE)

Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

LIST OF ABBREVIATIONS AND SYMBOLS

ARP	Address Resolution Protocol
BA	Binding Acknowledgement
BM	Balancer and Monitor
BU	Binding Update
CGA	Cryptographically Generated Address
CN	Correspondent Node
CoA	Care-of Address
CoT	Care-of Test
CoTI	Care-of Test Init
CPA	Certification Path Advertisement
CPS	Certification Path Solicitation
DoS	Denial of Service
DHAAD	Dynamic Home Agent Address Discovery
DHCP	Dynamic Host Configuration Protocol
HA	Home Agent
HoA	Home Address
HoT	Home Test
HoTI	Home Test Init
ICMP	Internet Control Message Protocol version 4
ICMPv6	Internet Control Message Protocol version 6
ID	Identifier
IETF	Internet Engineering Task Force
IGMP	Internet Group Membership Protocol
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
LAN	Local Area Network
MHADS	Multiple Home Agents Deployment Scheme
MITM	Man in the Middle
MIP	Mobile IP
MIPv6	Mobile IPv6
MN	Mobile Node
ND	Neighbor Discovery

NA	Neighbor Advertisement
NS	Neighbor Solicitation
RA	Router Advertisement
RO	Route Optimization
RR	Return Route-ability
RS	Router Solicitaton
RTT	Round Trip Time
SEND	SEcure Neighbor Discovery
TCP	Transmission Control Protocol
VHARP	Virtual Home Agent Reliability

CHAPTER 1

INTRODUCTION

1.1. Motivation

In the internet world, when a node requires communication with another node where an IP address is needed for identify the node to talk to. An IP address is not changed frequently when a node remains on the same network resemble a workstation device (such as personal computer, computer server, router and network attached storage (NAS)). However, it seems that mobile devices (for example a mobile phone, laptop and tablet) may move to different sub-networks. So mobile devices are alternate IP addresses when they change a network domain. Consequently, it causes a connection fail from the previous network. To solve this problem we have a Mobile IP solution.

A mobile device with Mobile IP can move to another network and keep connecting to the other end node. Mobile IP allows a connection to remain when a mobile node moves to another network and consequently requires a new network address. However, connections are often very short, and most nodes move much less frequently than the average connection period. This exposes a problem with Mobile IP. Whenever the mobile node is away from its home communications routed via the mobile node's home agent which leads to triangle routing. This increases packet delays and the possibility of loss. Also, this may lead to nodes disabling Mobile IP to avoid that cost, since usually few of their connections obtain any benefit.

Mobile IPv6, which is an extension of Mobile IPv4, adds route optimization, which allows a mobile node and its correspondent node to be able to communicate directly. Whenever a mobile node is away from home, it sends a binding update to the correspondent node to allow the correspondent node to associate the mobile node's home address, which it uses as a connection identifier with its care of address which gives the packet destination. With route optimization, it adds some overheads to packets, but this is a minor cost, more significantly the correspondent node is required to maintain a large binding cache containing the bindings for all mobile nodes it communicates with, though very few of those will normally move while communicating with the correspondent node. This may lead to popular correspondent node's refusing route optimization and so leaving even Mobile IPv6 with only triangle routing as a solution. These costs apply to all communications by mobile nodes that are away from home. This is the problem we seek to solve.

1.2. Objective

1) To investigate a method to reduce the costs associated with Mobile IP to mobile nodes that are away from home but not currently moving .
2) To design and experiment the solution so a mobile node can automatically find a home agent, and obtains its agreement to serve as home agent, and to configure the required security association between the home agent and mobile node.

1.3. Advantages

1) Designed solution for automatically finding a home agent, obtaining its agreement to serve as home agent, and configuring the required security association between the home agent and mobile node,
2) Guidelines for detecting home agents that are no longer of any use so they can be deleted,
3) Procedures to choose a home agent, or home address, when a local home agent is not available,
4) Prototype implementation to test the solution designed.

1.4. Scope of work

1) To investigate the multiple home agents,
2) To design the mechanism for multiple home agents,
3) To experiment the designed mechanism for multiple home agents on UNIX system for Mobile IPv6 only and not concerned with movement detection, rapid handover (reducing delay), home agents redundancy, anything related to route optimization and Mobile IP for connection surfing moving,
4) To test that the implemented mechanism can automatically find and configure an association between mobile node and home agents when the mobile node moves to a new network,

5) To experiment to find a way to detect home agents that are no longer of any use so they can be deleted,
6) To investigate choices of home address when no local home agent is available. These tasks may be tested using simulation to permit a large network to be evaluated.

1.5. Work plan

1) To study and investigate about principles of Mobile IP, Mobile IPv6 and multiple home agents,
2) To survey and investigate about related work about multiple home agents,
3) To design the mechanism for automatically finding and configuring an association between a mobile node and a home agent when the mobile node moves to a new network,
4) To implement the designed mechanism for multiple home agents.
5) To design the experiment for detecting home agents that are no longer of any usage (so they can be deleted). We seek an answer for mobile node choosing a home agent to use when the network has no home agent available,
6) Writing the final thesis report.

1.6. Outline

This thesis report consists of six chapters as follows:

In the first chapter, we describe motivations and objectives of our work. We also present the advantage achievement of our solution.

In the second chapter, we describe background information of for this domain work. The methods of IPv6 protocol for mobile device communicate with another node on internet are presented, for example, how a mobile device keeps a connection when it moves, Neighbor Discovery protocol for IPv6 node to find neighbor nodes and routers. Finally IPv6 node uses a SEcure Neighbor Discovery for increasing security of Neighbor Discovery protocol is given.

In the third chapter, we describe some problems of standard Mobile IPv6 when a mobile device has a movement with a cost of Mobile IPv6. Route Optimization extends a function of Mobile IPv6, it provides a shot routed path of Mobile IPv6 when a mobile device moves to new network. Unfortunate, more routing cost is expected. Finally we propose our solution according to above issues and problems statements. We present our designed protocol with scenarios and signal flows.

In the Fourth chapter, we present our experimenting and testing on the solution that we describe in chapter 3. Automated HA configuration for MIPv6 protocol algorithm is shown.

In the last chapter of this thesis, we summary our work and discuss some challenging issues that may need to work out in the future.

CHAPTER 2

BACKGROUND

In this chapter, we give a basic introduction to IP Mobility support for IP. This allows communication to continue when a node is moved to a new network. Then we explain the principles of IPv6 and Mobility support for IPv6. Mobility support for IPv6 operation includes the Route Optimization and Dynamic Home Agent Address Discovery protocols . We also describe the Neighbor Discovery protocol for automatic configuration.

2.1. Internet Protocol Version 6

IP addresses are dual purpose. One use is to identify end points involved in communications. They are also used to provide the location of an end point for packet delivery. Because of the second use, when a node moves its IP address must change. Because of the first use, that change causes existing communications to fail. Mobile IP was created to overcome this problem, it allows communications to continue after a node has moved, and so obtained a new address. Mobile IP (MIP) is defined in RFC 3220 [4].

IPv6 is designed to solve the problems now experienced with IPv4, and provides the ability to support more addresses. In addition some improvements were made, for example the header format is simpler than that of IPv4. This can reduce the processing time required for each packet at routers along network path. IPv6 also supports an auto-configuration mechanism for hosts.

IPv6 addresses are 128 bits each (IPv4 has 32 bit addresses). IPv6 address notation writes the addresses as 8 blocks of 16 bits in hexadecimal separated by colons rather than IPv4's notation of 8 bit fields (4 blocks), in decimal, separated by periods.

Example :

3ffe:a:0:0:0:0:123:456

From the example IPv6 address contain as 8 block of 16 bits in hexadecimal. If more than one all zero blocks are adjacent then you can write more concisely by using "::" for one such block. So from the example IPv6 address can be written briefly as "3ffe:a::123:456"

IPv6 allows 340 billion billion billion billion (3.40 x 10^{38}) unique addresses. So it is suitable for next generation device (such as mobile device). Because mobile device need a unique IP address and the numbers of them are increasing very quickly.

2.2. Mobility Support in IPv6

An IP address is a purpose to identify nodes for communicating on the internet. Also, the node gets a new IP address from that network's the node move. If an IP address's node change. It causes existing communication to fail. The IEEE proposed RFC 3220 [4] is a Mobile IP protocol to solve this problem. It can keep communication to continue after a node move.

Mobile IPv6 (MIPv6) [5] is an adaptation of the MIP protocol to IPv6. However, as MIPv6 was designed concurrently with IPv6, unlike MIP and IPv4, MIPv6 could assume that all IPv6 nodes would at least be aware of the protocol. This allowed enhancements such as Route Optimization (RO) that were not practical in MIP. Before introduction a basic operation of MIPv6, we are recommend try to understand terminology as follow below :

- Mobile Node (MN) is a moving IPv6 node.
- Home Address (HoA) is an IP address of MN. It is unchanged even if MN move to another network. The network that MN receive a HoA from router we call home network.
- Another network that is not a home network, we call a foreign network.
- When the MN move to foreign network and gets a new an IP address that IP address is a Care-of Address (CoA).
- When the MN stay at foreign network. It need to send message to special node at home network. The message contains the new an IP address (CoA) from router of foreign network. This message is called a Binding Update (BU). The special node at the home network is called a Home Agent (HA).
- When the HA get a BU from MN. It need to send back a message to the MN which call a Binding Acknowledgement (BA) message.
- A node is communicating with a MN and support mobile IPv6 is called a Correspondent Node (CN).

2.2.1.Operation

A MIPv6 MN has as its HoA the same value as its CoA when it is on its home link. Packets from CNs are sent to the MN using its HoA as the destination address. They reach the home network using the conventional Internet routing mechanism, where they are delivered to the MN as to any other node. The MN replies directly to the CN using its HoA as the source address. When the MN moves from the home link to a foreign link, it will obtain a CoA using normal IPv6 mechanisms. It can use a methods of IPv6 Neighbor Discovery (ND) [6] for automatic (stateless) address configuration. Or it can use statefull, auto-configuration using the DHCPv6 protocol. The MN will then send a BU message to its HA so the HA can update its binding cache and remember the association between the MN's HoA and its current CoA. The HA will reply to this binding registration with a BA message to inform the MN of the result of its update. Once this has completed the MN can resume communications with the CN. In Mobile IPv6 it has two choices for how this is accomplished.

2.2.2.Bidirectional

The first method is bidirectional tunneling show on figure 2.1, a packet from the CN will be sent to home network first. Then HA will intercept the packet and will tunnel it to the MN. The HA uses proxy Neighbor Discovery [6] to intercept the IPv6 packet, that was sent to the MN on the home link. To reply to the CN, packets are tunneled between MN and HA (reverse tunnel) so they can be routed to the CN from the home link. The advantage of bidirectional tunneling method is that the CN does not need to participate at all, the MN's CoA is not known to it.

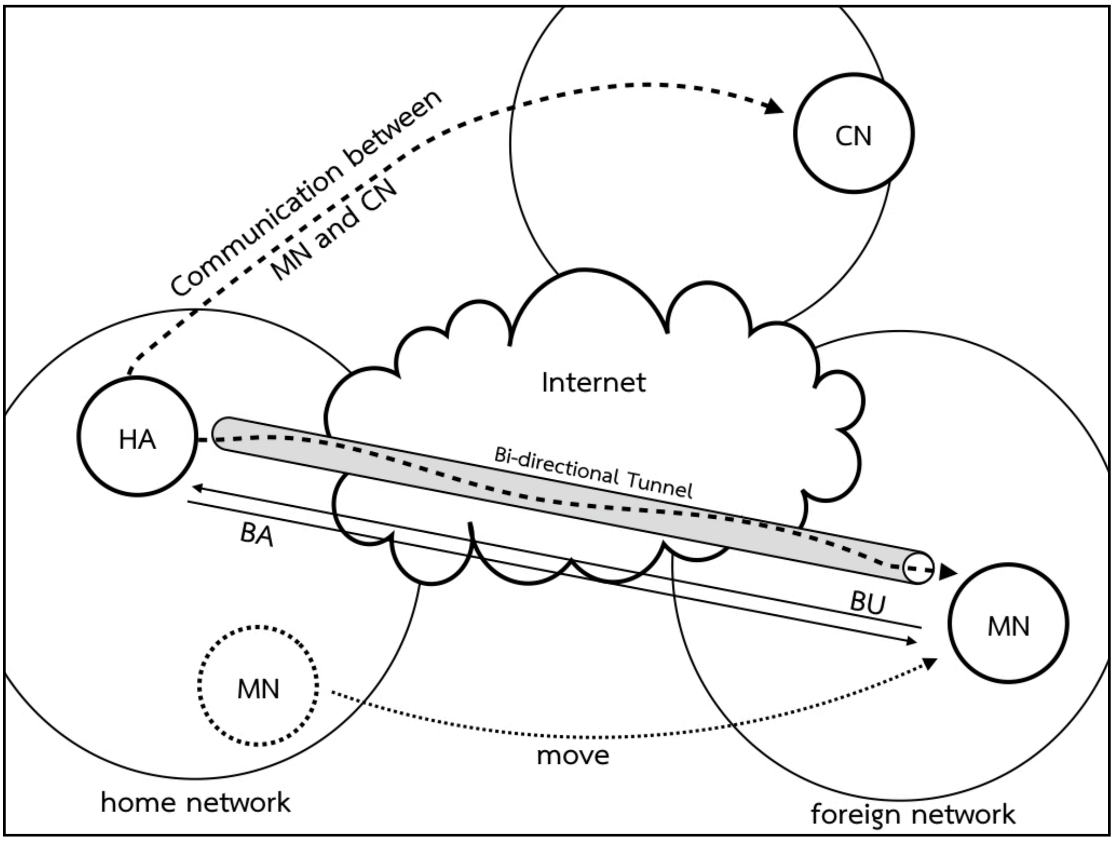


Figure 2.1 MIPv6 operation

The second method is Route Optimization (RO) which is described in the next section. Also MIPv6 can support multiple home agents, which is described in the subsequent section.

2.2.3.Route Optimization

When using MIPv6, and with support of the CN, RO can be used. With this, the MN sends a BU to the CN. However before MN and CN can establish bi-direction tunnel to communicate with each other, the CN needs a method to verify HoA and CoA that MN is its owner. The process of MIPv6 to verify that is call Return Routability (RR) [7]. The MN need to send two message to CN, first is a Home Test Init (HoTI) and second is a Care-of Test Init (CoTI). The HoTI is send from HA of MN via HoA (using the path via the HA) and the CoTI is sent from MN to CN directly via CoA. The CN responds with two message which are Home Test (HoT) and Care-of Test (CoT) messages. Once this has been accepted the CN knows the mapping between the MN’s HoA, its identity, and its CoA, its current location, and is able to directly send to the MN without a detour via the HA. When the CN needs to send a packet from the transport layer to any IPv6 destination it first checks the binding cache for the destination address. If found, the CN will use an IPv6 routing header [3] to direct the packet to the HoA via the CoA. The MN sends HoA information to CN by use of a new option of MIPv6, a Destination Options Header as the Home Address option (HAO). So the CN can communicate directly to the MN by use a Routing Header Type 2 (RTHDR2). This results in the packet traveling directly to the MN at its CoA, where it then “routes” the packet to itself at its alternate address. Advantages to using RO include a shorter transmission path between MN and CN, and less congestion at the HA. Figure 2.2 show a MIPv6 with RO operation.

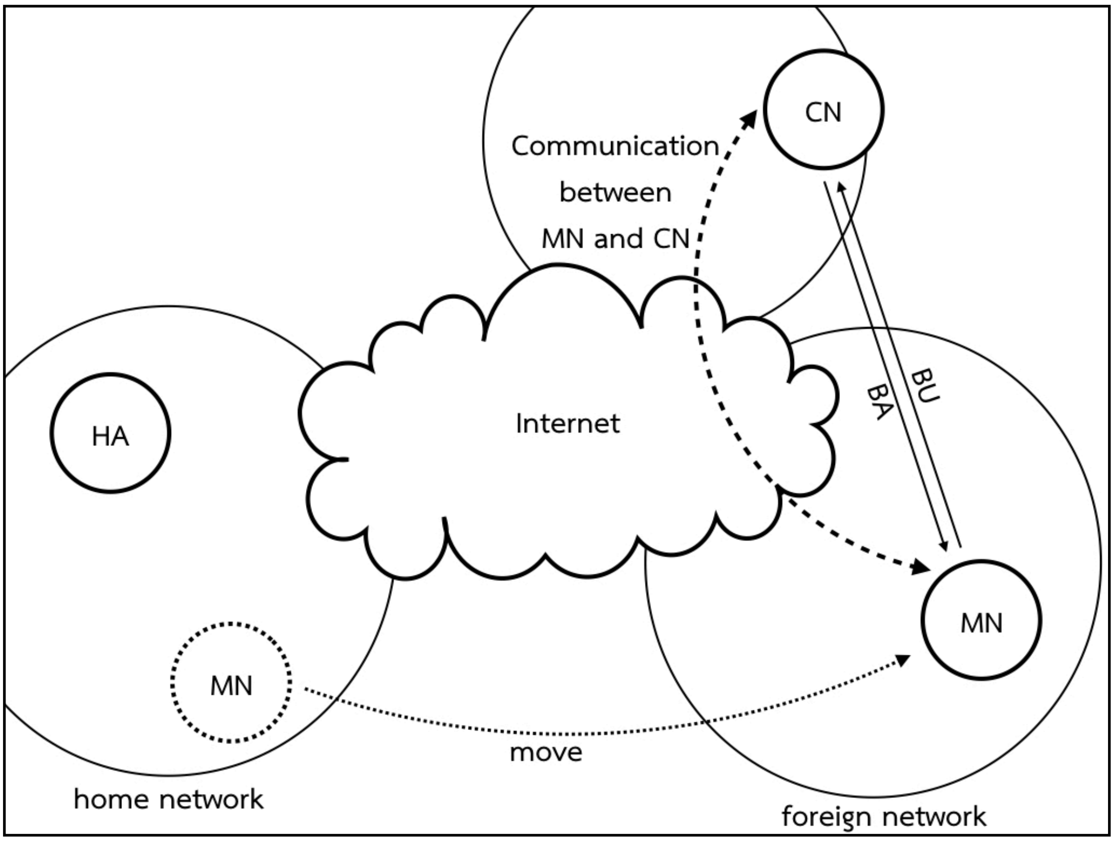


Figure 2.2 MIPv6 with RO operation

2.2.4.Dynamic Home Agent Address Discovery

There is some support for multiple HAs in MIPv6, intended to support reconfiguration of the home network. Such a reconfiguration could lead to a MN that is away from home being unable to locate its HA, as it does not know the new HA IP address. The MN can use Dynamic Home Agent Address Discovery (DHAAD) to discover the IP addresses of HAs. The MN sends a Home Agent Address Discovery message to one of the HA’s using a well known anycast address [8]. The anycast address is constructed from the prefix of the network on which the HA is expected, which is the same as the prefix of the HoA of the MN, and a defined constant. The DHAAD procedure causes the HA that receives the Discovery message to reply to the MN informing it of the addresses of HAs on the selected prefix. This allows a MN to regain contact with its HA. This mechanism does not provide any configuration, the MN must already know its HoA, and thus the prefix of its HoA, the MN must also already have a relationship with the HA in order to successfully use the HA address information discovered.

2.3. Security of Mobility Support in IPv6

When the MN is moved away from the home network. It is needed to send bind to HA in order to update the current IP address. Then the HA updates a binding cache table, and it also forwards CN's packets to MN. If this communication isn't authentic and verified information between MN and HA, or if the communication between HA and MN are not secured and encrypted, this case is possible to be attacked such as Man-in-the-Middle attacks, Hijacking, and Denial of Service attacks. We can use the tunnel between HA and MN for security, and then an IPsec tunnel is configured. The IPsec ESP protocol is required for Mobile IPv6 messages [39]. This protocol is used to provide privacy and authentication services for the IP layer. It allows a pair of nodes which are communicating to use mechanisms to provide security appropriate for the communication. The following data flows have to be secured [40]:

1. The binding update and binding ack between HA and MN.
2. HoTI and HoT messages sent via HA during the Return Routability process.
3. The ICMPv6 message prefix discovers between HA and MN.

Control messages between HA and MN need to contain authentication, integrity, proper sequence and anti-replay protection. If they are not protected, the MN and CN may be get into Man-in-the-middle, hijacking and denial of service attacks. Therefore, Security Association (SA) is required between HA and MN. Also, IPsec protocol is not responsible for protection to control the sequence of messages. The correct sequence is given in BU and BA messages. When the Internet Key Exchange (IKE) [43] is in condition of higher protection from replaying attack, it is a key distribution for IPsec, which is used for either pre-shared secrets or public keys for the key exchange.

2.3.1.False Binding Update Attack

An attacker is claimed to be a given node at a given address and then try to steal traffic which destines to that address [41]. The MN sends a bind to HA for current updated address when it is away from home shown in Figure 2.3. The MN can continue the communication with the CN because the HA knows MN's address presently and forwards packets from the CN to the MN. If an attacker wants to know some information from the CN, the attacker can send spoof BU to the HA. In consequence, the HA can misapprehend the MN's BU and MN's address presently. In addition, the HA forwards a packet to the attacker instead of the MN. Therefore, it is crucial to protect signaling messages between HA and MN.

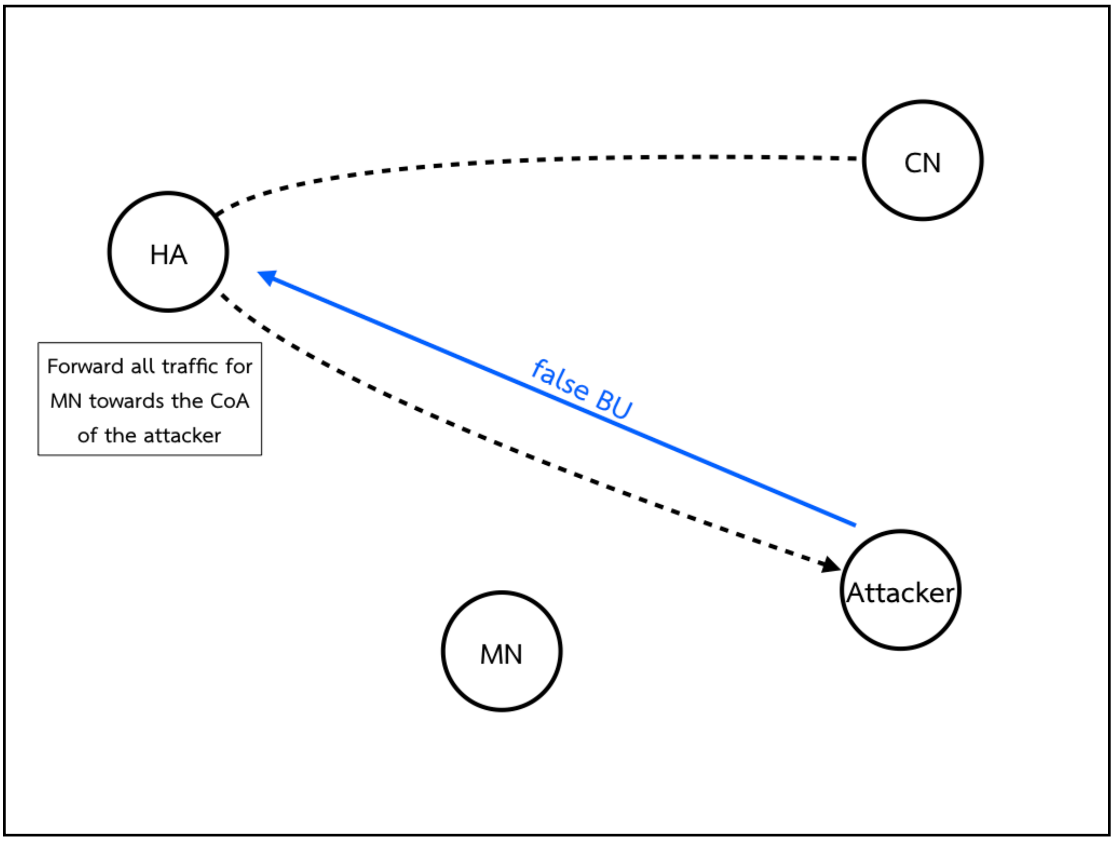


Figure 2.3 An un-protected BU for interactions between HA and MN.

Moreover, route optimization is a way of communication between MN and CN [42]. If BU is not authentic, an attacker can send spoofed BUs to both MN and CN. All nodes that support the correspondent node function would be vulnerable to this attack shown in Figure 2.4. Node A communicates with node B. An attacker can re-direct the packets to an address C by sending a BU to node B. The HoA in the BU would be A, and the CoA would be C. After that, node B receives the BU and creates a binding cache for this communication. The packets from node B are direct to node A which are sent to the CoA of a node. As a result, node A is not able to receive any packets destined to its address from node B.

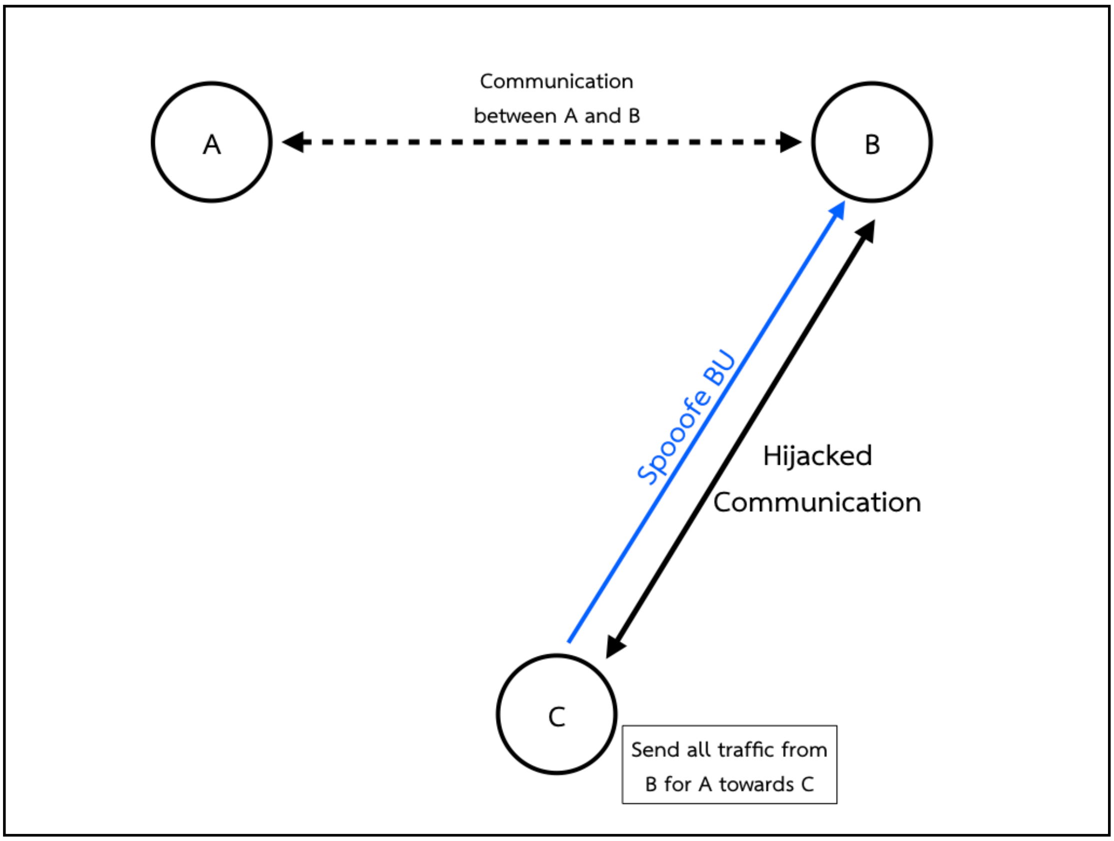


Figure 2.4 An un-protected BU attack

2.3.2.Man in the Middle

If an attacker spoofs a BU from the MN, the attacker will re-direct traffic between the HA and the MN [41] shown in Figure 2.5. If an attacker is node C, it can spoof the packets destined to node B via sending a spoofed BU to node A. It can be thus fake to be node B and steal the communication between node A and B. This situation happens when the MN is away from home. Interestingly, the MN uses a mobile IP to keep a connection with CN, so a malicious user impersonates the HA to deceive MN and CN.

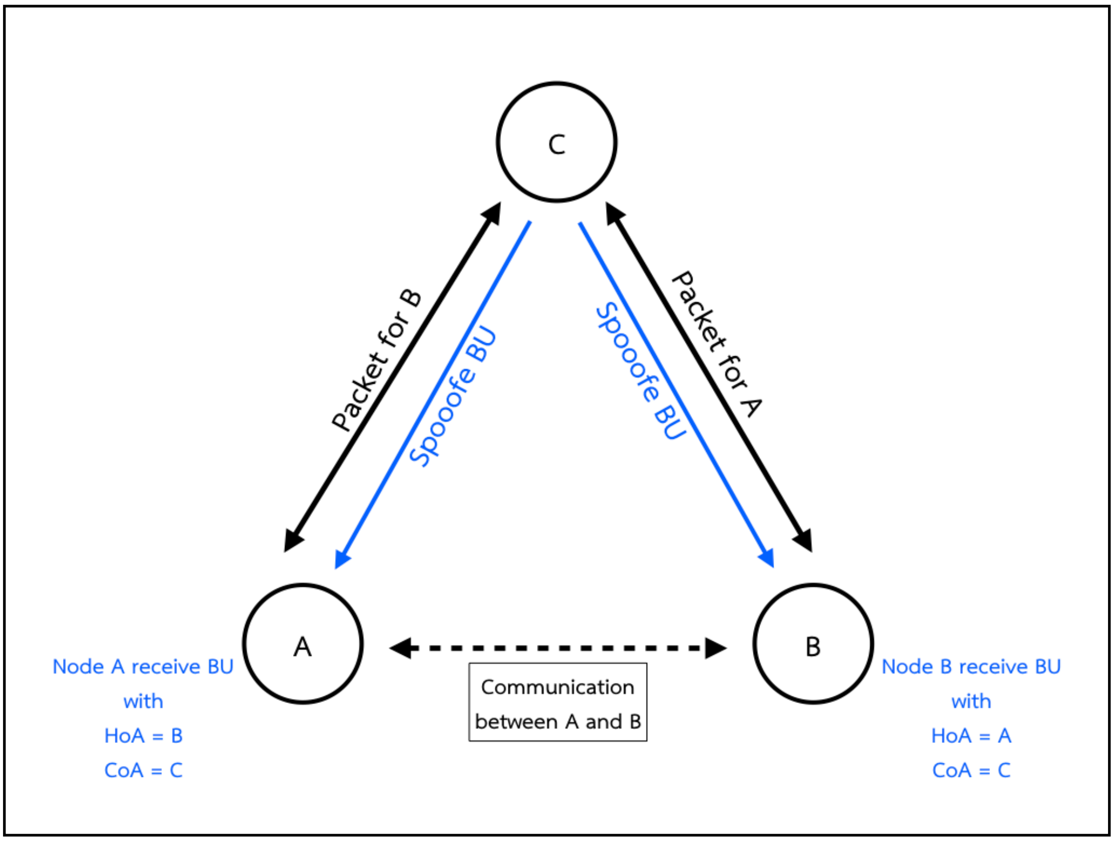


Figure 2.5 Man-in-the-Middle attack.

2.3.3.Denial of Service

An attacker impersonates a BU of the victim, the MN and sends this BU to a site with a large data stream. When the attacker sends a request to a video stream server, that server sends the video stream which is massive to the attacker's HoA. Consequently, the MN gets a large number of data from multiple servers shown in Figure 2.6. The (a) shows MIPv6 operation, and (B) exhibits a show route optimization operation.

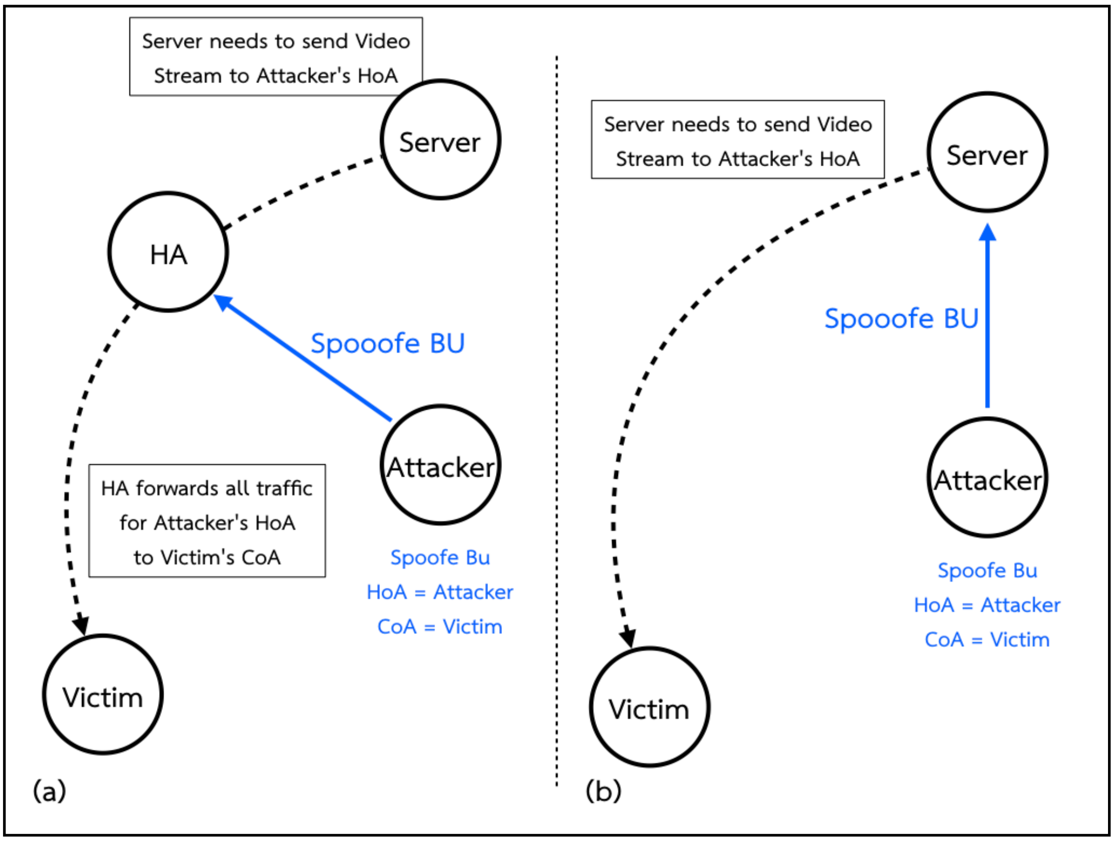

Server needs to send Video Stream to Attacker's HoA
Server
HA
Spooofe BU
Attacker
HA forwards all traffic for Attacker's HoA to Victim's CoA
Spoofe Bu
HoA = Attacker
CoA = Victim
Victim
(a)
Server needs to send Video Stream to Attacker's HoA
Server
Spooofe BU
Attacker
Spoofe Bu
HoA = Attacker
CoA = Victim
Victim
(b)

Figure 2.6 Flooding Attack.

2.4. Neighbor Discovery

Neighbor Discovery (ND) protocol is a part of Internet Control Message Protocol version 6 (ICMPv6). The ND protocol is an essential part of IPv6 since it contains such functions from the previous protocol as Internet Control Message Protocol version 4 (ICMP), Internet Group Membership Protocol (IGMP), and Address Resolution Protocol (ARP) [9]. A node uses the ND protocol to find a link-layer address, neighbor host and router on the same link in IPv6 wired or wireless local area network (LANs) [10] and maintain reachable information about the path to a native neighbor.

When the node on the IPv6 environment is applied, especially for mobility support in IPv6. The node on IPv6 can use the ND protocol for automatic configuration called IP auto configuration [11] [12]. Even if a Dynamic Host Configuration Protocol (DHCP) [13] can automatically allocate IP for the node on the IPv6 network, that network is needed to have a DHCP server to provide a service. The functions of the ND protocol are shown as follows:

1) Address resolution.

2) Redirection.
3) Neighbor unreachability detection.
4) IP auto configuration.

The ND protocol is defined as five types of messages as shown in Figure 2.7.

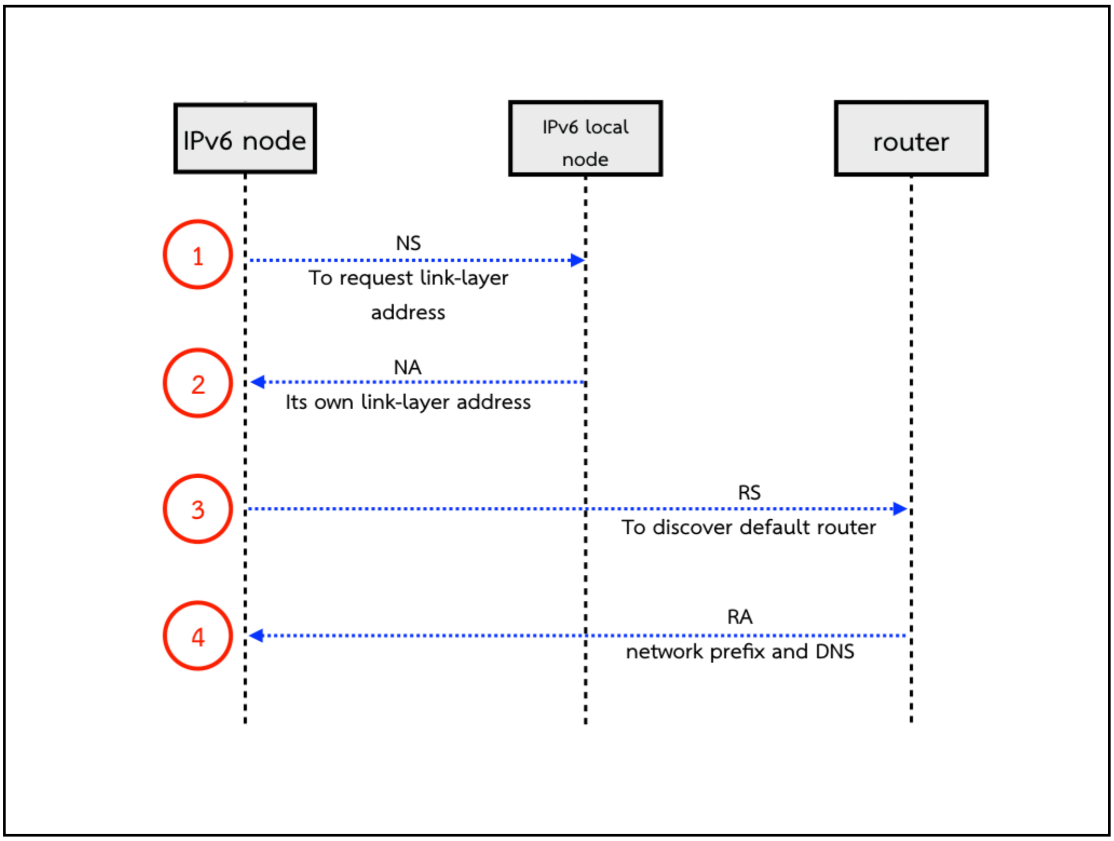

IPv6 node
IPv6 local node
router
1
NS
To request link-layer address
2
NA
Its own link-layer address
3
RS
To discover default router
4
RA
network prefix and DNS

Figure 2.7 ND protocol operation.

1) Neighbor Solicitation (NS): When the node comes to new network at the first time, a link-layer address of a neighbor node is requested via NS message.
2) Neighbor Advertisement (NA): When a neighbor host gets the NS message from the node on the same network, the NA message is replied to it with its link-layer address.
3) Router Solicitation (RS): The node sends the RS message on a network to a discovery default router and lean some network information such as network prefix and DNS, which is used for IP auto configuration.
4) Router Advertisement (RA): A router receives the RS message from the node, and the RA message is replied with network prefix and DNS.

5) Redirect: Redirect is used for a router to notify other nodes.

When the IPv6 node moves to the IPv6 link-local network, there are some steps of method IP auto configuration as follows:

1) The IPv6 node generates a temporary IPv6 link-local address by its network interface identifier such as MAC address.
2) The IPv6 node broadcasts the NS message to verify the temporary address. If another node on the network uses it, the IPv6 node will repeats step (1).
3) The IPv6 node broadcasts the RS message to a discovered default router. When a router gets the RS message, the RA message replied with IP configuration information.
4) The IPv6 node configures IP by the information of RA message.

2.5. Secure Neighbor Discovery

According to the mentioned details in item 2.4, an algorithm of the ND protocol to auto IP configuration is described. If this algorithm is not secure enough between nodes or a node and a router, a malicious user can attack an ND message on the network due to the fact that it can impersonate legitimate hosts or routers easily by spoofing the ND protocol message [14][15]. To address the security problem, the security of vulnerability, four kinds of attacks, in the ND protocol is represented below.

1) The host impersonating an attack: A malicious user impersonates a legitimate host by sending an NA message to a victim node. When this attack causes, it can be possible for an attacker to ride communication between the node and the victim like a man in the middle as displayed in Figure 2.8.

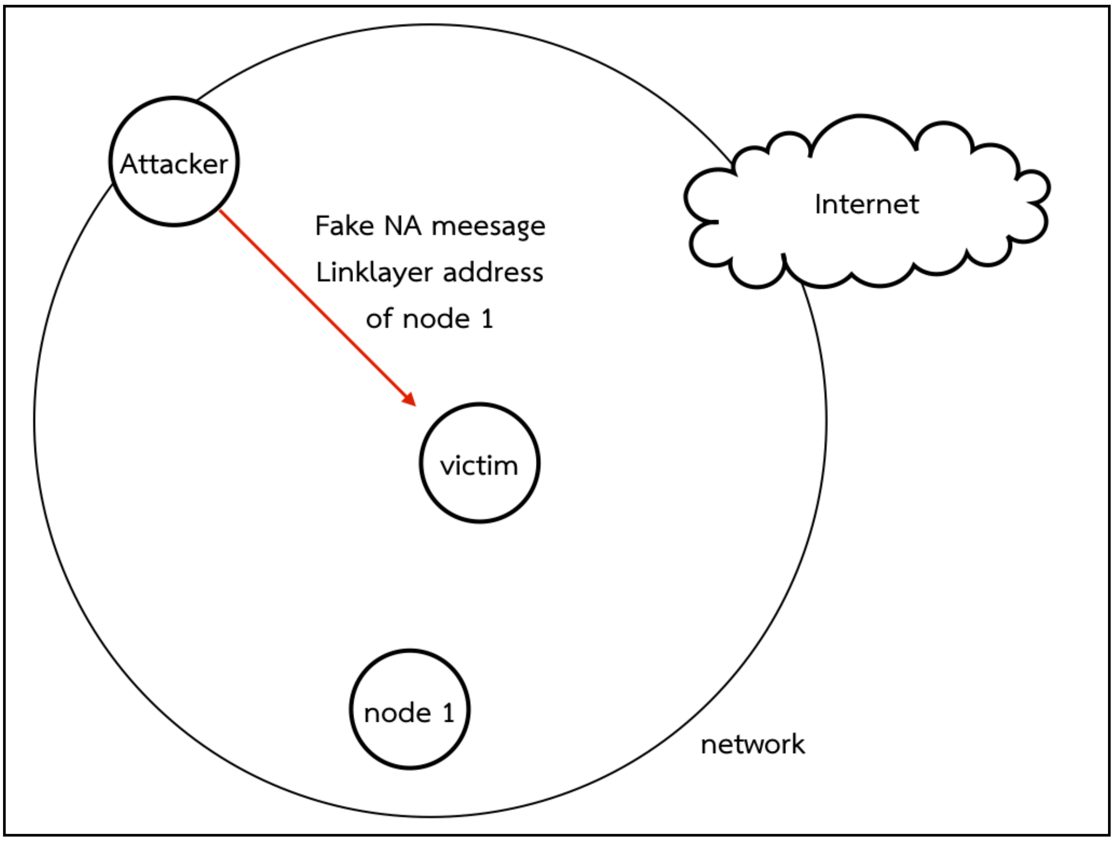


Figure 2.8 Host impersonating Attack

2) The router impersonating an attack: A malicious user impersonates a legitimate router by broadcasting an RA message on a local area network; therefore, a network is caused to paralyze as presented in Figure 2.9.

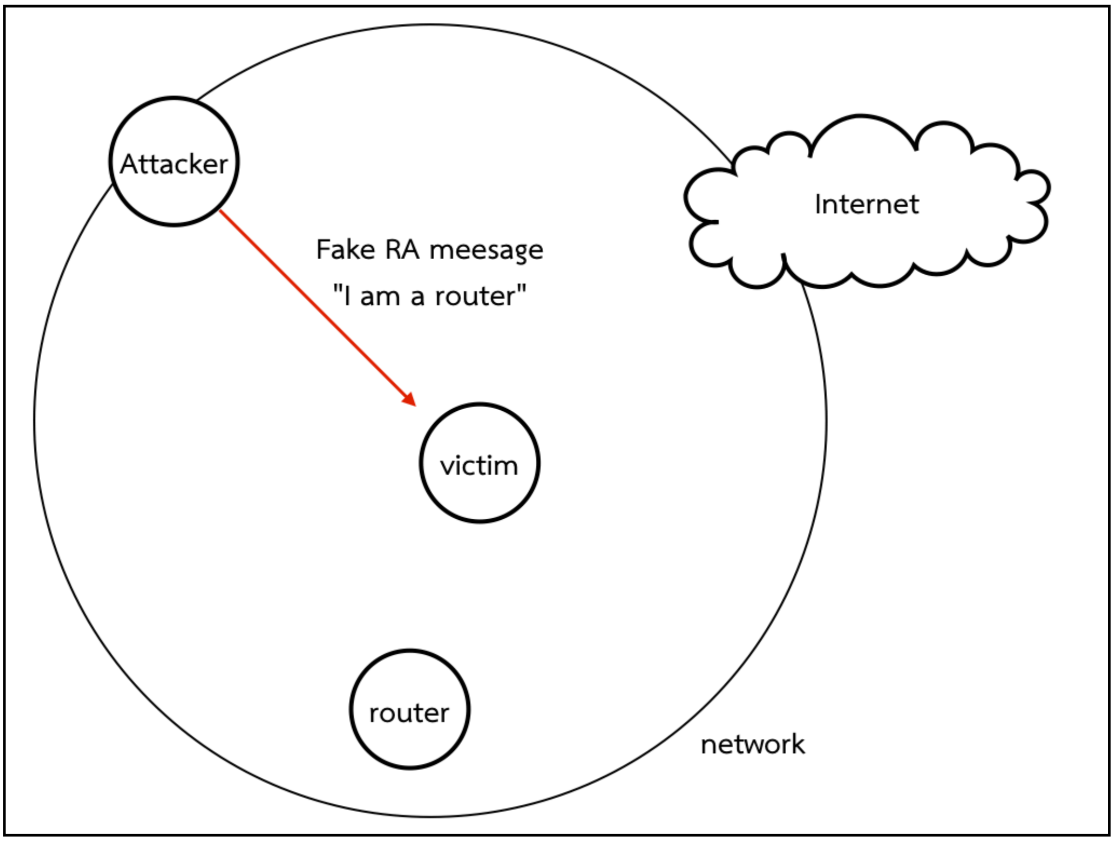


Figure 2.9 Router impersonating Attack

3) The replay attack: A malicious user impersonates an ND protocol message. Since a malicious user duplicates the ND protocol message on the network, it can replay this packet to a victim node. A malicious user can also replay a big number of the ND protocol messages to this victim node. The replay attack additionally performs DoS attack as shown in Figure 2.10.

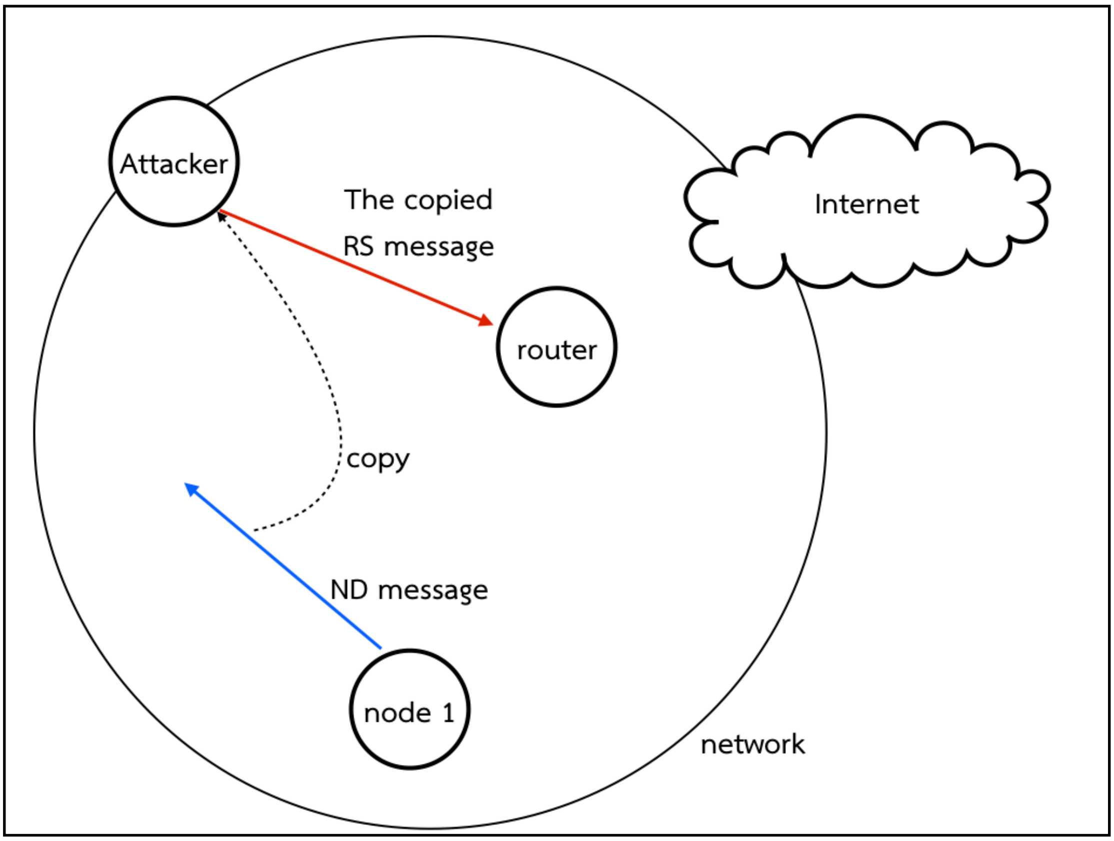


Figure 2.10 The Replay attack

4) The ND-DoS attack is a kind of DoS attack [16][17]. If a malicious user stays at an external network, it makes an IPv6 access to a router broadcasting a massive number of ND protocol messages which can continue to the victim network. Therefore, this attacker fabricates a destination for the IP address of the IP packet. In fact, the router is obligated to resolve the invalid destination IP address by broadcasting an ND protocol message on the network. The victim network is led to reduce the Quality of Service (QoS) for standard network traffic as shown in Figure 2.11.

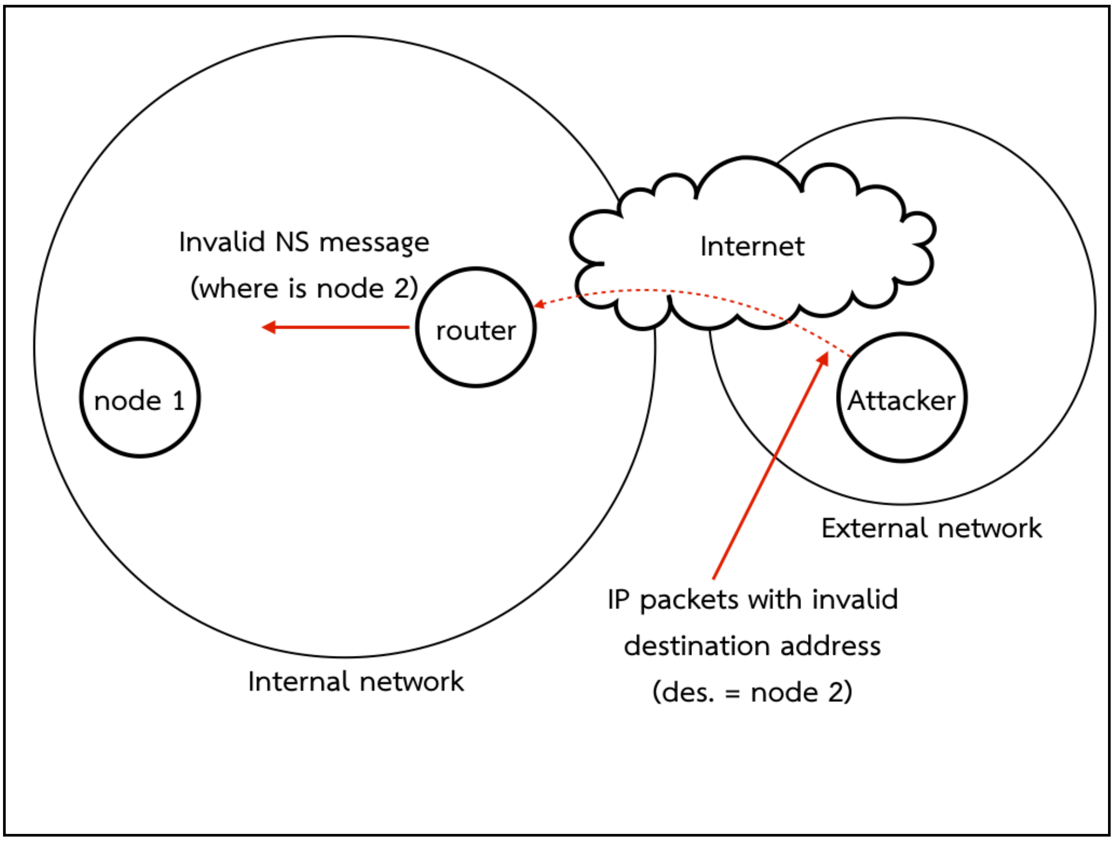


Figure 2.11 ND-DOS attack

Secure Neighbor Discovery (SEND) was proposed by the Internet Engineering Task Force (IETF) [18]. Its essential functions are presented below.

1) Address ownership proof mechanism: This algorithm is certainly the sender of the ND protocol messages using the Cryptographically Generated Address (CGA). The CGA contains a subnet prefix and an interface identifier (ID). The interface ID is made from a hash function with a public key and an auxiliary parameter, so the Interface ID = hash (public key, parameters) and CGA = concatenation (subnet prefix, interface ID). According to the SEND protocol a CGA option message is defined for a public key and an associated parameter. The node can verify between the public key and the source IP address itself.
2) The ND protocol message protection: A malicious user can steal a public key from other nodes on a network and modify ND protocol messages. The SEND provides a public key with RSA signature in order to protect the ND protocol from this situation, known as Digital signature. The sender is needed to include the digital signature in RSA signature option message in

the ND packet. Then the receiver can verify the integrity and the authentication of the sender.

3) The replay attack prevention: The SEND defines ND options. Firstly the timestamp is used to verify an advertisement message which has not been a replay. Secondly, the nonce (random number) is used to verify a solicit advertisement message that is a fresh response.
4) The router authority certification: This mechanism is proposed as a defense against an unauthorized router, rather than for the protection of ND protocol messages. A router provides the ND protocol message to host with a certificate and a trust anchor. The node can prove itself as an authorized router. Two new options in the ND protocol messages are proposed by the SEND protocol. First the Certification Path Solicitation (CPS) is used by host to request certification and a trust anchor of the router. Secondly, the Certification Path Advertisement (CPA) is used by a router reply to a CPS. These messages allow a host to verify that a router has been authorized by the local network management to act as a router for this network - the host can follow the trust chain to verify that the authorizing network manager is authorized by the owner of the network, and in turn that they are indeed the organization to whom the network address space has been delegated. In combination, and together with the other secure neighbor discovery messages described earlier that allow proof of the ownership of the source address of packets, these allow a host to verify that a node claiming to be a router on the network has been authorized to act as a router on this network.

CHAPTER 3

PROBLEM STATEMENT

In this chapter, we describe the costs and benefits of the standard MIPv6 protocol described in section 3.1. From this we observe that MIPv6 imposes costs upon all connections when the MN is away from its home network.

Route optimization (RO) can reduce those costs by allowing direct connections between MN and CN. However we fear that the costs imposed upon CNs, particularly popular ones, to support RO will lead to many of them refusing RO attempts. Section 3.2 describes the reasons for this.

We also survey related work with a multiple HAs. The majority uses of multiple HAs focus on load balancing and backup mechanism HAs of MN. That method has multiple HAs on the home network only. That is a different problem then the one investigated here with a different solution. We purpose MN associating with multiple HAs, one on each foreign network to reduce the costs of using MIPv6.

3.1. Triangle Route on Mobile IP

When the MN is away from home, all packets travel from the CN to the MN through the HA as shown in figure 3.1. The transmit time of packets via the HA is longer than the transmit time of a packet directly between HA and MN. These are effects of triangle route [19]. The worst case cost of the triangle route problem is shown in figure 3.2, when the MN moves to CN's network, but the MN sends all packets via the home network.

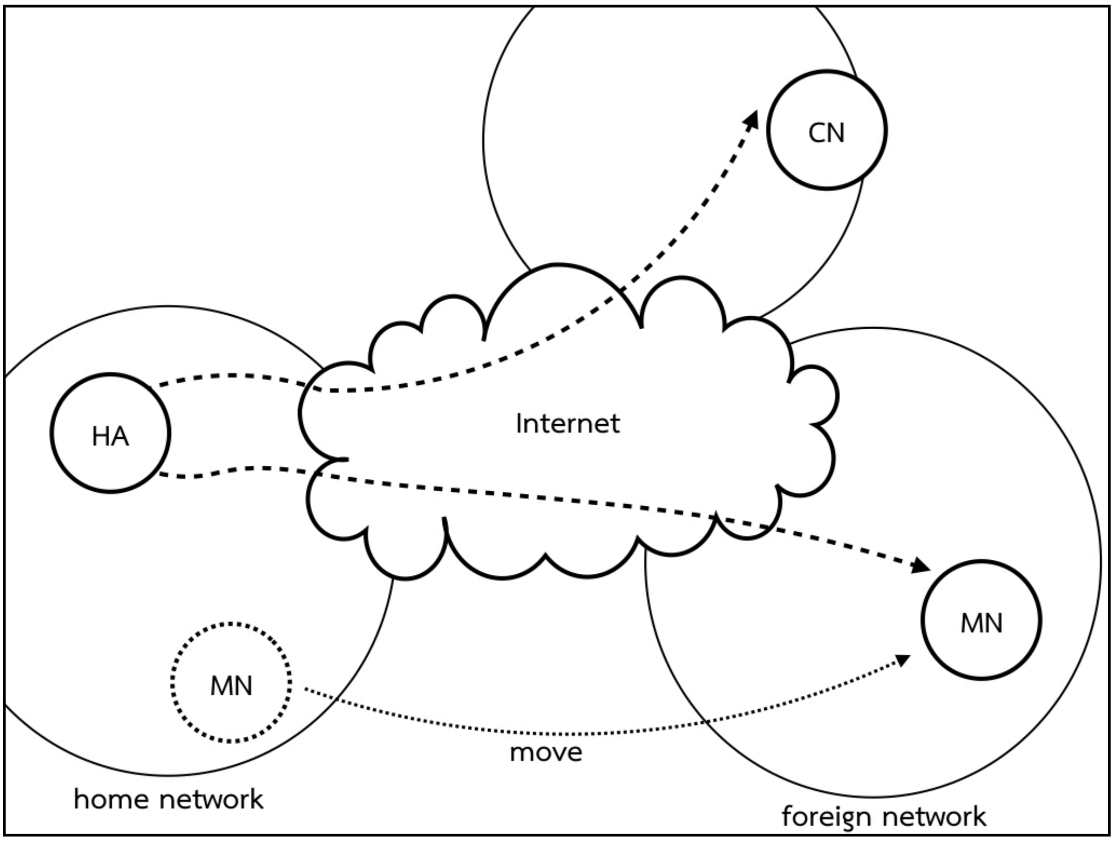


Figure 3.1 Triangle route problem.

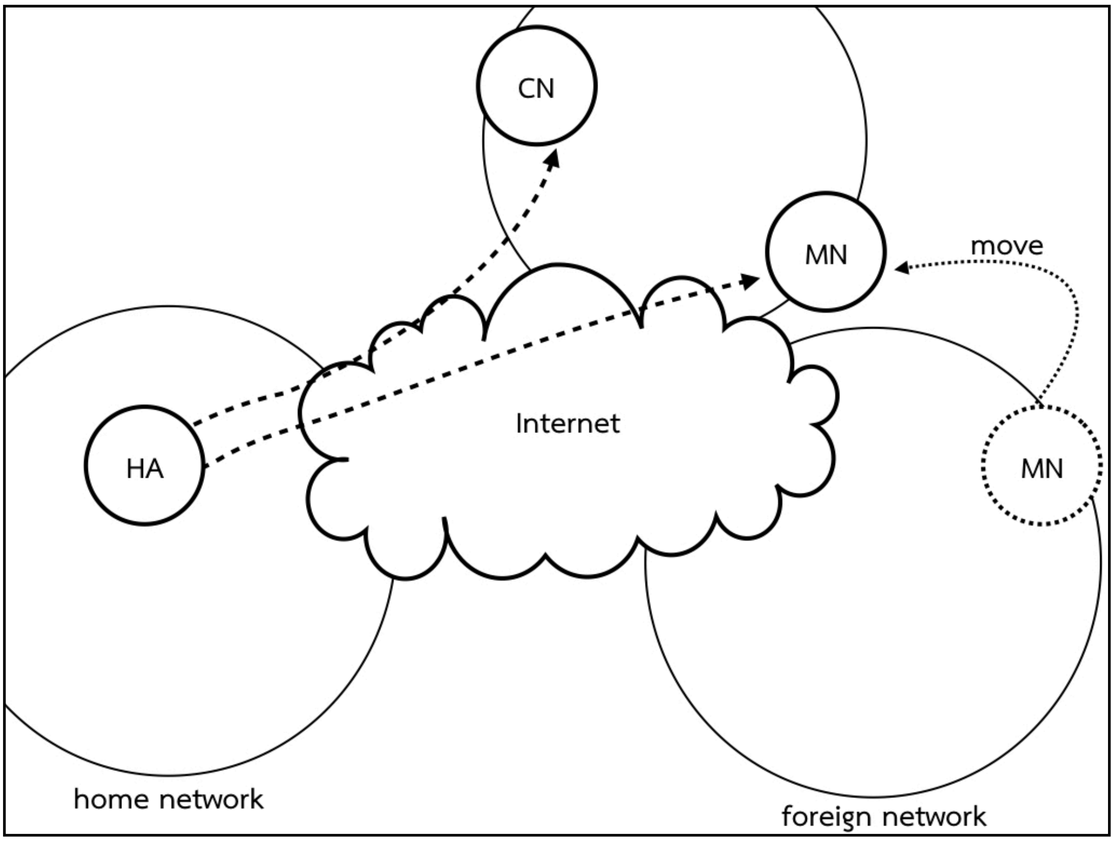


Figure 3.2 The worst case of triangle route problem.

1) Packet delay added by triangle routing causes problem with real time applications, for example Video Conference, Multicast Video.
2) The performance of the Transmission Control Protocol (TCP) depends upon the Round Trip Time (RTT) [35] between the client and server. A longer RTT means a slower maximum transmit rate. Further, more hops means a bigger chance of packet loss, and retransmission.
3) Packet loss because MNs send packet to CN via the HA. It uses more bandwidth and can cause traffic congestion at the HA.
4) Tunnel between HA and MN reduces MTU of that link, which may reduce path MTU between CN and MN (usually will). Which means more packets transmitted for connections that use long packet (TCP) or more fragmentation for large UDP (eg. DNS with DNSSEC). Both lead to increased congestion and packet losses as shown in figure 3.3.

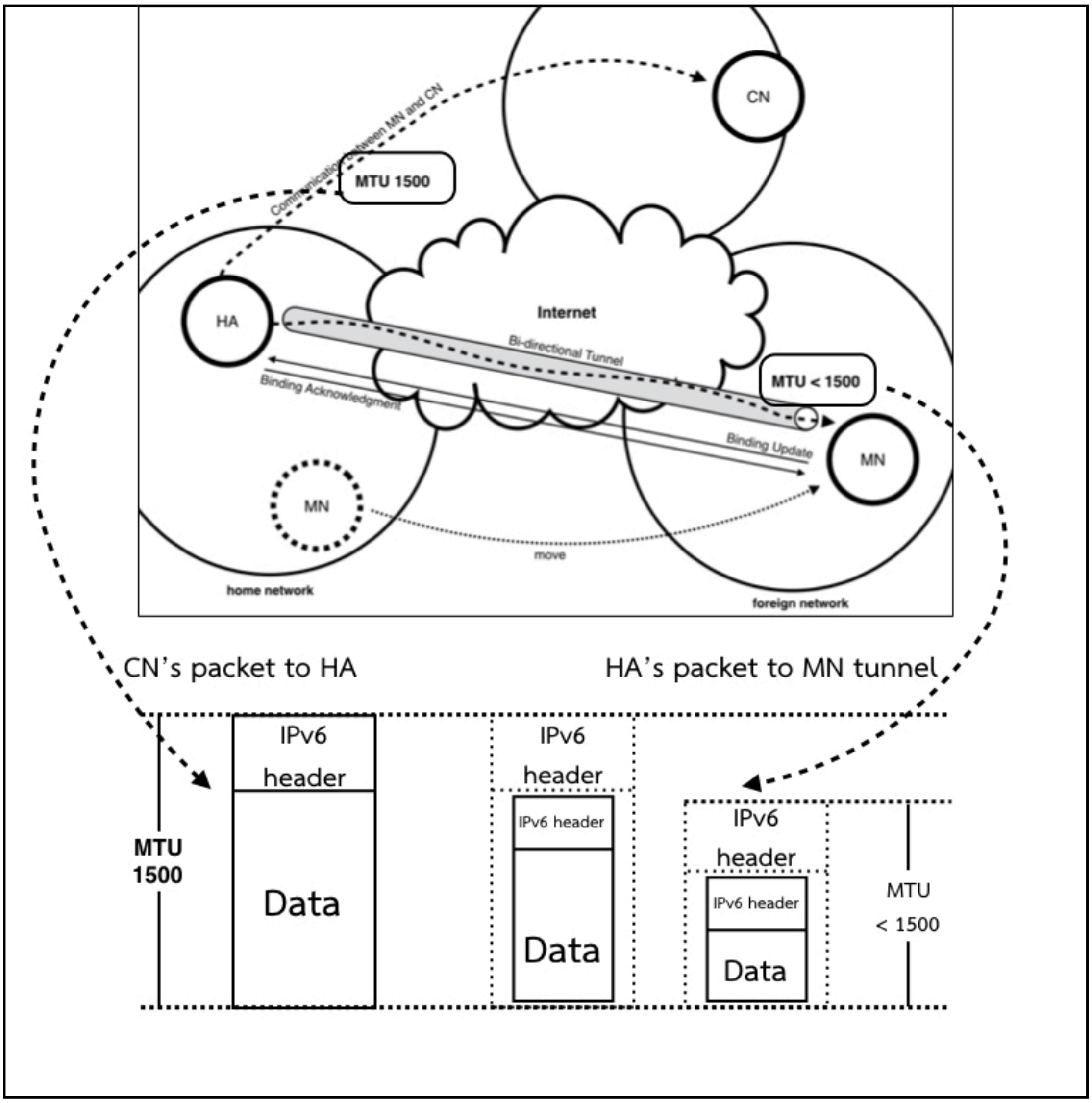


Figure 3.3 Fragmentation packet problem.

3.2. A large binding cache on CN

When an MN is away from home it can use route optimization to contact a CN. The CN must remember the CoA of all the MNs in contact with it. That could make the binding cache of the CN large, especially if the CN is a popular server as shown in figure 3.4. This might cause the CN to refuse to use route optimization, in which case the MN has only triangle routing and increased delays, as a solution.

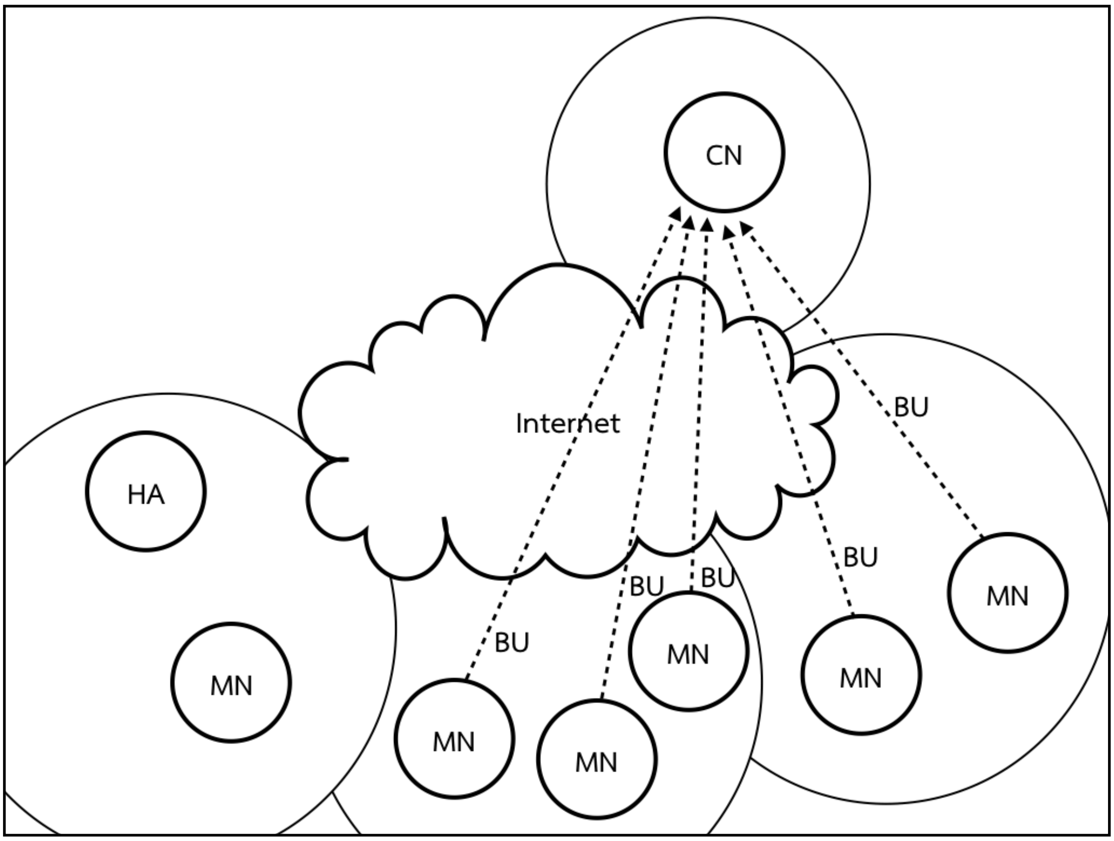


Figure 3.4 Binding cache of CN get large.

3.3. Proposed Solution

We observe that there is no significant cost to using Mobile IP when the MN is at home (shown in figure 3.5), only when it is away from home does the triangle route, or route optimization cost appear. So a solution to the problem would be for the MN to always be at home. That is, whenever a MN moves, its new location becomes a new home if possible. This requires the MN to find and associate with a HA in its new network. The MN will then have multiple HAs and home networks (shown in figure 3.6).

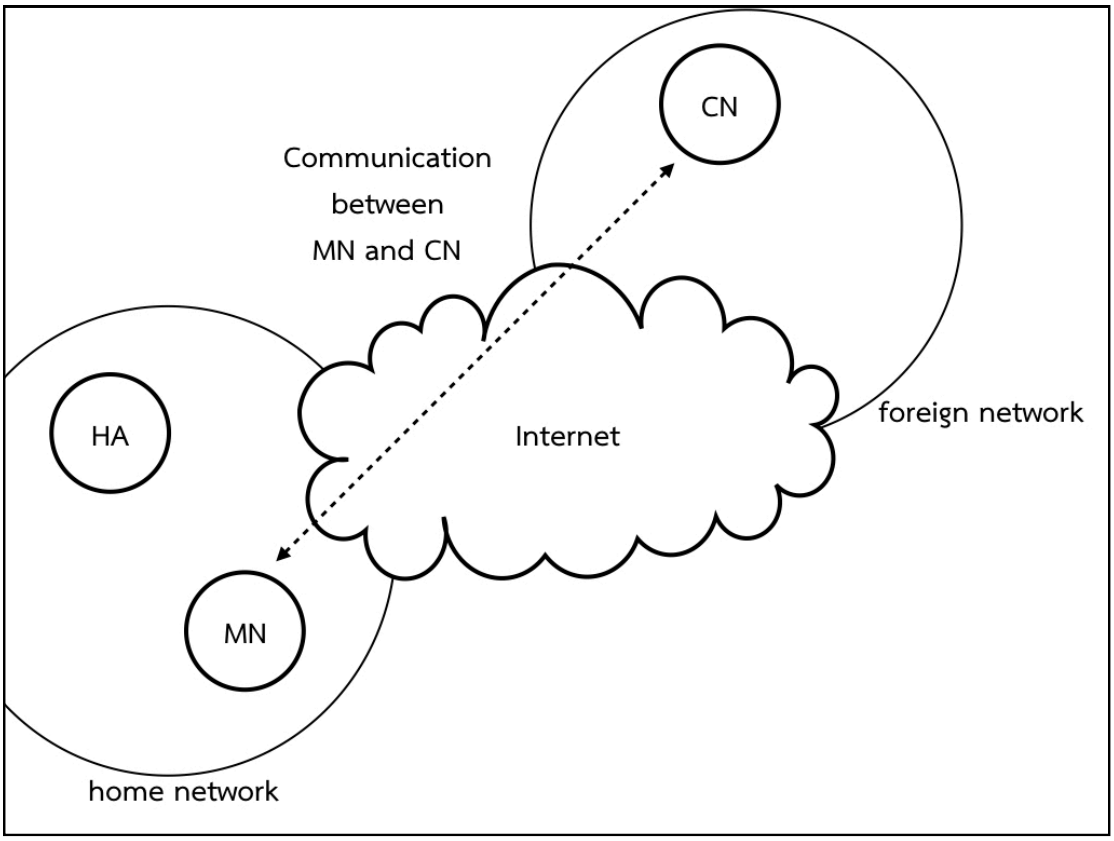


Figure 3.5 The MN can communication directly with CN at home network.

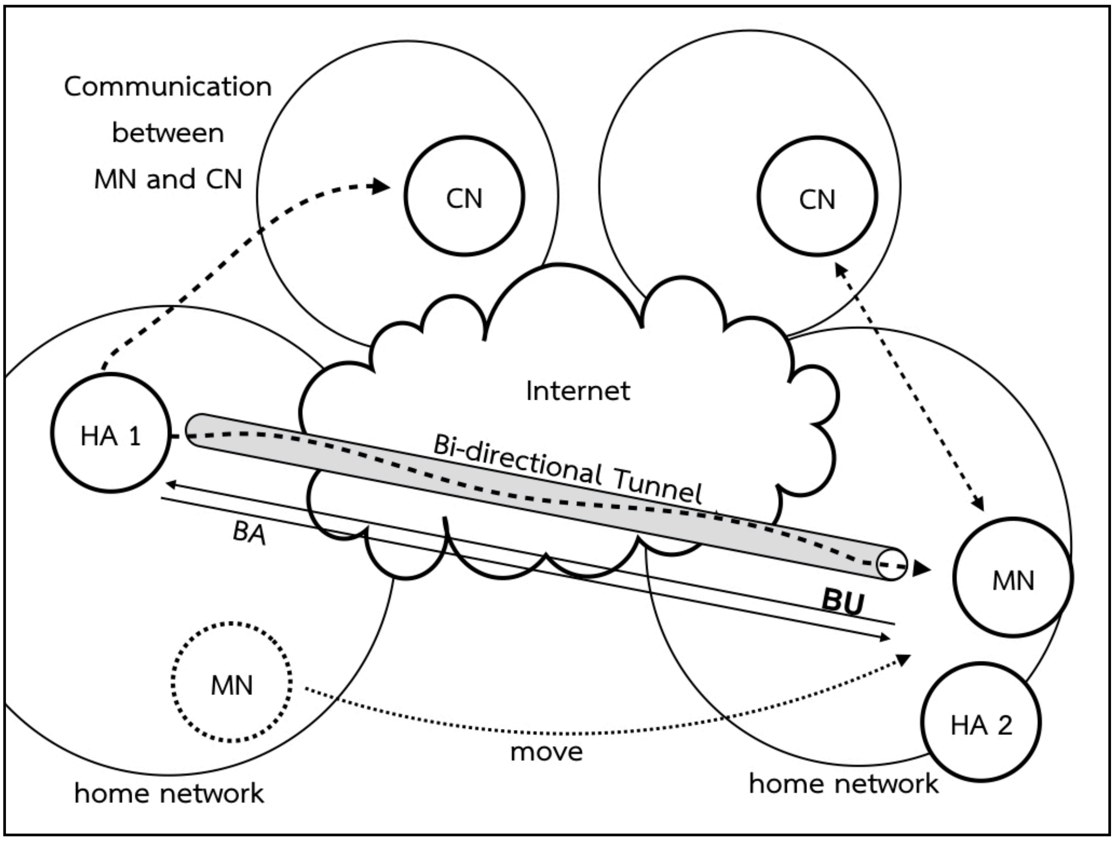


Figure 3.6 The MN has multiple HAs on multiple home network .

To do this the MN must first locate a HA on its new network, and then configure an association between itself and the HA. Current implementations involve people in this process, for our scheme to work everything must be automated. This includes finding a HA, obtaining its agreement to serve as HA for this MN, and configuring the required security association between the HA and MN.

Once the MN is able to add new HAs as required it might eventually obtain many HAs. Eventually some of those will be serving no useful purpose and should be removed. We need to find a way to detect HAs that are no longer of any use so they can be deleted.

Not all networks will have a HA available for the MN to use. In that case Mobile IP will work as it has before, including the extra costs. But now the MN has a new problem. When it initiates communications it needs to use a home address, but it now has several available - which is the best for it to choose? We need to seek an answer to this to provide guidance to MN implementors - while any choice would be functional, an optimal choice would be better.

3.4. Multiple HAs on single home network

A HA is required for the MIPv6 protocol [5]. The HA has three important functions as follows:

1) To process registration requests.
2) To maintain MN's location.
3) To tunnel packets forward between HA and MN.

When a MN uses the mobility function of MIPv6, the HA is an important element of this service. If the HA is dead or connection is lost between MN and HA, the MN can't use the mobility function of the MIPv6. If the MN is relying upon the mobility functions it would become isolated from the network. That is, unless the MN can find a new HA, or if it remains connected to its home network.

A single HA for a MN causes of performance bottlenecks (shown in figure 3.7) and is a single point of failure (shown in figure 3.8). When a HA provides service for various MNs. It can cause performance bottleneck problem such as high CPU load and traffic on HA and HA's network. The second problem is single point of failure. When the connection between a HA and a MN is lost, for example if a HA is dead, or the HA's network is down. Then the connection between MN and other nodes will fail. So MIPv6 can't work without a HA.

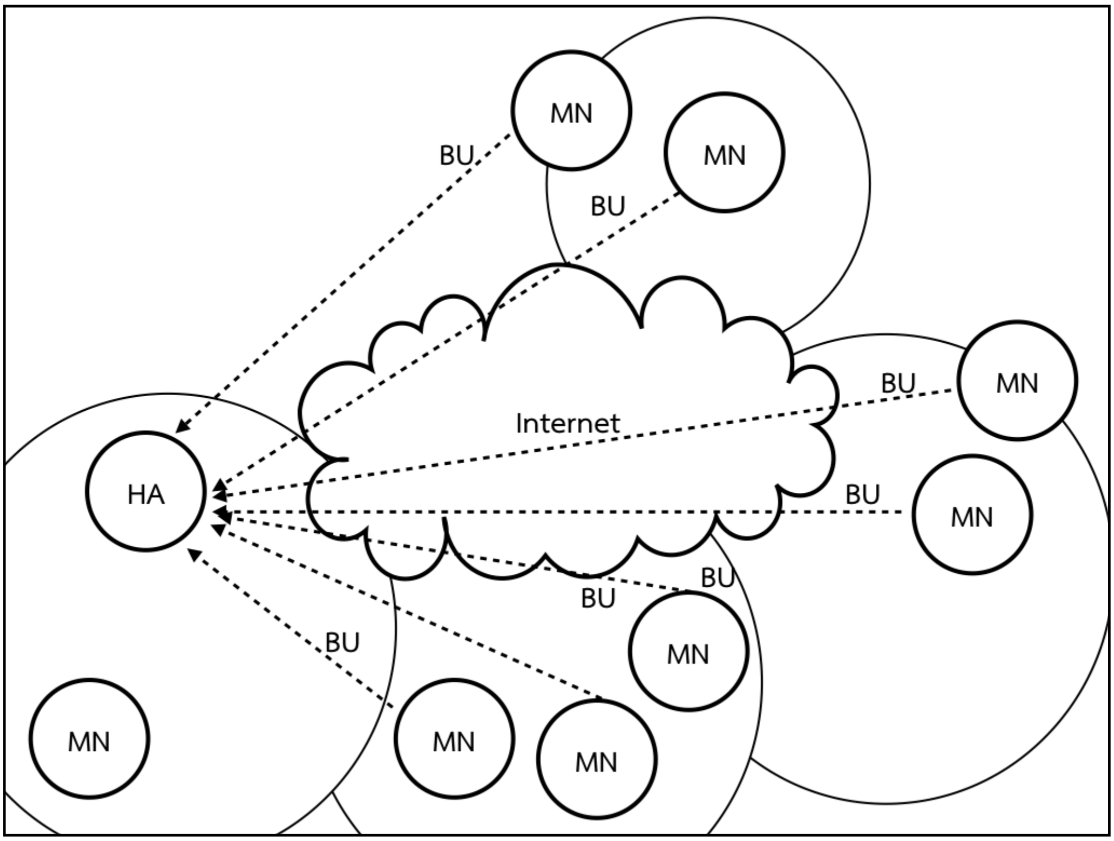


Figure 3.7 a performance of bottlenecks on MIPv6

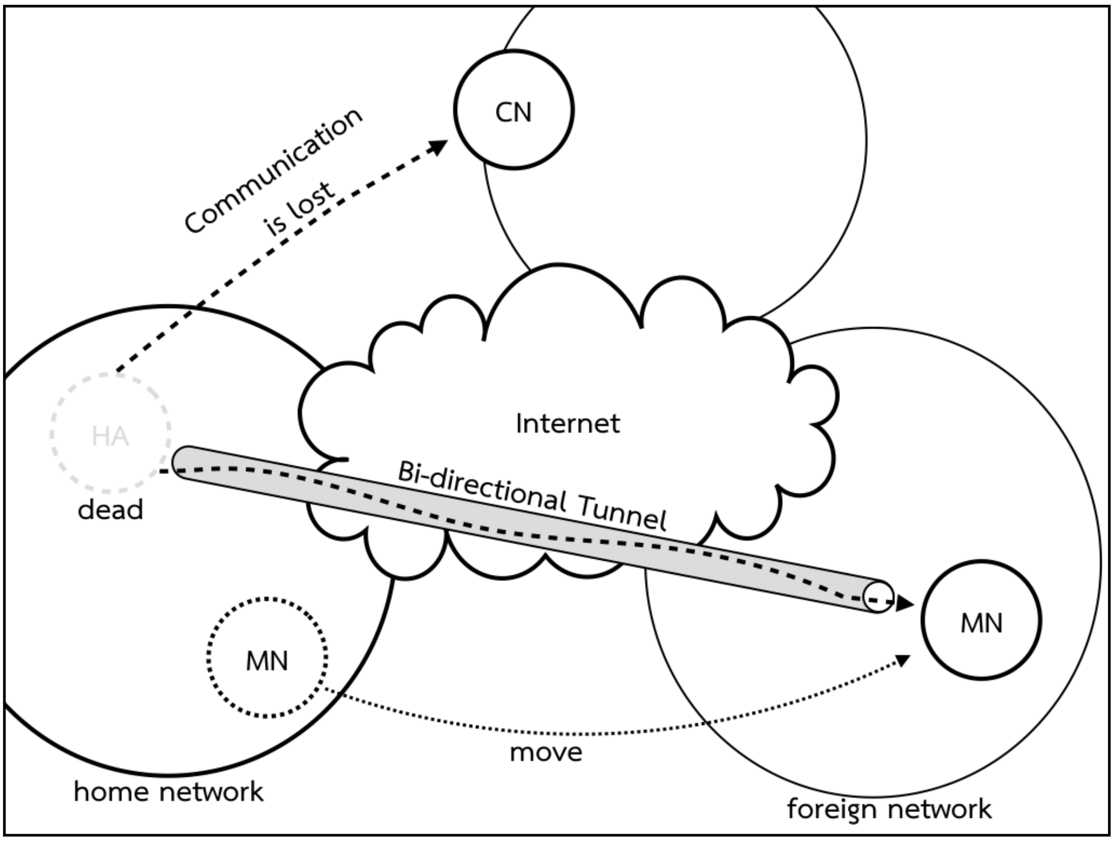


Figure 3.8 a single point of failure on MIPv6

This section surveys the literature regarding other proposals that utilize multiple HAs. The primary focus of this work is the HA, dealing with reliability (movement detection) and performance (handover delay reduction) issues, unlike our proposal where the primary concern is the MN, and the use of multiple HAs is just a tool to reduce the cost of Mobile IP for MNs. Other proposals have multiple HAs on only one home link. But we proposed multiple HAs from all visited networks of the MN. The proposals described here are independent of our work, and could usefully be used in conjunction with our proposal.

3.4.1.Single point of failure

First we describe proposed mechanisms in the area of multiple HAs for backup or redundancy. If the HA associated with MN is down the standard MIPv6 protocol takes a long time to find a new HA and resume communications. This is because the MN dosen't know the HA is down until the MN sends a registration or binding update to the HA. The solutions proposed in [20][21][22][23] for this single point of failure problem use the Dynamic Home Agent Address Discovery (DHAAD) [5] protocol. T. Narten, E. Nordmark and W. Simpson [6] describe an improved Neighbor

Unreachability Detection. We call that ICMP Home Agent Unreachability Detection. This protocol improved HA discovery time of MN. First, the HA finds another HA in the same network is unreachable via the route. So the HA put IP address of unreachable HA to the ICMP Unreachability Message and notifies the MN. Second, the MN get the ICMP unreachability message, then the MN find a new HA on the home network. When a MN knows a HA has disappeared, the MIPv6 provides the DHAAD protocol to find a new HA. But disadvantages of this protocol are

1) Increased processing delay.
2) Extra signaling overhead.

For active and backup HA the solutions proposed in [21][24][25][26][27] provide a MN the means to keep continuously working using the MIPv6 protocol while a HA is unreachable. The active HA manages registration from the MN and synchronizes with a backup HA. Disadvantages of this solution are

1) A registration packet of MN is low latency.
2) A significant of level overhead.
3) It causes a service interruption between detection and recovery in the case of HA failure.

H.Ahn, CS.Hwang. [28] proposed a stable storage solution for solving the registration synchronization problem. This solution is based on the assumption that the stable storage is silent. But it has a delay in HA failure detection and a service interruption problem.

- **Simultaneously Update Home Agents**

H. Shi and S. Goto [29] proposed a new mechanism of multiple HAs. The MN has more than one HA and sends BU messages simultaneously (Figure 3.9). When the MN sends a BU, the MN will simultaneously send it to both HA. When one of the HA goes down or fails the other takes over its workload, and a CN which is sending the data can immediately send via the HA that is not down.

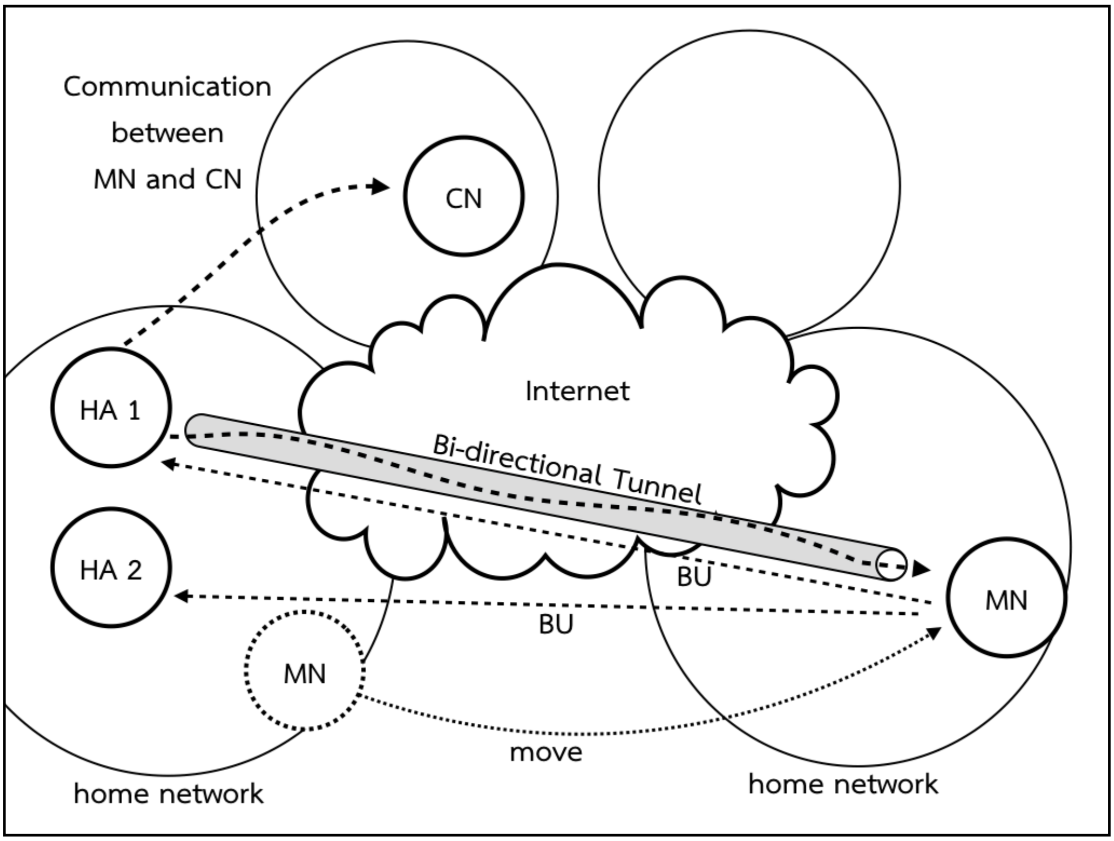


Figure 3.9 multiple HAs with binding update to all HA

This mechanism allows for fast HA recovery. but a limitation of this mechanism supports sending BUs to multiple HAs simultaneously at the cost of the extra traffic required. The BU format is altered to support this, and a new ICMP “Home Agent Unreachable” error message is required.

- **Home Agent Reliability Protocol**

In 2009, IETF proposed a Home Agent Reliability Protocol [26]. The Home Agent Reliability Protocol supports multiple HAs for MIPv6 (Figure 3.10). The MN sends a BU to the active HA only. The active HA and standby HA will synchronize information. If the standby HA discovers the active HA has failed, it sends a message to each known MN to inform it. And when the MN discovers the active HA is unavailable, if no reply is received to a BU, or if it is told by the standby HA, it exchanges HA switch messages with the standby HA. If the binding information of the standby HA is valid it will change its status to active HA. If the binding information of the standby HA is not valid it will have to wait upon a binding registration from MN before changing to become the active HA.

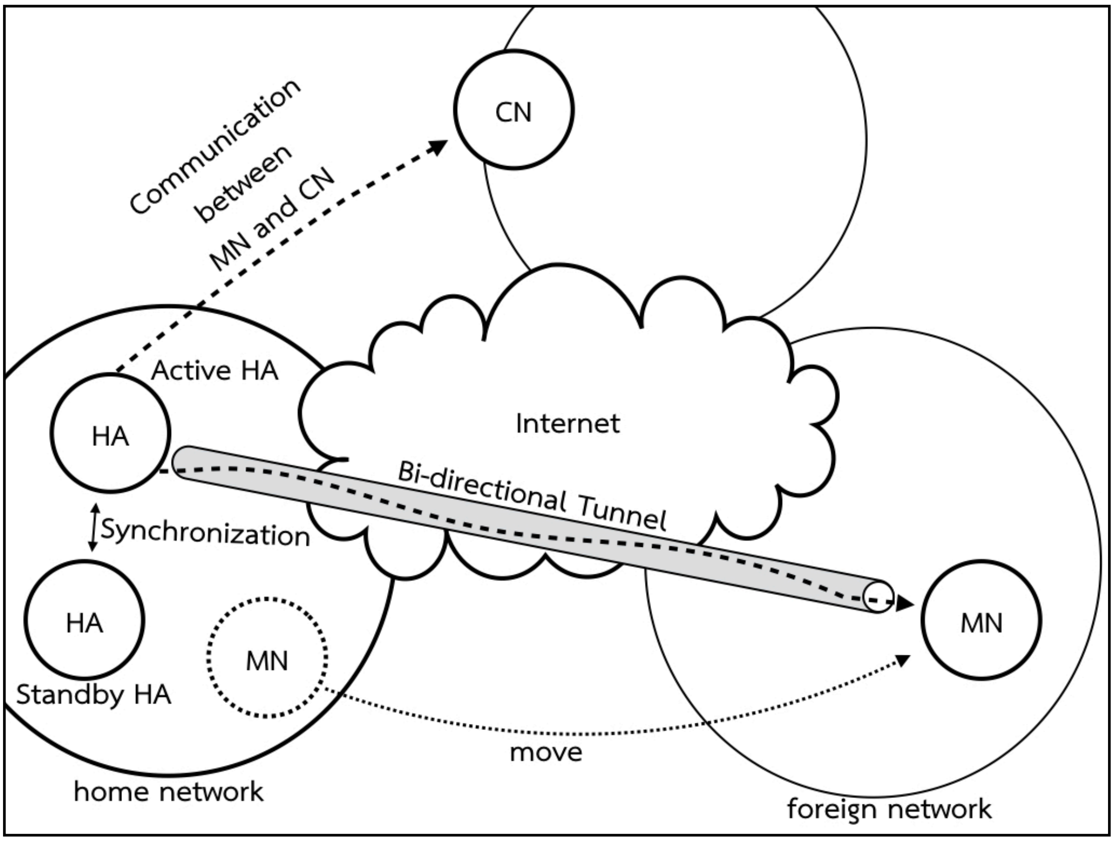


Figure 3.10 Home Agent Reliability Protocol

A limitation of this mechanism: the nature of the synchronization used leads to times when the active and standby HAs hold inconsistent information, similar to what can occur with primary and secondary DNS servers.

3.4.2.Load Balancing and Failure Detection Issues

Next we describe proposed mechanisms in the area of multiple Home Agents for load balancing and failure detection. A single HA of MIPv6 serves multiple MNs for their mobility function. If a HA has a lot of MNs, this can cause heavy CPU load, and much traffic on the home network. So F. Heissenhuber, W. Fritsche and A. Ried [30] proposed the HA redundancy and load balancing solution as a failure detection and recovery mechanism. This solution is transparent to the MN. Another proposal is the Virtual Home Agent Reliability (VHARP) [24][31][25] protocol. The VHARP protocol uses multiple HAs on the home network. All the HAs send a Heart Beat signal to inform the state of each HA. So the VHARP provides transparent failure detection, a recovery mechanism, a reduces message exchange and secures data transfer. In addition to it improves a overall system and a application performance. And it reduces the workload of HAs on a network. If more packets arrive at a HA than a set arrival rate threshold, then that load is divided to other HAs. Every registration request of MNs process by active HA every time. Moreover they improve the VHARP method by additional policy [9]. Those are timer and counter. The HA chooses policy amongst random, round-robin and shortest queue. And then A. Vasilache, J. Li, and H. Kameda [32] proposed double-threshold for load balancing. However, the VHARP method has disadvantages.

1) These methods are based on passive overload transfer. It does not react until arrival the rate of HA's packet is more than the threshold. This solution can't protect from the bottleneck problem.
2) The MN can't define the threshold value because VHARP works transparently to the MN. So it causes processing delays and overhead problems.
3) The VHARP with double threshold need HA's load information on home network. Because each a threshold has a information and value to keep a number. So more threshold and HAs to increase signaling overhead and synchronization problems on the home network.

- **Multiple Home Agents Deployment Scheme**

A Multiple Home Agents Deployment Scheme (MHADS) [33] enhances service availability for MIPv6, as shown in Figure 3.11. This mechanism enhances service availability and improves performance, when the home network is overload, besides, to occur to HA failure. The edge router of the home link is extended to support the Balancer and Monitor (BM) function. The BM function acts as a balancer. It is responsible for protecting HA overload by choosing the best HA for each MN's registration. Fault-tolerance, as a monitor and detection HA failure, its takeover and HA failure recovery.

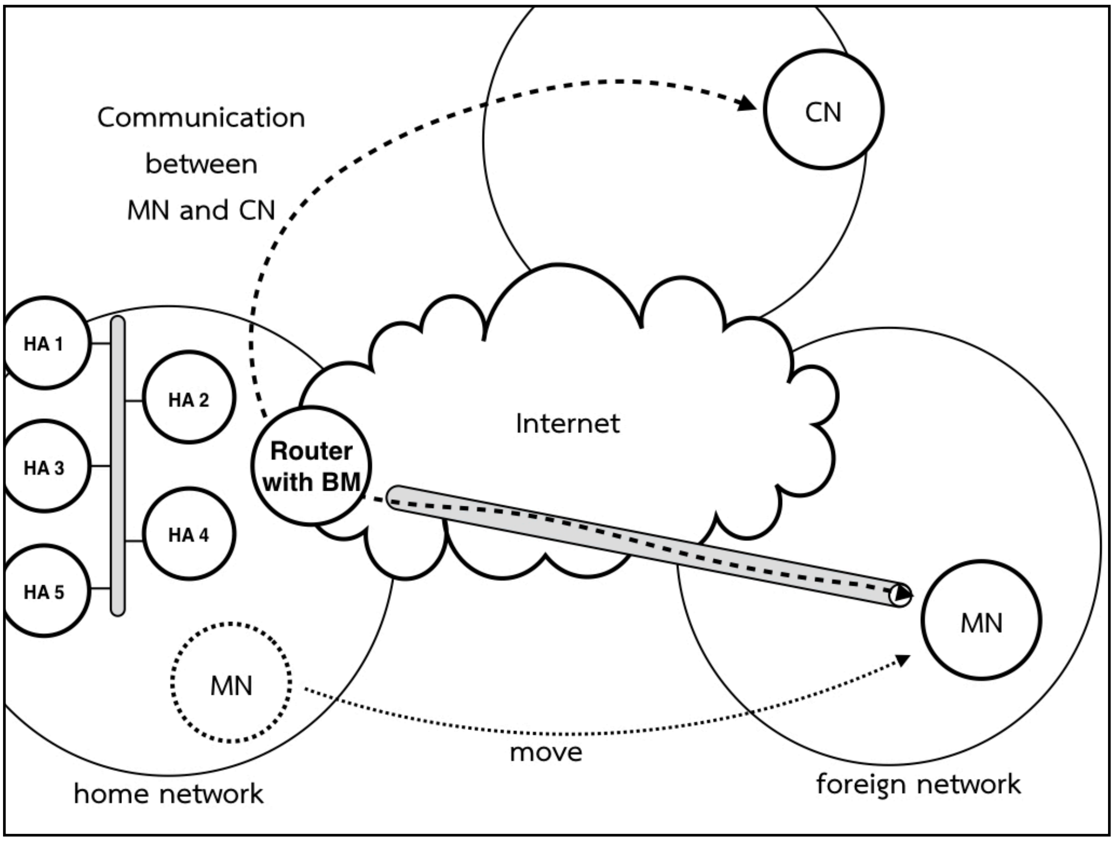


Figure 3.11 MHADS Architecture

CHAPTER 4

DESIGN AND IMPLEMENTATION

Chapter 3, we showed the cost of Mobile IP when an MN is away from its home network. Also, we showed that there is no a cost of Mobile IP at the home network. So we proposed Mobile IPv6 with multiple HAs and home networks to reduce the cost of mobile IP. The standard Mobile IPv6 from IETF [5] only supports multiple HAs on one home network.

When a user uses Mobile IP, they need assignation of two variables for configuration of the Mobile IP program [34]. First, it’s a home prefix of the network. The home prefix allows discovery of HAs for the MN on the home network. Second is home address (HoA) of MN. The MN has a one home network. Thus it needs configuration only one time. However, we have multiple home networks. If the MN moves to a new network we assume that becomes a new home network. Then the user needs to configure a home prefix and home of address for this home network.

Moreover, the user repeats this method every time the MN moves. This situation is complicated and not practical. Perhaps the user doesn’t need it. So they came back to standard Mobile IPv6 with a cost. So we proposed a new protocol for resolution above problem. A requirement of the protocol is

1) To find a HAs on the local network automatically. If anything is needed here it needs to be something different.
2) To arrange for a HAs. An MN’s multiple HAs to need security between MN and HA. Because we can’t trust a foreign network unlike the true home network, thus we need a method for verification and authentication between MN and HA.

Therefore we design the “Automated HA configuration for MIPv6" protocol for a resolution of that requirement. First section we describe algorithm of this protocol. The second section we introduce the KAME software for mobile IP operation. It is open source software. So we adjust the KAME [34] for the Automated HA configuration for MIPv6 protocol. So we prove the protocol can be possible.

Finally, this protocol doesn’t concern session alive between HA and MN. Also, it doesn’t need immediate fault detection and binding update. We make a new connection with new HA to reduce the cost of MIPv6. Moreover, the MN can keep a connection from the previous network. So we get an advantage of protocol after finishing the procedure.

4.1. Protocol Design

We divide three-part of the protocol as shown in figure 4.1. A node's packet doesn't remember state show on dash line.

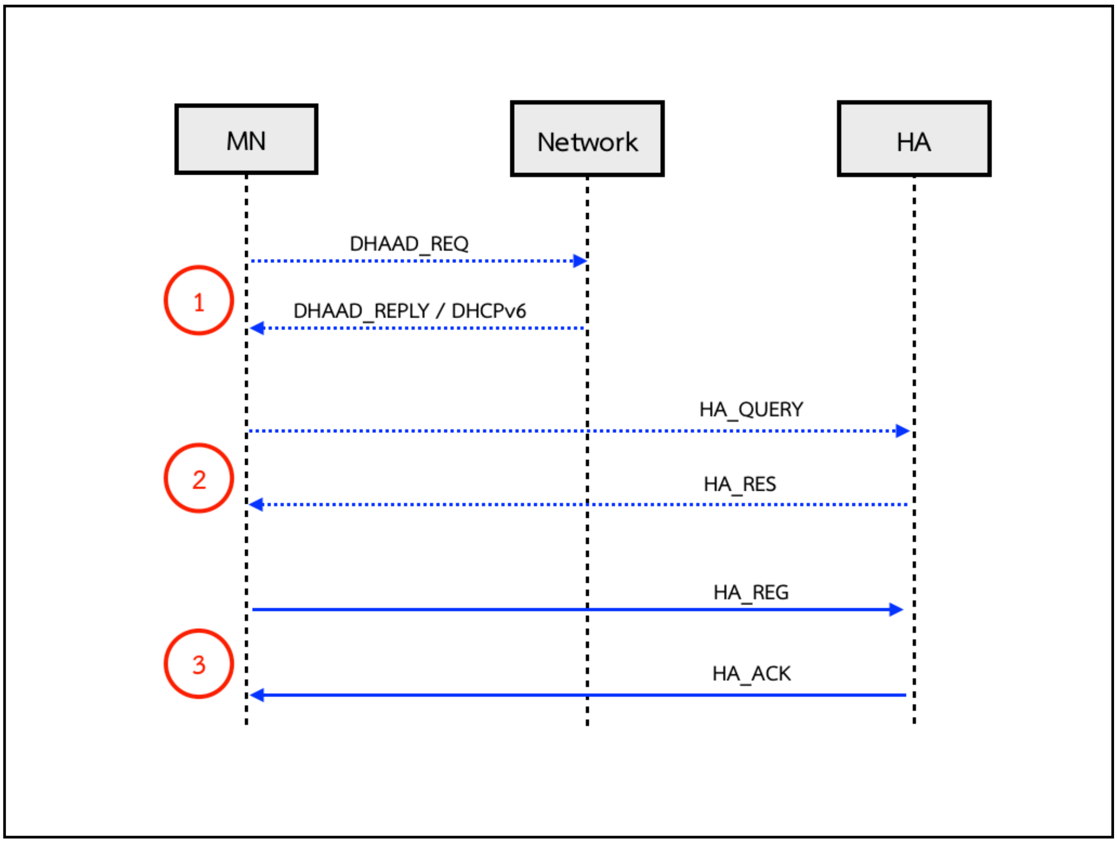


Figure 4.1 Automated HA configuration for MIPv6 protocol diagram

1) The MN discovery and request HAs from the network.
2) The MN is exchange information for HA's authentication and verification.
3) The MN establishes HA for HA's register and shares a secret key.

4.1.1.A MN discovery and request HAs from network.

When The MN moves to a foreign network the MN obtains a new IP address for identification on the internet. The MN has two methods to get an IP address from networks. The first is to use the Dynamic Host Configuration Protocol for IPv6

(DHCPv6) protocol from DHCPv6 server. The second is using the Neighbor Discovery (ND) protocol. When the MN goes to a new network it gets a new IP address (CoA). The MN sends a BU to HA for notice about new IP address. If a network provides state full IPv6 address from DHCPv6 server, the MN can get an HA address list from DHCPv6 protocol.

But not all networks provide a DHCPv6 server. The MN uses Neighbor Discovery (ND) to fill an IP address using stateless method. We pick a DHAAD protocol from standard MIPv6 to find HAs address list. Usually, DHAAD allows the MN to locate its HA in case its address might have altered (or the original HA has been replacing for one reason or other by a new one). So the standard MIPv6 send a DHAAD protocol back to the home network to find new HA. Where the HA already knows the MN, and it knows that HA. However, we use a DHAAD protocol to find HAs address list on the current network. The HA never know the MN before. Also, the MN doesn't even know if any HA exists on its current network. However, a DHAAD allows the MN to find any HA that exists on a network. After that, the MN creates HA anycast address from MN’s prefix. Then the HA replies with a list of HA addresses to the MN.

The HA discovery could also be complete by DHCPv6. So the Service Location Protocol (SLP) could be another method. Which method is used for multiple HAs purpose. The reason we concentrate on DHAAD for this purpose, because it is part of the MIPv6 protocol set, so any HA that exists understand DHAAD and respond. Since we are using the MIPv6, it is more rational to depend upon MIPv6 protocols, than rely upon other unrelated protocols.

If a MN doesn't reply with a DHAAD_ACK packet from HA’s network, and that packet is request times out, then the MN repeats its request until the MN moves. In case where the MN doesn’t receive any a DHAAD_ACK, it can be possible that the network does not have HAs service. So we need to use standard MIPv6 to continue working.

4.1.2.MN information for HA’s authentication and verification.

When the MN receives a HA’s address from DHCPv6 or DHAAD_ACK, the MN sends a HA_QUERY packet to request MIPv6 service from each HA. The purpose of packets are

1) Request, A packet indicate HAs for MIPv6 service.
2) MN’s authentication, A MN send information to HA for authentication. However, only information is public (not private).

The HA reply a HA_RES packet to MN. The HA doesn’t remember any state of the packet. In the case of the MN doesn’t send the information again perhaps the MN didn’t choose that HA. It might do this if other HAs respond faster or make less information requests. So the MN can send multiple HA_QUERY packets to the many HAs at the same time. Also, a network provides multiple HAs.

When the MN gets a HA_RES packet, it needs to examine and validate information from the HA. Validation is done using the same method as Secure Neighbor Discovery[18], described briefly in section 2.5, uses to authorize a node to be a router on the network. The HA includes its certificate in the HA_RES packet, along with other data, and signs the packet with the private key associated with the public key included in the certificate. The MN uses the same trust chain as is used by SEND to authenticate routers, except with different, or perhaps additional if the HA is also a router, authorizing data in the certificate. Once verified, the MN is certain that the HA is an authorized HA on the local network.

If a result of validation is failing, or the answer is “no”, we can ignore a packet from HA. Because answer “no” is mean empty data and empty a certification. We are concern answer and validation information from HA. We do want to know is that the HA authorized by the network. Because the HA that offers service for the MN will be forwarding all of MN packets, and seeing them unencrypted (encryption runs between MN and HA). They try to emulate a HA_RES packet from HA. So the malicious HA could do anything it likes to our data (man in the middle attack). Finally, the MN ignore it. Second the HA we discovered earlier might not understand this new protocol, and not reply at all. Which the MN treats the same as it saying "no". So the MN wait for an acceptable answer reply HA’s a HA_RES packet.

If the MN can validate the HA’s authentication from the HA_RES packet, then we need to configure encryption data between HA and MN to protect from a malicious user man-in-the-middle attack, because we send private information after validation succeeds. The MN can choose an encryption algorithm from those offered in the HA_RES packet. Actually the method algorithm uses is like the web and the HTTPS authentication [36] on a web browser. We have a client that needs to trust that a server is what it says it is before a client sends it any private information.

However, we have a different problem. In the web, what matters is that a client is connecting to the server (web server) that the user selected. It's complete by that server having a certificate, signed by some certificate authority that we trust, that relates the name of the site with the server claiming to be that site. However in MIPv6 we don't care what anything is a called, and so a web server type certificate is useless. We want to know is that the real HA that belongs to and has been authorized by, the network we currently connected. That has been authorized to be the HA on that network. It is not just some random host that happens to be connected and is pretending to be the HA as a MN connecting to this network, we are already trusting the network's management to handle our data. All our packets to the world are passing through that network's routers. Therefore, we need is a method by which we can validate that the network manager is authorized the HA. That is the same issue as making sure a router on the network authorized to be a router and has been authorized to act that way. Fortunately, the secure neighbor discovery has already solved that problem, and we can copy the method used to validate a system as the HA just as quickly as validating it as a router. So that's what we will do.

We show an algorithm for symmetric encryption key between HA and MN at figure 4.2.

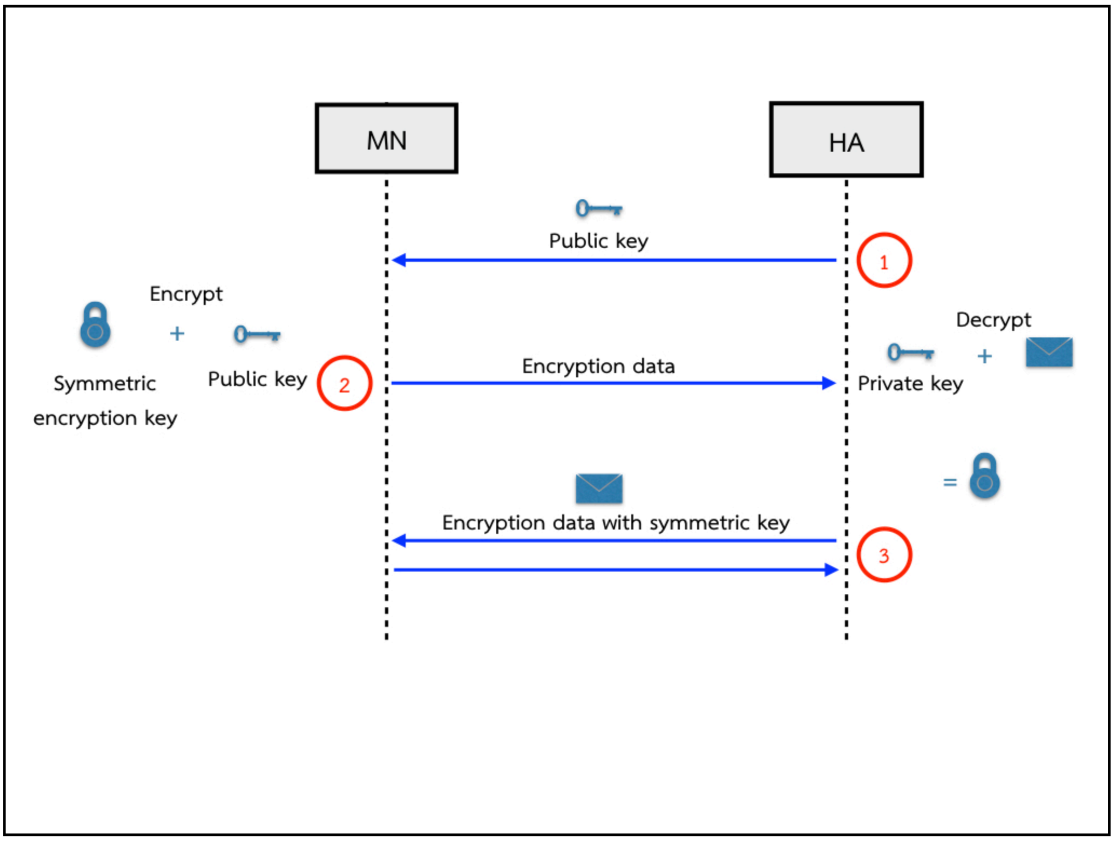


Figure 4.2 Encryption data with symmetric key

1) The MN gets the HA’s public key. It is the same one from the certificate that the MN used to validate the trust of the HA. So the MN know can

believe that HA. Then only the trusted HA able to decrypt the MN’s data. Also, HA's public key is the one being used to encrypt by the MN when it is sending the symmetric encryption key in the next step.

2) The HA's public key includes a list of symmetric key protocol that it supports, so the MN can pick one that both HA and MN understand. Then the MN uses the HA’s public key to encrypt a random symmetric encryption key and sends it to the HA.
3) The HA decrypts the symmetric encryption key by HAs private key. Also, then all communication between HA and MN use encryption with the symmetric key.

In the case of HA’s answer is “YES”. The MN’s information is enough in a HA_QUERY packet. We can share the secret key to encryption data between HA and MN. Also, then we go to a HA_REG packet next step.

Not all networks provide HAs service for everyone. The network no HA can be found. None the MN is willing to trust the HA. No HA that the MN trusts are willing for the MN we can’t to act as the HA for the MN, we can't set up the HA on the network we are visiting, and normal MIPv6 applies. Other cases, maybe they provide HAs and MIPv6 for member only. Maybe if you pay money, then they provide a service. The second case is “MORE INFORMATION” answer. Because the HA request a piece of secret information, we can authenticate username and password for a member of the network. Alternatively, we can use credit card information to pay for HAs service. HA can reject the MN's information. If the information is wrong. The HA should inform the MN. They don’t repeat the information. So this is a reason to encrypted data between MN and HA before exchange information. Actually, the HA doesn’t remember state both packets HA_QUERY and HA_RES, a later HA_QUERY imply includes extra information requested by an earlier HA_RES, and will be encrypted.

4.1.3.A MN establish HA for HA’s register.

We have the second half of the problem, and once we trust the HA is correctly authorized, then we need secure communications with it. We might need to send private data to the HA to allow it to make its policy decision whether or not to act as the HA for us. Also, once the HA - MN bond is formed, we need a secure encrypted tunnel between them (that is just a standard MIPv6, or MIP v4 or v6, operational requirement). Finally, the MN sends a HA_REG packet to HA to register. Also, the HA should reply to a HA_ACK packet to MN.

Moreover, then everything between MN and HA are encrypted with a tunnel. When the is MN away from home the HA captures and forwards any packet to the MN shown in figure 4.3.

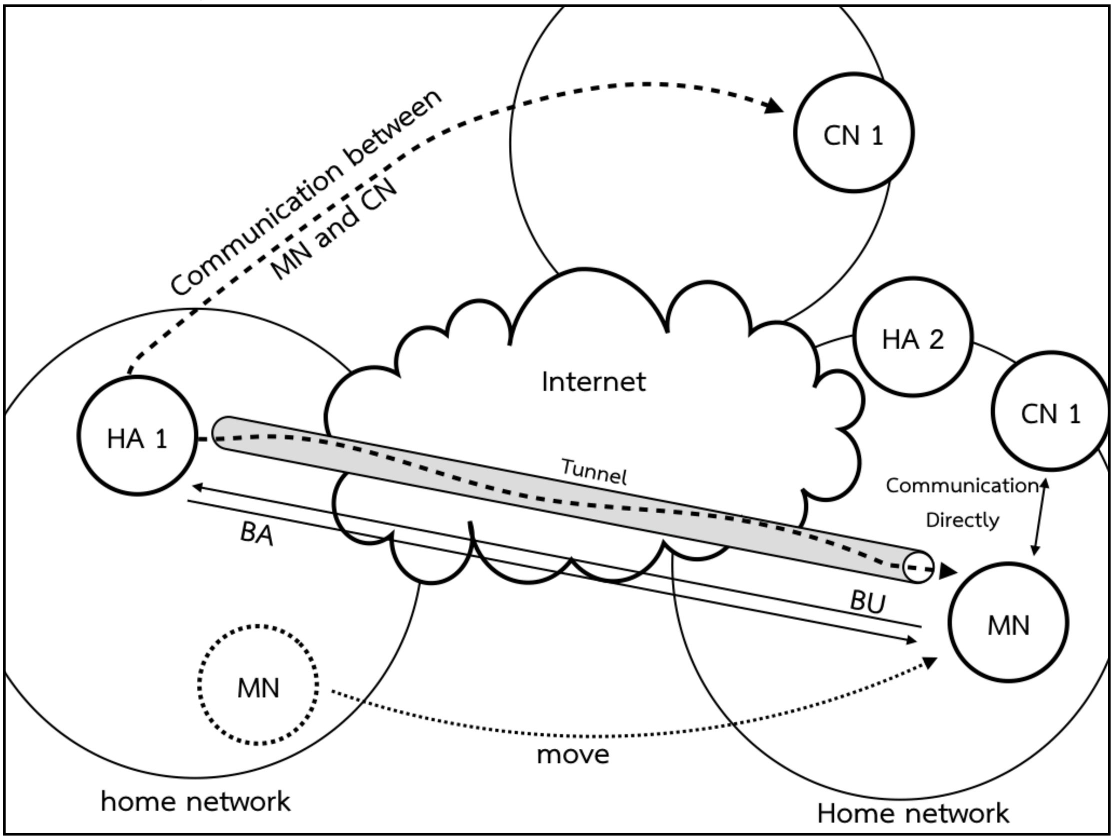


Figure 4.3 a secure encrypted tunnel between HA and MN.

4.2. Implementation

In this part, we show the implementation of the Automated HA configuration protocol in a real scenario (not a simulation). We implement that based on KAME software. So we demonstrated that our us protocol is practical and possible under the conditions as follows:

1. The MN can enable multiple HA and multiple home network.
2. The MN communicates with new CoA and keeps communication with each network. So the MN can send BU to each HA.

We implemented the DHAAD protocol in mnd and babymdd process, it is executed whenever babymdd has detected that the MN has moved, and obtains addresses of home agents on the new network to which the MN has connected.

We also implemented our new the Automated HA configuration protocol in mnd and babymdd protocol. When a HA is located on the new network this protocol configures an association between the MN and newly discovered HA. Our implementation omits the security aspects of the protocol - there is nothing new in these parts, they would simply be copies of code from SEND or web browser authentication (SSL). So we are known to function correctly. Because our implementation is entirely based within a private virtual network, not combated to the real internet, there are no actual threats to be avoided.

4.2.1.Overview of the KAME

The Automated HA configuration for MIPv6 that is required by the implementation is a modification to the existing SHISA mobility stack [34] on BSD [44]. Fortunately, they [34] develop KAME project to provide MIPv6 on IPv6 node for BSD[44] operating system. Also, KAME project is provided as open source. So we can develop it to use for our project. SHISA is implemented on top of the KAME IPv6 stack[37]. So the SHISA implementation provides the functions for Mobile IPv6 to function as MN, HA and CN.

SHISA contains several user space programs and a modified kernel shown in figure 4.4. Also, table 4.1 shows the programs of the SHISA stack [37]. We are interested in the mnd, had and babymdd programs. The babymdd program provides a simple movement detector for the MN. It sends a signal to indicate that the MN moved. The had program provides the HA functions and manages the signalling processing on the HA side. The mnd program provides mobile functions to the IPv6 node.

Table 4.1 SHISA programs

Program	Function
mnd	The mobile host functions
had	The HA functions (for MIPv6)
babymdd	A simple movement detector of MN
cnd	The CN functions.

The information of the interface is to work in kernel and user space programs. They [34] provide the new implemented Mobility socket domain (AF_MOBILITY). Also, the mechanism and message formats are likely the Routing socket [45] to exchange mobility-related information events between user space programs. The socket is also used as a broadcasting channel from the kernel to user space programs. So the movement detection program (babymdd) uses this socket to notify other programs.

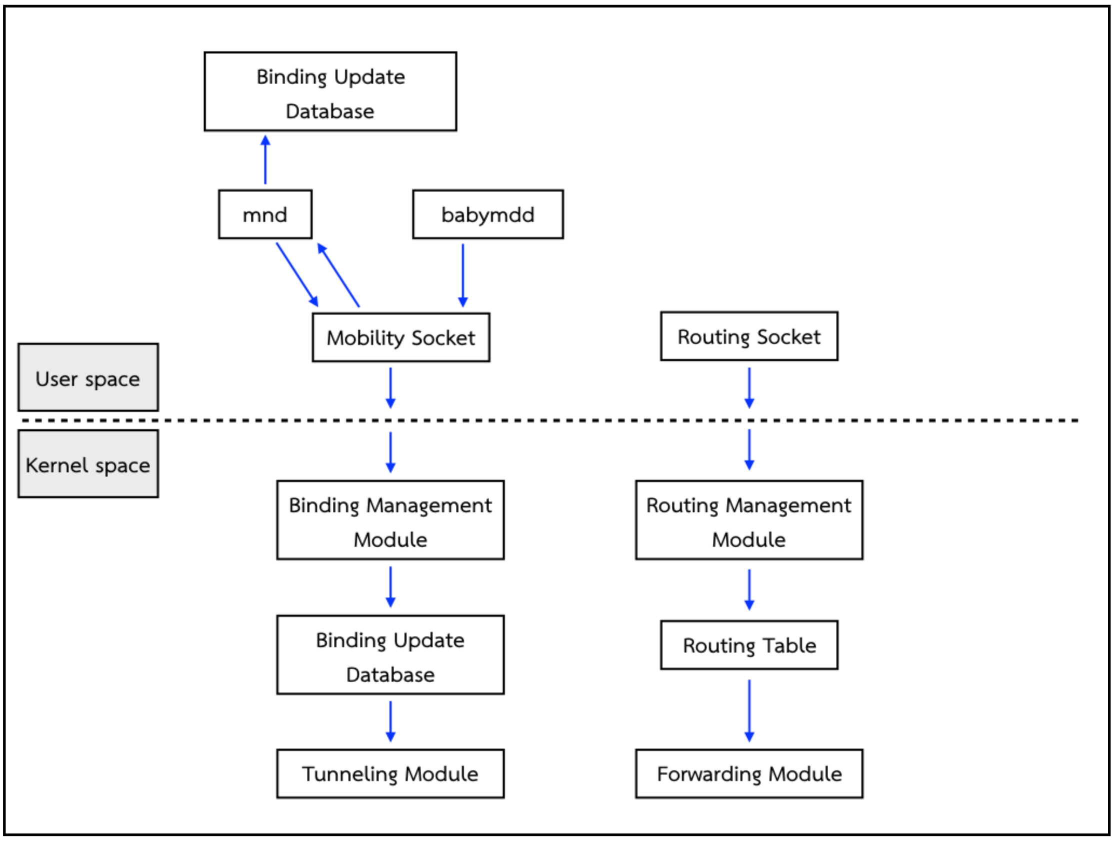


Figure 4.4 The relationship of the SHISA modules.

4.2.2.Sending and Receiving Signaling Messages

All the singling messages processes the user space programs. The Mobility Header takes the singling messages. To support this header format, SHISA implemented customary in the kernel and used the raw IPv6 packet. Figure 4.5 shows the output flow of tunneling packets on the MN when the MN sends a tunneled packet to the mip6_tunnel_output() function.

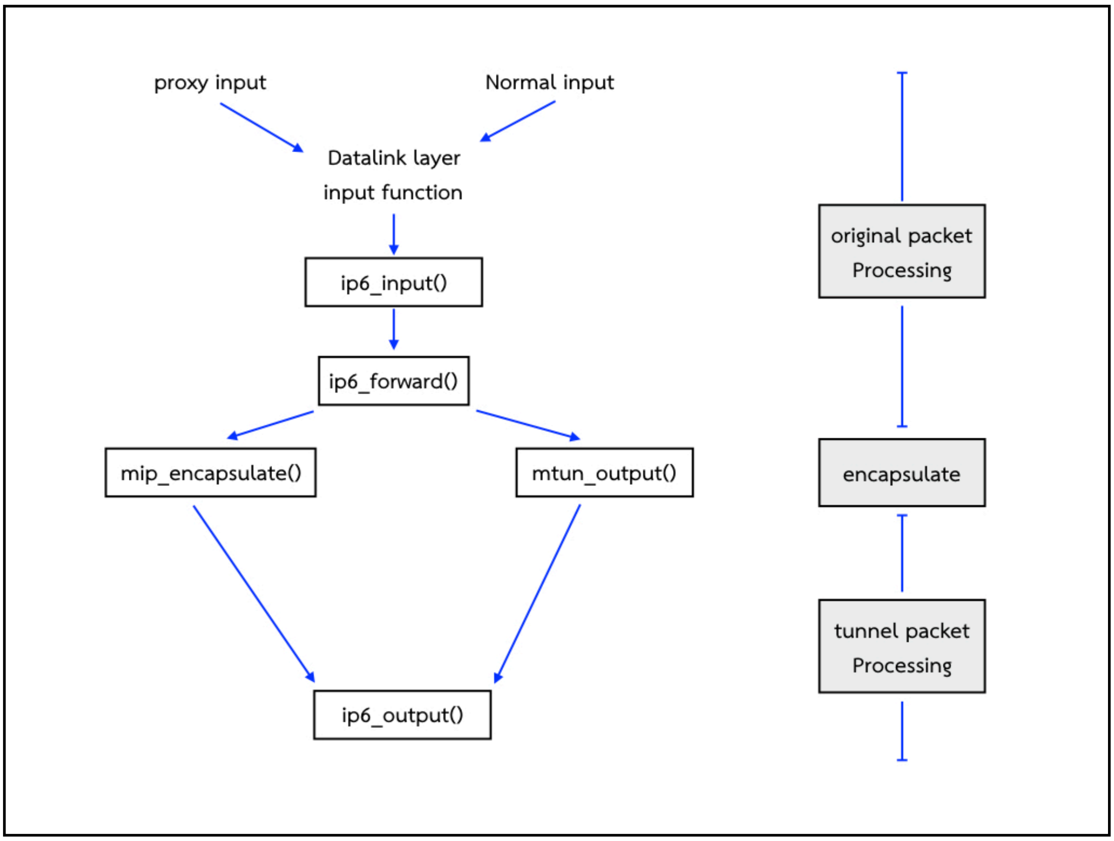


Figure 4.5 The output flow of a tunneled packet on a MN.

Figure 4.6 shows the input flow of tunneling packets on the MN. The tunneled packets sent to the MN itself are processed by mip6_tunnel_input(). For forwarding packets to mobile MNs, the mtun_input() function handles tunneled packets.

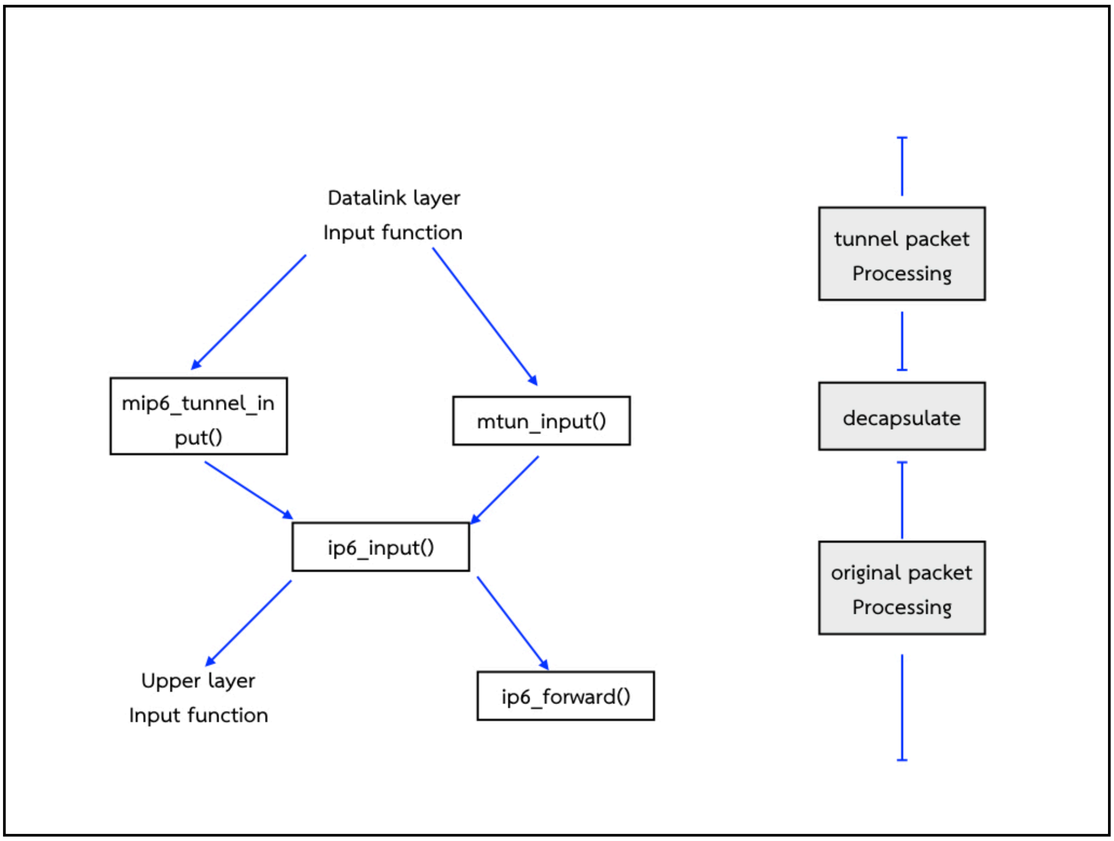


Figure 4.6 The input flow of a tunneled packet on a MN.

The HA intercepts the packets to the HoA of the MN. The proxy start after the HA receives a BU message for registration from the MN. The intercepted packets are forwarded using the tunneling mechanism. So then the packet sent to HoAs. Figure 5.7 shows the flow. The input processing of the tunnel packet at the HA is a simple forwarding process. The only distinction is are the packets input by function mip6_tunnel_input().

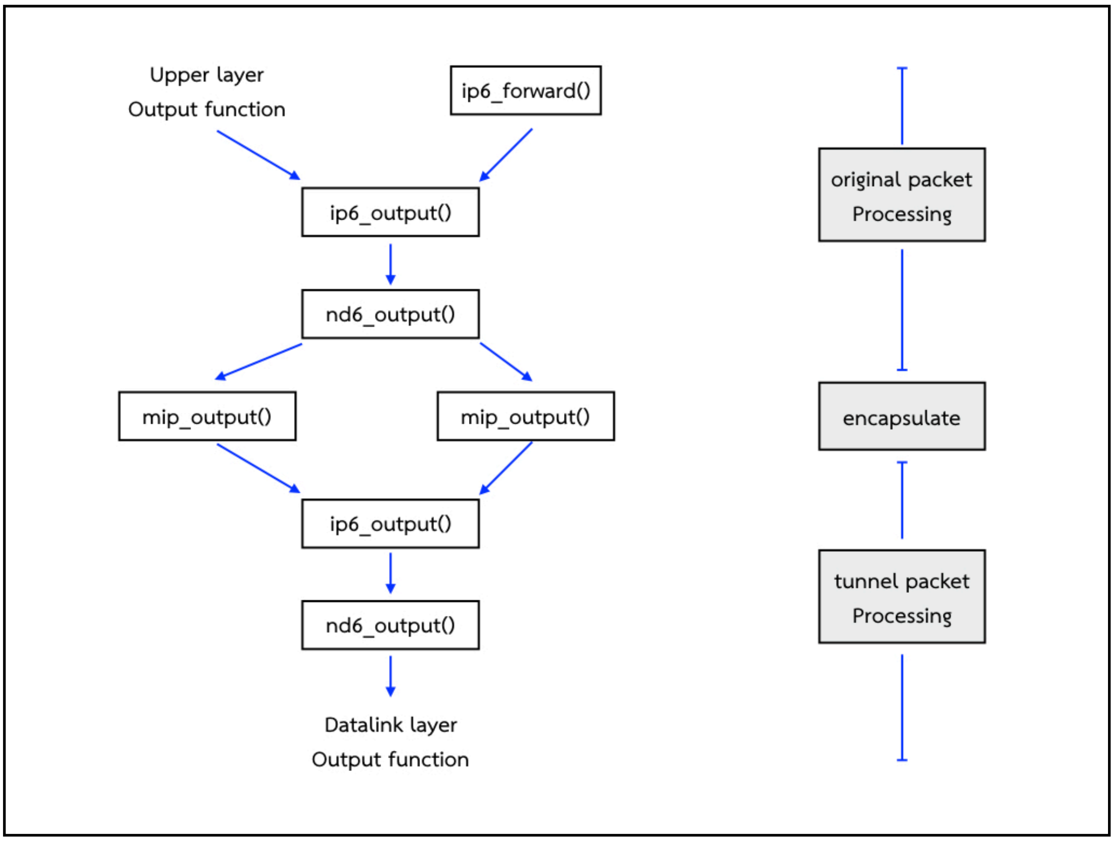


Figure 4.7 The output flow of a tunneled packet on a HA.

4.2.3.The process to control message and signal

In KAME, the two processes necessary to enable mobile IP on MN are mnd and babymdd processes. The babymdd process notifies the MN when node move (MN's IP address changed) via a signal to mobility socket in user space. Then it chooses a primary CoA (IP's new network), and it removes the last CoA address from the physical interface. The trigger of the mnd process is when it receives a signal from the babymdd process. The mnd process creates a tunnel for communication with HA, and it sends a BU to HA for notification about new CoA, when the MN moves to the foreign network. On the other hand, the MN deletes the tunnel connected to a HA when the MN returns to the home network. We show a flowchart of the mnd and babymdd processes in figure 4.8.

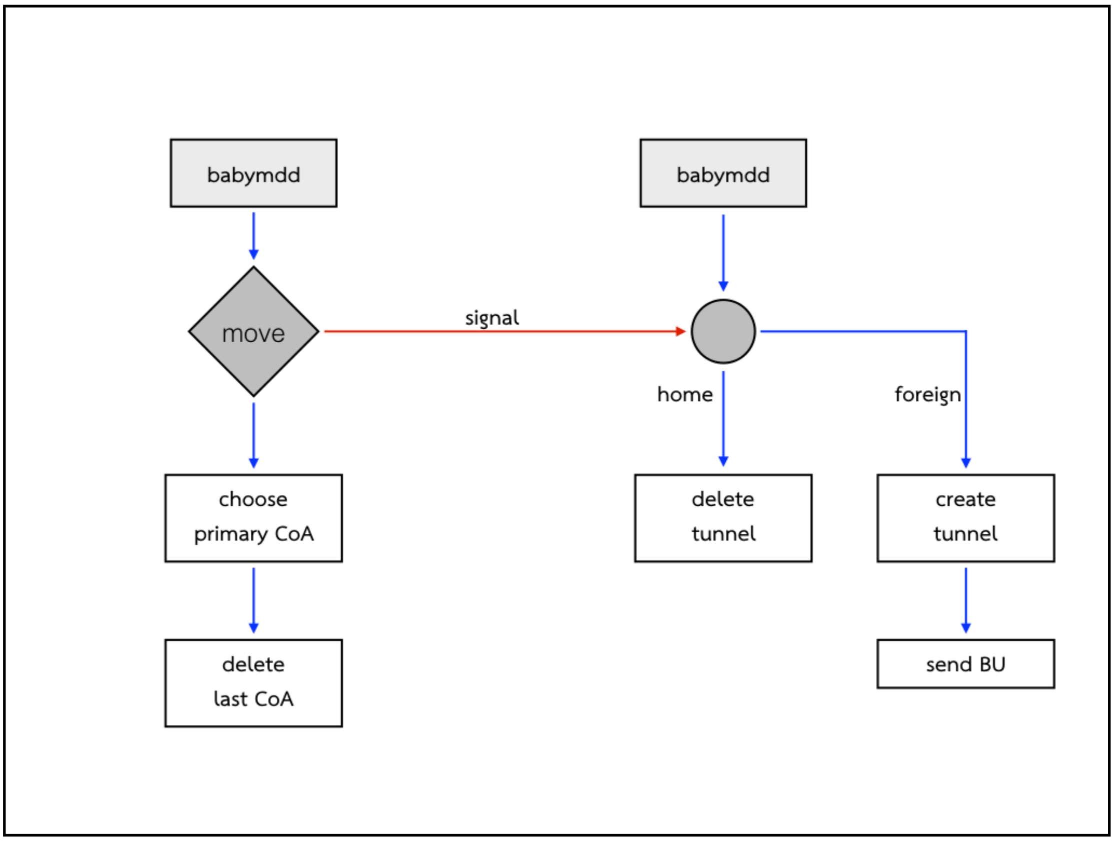


Figure 4.8 flow chart of babymdd and mnd process

Figure 4.8 shows that the mnd process controls a tunnel to connect to HA. This process creates virtual interface named "mipX" (X is a number starting at 0) for a tunnel with HAs in KAME software. There is one virtual mip interface for each HA. So the MN needs multiple mip interfaces to support multiple HAs shown figure 4.9. We implement on FreeBSD OS version 5.4. So figure 4.10 shows several mip interfaces of MN which has two home networks and two tunnels, one to each HA on the FreeBSD system.

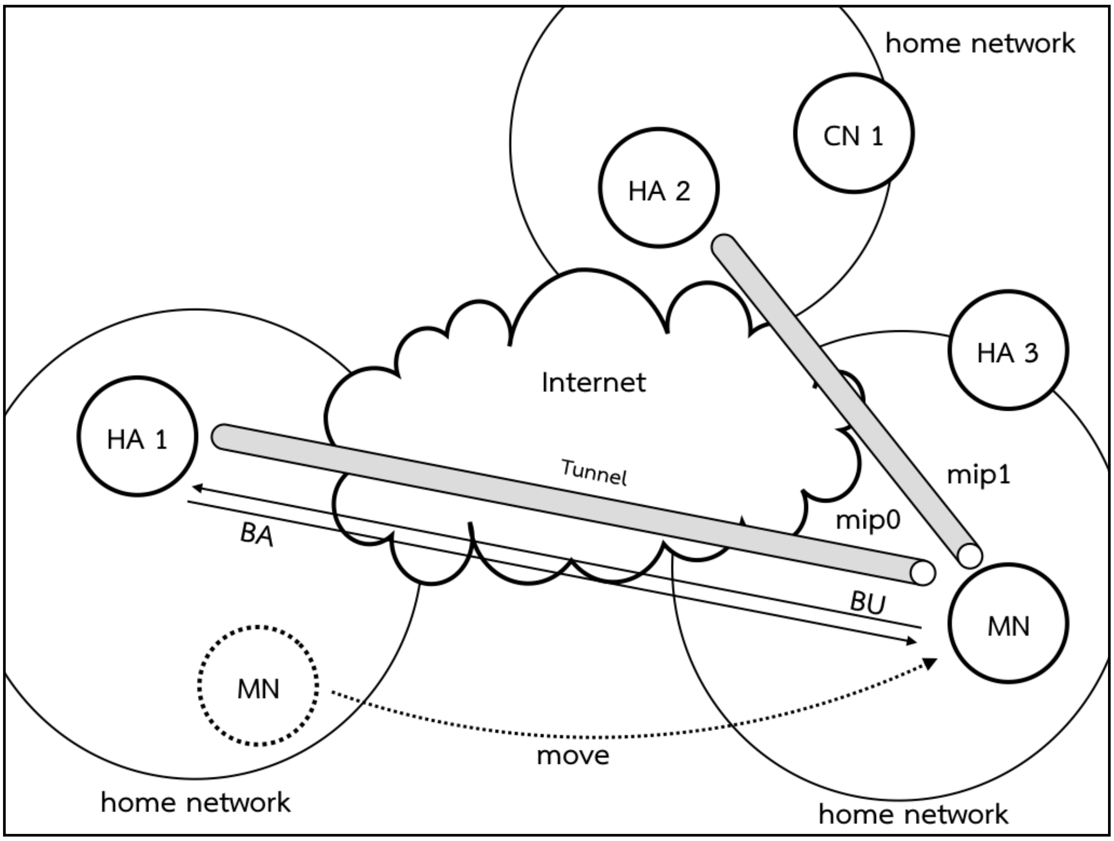
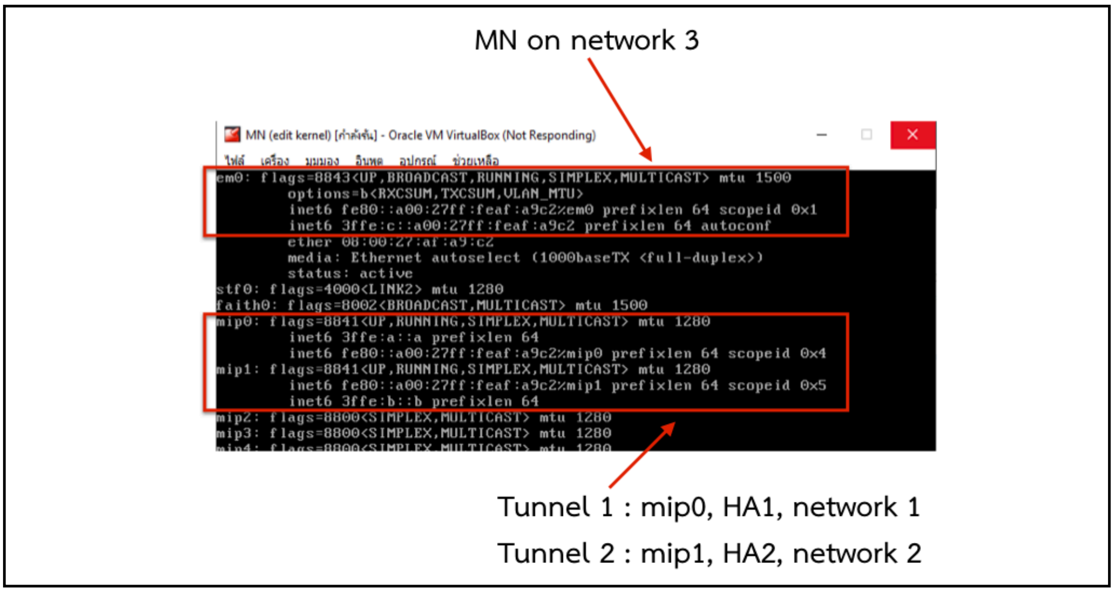


Figure 4.9 virtual interface (mip) to create tunnel


MN on network 3
MN (edit kernel) [กำลังรัน] - Oracle VM VirtualBox (Not Responding)
ไฟล์ เครื่อง มุมมอง อินพุต อุปกรณ์ ช่วยเหลือ
Tunnel 1 : mip0, HA1, network 1
Tunnel 2 : mip1, HA2, network 2

```
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=b<RXCSUM,TXCSUM,VLAN_MTU>
        inet6 fe80::a00:27ff:feaf:a9c2%em0 prefixlen 64 scopeid 0x1
        inet6 3ffe:c::a00:27ff:feaf:a9c2 prefixlen 64 autoconf
        ether 08:00:27:af:a9:c2
        media: Ethernet autoselect (1000baseTX <full-duplex>)
        status: active
stf0: flags=4000<LINK2> mtu 1280
faith0: flags=8002<BROADCAST,MULTICAST> mtu 1500
mip0: flags=8841<UP,RUNNING,SIMPLEX,MULTICAST> mtu 1280
        inet6 3ffe:a::a prefixlen 64
        inet6 fe80::a00:27ff:feaf:a9c2%mip0 prefixlen 64 scopeid 0x4
mip1: flags=8841<UP,RUNNING,SIMPLEX,MULTICAST> mtu 1280
        inet6 fe80::a00:27ff:feaf:a9c2%mip1 prefixlen 64 scopeid 0x5
        inet6 3ffe:b::b prefixlen 64
mip2: flags=8800<SIMPLEX,MULTICAST> mtu 1280
mip3: flags=8800<SIMPLEX,MULTICAST> mtu 1280
mip4: flags=8800<SIMPLEX,MULTICAST> mtu 1280
```

Figure 4.10 show mip interface on FreeBSD

The KAME software refers to the Mobility Support in IPv6 (RFC 3775) [5]. If the MN away from home network, then it needs to communicate via HA at home network (triangle route). The MN get a cost and a benefit from this situation (shown in figure 4.11). A benefit of this is the MN can continue existing communications. A cost is the triangle route between the MN and the CN. However, the MN can provide a communication directly with any nodes when it stays at the home network. So we proposed to have multiple HAs and home networks to make every network that the MN moves to a home network. The MN can dispose of a cost of MIPv6 for connections that don’t require mobile IP secure, and it can get a benefit to keeping a communication for those connections that do (shown in figure 4.12).

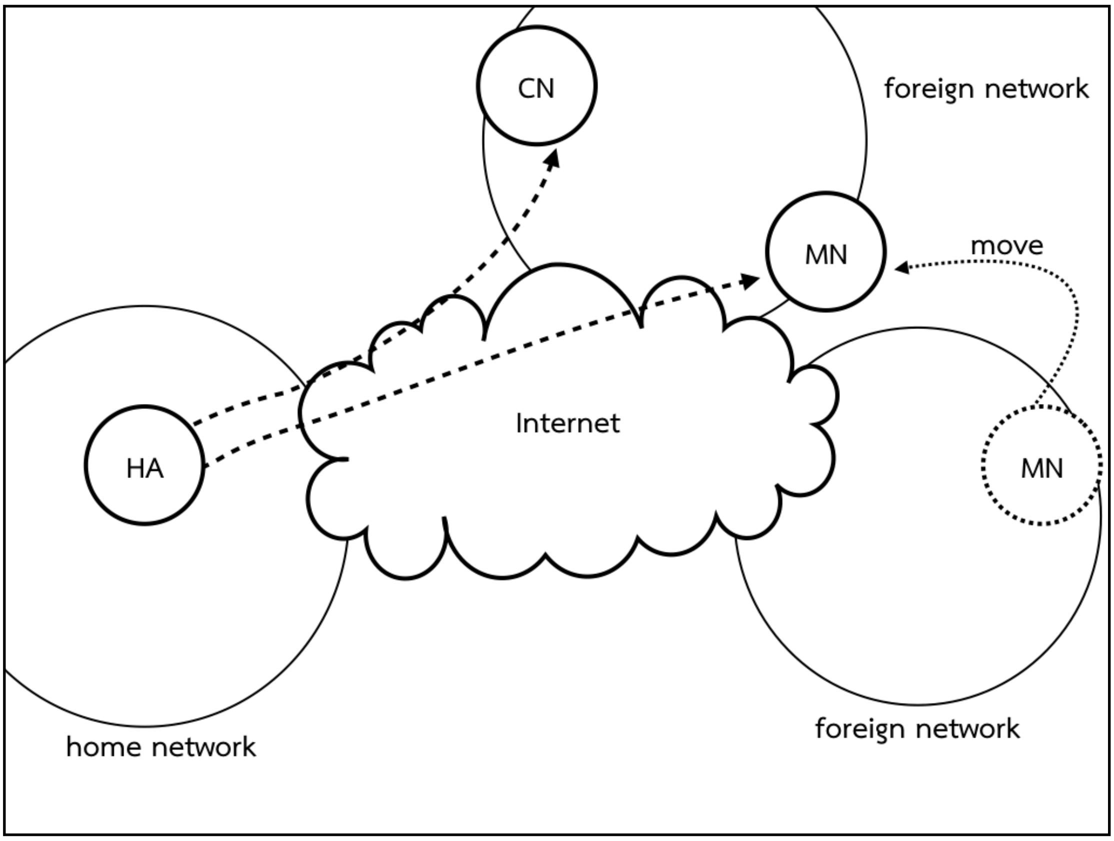


Figure 4.11 A worst case of cost for triangle route on MIPv6

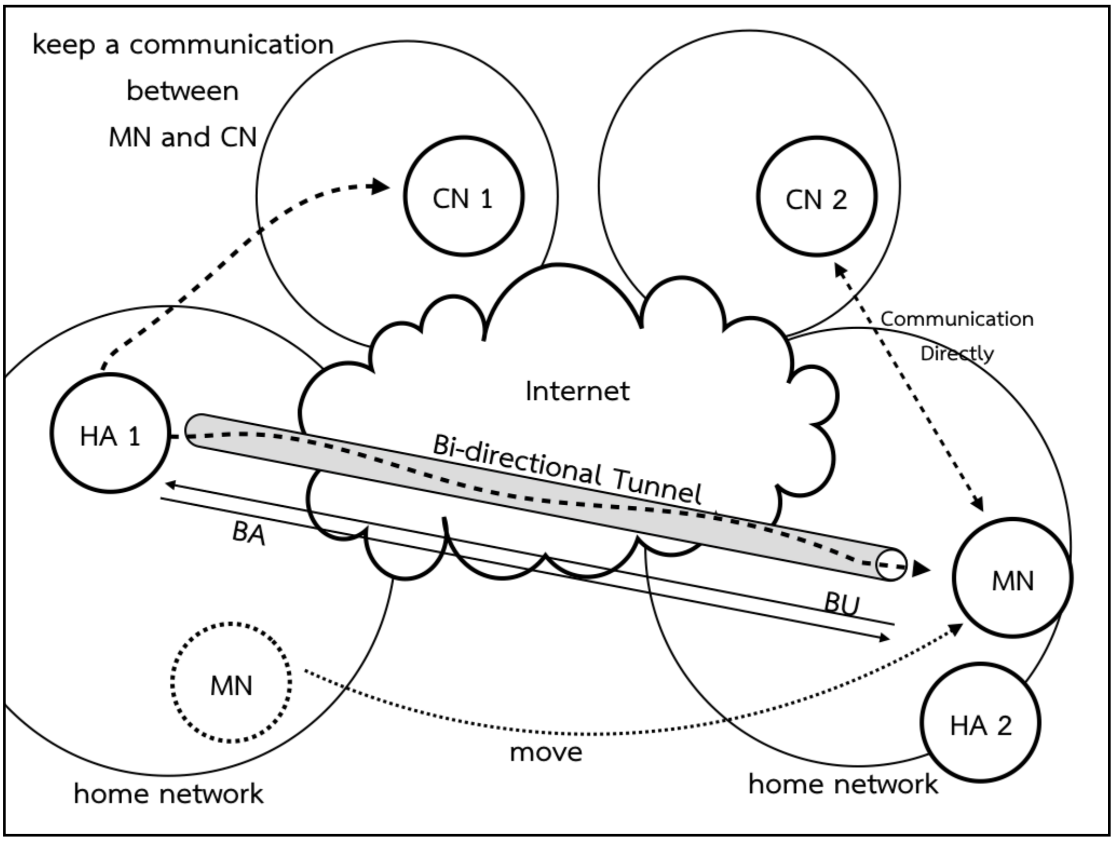


Figure 4.12 The MN on multiple HAs and home network.

We determined to use a multiple mnd and babymdd process with each tunnel. It benefits flexibility and is convenient to maintain. Also, we need to edit the KAME software routing table operations. A cost problem chooses a route to tunnel only when the MN away from home network. However, we have a multiple home networks. So the MN needs to choose an IP address from that network (home network).

4.3. Testing And Evaluation

In this part, we describe the testing and the evaluation of our solution in multiple HAs and home networks. The solution is evaluated using an implementation of the designed solution. First, we describe a network topology for a test scenario. Second, we experiment with three different scenarios to compare between standard MIPv6 and our solution. First scenarios, the MN starts at home network. Following scenarios, the MN moves to network number two, and it has two HAs. Finally, the MN moves to network number three, and it has three HAs.

4.3.1.Network topology

Our solution design a protocol for multiple HAs and home networks. The implementation is used to verify the design of our solution. Figure 4.13 shows the network topology for the testing. We have three networks connected. Also, every network has a HA. We implement on a virtual machine (Oracle VM Virtualbox program). This requires only one physical computer to run many virtual computers on the virtual machine, it is flexible and comfortable to setup.

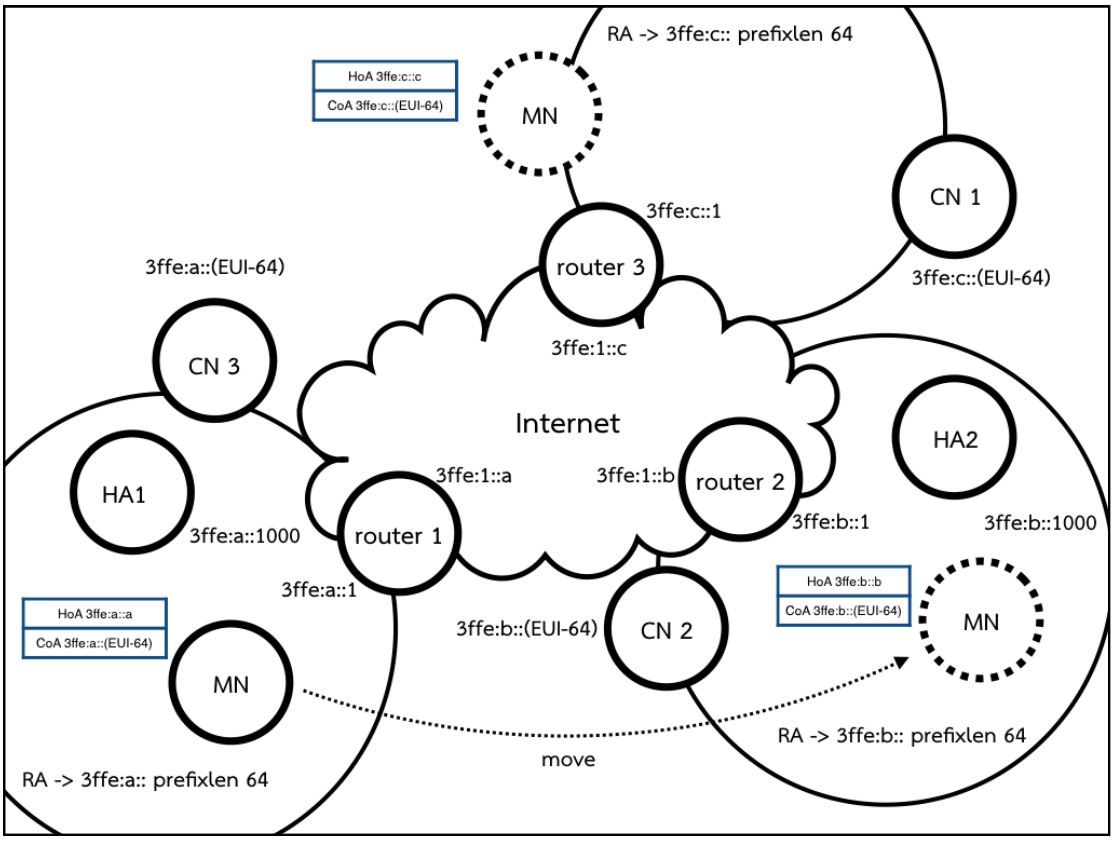


Figure 4.13 The network topology for testing

4.3.2.Scenario

We compare the MIPv6 protocol and our protocol. So every the MN move is to a new home network for our protocol. However, only scenario 1 is a home network for the MIPv6 protocol. Also, we show a different IP address to communicate a new session between the MIPv6 and our protocol, how our protocol can retain communications when the MN moves.

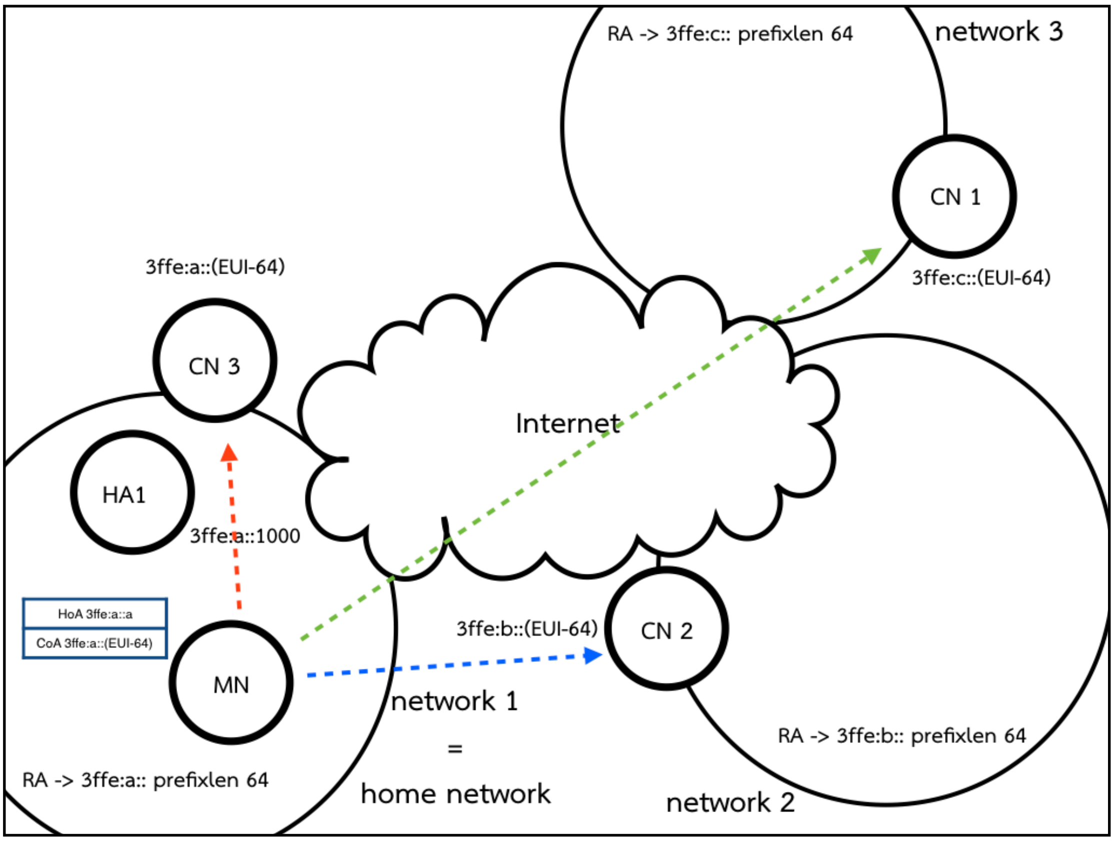
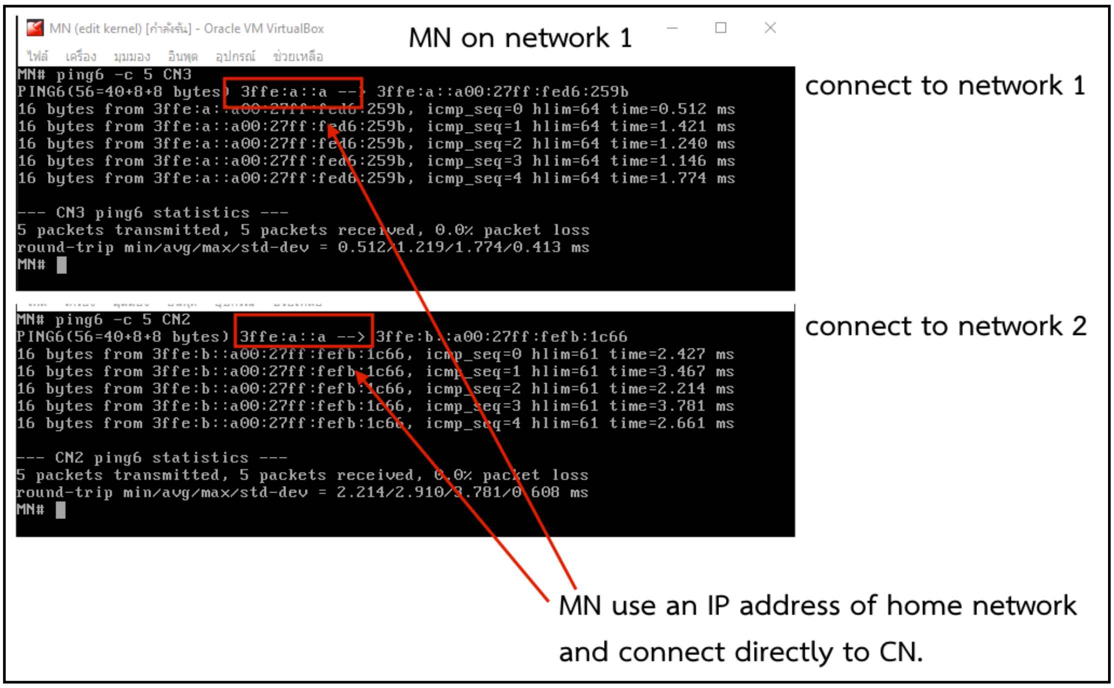


Figure 4.14 scenario 1

The first scenario is shown in figure 4.14. The MN stays at home network (prefix 3ffe:a::). So there is no difference between the MIPv6 and our protocol. The MN can start with an IP address from that network. Figure 4.15 shows the result.


MN (edit kernel) [กำลังรัน] - Oracle VM VirtualBox
ไฟล์ เครื่อง มุมมอง อินพุต อุปกรณ์ ช่วยเหลือ

MN on network 1

connect to network 1

```
MN# ping6 -c 5 CN3
PING6(56=40+8+8 bytes) 3ffe:a::a --> 3ffe:a::a00:27ff:fed6:259b
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=0 hlim=64 time=0.512 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=1 hlim=64 time=1.421 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=2 hlim=64 time=1.240 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=3 hlim=64 time=1.146 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=4 hlim=64 time=1.774 ms

--- CN3 ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.512/1.219/1.774/0.413 ms
MN# █
```

connect to network 2

```
MN# ping6 -c 5 CN2
PING6(56=40+8+8 bytes) 3ffe:a::a --> 3ffe:b::a00:27ff:fefb:1c66
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=0 hlim=61 time=2.427 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=1 hlim=61 time=3.467 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=2 hlim=61 time=2.214 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=3 hlim=61 time=3.781 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=4 hlim=61 time=2.661 ms

--- CN2 ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 2.214/2.910/3.781/0.608 ms
MN# █
```

MN use an IP address of home network and connect directly to CN.

Figure 4.15 MN on home network.

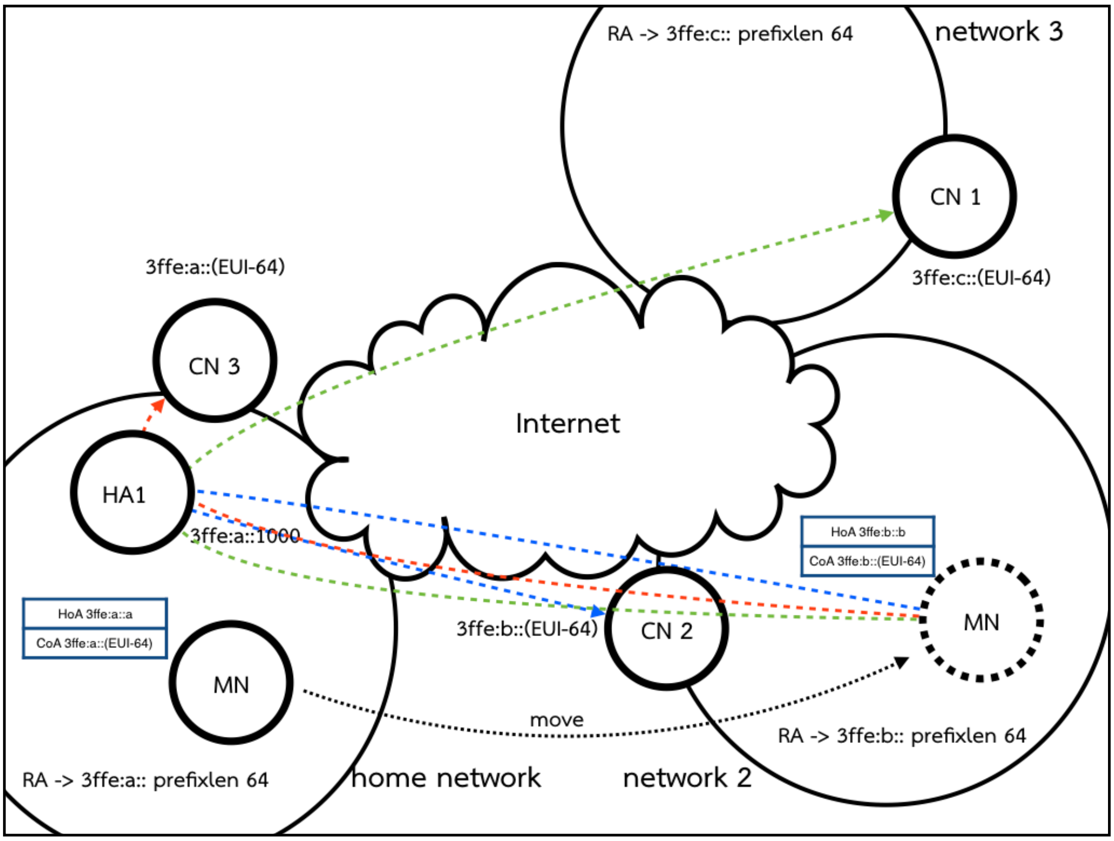


Figure 4.16 scenario 2 on MIPv6

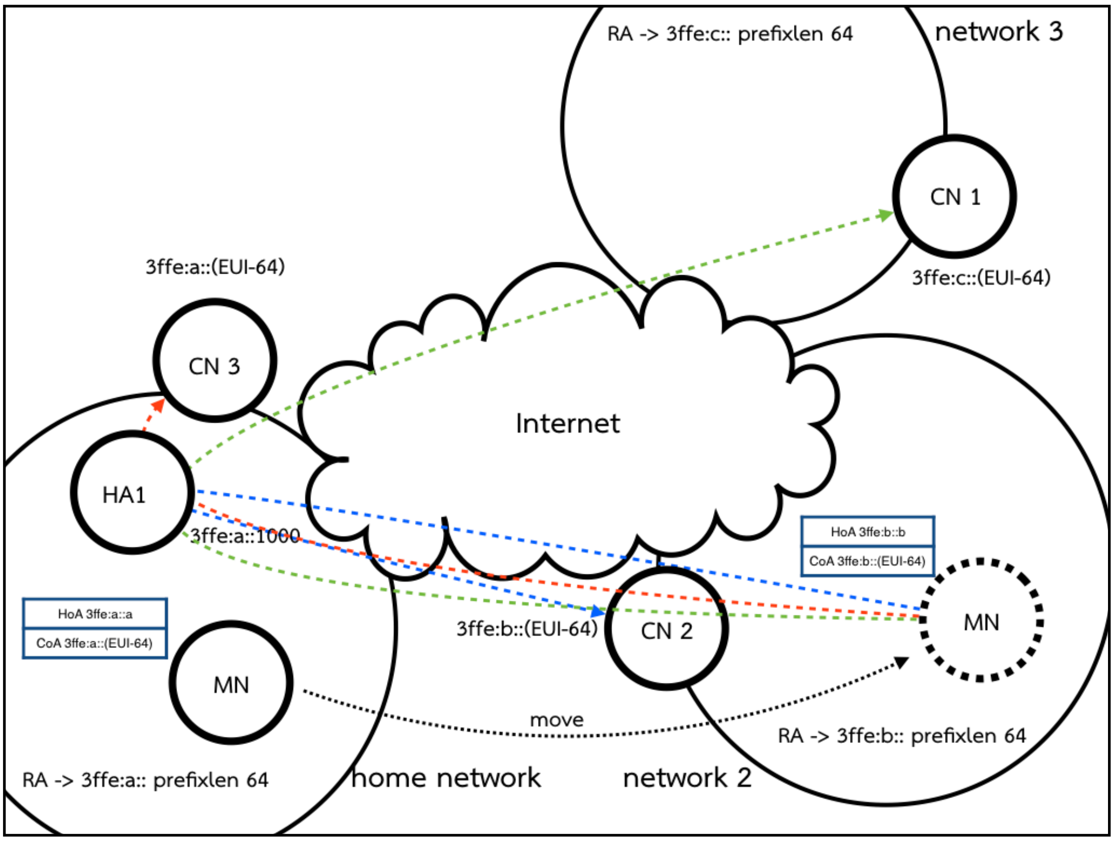
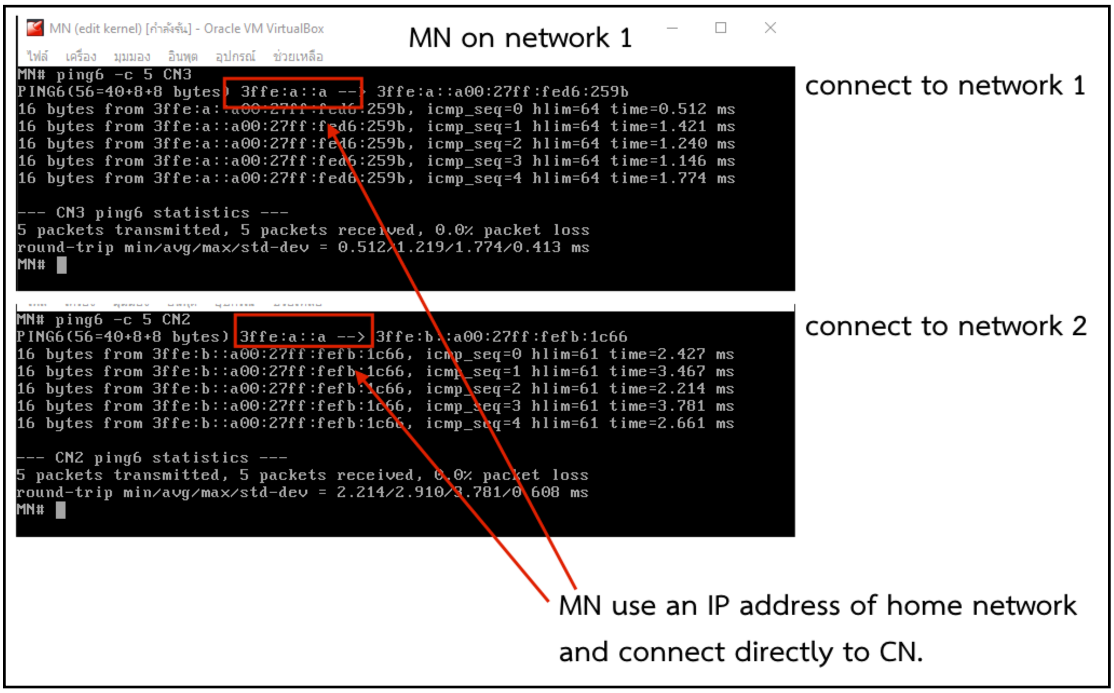


Figure 4.17 scenario 2 on us protocol

The second scenario: the MIPv6 protocol is shown in figure 4.16 and our protocol is shown in figure 4.17. The MN moves to network number 2 (prefix 3ffe:b::). So it is now at a foreign network for the MIPv6 protocol. The MN connects to every node via HA from the home network for a new connection. On the other hand, the MN uses a new IP address from the new network to start any new connection to any node for our protocol. Also, both protocols can retain the existing communications. Figure 4.18 shows the result of the MIPv6 protocol and figure 4.19 for our protocol.



MN (edit kernel) [กำลังรัน] - Oracle VM VirtualBox
ไฟล์ เครื่อง มุมมอง อินพุต อุปกรณ์ ช่วยเหลือ
MN on network 1
connect to network 1
connect to network 2
MN use an IP address of home network
and connect directly to CN.

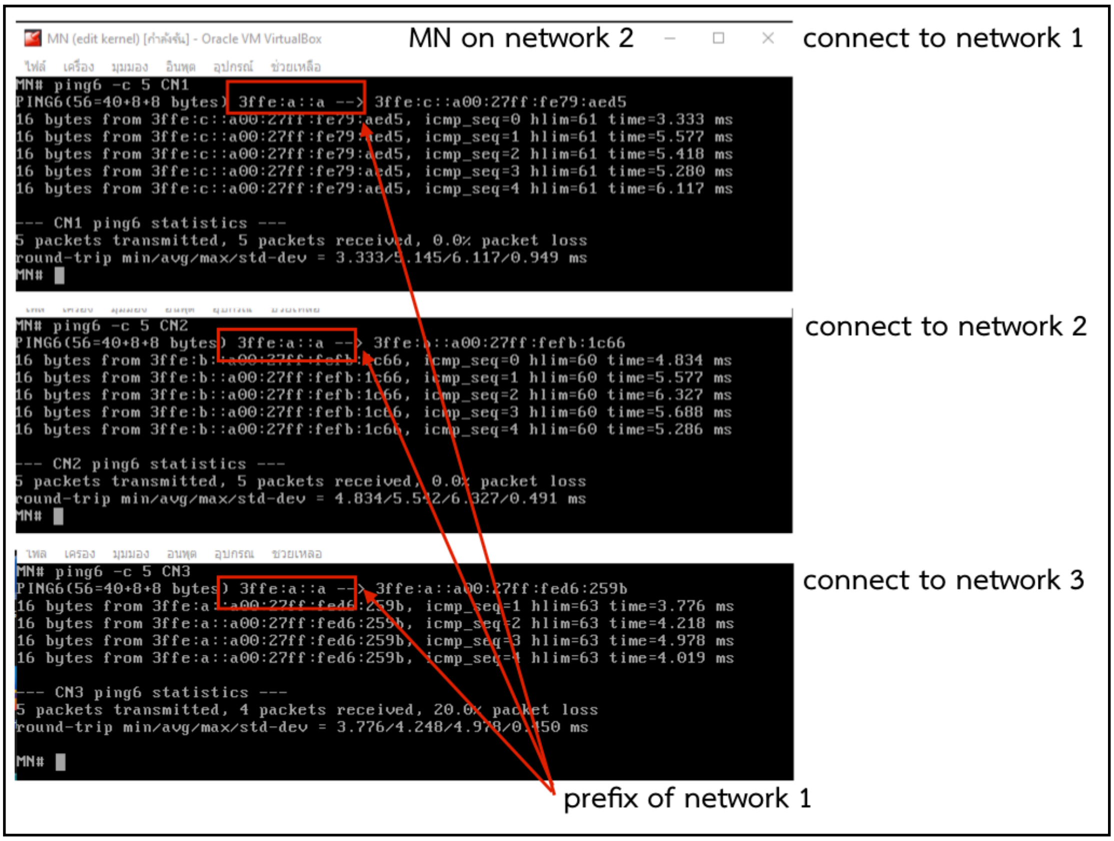
```
MN# ping6 -c 5 CN3
PING6(56=40+8+8 bytes) 3ffe:a::a --> 3ffe:a::a00:27ff:fed6:259b
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=0 hlim=64 time=0.512 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=1 hlim=64 time=1.421 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=2 hlim=64 time=1.240 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=3 hlim=64 time=1.146 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=4 hlim=64 time=1.774 ms

--- CN3 ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.512/1.219/1.774/0.413 ms
MN#
```

```
MN# ping6 -c 5 CN2
PING6(56=40+8+8 bytes) 3ffe:a::a --> 3ffe:b::a00:27ff:fefb:1c66
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=0 hlim=61 time=2.427 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=1 hlim=61 time=3.467 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=2 hlim=61 time=2.214 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=3 hlim=61 time=3.781 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=4 hlim=61 time=2.661 ms

--- CN2 ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 2.214/2.910/3.781/0.608 ms
MN#
```

Figure 4.18 the result of scenario 2 of MIPv6 protocol



MN (edit kernel) [กำลังรัน] - Oracle VM VirtualBox
ไฟล์ เครื่อง มุมมอง อินพุต อุปกรณ์ ช่วยเหลือ
MN on network 2
connect to network 1
connect to network 2
connect to network 3
prefix of network 1

```
MN# ping6 -c 5 CN1
PING6(56=40+8+8 bytes) 3ffe:a::a --> 3ffe:c::a00:27ff:fe79:aed5
16 bytes from 3ffe:c::a00:27ff:fe79:aed5, icmp_seq=0 hlim=61 time=3.333 ms
16 bytes from 3ffe:c::a00:27ff:fe79:aed5, icmp_seq=1 hlim=61 time=5.577 ms
16 bytes from 3ffe:c::a00:27ff:fe79:aed5, icmp_seq=2 hlim=61 time=5.418 ms
16 bytes from 3ffe:c::a00:27ff:fe79:aed5, icmp_seq=3 hlim=61 time=5.280 ms
16 bytes from 3ffe:c::a00:27ff:fe79:aed5, icmp_seq=4 hlim=61 time=6.117 ms

--- CN1 ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 3.333/5.145/6.117/0.949 ms
MN#
```

```
MN# ping6 -c 5 CN2
PING6(56=40+8+8 bytes) 3ffe:a::a --> 3ffe:b::a00:27ff:fefb:1c66
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=0 hlim=60 time=4.834 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=1 hlim=60 time=5.577 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=2 hlim=60 time=6.327 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=3 hlim=60 time=5.688 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=4 hlim=60 time=5.286 ms

--- CN2 ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 4.834/5.542/6.327/0.491 ms
MN#
```

```
MN# ping6 -c 5 CN3
PING6(56=40+8+8 bytes) 3ffe:a::a --> 3ffe:a::a00:27ff:fed6:259b
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=1 hlim=63 time=3.776 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=2 hlim=63 time=4.218 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=3 hlim=63 time=4.978 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=4 hlim=63 time=4.019 ms

--- CN3 ping6 statistics ---
5 packets transmitted, 4 packets received, 20.0% packet loss
round-trip min/avg/max/std-dev = 3.776/4.248/4.978/0.450 ms

MN#
```

Figure 4.19 the result of scenario 2 of us protocol

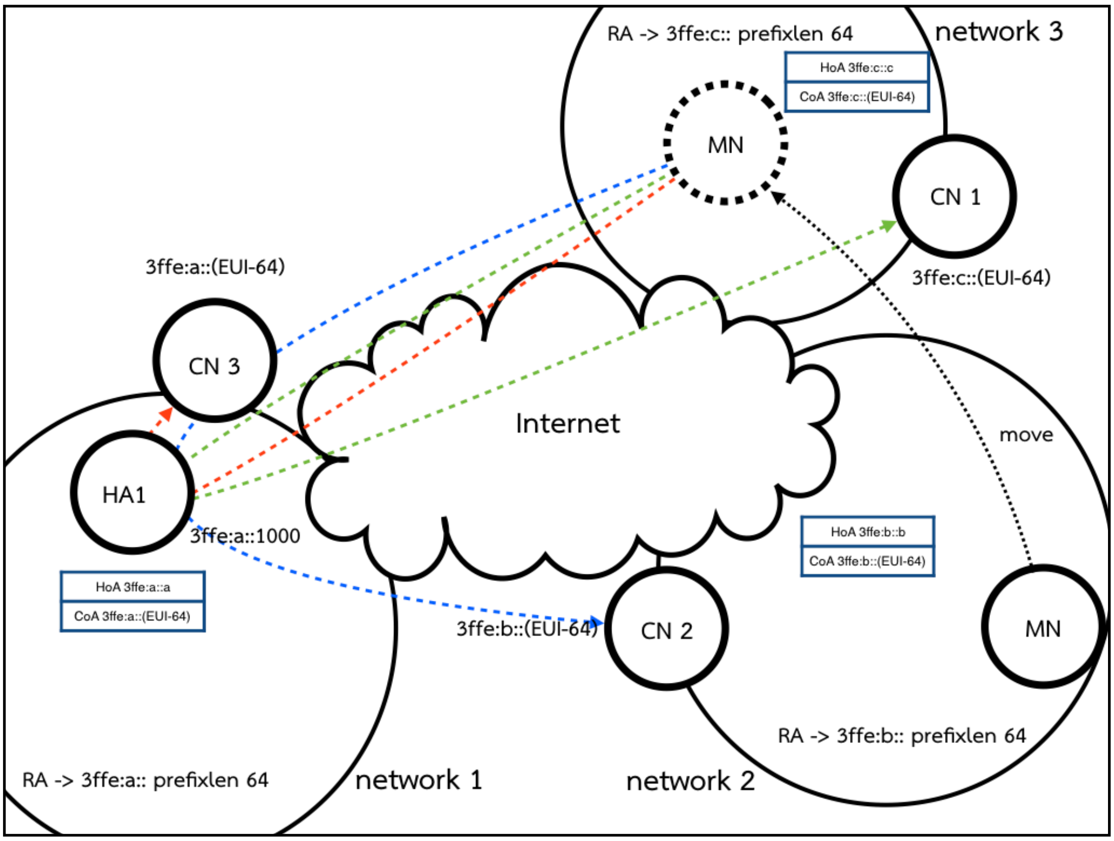


Figure 4.20 scenario 3 on MIPv6 protocol

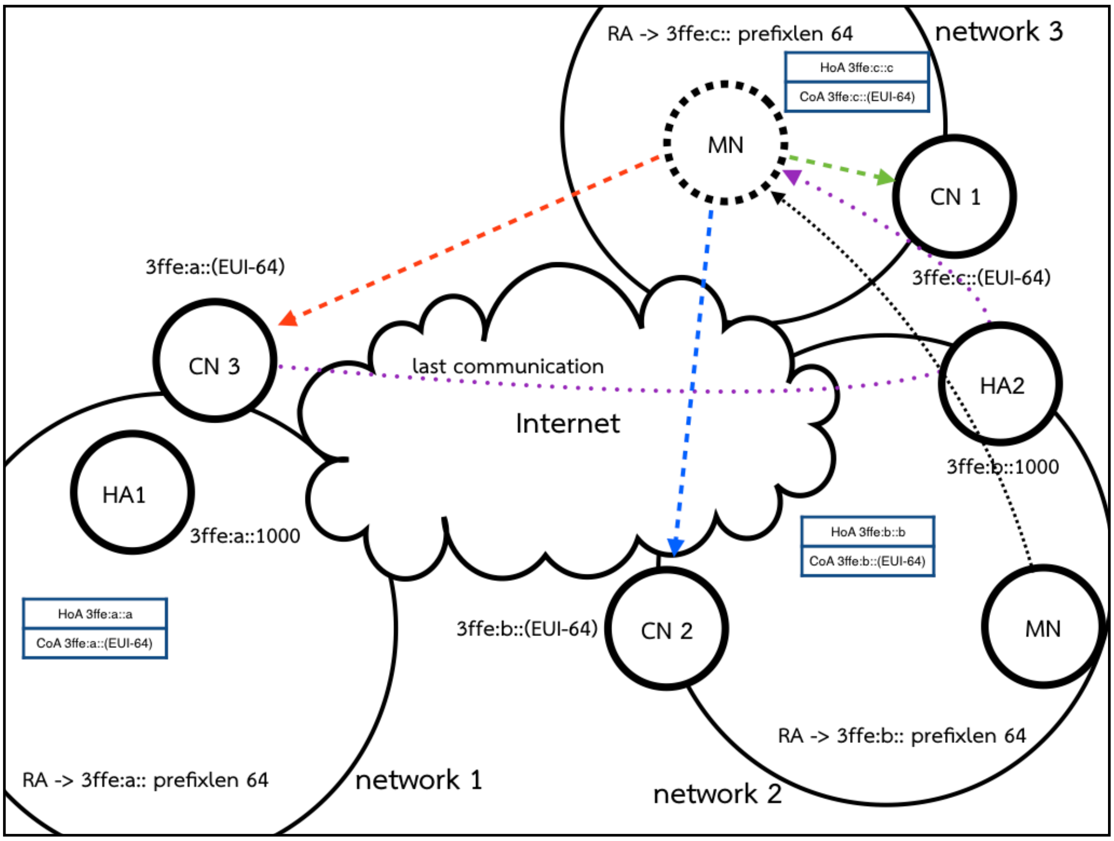
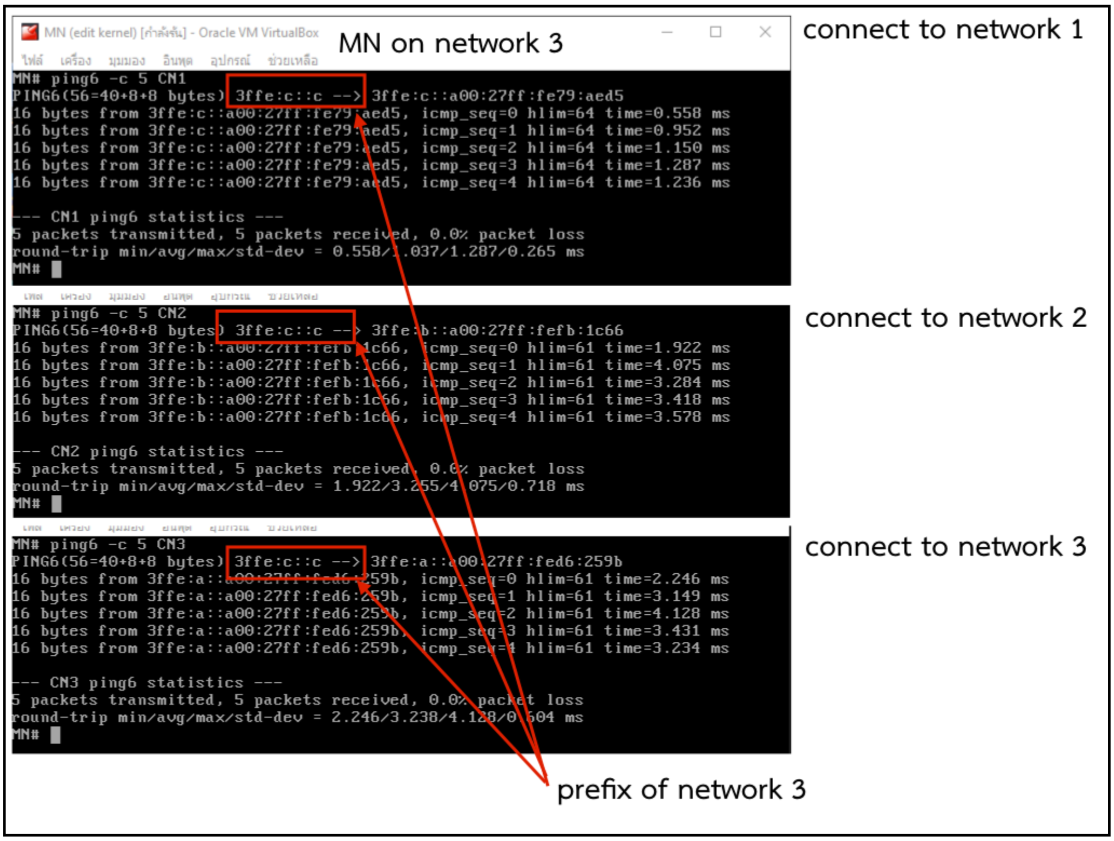


Figure 4.21 scenario 3 on us protocol

The third scenario: the MIPv6 protocol is shown in figure 4.20 and our protocol is shown in figure 4.21. The MN moves to network number 3 (prefix 3ffe:c::). So it is another foreign network for the MIPv6 protocol. A condition like scenario 2 for the MIPv6 protocol. However, we have a 3 HAs for our protocol. So the MN can keep communicating using both HA1 and HA2. Also, the MN can start a new session using its IP address from network number 3. Figure 4.22 shows the result of our protocol. Figure 4.23 shows multiple tunnels for multiple HAs of our protocol.


MN (edit kernel) [กำลังรัน] - Oracle VM VirtualBox
ไฟล์ เครื่อง มุมมอง อินพุต อุปกรณ์ ช่วยเหลือ
MN on network 3
connect to network 1
connect to network 2
connect to network 3
prefix of network 3

```
MN# ping6 -c 5 CN1
PING6(56=40+8+8 bytes) 3ffe:c::c --> 3ffe:c::a00:27ff:fe79:aed5
16 bytes from 3ffe:c::a00:27ff:fe79:aed5, icmp_seq=0 hlim=64 time=0.558 ms
16 bytes from 3ffe:c::a00:27ff:fe79:aed5, icmp_seq=1 hlim=64 time=0.952 ms
16 bytes from 3ffe:c::a00:27ff:fe79:aed5, icmp_seq=2 hlim=64 time=1.150 ms
16 bytes from 3ffe:c::a00:27ff:fe79:aed5, icmp_seq=3 hlim=64 time=1.287 ms
16 bytes from 3ffe:c::a00:27ff:fe79:aed5, icmp_seq=4 hlim=64 time=1.236 ms

--- CN1 ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.558/1.037/1.287/0.265 ms
MN#
```

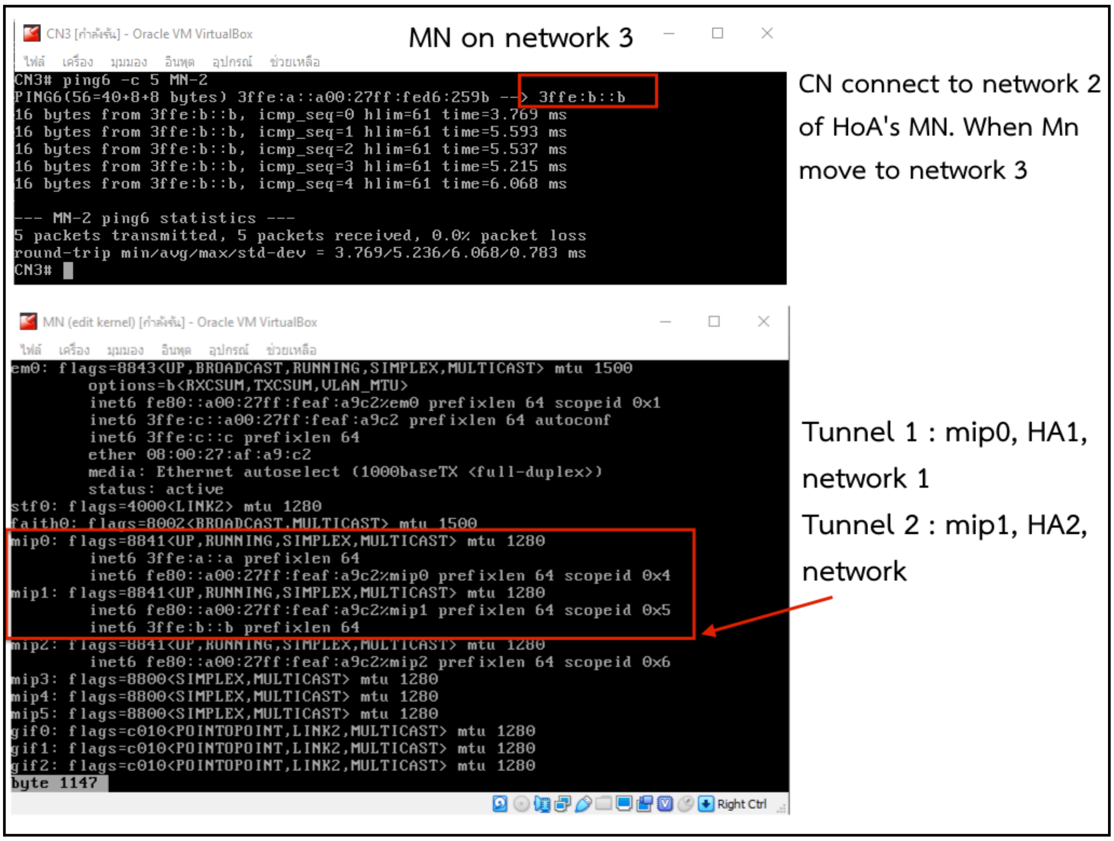
```
MN# ping6 -c 5 CN2
PING6(56=40+8+8 bytes) 3ffe:c::c --> 3ffe:b::a00:27ff:fefb:1c66
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=0 hlim=61 time=1.922 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=1 hlim=61 time=4.075 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=2 hlim=61 time=3.284 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=3 hlim=61 time=3.418 ms
16 bytes from 3ffe:b::a00:27ff:fefb:1c66, icmp_seq=4 hlim=61 time=3.578 ms

--- CN2 ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 1.922/3.255/4.075/0.718 ms
MN#
```

```
MN# ping6 -c 5 CN3
PING6(56=40+8+8 bytes) 3ffe:c::c --> 3ffe:a::a00:27ff:fed6:259b
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=0 hlim=61 time=2.246 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=1 hlim=61 time=3.149 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=2 hlim=61 time=4.128 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=3 hlim=61 time=3.431 ms
16 bytes from 3ffe:a::a00:27ff:fed6:259b, icmp_seq=4 hlim=61 time=3.234 ms

--- CN3 ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 2.246/3.238/4.128/0.604 ms
MN#
```

Figure 4.22 The result of scenario 3 of us protocol


CN3 [กำลังรัน] - Oracle VM VirtualBox
ไฟล์ เครื่อง มุมมอง อินพุต อุปกรณ์ ช่วยเหลือ
MN (edit kernel) [กำลังรัน] - Oracle VM VirtualBox
ไฟล์ เครื่อง มุมมอง อินพุต อุปกรณ์ ช่วยเหลือ
Right Ctrl

MN on network 3

```
CN3# ping6 -c 5 MN-2
PING6(56=40+8+8 bytes) 3ffe:a::a00:27ff:fed6:259b --> 3ffe:b::b
16 bytes from 3ffe:b::b, icmp_seq=0 hlim=61 time=3.769 ms
16 bytes from 3ffe:b::b, icmp_seq=1 hlim=61 time=5.593 ms
16 bytes from 3ffe:b::b, icmp_seq=2 hlim=61 time=5.537 ms
16 bytes from 3ffe:b::b, icmp_seq=3 hlim=61 time=5.215 ms
16 bytes from 3ffe:b::b, icmp_seq=4 hlim=61 time=6.068 ms

--- MN-2 ping6 statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 3.769/5.236/6.068/0.783 ms
CN3# 
```

CN connect to network 2 of HoA's MN. When Mn move to network 3

```
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=b<RXCSUM,TXCSUM,VLAN_MTU>
        inet6 fe80::a00:27ff:feaf:a9c2%em0 prefixlen 64 scopeid 0x1
        inet6 3ffe:c::a00:27ff:feaf:a9c2 prefixlen 64 autoconf
        inet6 3ffe:c::c prefixlen 64
        ether 08:00:27:af:a9:c2
        media: Ethernet autoselect (1000baseTX <full-duplex>)
        status: active
stf0: flags=4000<LINK2> mtu 1280
faith0: flags=8002<BROADCAST,MULTICAST> mtu 1500
mip0: flags=8841<UP,RUNNING,SIMPLEX,MULTICAST> mtu 1280
        inet6 3ffe:a::a prefixlen 64
        inet6 fe80::a00:27ff:feaf:a9c2%mip0 prefixlen 64 scopeid 0x4
mip1: flags=8841<UP,RUNNING,SIMPLEX,MULTICAST> mtu 1280
        inet6 fe80::a00:27ff:feaf:a9c2%mip1 prefixlen 64 scopeid 0x5
        inet6 3ffe:b::b prefixlen 64
mip2: flags=8841<UP,RUNNING,SIMPLEX,MULTICAST> mtu 1280
        inet6 fe80::a00:27ff:feaf:a9c2%mip2 prefixlen 64 scopeid 0x6
mip3: flags=8800<SIMPLEX,MULTICAST> mtu 1280
mip4: flags=8800<SIMPLEX,MULTICAST> mtu 1280
mip5: flags=8800<SIMPLEX,MULTICAST> mtu 1280
gif0: flags=c010<POINTOPOINT,LINK2,MULTICAST> mtu 1280
gif1: flags=c010<POINTOPOINT,LINK2,MULTICAST> mtu 1280
gif2: flags=c010<POINTOPOINT,LINK2,MULTICAST> mtu 1280
byte 1147
```

Tunnel 1 : mip0, HA1, network 1

Tunnel 2 : mip1, HA2, network

Figure 4.23 the multiple tunnel for multiple HA of us protocol

This section describes the trade-offs of our protocol with regard to storage, speed and computation. First, keeping a list of Home Agents, and the MN's associated Home Addresses, is a requirement of our protocol. The advantage we gain is that for new connections (aside from the HA cost) there are no overheads, unlike new connections established by a normal MIPv6 when it is away from home. Whether we end up saving packets or time or not will depend upon the nature of the communications, a mobile node that moves, establishes a HA at the foreign network (making it a new home network) but which then does nothing else before it moves again, will have done all that work getting a new HA and then sending it BU's until we drop it as a HA for no benefit at all. On the other hand, when the MN is at the new network for an extended period and makes many new connections, which transfer lots of data, the savings in both the amount of data transferred by the node, even the tunnel overheads end up being noticeable, and the savings to the rest of the network, with less unnecessary traffic can be considerable. Perhaps even more importantly, this method makes it more reasonable for a node to enable MIP in the first place, and by so doing avoid broken communications, which might mean restarting connections causing duplication of previously transmitted data, by allowing MIPv6 to work to keep connections alive after the node has moved, which it obviously cannot do if MIP is not being used because the costs of enabling it seem to be too high to bear.

Second, sending BUs to multiple HAs is required by our protocol. This cost is something that we actively want. Ideally we will keep the list of HAs as small as possible, to keep this overhead as low as possible, but the issue of dropping HAs once established, that is choosing which HA can be dropped, has not yet been investigated, so for now once we have a new HA we have it forever. That will become a significant cost over time if it is not dealt with.

Third, there are overheads involved in finding a local HA (using DHAAD) and in automating the establishment of the MN to HA association. There are three cases to consider: No Mobile IP at all, traditional MIPv6, and our protocol. In all cases the MN needs to detect it has moved, get attached to the new network (which may involve entering WPA passwords, or various other authentication requirements) then once the link level is established, send a RS to get a RA and discover the default router (and if it is doing SEND, validate it). Then it needs to acquire an address using DHCPv6 or stateless address auto configuration and perform the Duplicate Address Detection algorithm.

After that in Mobile IP is not being used, the "mobile" node can start using the network immediately. For MIPv6 we next need to send a BU to the (one) HA, before which (or perhaps in parallel) send DHAAD to the home network to discover the HA's current address. Then wait for the answering BA to be received. After that has occurred, it can start making new connections.

For our protocol we do DHAAD, then the Automated HA Config protocol. That is certainly at least 2 round trips (one for DHAAD's request and reply, and one for the HA_REQ and reply) though more likely at least 2 for AHAC (with possibly even some user interaction involved to supply info needed for registration). Ignoring the user interaction that means 3 RTT's probably, along with perhaps more for HA authentication if that is being done. But if it is, SEND is probably enabled as well, and most of the overheads will have already been paid as part of the router authentication that would have already been done - the trust anchor and chain will the the same for the two cases, all we need additional for the HA case is to verify the signature on the certificate - which should be the same one that authenticated the router, so that just means computation, no more packets. Then we need to send a BU and get a BA to finalize the HA setup.

The advantage is that all these packets (other than the SEND related ones which we should not normally need - because that has already been done) are local LAN packets, and so all have very short RTT's, and also consequently allow for very short retransmit delays in the case of no reply, unlike standard MIPv6, where even if the node has only moved a short distance, and the RTT is small, the MN cannot know that until after it has actually sent a packet and received a reply so at least initially it needs to allow a lengthy delay (perhaps a second) between transmit and retransmit, just in case the movement was to a long (network route) distance from the home network.

Normally the HA lives in the router, one additional HA discovery method might be to have the router announce it is a HA in its RA packet, and have it authenticated as a HA in the exact same authentication that authenticates it as a router -- no additional packets or computation are needed at all in that case (the authentication could be combined even if the RA isn't enhanced to include an "I am a HA" bit).

Fourth, our protocol adds additional computation costs. The Automated HA configuration for MIPv6 protocol requires additional CPU usage not required previously, at both the MN and the HA. For the MN this may have battery usage implications (more power consumption) - for the HA the chief cost (apart from whether or not it has sufficient CPU cycles available) is that it will need to have a private key available (and secure internally) that it can use to sign packets. If it is doing SEND, it would need that anyway, and if it isn't, it probably doesn't really care about security, so having a private key less secure than perfectly, might be acceptable.

Fifth, we need to consider if we have made communications slower in the case that the new protocol is not available at a foreign network, either because there is no HA, or the HA has not been upgraded to understand AHAC, or when the HA refuses to act on behalf of a newly arrived MN. We need to determine whether we have slowed things down by attempting the new protocol and failing. That means whether or not we can get a "no" response from the HA quicker than we get the BA from the old (or original) HA. If we actually get a "no" response then that should certainly be faster than the BA coming back, as it is just 1, or 2, LAN RTTs compared with an Internet RTT. So, when most HAs support the new protocol, and most networks have an HA (probably as part of their router) - that is, if MIP becomes popular - in the cases where there is no available (willing) HA, we should be adding no extra delay, and as soon as the MN gets its BA from the old HA and can resume operating, it should already know there is no local HA to assist.

The cases where there is no HA on the network at all, or none which understands the new protocol are harder to evaluate, as (in the first case) that means sending several DHAAD requests, with waits for replies that never come before retransmits - and it all depends upon how long we wait, and how many times we retransmit, for which we have no answers (no experiments were performed for that case). Similarly if a HA exists, but does not know the protocol, there will be HA_REQ packets transmitted and not answered, with similar delays.

One reasonable approach might be to resume using the network as soon as the BA from the old HA is received (which is as early as MIP allows) or if the MN has completed the new protocol, allow new connections when that happens. That means that using the new protocol can reduce the delays, but cannot increase them. The cost of this is that some new connections may end up using the old HA (and so be triangle routed, or need RO applied) when if they had simply been delayed a little until the new protocol succeeds would have not needed that. Of course, since at that point we don't know whether the new protocol will ever succeed, so delaying is a gamble.

It would also be worth pointing out that all of this is a transitional problem - once this new protocol becomes popular, and MIP is used more, networks will have HAs and those HAs will support this protocol at least with the ability to send a "no" reply, which removes the ambiguities and means no delays beyond those with normal MIP. On the other hand, if this new protocol does not become popular, then no HAs will use it, so no MNs will bother implementing it, and the whole question of its costs becomes irrelevant.

The method of overhead analysis is to compare traditional MIPv6 with our protocol. For each individual MN, it only needs to be associated with as many HAs as benefit. If the overheads of having the HAs are greater than the benefits being achieved, then the HAs can be released and the cost goes away. So any scalability issues never affect individual nodes. But for the network as a whole, we have potentially MN multiplied by HA (MN x HA) of HA associations, instead of just one for each MN. We might need more HAs to cope with the load. However, the load that needs to be handled is just the BU/BA and the list of received BUs, the costs of which are relatively insignificant. There will be a net decrease in the number of connections being forwarded through HAs (each connection only ever flows through one HA, so assuming a constant number of connections, that number can’t increase).

This changes which HA is being used (which does not affect scalability) and that many connections need no HA with the new protocol unless the node has moved while the connection is alive. Since many connections are short and begin and end between MN movement events, all of those connections never use a HA, where all connections with regular MIPv6 and a MN which is away from home do. So this new protocol actually reduces the costly part of HA usage, and improves the scalability of the network. Further, since less bytes need to be transmitted when no MIPv6 overheads are added (no triangle routes, no tunnels, no added encryption and no extra RO header), network bandwidth is also saved. Even if more HAs are needed, those are computers, and hence relatively cheap (even when all the running costs are included), compared with bandwidth which is expensive. The new protocol is more scalable than regular MIPv6.

The harder comparison is against not using MIPv6 at all. In that case we have no HAs at all, no BU/BA packets, and no MIPv6 triangle routing or RO overheads to deal with. That is, we necessarily are adding costs (even if less than regular MIPv6) over not having MIPv6 at all. The benefit is in not having connections fail, which in the best case means repeated (or sometimes where possible, restated) connections, and the overheads involved with all of that, and in the worst case total failure of the ability to communicate if a connection is never able to be completed between movements of the MN. That cost is intangible, and impossible to measure.

CHAPTER 5

CONCLUSION AND DISCUSSION

This chapter presents the conclusion of the work applying multiple HAs and home networks to Mobile IPv6. It summarizes the advantages and limitations of the developed module. Besides, it presents a discussion of this work.

5.1. Conclusion

The MN uses an IP address to communication for identity on the network. The MN obtains a new IP address when it moves to a new network, that causes a problem from MN's IP address change. Any active connection (session) is a lost. Also, communication is gone. MIPv6 can solve that problem. It can keep a connection between the MN and a CN by HA on a home network. The HA capture CN's packet, then forward that to the MN. We get a benefit from MIPv6. Also, we get a cost. The MIPv6's cost the MN away from its home network. We need to send any packet via home network every time to keep connections alive.

When the MN stay at home network, we observe no significant cost from MIPv6. The MN can communicate directly to a CN. So we propose a multiple home network for the MN. Also, the MN gets a multiple HAs each network. This solution can reduce the cost of MIPv6.

Finally, we need a new protocol for managing multiple homes and HAs. So we propose the "Automated HA configuration for MIPv6" protocol to this task. This protocol automation finding a HAs on a foreign network. It has a security method for exchanging information between HA and MN to protect from a malefactor. In future 5.1, we show a design network for ten “Automated HA configuration for MIPv6” protocol.

With this system we contribute a method that allows Mobile IP to be more practical, as it would no longer impose any significant costs upon ordinary connections initiated after a node has moved.

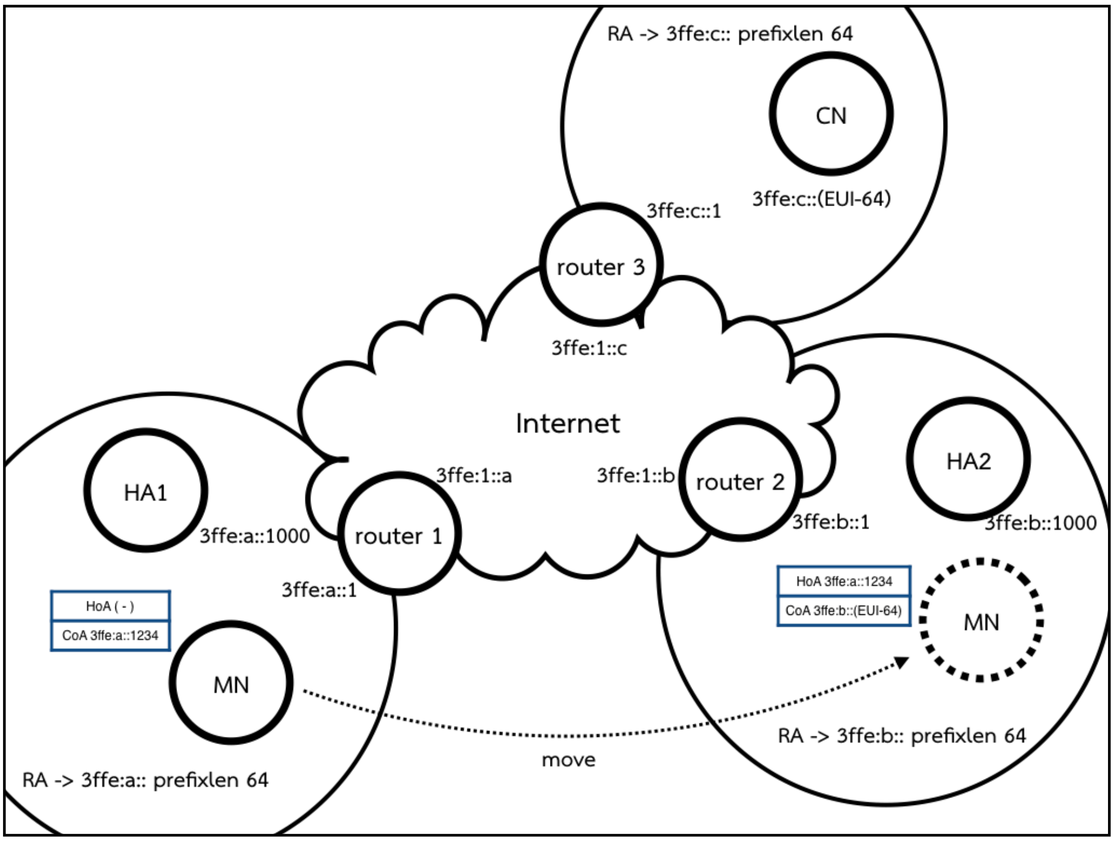


Figure 5.1 Design network fort mechanism.

5.2. Discussion

We have two issues when the MN uses a multiple HAs. First, the MN go to a network that has no available HA on the network. That HA can't found. Or no HA the MN is willing to trust. Or the HA that the MN trusts is not willing to act as the HA for the MN. So we can't set up the HA on the network we are visiting, and normal MIPv6 applies. We mostly simply ignore this case, our method can’t help, but there is one significant difference between normal MIPv6 and MIPv6 with our project. That is, in normal MIPv6, the MN has just one HA to use. Here we might have already visited several networks where we could connect to a new HA. So the MN in our project might already have several associated HA's that it can use shown in figure 5.2.Figures.

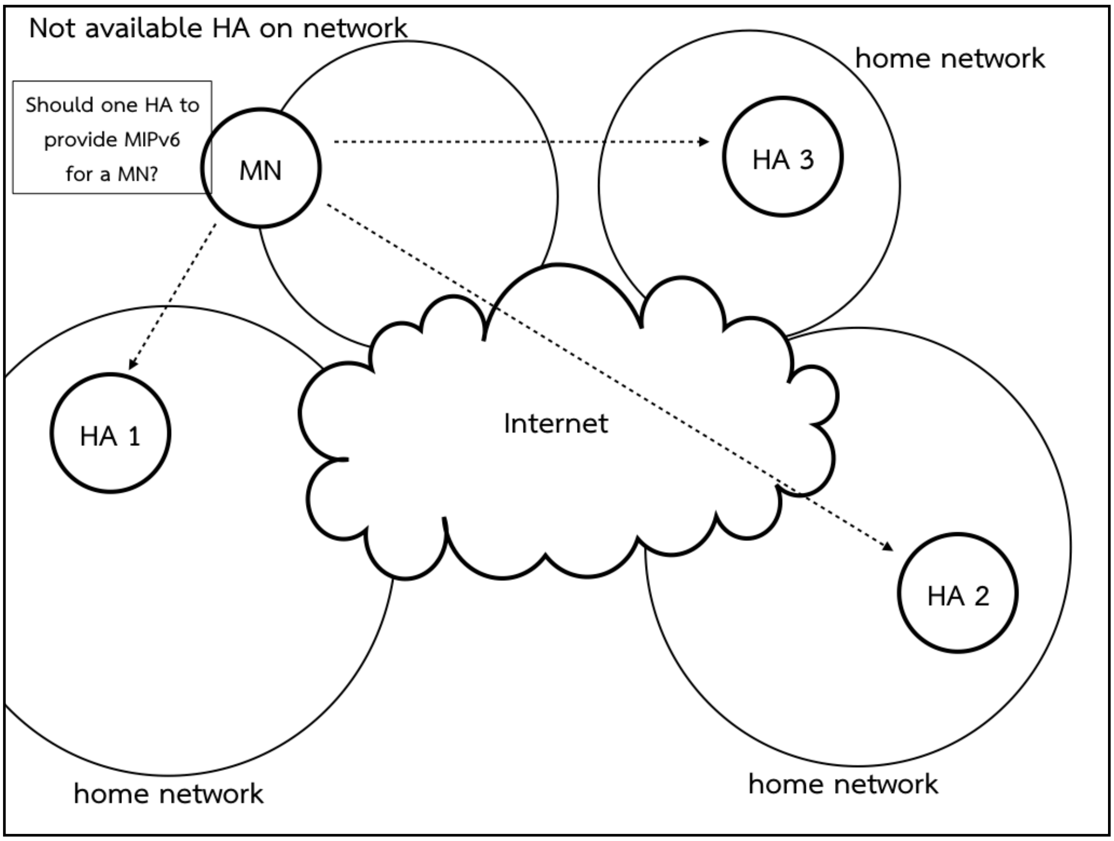

Not available HA on network
Should one HA to provide MIPv6 for a MN?
MN
home network
HA 3
HA 1
Internet
HA 2
home network
home network

Figure 5.2 Should one HA to provide MIPv6 for a MN.

The MN can find a new one HA on the network. It always uses that one for new sessions (existing session always use whatever HA they picked when they started - that's a MIP requirement, as the address the CN sees for the MN comes from the selected HA's network). However, when there is no available HA on the network, we are visiting, and we need to pick one of the HA's we are associated with and use that one for new sessions we create. A study of the methods used to solve that problem has been out of the scope of our project. The implementation of the new protocol, we always use the first HA association. The MN established for connections when a local HA is not available, and leave it for future study to find a better algorithm to choose. The issues that are relevant to the choice.

1. We want to pick the HA that minimizes the network cost. One that is close to either the current network (foreign network) or that is close (or even at) the CN network would be suitable for this.
2. We want one that isn't overloaded if possible.

Second, the MN may obtain many HAs when it has moved much times. We can't delete the HA until a connection completes. We want to select the HA that already being used by other connections rather than idle one. That way, we can drop the idle one if we find. We have too many HA's for saving the cost of updating our location with each of them every time we move the show in figure 5.3. We need to list all of the packets in the new protocol and indicate parameters each one might include — for example, list all of the packet, parameter in packet, kind of data.

The method of choosing old HA’s once they are no longer needed to be released from being a HA is also for future study. Also not a potential problem: even offer a connection ends a CN might remember the MN’s address so it can later establish a return connection. Though the use of NAT with IPv4 makes that kind of usage unlikely land so also unlikely for IPv6.

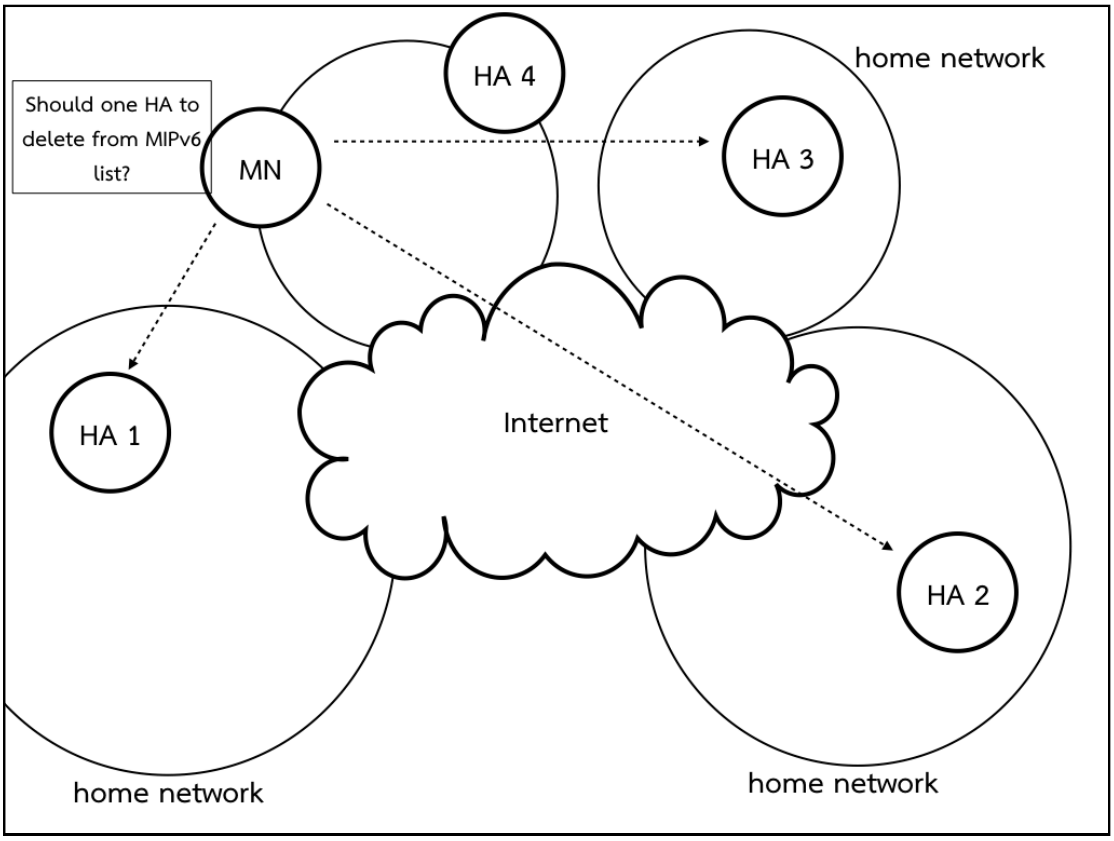


Figure 5.3 Should one HA to delete from MIPv6 list?

REFERENCES

[1] J. Postel, “Internet Protocol”, RFC 791, September 1981.

[2] M. Ramey and K.Kiami, “Less than 10% of IPv4 addresses Remain Unallocated,” Number Resource Organization

[3] S. Deering and R. Hinden, “Internet Protocol Version 6 (IPv6) Specification,” RFC 2460, December 1998.

[4] C. Perkins, “IP Mobility Support for IPv4,” RFC 3220, January 2002.

[5] D. Johnson, C. Perkins and J. Arkko, “Mobility Support in IPv6,” RFC 3775, June 2004

[6] T. Narten, E. Nordmark and W. Simpson, “Neighbor Discovery for IP Version 6 (IPv6),” RFC 2461, December 1998.

[7] M. Dummore, “Final Mobile IPv6 Support Guide,” 6NET IST-2001-32603, December 2004.

[8] D. Johnson and S. Deering, “Reserved IPv6 Subnet Anycast Addresses,” RFC 2526, March 1999.

[9] A. Conta and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," IETF, RFC 2463, 1998.

[10] T. Narten, E. Nordmark, and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)," IETF, RFC 2461, 1998.

[11] S. Hagen, IPv6 Essentials, O'Reilly, The second edition, 2006.

[12] Y. Tseng, J. Jiang, J. Lee, "Secure Bootstrapping and Routing in an IPv6-Based Ad Hoc Network.," Proc. of ICPP Workshops, 2003, pp. 375-383.

[13] R. Droms, J. Bound, B. Volz, T. Lemon, C. E. Perkins, "Dynamic Host Configuration Protocol for IPv6 (DHCPV6)", IETF, RFC 3315, 2003.

[14] P. Nikander, J. Kempf, E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats," IETF, RFC 3756, 2004.

[15] J. Arkko, T. Aura, J. Kempf, V. Mantyla, P. Nikander, and M. Roe, "Securing IPv6 Neighbor and Router Discovery," Proc. of the 3rd ACM workshop on Wireless security, 2002, pp. 77-86.

[16] X. Geng and A. B. Whinston, "Defeating Distributed Denial of Service Attacks," IT Pro, 2000, pp. 36-41.

[17] G. An and J. S. Park, "Packet Marking Based Cooperative Attack Response Service for Effectively Handling Suspicious Traffic" Lecture Notes in Computer Science, Vol. 4318, 2006, pp. 182-195

[18] J. Arkko, J. Kempf, B. Zill, and P. Nikander,"SEcure Neighbor Discovery (SEND)," IETF, RFC 3971, 2005.

[19] 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.

[20] F. Heissenhuber, W. Fritsche, and A. Riedl, “HA redundancy and load balancing in mobile IPv6,” in Proc. 5th International Conf. Broadband Communications, Hong Kong, 1999.

[21] Wakikawa, R., Devarapalli, V. and P. Thubert, “Inter home agents protocol (HAHA)”, IETF draft-wakikawa-mip6-nemo-haha-00.txt (work in progress), October 2003.

[22] Hui Deng, Brian Haley, Xiaodong Duan, Rong Zhang and Kai Zhang, “Load balance for distributed home agents in mobile IPv6”, draft-deng-mipv6-ha-loadbalance-02.txt (work in progress), October 2004.

[23] H. Deng, X. Huang, “A hybrid load balance mechanism for distributed home agents in mobile IPv6”, Proceeding of PIMRC, 2003.

[24] J. Faizan, H. El-Rewini, and M. Khalil, “VHARP: Virtual home agent reliability protocol for mobile IPv6 based networks”, in Proc. International Conf. on Wireless Networks, Communications, and Mobile Computing, Wireless Com 2005, Hawaii, USA, June 13-16, 2005.

[25] J. Faizan, H. El-Rewini, and M. Khalil, “Introducing reliability and load balancing in mobile IPv6 based networks”, Wireless Communications and Mobile Computing, Nov. 2006.

[26] R. Wakikawa. “Home agent reliability protocol”. IETF draft-ietf-mip6-hareliability-03.txt, November 2007. (working in progress)

[27] Jong-Hyouk Lee, Tai-Myoung Chung. “Performance evaluation of distributed multiple home agents with HAHA Protocol”. International Journal of Network Management, Vol. 17, Pp 107-115. 2007.

[28] H.Ahn, CS.Hwang. “Efficient fault-tolerant protocol for mobility agents in mobile IP”. International Parallel and Distributed Processing Symposium (IPDPS). 2001.

[29] H. Shi and S. Goto, "An Implementation of Multiple Home Agents Mechanism in Mobile IPv6," Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007. TridentCom 2007. 3rd International Conference on, Lake Buena Vista, FL, 2007, pp. 1-9.

[30] F. Heissenhuber, W. Fritsche and A. Ried. “Home Agent Redundancy and Load

Balancing in Mobile IPv6”. in Proc. 5th International Conf. Broadband Communicatons Hong Kong. 1999.

[31] J. Faizan, H. El-Rewini, and M. Khalil, “Efficient dynamic load balancing for multiple home agents in mobile IPv6 based network”, In Proceedings of IEEE International Conference on Pervasive Services ICPS 2005, Santorini, Greece, July 11-14, 2005.

[32] A. Vasilache, J. Li, and H. Kameda, “Threshold-based load balancing for multiple HAs in mobile IP networks,” Telecommunications Systems, vol. 22, issue 1-4, pp. 11-31, January-April, 2003.

[33] Zhang, Hanwen, Chao Ma, Yujun Zhang, and Zhongcheng Li. "A multiple home agent deployment scheme to enhance service availability for MIPv6." In Communication Systems, 2008. ICCS 2008. 11th IEEE Singapore International Conference on, pp. 1422-1426. IEEE, 2008.

[34] S. Keiichi. 2006. SHISA: the IPv6 mobility framework for BSD operating systems. International Multi-Conference on Computing in the Global Information Technology - (ICCGI’06) : 2-2.

[35] Welcher, Peter. 2009. TCP/IP performance factors. Net Craftsmen. https:// www.netcraftsmen.com/tcpip-performance-factors. (June 5, 2019).

[36] E. Rescorla. "HTTP Over TLS". RFC 2818, May 2000.

[37] WIDE project. 2007. SHISA. http://www.mobileip.jp. (January 15, 2019).

[38] Keiichi, Shima. 2006. SHISA: The IPv6 mobility framework for BSD operating systems. IPv6 Today - Technology and Deployment (IPv6TD'06).

[39] J. Arkko, V. Devarapalli and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling between Mobile Node and Home Agent," RFC 3776, June 2004.

[40] O'Reilly, IPv6 Essentials 0-596-10058-2. Chapter 11: Mobile IPv6.

[41] P. Nikander, J. Arkko and T. Aura, "Mobile IP version 6 Route Optimization Security Design Background," RFC 4225, December 2005.

[42] T. Aura. "Mobile IPv6 Security," Proceeding of the Security Protocols, 10th International Workshop, Cambridge, UK, April 2002.

[43] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)," RFC 2409, 4November 1998.

[44] FreeBSD [Online], http://www.freebsd.org

[45] Keith Sklower. A Tree-based Packet Routing Table for Berkeley UNIX. In Proceedings of the Winter 1991 USENIX Conference, pages 93-103. USENIX Association, January 1991.

VITAE

Name	Mr. Wuttipon Noopetch
Student ID	5710120069

Educational Attainment

Degree	**Name of Institution**	**Year of Graduation**
Bachelor of Engineering (Computer Engineering)	Prince of Songkla University	2008

List of Publication and Proceedings

W. Noopetch, S. Kamolphiwong, “Automated HA configuration for MIPv6,” in Proceeding of The Ninth International Conference of Genetic and Evolutionary Computing (ICGEC 2015), Myanmar, Aug. 2015