# UNDERSTANDING IPv6

Youngsong Mun[1] and Hyewon K. Lee[2]
[1]Soongsil University, Seoul, Korea; [2]Daejin University, Kyungki, Korea

Springer

Youngsong Mun
Soongsil University
Seoul, Korea

Hyewon K. Lee
Daejin University
Kyungki, Korea

Library of Congress Cataloging-in-Publication Data

Mun, Youngsong, 1960-
Understanding IPv6 / Youngsong Mun and Hyewon K. Lee.
p. cm.
Includes bibliographical references and index.
ISBN 0-387-25429-3 (alk. paper)
1. TCP/IP (Computer network protocol) 2. Internet. I. Lee, Hyewon K., 1975- II. Title.

TK5105.585.M863 2005-04-04
004.6'2--cd22

2005046556

ISBN 0-387-25429-3 e-ISBN 0-387-25614-8 Printed on acid-free paper.
ISBN 978-0387-25429-6

Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 11366232

springeronline.com

# Contents

# Preface

IPv6 is a new Internet protocol which is expected to replace current IPv4 protocol. IPv6 has been implemented in experimental networks in many countries and even IPv6 commercial services are available in some countries. Since the current internet protocol IPv4 has been developed for small sized closed networks, it has faced various problems when it has been applied to the worldwide public networks.

Current IPv4 protocol becomes too complex since the various additional protocols must be installed with IPv4 for IP based equipments to operate properly. Several protocols such as Address Resolution Protocol (ARP) and Internet Control Message Protocol (ICMP) are needed to be operated with IPv4 for the proper operation of IPv4 enabled devices. Address shortage is the most serious problem in IPv4. There is tremendously increasing number of Internet users in developing countries. There are also new emerging applications requiring enormous number of IP addresses. Address shortage problems must be overcome to accommodate these advanced applications and address demands of many countries. Network Address Translation (NAT) and Classless Inter Domain Routing (CIDR) are devised to increase the effective number of IP addresses. However, they can not be a permanent solution to the address shortage problem.

While original IPv4 is designed based on simplicity and additional functions have been defined in separate protocols, IPv6 integrates these additional functions into itself. Additionally, IPv6 itself holds lots of new or modified features. Autoconfiguration feature is very attractive in terms of network management cost. IPv6 also has the strong support of mobility as well as security. IPv6 offers practically infinite number of IP addresses.

Since features of IPv6 are complex and diverse, readers should go through a number of related documents defined by Internet Engineering Task Force (IETF) to understand this protocol. Understanding IPv6 is especially difficult due to the fact that there are many related protocols to be mastered and IPv6 standards itself are not fixed yet. Thus, this book tries to be inclusive of all necessary materials to understand IPv6. Through 12 chapters, this book helps students and engineers to see IPv6 world easily and quickly. This book contains most up-to-date information as well as the general knowledge about IPv6.

Important recent change in scoped address architecture such that site local address is recently deprecated must be noted. Optimistic duplicate address detection as well as general duplicate address detection is covered in this book. Each transition mechanism has its own role. However, among various transition mechanisms, Teredo is considered as the most promising one which will have the widest application. However, since Teredo is recently proposed, it is only covered in the most recent books. Domain name system (DNS) is actually not the integral part of Internet protocol. However, popular applications such as web service can not be operated without DNS. A new record type for DNS for IPv6 'AAAA' must be studied. Hierarchical mobility management as well as fast handoff is very important for mobile IPv6 for the seamless mobility management. Early binding update scheme is also covered in our book. Security features such as Virtual Private Network (VPN) traversal and firewall traversal are essential features for the real deployment of mobile IP. Content of each chapter is as follows.

Chapter 1 firstly explains reasons to develop IPv6. Address shortage problem is the most outstanding reason. To integrate various additional protocols which are needed for IPv4 is the other reason. Chapter 2 covers protocol architecture. IPv4 and IPv6 header formats are compared. IPv6 header format is explained in detail. Chapter 3 covers addressing architecture. It covers scoped address architecture as well as address types such as unicast, multicast, and anycast address types. IPv6 address with embedded IPv4 address is also explained.

Chapter 4 covers ICMPv6 in detail. IPv6 requires ICMP as IPv4 does, but several changes are made for IPv6. New protocol, ICMPv6 is defined in RFC 2463. ICMPv6 is also mainly used to report errors encountered in processing packets, and performs diagnostic functions. This protocol plays very important role in Neighbor Discovery (ND) protocol and Path MTU Discovery protocol.

Core features of IPv6, neighbor discovery and address autoconfiguration are covered in Chapter 5 and Chapter 6, respectively. IPv4 hosts needs ARP, RARP protocols to be connected to networks. Those separate protocols are integrated into the Neighbor Discovery feature of IPv6. Neighbor Discovery protocol of IPv6 includes new features such as Neighbor Unreachabilty Detection (NUD) and Duplicate Address Detection (DAD) as well as traditional address resolution features. Address autoconfiguration is one of the most attractive features of IPv6 since it can significantly save network management costs. Autoconfiguration is composed of a sequence of processes which create a link-local address, verify and guarantee the uniqueness of assigned addresses, determine which information should automatically be configured, and decide whether stateless, stateful or both address configuration mechanism would be adopted. It is hard to consider IPv6 and autoconfiguration separately. Chapter 6 explains how to build link-local and global addresses, and how to guarantee uniqueness on the address assignment. Both stateless and stateful address autoconfiguration are explained. Optimistic DAD to reduce the address configuration time is also explained in this chapter.

Chapter 7 covers DHCPv6. For the conventional role of DHCP in IPv4 to configure newly introduced hosts, DHCP allows managing address resources and related information in a concentrated manner, resulting in reducing network management costs. DHCP in IPv6 is widely accepted as a vehicle to solve various configuration tasks including bootstrapping and prefix delegation in addition to the conventional role of DHCP in IPv4. DHCPv6 is classified as a stateful configuration protocol.

Gradual deployment of IPv6 while providing uninterrupted IPv4 services is expected to happen. Thus IPv4 and IPv6 will exist together for a significant amount of time through transition mechanisms. Various interconnection mechanisms provide interoperability between IPv4 and IPv6 entities throughout the IPv4/IPv6 mixed network environment. They may be classified into two groups; tunneling mechanisms and translation mechanisms. Tunneling mechanisms help isolated IPv6 nodes or IPv6 sites to communicate over IPv4 networks, and translation mechanisms allow IPv4 and IPv6 nodes to communicate. Transition mechanisms are covered in detail in Chapter 8.

DNS has been developed as a systematical delegation model and distributed domain database with hierarchy. Hierarchical name space supports flexible structure with additional extensions. Distributed database architecture augments management efficiency. DNS is actually not the integral part of internet protocol. However, popular applications such as web service can not be operated without DNS. New record type, 'AAAA' is defined to support 128-bit IPv6 address in DNS. The 'AAAA' record is just an extension of present 'A' record type. Thus most DNS entities will handle the new record type without much trouble. Chapter 9 covers modifications in DNS for IPv6.

Chapters 10, 11, and 12 are related to mobile IPv6 (MIPv6). Basic mobile IPv6 is covered in detail in Chapter 10. Route optimization is supported in MIPv6 while that is not true anymore in mobile IPv4. For the basic authentication support in case of route optimization, return routability mechanism is devised. Foreign agents are not necessary in MIPv6. Mobile nodes can form Care of address (CoA) using IPv6 stateless autoconfiguration protocols. Routing Header should be used to send packets from correspondent nodes to mobile nodes. For the MIPv6, type 2 Routing Header is defined.

Chapter 11 covers enhanced handover schemes for MIPv6. They are optimization mechanisms for the fast and seamless handover such as hierarchical MIPv6 (HMIPv6), fast handoff, and early binding update. HMIPv6 requires new entities called Mobility Anchor Points (MAPs) in the visiting network. MAP acts as a local home agent. Mobile nodes have to perform binding updates to home agents and correspondent nodes only when it firstly enters into a MAP domain. When mobile nodes move inside the MAP domain, binding updates to the MAP are only needed. In the fast handover, several portions of the layer 3 handover are performed in advance prior to the handover, such as the new care-of address configuration and the movement detection to reduce the handover latency. Early binding update scheme has been proposed to reduce latencies due to return routability procedure.

Important security issues such as VPN traversal, firewall traversal and cryptographically generated address are covered in Chapter 12. Since return routability scheme in MIPv6 bears shortcomings such as long latencies and weakness to the man-in-the-middle attack, better security mechanisms for the proof of ownership of home addresses are needed. Cryptographically generated address receives widest consensus as optional security mechanism for MIPv6.

This book has been developed through the several semesters of graduate class in the School of Computing at Soongsil University. We would like to thanks to all graduate students who have taken our classes. We would like to give special thanks to Dr. Young-Jin Kim who is a IPv6 Forum Technical Directorate for his enlightening discussions.

This book is in no way faultless. Thus, we will appreciate readers who report any errors in this book to us.

Youngsong Mun
Hyewon Keren Lee
February 2005

Chapter 1

# THE AGE OF IPv6

## 1.1 INTRODUCTION

The current internet protocol, IP version 4 or shortly IPv4 was born to interconnect educational and government institutions in United States. Thus, the original IP protocol is naturally light and simple. Since the IPv4 based Internet is widely used throughout the world, current status of IPv4 is beyond imagination of the initial motivation to create it.

As IP protocol is applied to the bigger network, internet engineers have realized that the original features of IP protocol are not enough. This becomes getting worse when IP protocol is adopted to open public networks. For the commercial network service, IP protocol needs more features to satisfy user demands for a better service. Since IP protocol was defined in Request for Comments (RFC) 791 in 1981, a lot of extensions and additional protocols have been added, such as path Maximum Transmission Unit (MTU) discovery, super-netting, Address Resolution Protocol (ARP), Dynamic Host Configuration Protocol (DHCP), Network Address Translation (NAT), etc. whenever necessity arises. These extensions as well as additional protocols are defined in separate documents by Internet Engineering Task Force (IETF).[1]

[1] The IETF is a large open international community of network designers, operators, vendors, and researchers related to the evolution of the Internet architecture and the smooth operation of the Internet.

However, cooperating with new extensions and additional protocols may not be appropriate any more due to the huge overhead for compatibility with existing protocols. In other words, simple and light IPv4 protocol working with lots of additional protocols becomes very complicated to be implemented and operated properly on various devices in Internet.

Even worse, Internet users have increased by the geometric progression every year, which incurs shortage in address resources. IPv4 uses 32-bit identifier so that there are 4.3 billion identifiers numerically, but address allocation based on classes allows only 0.5 or 1 billion address space. CIDR[2] is devised to overcome the inefficiency caused by the assignment by classes. Address increase by adopting CIDR is minimal. Using private addresses may also increase the effective address space. For this purpose NAT is devised. NAT maps public IP addresses with IP private addresses. However, it has shortcomings such as preventing end to end communications. Web server can not be operated with private addresses.

Applications requiring enormous number of public IP addresses are emerging such as p-to-p communications, telematics and home networkings. In the home networking it is desirable for every appliance in the home to have public IP address. Various parts of cars can have IP addresses in the future telematics. Even in the cellular communications, mobile terminals are desirable to be IP terminals to provide advanced services such as push services. In 4G mobile communications, all mobile terminals are expected to be IP terminals.

Asia as well as Europe is facing IP address shortage problems. When considering the ratio of the number of IP addresses to the population, Asia and Europe have relatively much less number of IP addresses than US has. This could be very serious especially in Asia. China's population is approximately 1.2 billion while the number of allocated IP addresses is much less than 0.1 billion. There is no possibility that those countries will get enough public IP addresses they need in the future. This fact may create disaster as China becomes developed in a very fast pace. Japan and Korea also have IP address shortage problem. Simply, there is no way using IPv4 to accommodate billions of potential users and devices.

Now, many Internet experts predict that the current IPv4 address pool will be depleted in a few years if the current assignment trend is maintained. ASO[3] predicts that IPv4 address space will be depleted in 2007 when address

[2] CIDR (Classless Inter Domain Routing)

[3] ASO (Address Supporting Organization)[1]

allocation rate from Regional Internet Registry (RIR)[4] to Local Internet Registry (LIR) is assumed to be increased by 1.3 times, while BT and HITACHI predict it as 2006. Some argues that IPv4 address space will not be depleted until 2020 by using address saving techniques such as CIDR and private addresses. However, this analysis does not consider new emerging applications requiring enormous number of IP addresses as well as fast developing countries like China.

When we consider problems explained above, it is evident that we need a new IP protocol. Current IPv4 protocol becomes too complex since the various additional protocols must be installed with IPv4 to make IP based equipments operate properly. Address shortage problems must be overcome to accommodate advanced applications and address demands of many countries. New internet protocol, IPv6, formerly called next generation internet protocol, was proposed from early 1990s as a successor of IPv4 and published as RFC 1752.[2] Later on, IPv6 protocol is standardized in RFC 1883 which eventually becomes obsolete by RFC 2460.[3] While original IPv4 is designed based on simplicity and additional functions have been defined in separate protocols, IPv6 integrates these additional functions into itself.

There are innumerable user requirements, such as direct access to Internet, flexibility, scalability, stability, simple and effective operation and maintenance, security, mobility, etc. To embrace these various requirements, IPv6 is designed to support and integrate address autoconfiguration, address extensibility, effective hierarchical address management, renumbering, multicast, mobility, integrated security, and so on.

In a long term, current IP network is expected to evolve to IPv6. Evolution from IPv4 to IPv6 network is briefly sketched in Fig. 1-1. At the initial stage, some IPv6 experimental networks or just IPv6 terminals are present in IPv4 network. In the next evolution stage, some IPv6 islands show up, more and more connected IPv6 islands and IPv6 networks appear, and IPv6 networks will be predominant over IPv4 networks. Interoperability mechanisms are required, and signaling protocols will be introduced in this stage. Only IPv6 networks with some isolated IPv4 sites or terminals will exist at the final stage.

To introduce IPv6 to current IPv4 based networks, compatibility between IPv4 and IPv6 networks should be implemented at first, and address allocation policy from RIR to LIR and from LIR to ISP should be researched

[4] RIR is responsible to provide address allocation and registration to LIRs or ISPs. There are four RIRs, such as APNIC (Asia Pacific Network Information Centre), ARNIN (American Registry for Internet Numbers), LACNIC (Latin American and Caribbean Internet Address Registry), and RIPE-NCC (RIPE Network Coordination Centre).

and realized under natural evolution to IPv6. Besides, gradual progresses, such as application programs running on completely different surroundings or Domain Name System (DNS) to allow IPv6 query for IPv4 or IPv6 address or opposite lookup, should follow.[4]

Transition to IPv6 has been recently implemented. However, there are some Internet pioneers and experts who still have doubt about IPv6 and believe that NAT, Application Level Gateway (ALG) and DHCP protocols will help IPv4 in solving the address shortage problem resulted from 32-bit address. They also argue that employing IPv6 protocol for Internet is too impetuous because IPv4 can do whatever IPv6 does, and the Internet may become unstable because the IPv6 is not mature. There are also criticisms about equipments for transition mechanisms such as tunnel broker and NAT-PT because they may be the performance bottleneck and open new security holes.

Some arguments may be partly true. It is a kind of chicken and egg problems between infrastructure and applications. Since IPv6 is proposed and standardized, many experiments have been performed. 6bone[5] is exemplary experimental network. For the coexistence of IPv4 and IPv6, various transition mechanisms, such as tunneling and translation, have been proposed and standardized. Hardware including routers and transition equipments has been developed and tested by leading hardware vendors. However, these movements and experiments are not enough for commercial IPv6 infrastructure to be deployed. Further, applications to justify advantages and troubles in adopting IPv6 have not been even emerged. This is the main reason why IPv6 is mainly recognized as the means of overcoming address shortage problem, However, many countries labor to develop and spread IPv6. For example, IPv6 and mobile IPv6 on hardware chip, IPv6-ready appliances, many software solutions, various application programs, and compatibility with 3GPP/3GPP2 have been researched and implemented.

In North America, the networking hardware vendors, such as Cisco and Juniper, have successfully introduced IPv6 equipments into the market. Operating systems, such as Linux and Windows, also support IPv6 in their latest releases. Department of Defense of U.S plans to convert its network to IPv6 by 2009.

Since 1998, various research projects for IPv6 adoption have been performed in European nations. Nokia and Ericsson are leading the mobile technology and service development based on IPv6. 6WIND is manufacturing IPv6 routers.[6-9]

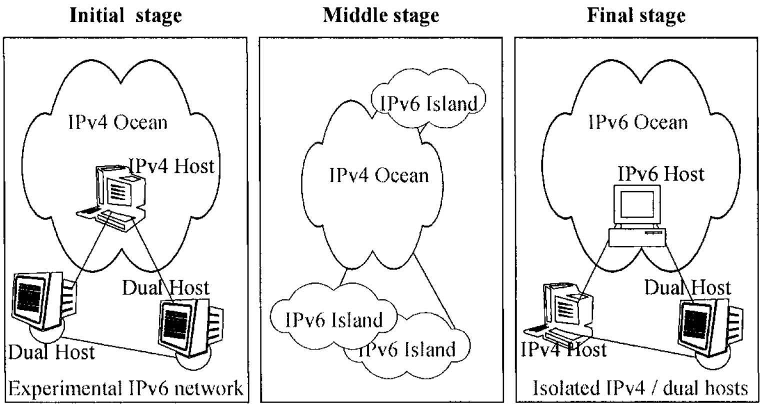


*Figure 1-1.* Evolution from IPv4 to IPv6 networks.

Japanese government gives tax cut to companies for buying IPv6 equipments. Hitachi is leading in manufacturing IPv6 routers in Japan. NTT provides the first commercial IPv6 connections. Japan is especially strong in experiments on combining consumer appliances with the IPv6.

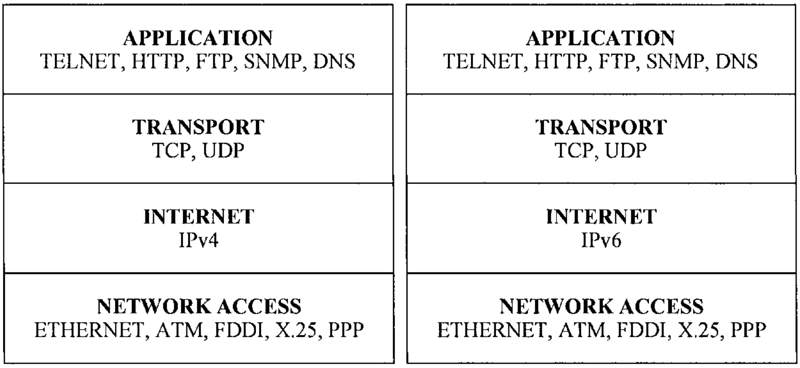
According to statistic data from CAIDA, 93% of IPv6 addresses are allocated to EU, North America, and Asia, where 37% to Europe, 35% to North America, and 21% to Asia by September 2003.[10]

IPv4 is a kind of inheritances from the old ages. To satisfy more requests from users, to supply many IP addresses, and to utilize current fast physical infrastructure, it's the time to move on. It seems that such hesitations with reasons stated above, absence on IPv6 applications, and no name lookup service for IPv6 mainly sit heavy on transition to IPv6.

Under the slogan, 'Ubiquitous, Everything over IP' to connect various devices on networks via Internet becomes a big issue in the world market. To provide plentiful addresses to billions of devices, shifting to IPv6 seems to be inevitable.

## 1.2 PROTOCOL STACK

In principle, no other layer except IP layer need be changed when IPv6 protocol is adopted instead of IPv4. This is because each layer is managed and operated independently. The TCP/IP[11] protocol stacks based on IPv4 and IPv6 are shown in Fig. 1-2. However, in the dual stack nodes, applications may aware desirable IP protocol stack.



| (a) TCP/IP protocol stack based on IPv4 | (b) TCP/IP protocol stack based on IPv6 |
|---|---|
| **APPLICATION**<br>TELNET, HTTP, FTP, SNMP, DNS | **APPLICATION**<br>TELNET, HTTP, FTP, SNMP, DNS |
| **TRANSPORT**<br>TCP, UDP | **TRANSPORT**<br>TCP, UDP |
| **INTERNET**<br>IPv4 | **INTERNET**<br>IPv6 |
| **NETWORK ACCESS**<br>ETHERNET, ATM, FDDI, X.25, PPP | **NETWORK ACCESS**<br>ETHERNET, ATM, FDDI, X.25, PPP |

*Figure 1-2.* TCP/IP protocol stack.

The main function of the internet layer in TCP/IP protocol stack is to provide unique network identifier over entire networks in the world, and route packets to a destination based on the destination address. IPv6 uses 128-bit length address, which provides virtually infinite address space. Autoconfiguration in every IPv6-enabled device is provided in IPv6. Neighbor discovery is also automatically provided for IPv6 nodes to find out neighbors on the same link.

Some extensions or changes are required in the application layer to adopt IPv6 protocol. For example, there are HTTP protocol for web service, SMTP protocol for e-mail service, and FTP protocol for file transfer. From the users' point of view, no changes should be required for existing applications when they adopt IPv6 protocol instead of IPv4. However, some application protocols need modifications or extensions if they include IP address in its data portion of a message. Typical protocol which needs extensions for IPv6 is FTP.

In case of the transport layer, no modifications are required to TCP or UDP to work over IPv6. Furthermore, same port numbers may be used in IPv6 environment. The port number is contained in Protocol field of IPv4 Header. However, the port number is delivered in the Next Header field of IPv6 Header.

At the data link layer, no changes are necessary because IPv6 protocol is independent on the physical medium and data link layer protocols. For example, the most widely used protocol in the data link layer, Ethernet, is also compatible with IPv6. Just different value in the Type field in Ethernet Header is defined: 0x0800 indicates IPv4 protocol while 0x86DD indicates

IPv6 protocol. However, for some special purposes, incorporation between IP and data link layers is desirable. For example, IP layer gets triggers from the data link layer for the fast handover in the Mobile IPv6. Triggers from the data link layer may expedite actions in the IP layer.

In addition to the motivation to cope with the address exhaustion problem, IPv6 has various advantages over IPv4. Address autoconfiguration capability is considered one of prominent features. This feature is expected to significantly reduce management costs. Protocols to guarantee Quality of Service (QoS) are separately defined in IPv4. However, IPv6 gives efforts to integrate this feature using the flow label. Integrated security is another important reason to evolve to IPv6. Mobility is optimized and secured in IPv6. Routing in the Mobile IPv4 can not be optimized due to the security reasons. These essential technical advantages of IPv6 will be described in detail in Chapter 2.

## 1.3 CONCLUSIONS

Many additional protocols are needed for the proper operation of IPv4 because original IPv4 was designed based on simplicity. This causes numerous problems such as the increased complexity of operation and scalability problem. However, IPv6, standardized at RFC 2460 integrates these additional functions into IP protocol. IPv6 also provides various advantages such as extended address space, autoconfiguration, simplified header format, interoperability, integrated security, and route optimization for mobile terminals. Some internet experts still argue about the necessity of IPv6. Some believe IPv4 is able to solve any problem or any additional functions which are provided by IPv6. However, transition to IPv6 world is inevitable because billions of potential users and devices requiring IP addresses are expected to emerge in the near future. Further, to satisfy innumerous user demands for the next generation Internet, IPv4 seems to be already complex to annex new functions through separated protocols.

## REFERENCES

1. ASO, http://aso.icann.org.
2. S. Bradner and A. Mankin, The Recommendation for the IP Next Generation Protocol, RFC 1752 (January 1995).
3. S. Deering and R. Hinden, Internet Protocol Version 6 (IPv6) Specification, RFC 2460 (December 1998).
4. F. Baker, E. Lear, and R. Droms, Procedures for Renumbering an IPv6 Network without a Flag Day, work in progress (February 2004).

5. 6BONE, http://www.6bone.net.
6. IPv6, http://www.ipv6.org.
7. IPv6FR, http://www.ipv6forum.com.
8. 6NET, http://www.6net.org.
9. IPv6TF, http://www.ipv6tf.org.
10. CAIDA, http://www.caida.org.
11. D. Comer, Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture (Prentice Hall, 1994).

Chapter 2

# PROTOCOL ARCHITECTURE

## 2.1 INTRODUCTION

IPv6 is a new version of Internet protocol which is expected to substitute IPv4. It is very difficult to predict exactly when IPv4 will eventually come to an end. However, most experts in internet protocol area expect that the coexistence of two protocols will be lasted for a long time. As a successor of IPv4, IPv6 will bring lots of advantage over IPv4 as follows:[1]

- Expanded addressing capabilities
  The size of IPv6 address is 128-bit length and four times as long as the address length of IPv4. This increased length enables various desirable features in IPv6 possible, such as hierarchical delegation and management of addressing space, virtually unlimited number of address assignment to internet devices, and autoconfiguration of internet devices. Exaggeratingly, it may be said that stars in the sky are addressable with IPv6 addressing space. Address autoconfiguration and huge addressing space are mostly emphasized points in IPv6.[2]
- Simplified header format
  The size of IPv4 Header is ranged from 20 bytes to 20 plus option field length, but the option field is highly variable to know or determine in advance from where the data field will start. Some fields in IPv4 Header are dropped or moved into Option Header to simplify and quantify to reduce the common processing cost of packets in IPv6.

<table>
<tr><td><b>Version</b></td><td><b>IHL</b></td><td><b>Type of Service</b></td><td colspan="3"><b>Total Length</b></td></tr>
<tr><td colspan="3"><b>Identification</b></td><td><b>Flags</b></td><td colspan="2"><b>Fragment Offset</b></td></tr>
<tr><td colspan="2"><b>Time to Live</b></td><td><b>Protocol</b></td><td colspan="3"><b>Header Checksum</b></td></tr>
<tr><td colspan="6"><b>Source Address</b></td></tr>
<tr><td colspan="6"><b>Destination Address</b></td></tr>
<tr><td colspan="5"><b>Options</b></td><td><b>Padding</b></td></tr>
</table>

(a) IPv4 Header format

<table>
<tr><td><b>Version</b></td><td><b>Traffic Class</b></td><td colspan="3"><b>Flow Label</b></td></tr>
<tr><td colspan="3"><b>Payload Length</b></td><td><b>Next Header</b></td><td><b>Hop Limit</b></td></tr>
<tr><td colspan="5"><b>Source Address</b></td></tr>
<tr><td colspan="5"><b>Destination Address</b></td></tr>
</table>

(b) IPv6 Header format

*Figure 2-1.* Header formats.

- Autoconfiguration
  The autoconfiguration is the most fascinating point in adopting IPv6. IPv6 enabled device is able to configure itself dynamically when it plugs in. IPv6 interface is given with several identifiers in accordance with scope or number of receivers, such as link-local address, global address, multicast address, anycast address, and so on. When an IPv6 device boots, it automatically configures its link-local address with several multicast addresses and gets or builds its global IP address.
- Providing Quality of Service (QoS)

To provide quality of service in data transmissions, the flow label is defined in IPv6. Flow labels are pre-defined labels to classify data packets to settle quality requests from communicating peers.[3] Type of Service (ToS) field in IPv4 Header format is defined for the similar purpose. However, most IPv4 routers do not support this field. Thus, Integrated Services (Int-Serv) and Differentiated Services (Diff-Serv) are devised to support QoS in IP networks. For end-to-end QoS support, all the routers on the path from the sender to the receiver must support QoS mechanism. However, in IPv4, these schemes meet difficulty in widely deployed. All the IPv6 routers must support IPv6 QoS mechanisms. Diff-Serv is one of the methods which are currently under consideration for the flow label.

- Integrated security
  For the network security in IPv4, IPsec is devised. It is widely used for the Virtual Private Network (VPN). However, its support is optional in IPv4. IPsec is mandatory in IPv6.
- Enhanced mobility
  Route optimization is possible in the Mobile IPv6. Authentication mechanisms for the mobile node are provided in route optimization process.

*Table 2-1.* Significant changes from IPv4 header to IPv6 header (length in bits).

| Field in IPv4 | Field in IPv6 | Descriptions |
|---|---|---|
| Version (4) | Same | The Version field specifies the IP version of packet and helps intermediate routers to determine how to interpret remaining packet. |
| IHL (4) | Obsolete | The 4-bit Internet Header Length specifies the length of the header in 8-byte units including Options. The minimum value is 5, and maximum header length will be 60 bytes because IHL field is 4-bit (15×8=60 bytes). When no options are used, IHL field is set to 5. |
| Type of Service (8) | Traffic Class (8) | Used to specify different types of IP packets and provide quality of service. |
| – | Flow Label | The Flow Label field is newly defined in IPv6. This field is used to handle different quality requests from users. |
| Total Length (16) | Payload Length (16) | The Total Length field specifies the total length of the IP packet including IP Header in bytes, and Payload Length field specifies the length of IP packet excluding IP Header. Extension Headers in IPv6 are also considered as IP payload. In IPv4, with Total Length and ILH fields, we know where the data portion of packet starts. |

*Table 2-1.* Cont.

| Field in IPv4 | Field in IPv6 | Descriptions |
|---|---|---|
| Identification (16) | Moved to Extension Header; Fragment Header | Identification specifies a value assigned by the sender and helps to assemble the fragmented packets. |
| Flags (3) | Moved to Extension Header; Fragment Header | Flags specify various control flag for fragmentation. |
| Fragment Offset (13) | Moved to Extension Header; Fragment Header | Fragment Offset specifies where a fragment belongs to. |
| Time to Live (8) | Hop Limit (8) | Time to live (TTL) field specifies how long a packet is allowed to remain in the Internet. In IPv6, Hop Limit replaces Time to Live and specifies how many hops a packet is allowed to be forwarded. |
| Protocol (8) | Next Header (8) | The Protocol field specifies the next protocol in the data field of a packet. In IPv6, Next Header field is defined, which indicates next following header type. |
| Header Checksum (16) | Obsolete | Header Checksum includes only checksum for header. Some header parts, such as TTL, may be changed. Thus Header Checksum should be recomputed and verified at each point wherever the header is processed. |
| Source Address (32) | Source Address (128) | This field specifies a source address. |
| Destination Address (32) | Destination Address (128) | This field specifies a destination address. |
| Option (variable) | Moved to Extension Header | Option field in IPv4 is a variable length field, and it has limits to express option enough. To substitute the Option of IPv4 Header, two Options Headers are defined in IPv6: Destination options and Hop-by-Hop options Headers. |

## 2.2 COMPARISONS OF IP HEADER FORMATS

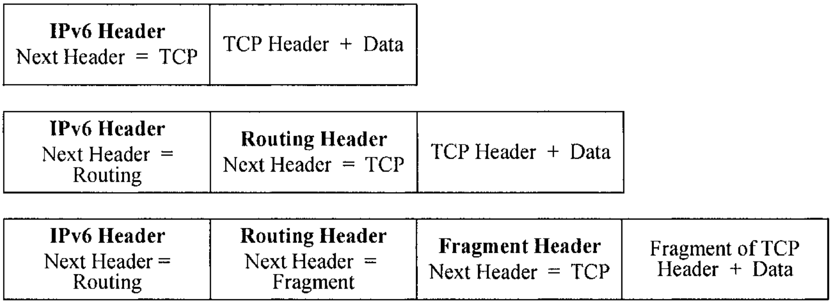
The IPv4 and IPv6 Headers are shown in Fig. 2-1. Table 2-1 compares fields in IPv4 Header with ones in IPv6 Header. Some fields from IPv4 Header are dropped completely and obsolete, moved to Extension Headers, or given with different field names with slightly modified functions. Besides, 20-bit Flow Label field is a newly introduced in IPv6 and still under development. This field is explained in detail in the following section.

## 2.3 EXTENSION HEADERS

In IPv6, several Extension Headers are defined to separate different optional internet-layer information. These headers are located between IPv6 Header and upper-layer header such as TCP or UDP Header. Each Extension Header is identified by distinct Next Header value, and one or more Extension Headers may be included in a packet. The Next Header field uses the same value as the value defined in RFC 790,[4] which is called assigned numbers. The assigned number is specified in Appendix A.

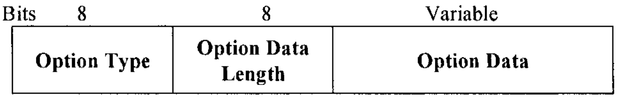
Processing the Next Header field of IPv6 Header provokes to process the first Extension Header following the IPv6 Header, and the Next Header field of Extension Header will do the same thing to the next Extension Header, and so on, until the upper layer header is present. Contents of each Extension Header notify whether processing next header is necessary. Thus, Extension Headers in a packet should be handled by the order of appearance in the packet.

Most of Extension Headers are handled by a node identified from the Destination Address field of IPv6 Header. Only Hop-by-Hop Options Header should be processed by all intermediate nodes including the final destination, because this Options Header contains information which must be examined at every single node along the delivery path to the destination.



| | | | |
|---|---|---|---|
| **IPv6 Header**<br>Next Header = TCP | TCP Header + Data | | |
| **IPv6 Header**<br>Next Header = Routing | **Routing Header**<br>Next Header = TCP | TCP Header + Data | |
| **IPv6 Header**<br>Next Header = Routing | **Routing Header**<br>Next Header = Fragment | **Fragment Header**<br>Next Header = TCP | Fragment of TCP Header + Data |

*Figure 2-2.* Example of Extension Headers in a packet.



| Bits 8 | 8 | Variable |
|---|---|---|
| **Option Type** | **Option Data Length** | **Option Data** |

*Figure 2-3.* General Options Header format.

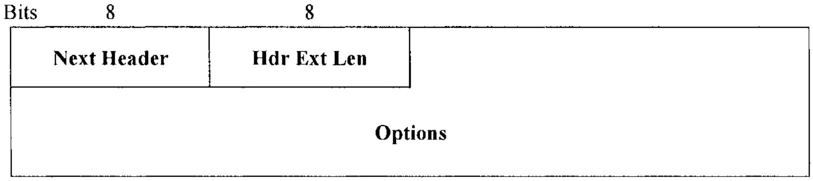


*Figure 2-4.* Hop-by-Hop Options Header format.

Seven Extension Headers are currently defined as follows: Hop-by-Hop Options, Routing, Fragment, Destination Options, Authentication, and Encapsulating Security Payload (ESP) Extension Headers. When more than one Extension Header is used in the same packet, Extension Headers are recommended to be appeared by the following order.

1. IPv6 Header.
2. Hop-by-Hop Options Header.
3. Destination Options Header.[5]
4. Routing Header.
5. Fragment Header.
6. Authentication Header.
7. Encapsulating Security Payload Header.
8. Destination Options Header[6] (for the upper layer).

Each Extension Header should appear only once in the same packet except Destination Options Header which may appear twice. Even if new Extension Headers are newly defined, the order of appearance should be regulated by the rule stated above. For example, Type 2 Routing Header which is especially defined to route packets between correspondent nodes and a mobile node should also follow the order for Routing Header. It is explained in detail in Chapter 10.

[5] Options to be processed by the destination specified in the IPv6 Destination Address field. If a Routing Header is included in a packet, the Destination Address field is exchanged to one of the listed address in the Routing Header in order. This option will be processed by each intermediate destination until the packet reaches the final destination. This address exchange on Destination Address field is specified in Fig. 2-7.

[6] Options to be processed by the final destination of a packet

## 2.3.1 Options Headers

Two Options Headers are defined in IPv6: Hop-by-Hop Options and Destination Options Headers. Both Options Headers carry a variable number of Type-Length-Value (TLV) fields. The header format is shown in Fig. 2-3.

Both the length of Option Type field and the length of Option Data Length field are 8-bit while the length of Option Data field is variable. The Option Data Length field contains the length of Option Data field in bytes.

### 2.3.1.1 Hop-by-Hop Options Header

The Hop-by-Hop Options Header carries optional information that should be processed by every node along the delivery path. Its format is shown in Fig. 2-4. The Next Header value of the preceding header should be set to 0 to identify that Hop-by-Hop Options Header.

- The Next Header field notifies the next header type following Hop-by-Hop Options Header.
- The Header Extension Length field specifies the length of Hop-by-Hop Options Header in 8-byte units except the Next Header field.

### 2.3.1.2 Destination Options Header

The Destination Options Header is used to carry optional information which should be examined or processed only by a destination node identified in the Destination Address field of IPv6 Header. When the Next Header value of the preceding header is 60, it indicates that Destination Options Header is following. Header format and the function of each field are identical to the Hop-by-Hop Options Header as shown in Fig. 2-4.

The Destination Options Header may appear twice if Routing Header is present in the packet. Once a packet arrives at the node specified in the Destination Address field, the address in the Destination Address field will be changed to another one selected from the address vector in the Routing Header. Address swapping is explained in Section 3.2.

## 2.3.2 Routing Header

The Routing Header is used to specify intermediate routers which must be gone through until the packet reaches the final destination. The function of Routing Header is very similar to the loose source routing of IPv4. The Next Header value of preceding header is 43. The Routing Header is shown in Fig. 2-5. The Routing Header should not be processed until the packet reaches the destination node identified by the Destination Address field of IPv6 Header.

- The Next Header field notifies the following header format by the Routing Header.
- The Header Extension Length field specifies the length of Routing Header in 8-byte unit except the Next Header field.
- The Routing Type is the identifier among variable Routing Header types.
- The Segments Left field specifies remaining nodes to be visited, and this number is equal to the number of address listed in the address vector of Routing Header.
- The variable-length Type Specific Data field is different according to the Routing Type value. When Routing Type value is 0, the Type Specific Data field is composed of reserved field and address lists as shown in Fig. 2-6.

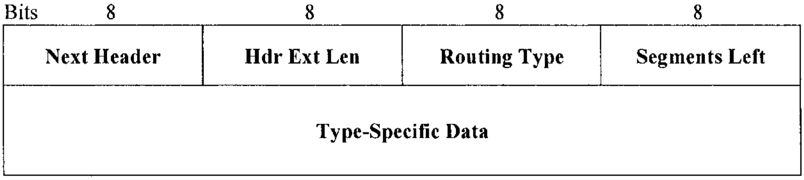
When Routing Type value is set to 0, the Routing Header format is as same as Fig. 2-6. The address vector in the Routing Header lists intermediate nodes to be visited along to the destination, and each address in vector is written as Address[1..*n*], as shown in Fig. 2-7. Remaining fields are identical to those of Fig. 2-5. If Routing Type value is 0, then any multicast address should not be contained in the address list of Routing Header or the Destination Address field of IPv6 Header.

For instance, address swapping between the Destination Address field of IPv6 Header and address selected among the address vector of Routing Header is as follows:

1. Source node *S* builds a packet $p_1$ and transmits it to destination *D* along the packet's delivery path $I_1$, $I_2$ and $I_3$ as shown in Fig. 2-7. The intermediate nodes to be visited ($I_1$, $I_2$ and $I_3$) are specified in the leftmost Routing Header.
2. When the packet $p_1$ arrives at the node $I_1$, this node will find out that it is not the final receiver of $p_1$ from Destination Options Header and Routing Header. At first, this node selects $i^{th}$ address from address vector as the next intermediate destination, where *i* is calculated from Eq. (1). Then, $I_1$ swaps the address in the Destination Address field of IPv6 Header with the selected address from the address vector of Routing Header. In this example, the selected address becomes $I_2$. $I_1$ decreases the value of Segment Left field by 1.
3. Now, the modified packet ($p_2$) by $I_1$ is transmitted to $I_2$.
4. The next intermediate destination node, $I_2$, handles the packet $p_2$ as the same way in the previous node.
5. When the packet reaches the final destination *D*, the Segment Left field will be changed to 0.

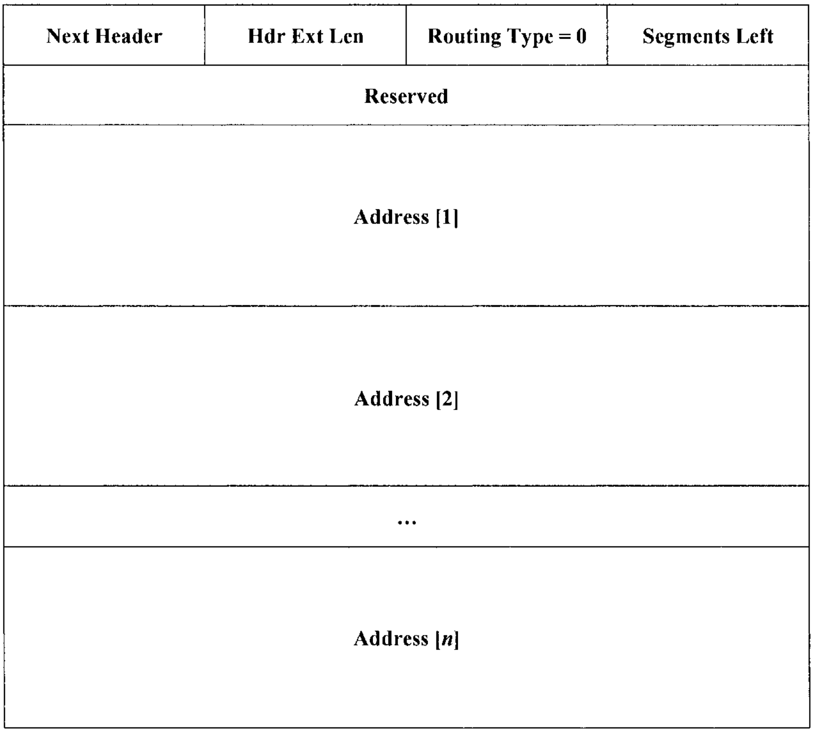
$$i = n - Segment\ Left + 1 \tag{1}$$

where $0 < i \leq n$ , and $n$ is the number of address listed in address vector.



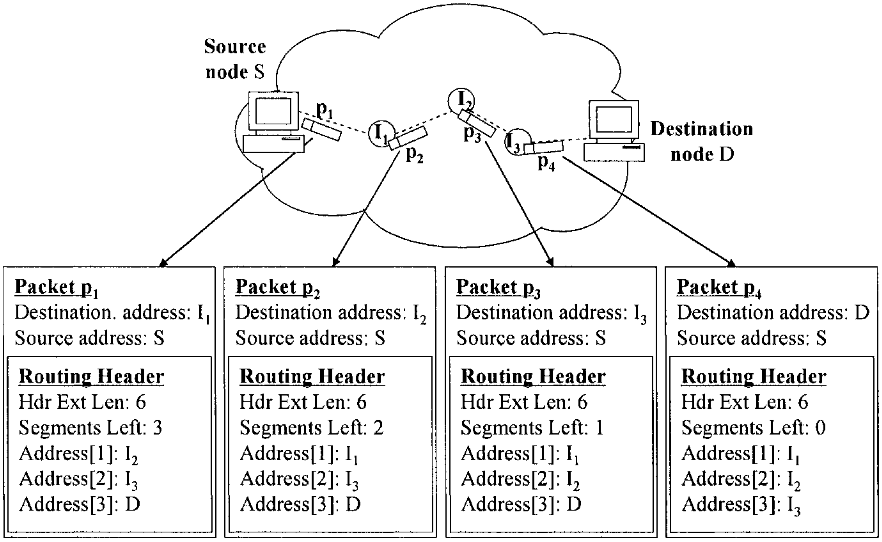
| Bits 8 | 8 | 8 | 8 |
|---|---|---|---|
| **Next Header** | **Hdr Ext Len** | **Routing Type** | **Segments Left** |
| **Type-Specific Data** | | | |

*Figure 2-5.* Routing Header format.



| **Next Header** | **Hdr Ext Len** | **Routing Type = 0** | **Segments Left** |
|---|---|---|---|
| **Reserved** | | | |
| **Address [1]** | | | |
| **Address [2]** | | | |
| ... | | | |
| **Address [*n*]** | | | |

*Figure 2-6.* Routing Header format when Routing Type field is set to 0.



**Packet $p_1$**
Destination. address: $I_1$
Source address: S

**Routing Header**
Hdr Ext Len: 6
Segments Left: 3
Address[1]: $I_2$
Address[2]: $I_3$
Address[3]: D

**Packet $p_2$**
Destination address: $I_2$
Source address: S

**Routing Header**
Hdr Ext Len: 6
Segments Left: 2
Address[1]: $I_1$
Address[2]: $I_3$
Address[3]: D

**Packet $p_3$**
Destination address: $I_3$
Source address: S

**Routing Header**
Hdr Ext Len: 6
Segments Left: 1
Address[1]: $I_1$
Address[2]: $I_2$
Address[3]: D

**Packet $p_4$**
Destination address: D
Source address: S

**Routing Header**
Hdr Ext Len: 6
Segments Left: 0
Address[1]: $I_1$
Address[2]: $I_2$
Address[3]: $I_3$

*Figure 2-7.* Example of packet delivery with Routing Header.

## 2.3.3 Fragment Header

In IPv4, when a packet size is too big to transmit by any node on the path to the next node, the node will fragment that packet before forwarding to the next node. The fragmentation in IPv6 protocol should be performed only by a source node. The Fragment Header is used to send a packet whose size is larger than the path MTU to a destination.

When a packet is too big to fit in the path MTU to a destination, a source node will divide the original packet into several fragments and send each fragment as a separate packet. If an intermediate router on the path to the destination finds a packet is too large to process, this router builds and returns ICMP error message to the source node to inform that the packet is too big to handle.

As the destination node receives a packet with Fragmentation Header, it waits until all fragmented packets are received. Once all the packets are received, the node will start reassembly to build the complete original packet. The end of fragmentation is recognized by the *M* flag in the Fragmentation Header. The Next Header value of the preceding header is 44. The Fragment Header is shown in Fig. 2-8.

- The Next Header field notifies the following header format by the Fragment Header, and this field uses the same value as the value defined in RFC 790.

- The Fragment Offset field is filled with 13-bit unsigned integer. The Offset is calculated as the relative distance measured in 8-byte unit from the beginning of the original packet.
- The RES field is 2-bit reserved field and initialized to 0 when a packet is transmitted.
- The *M* flag specifies whether there are more fragments, or it is the last one. If a packet with *M* flag set is received, it indicates the last fragment of one original packet. Once the receiver gets all fragmented packets including a fragment with *M* flag set, it starts reassembling them to build the original packet. Otherwise, the receiver should wait until it receives all fragmented packets.
- 32-bit length Identification field contains identification number generated by a source node. The identification number should be different from that of any other fragmented packet sent recently with the same source and destination address.

Original unfragmented packet may be classified into two parts: unfragmentable and fragmentable parts, as shown in Fig. 2-9 (a). IPv6 Header and Extension Headers which should be processed by each intermediate node to a destination belong to the unfragmentable part, and remaining Extension Headers and data portion belong to the fragmentable part. If the unfragmentable part is considered logically as a header, the fragmentable part can be treated as a data, and the data is divided into several fragments to fit into the path MTU, as shown in Fig. 2-9 (b). Except the last fragmented packet, the size of all fragmented packets is the multiple of 8 bytes.

Each fragmented packet consists of unfragmentable part, Fragment Header and one portion of fragmentable part as shown in Fig. 2-9 (c).

- Unfragmentable part is identically copied from that of the original packet except following changes:

  Payload length changes to the length of this fragmented packet only.

  The Next Header field of the last Extension Headers of unfragmentable part changes to 44.

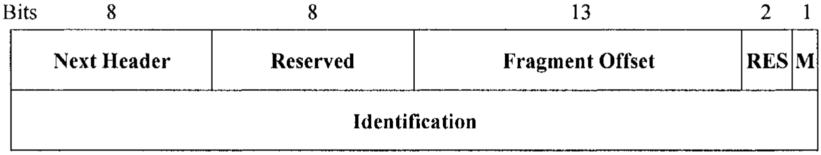
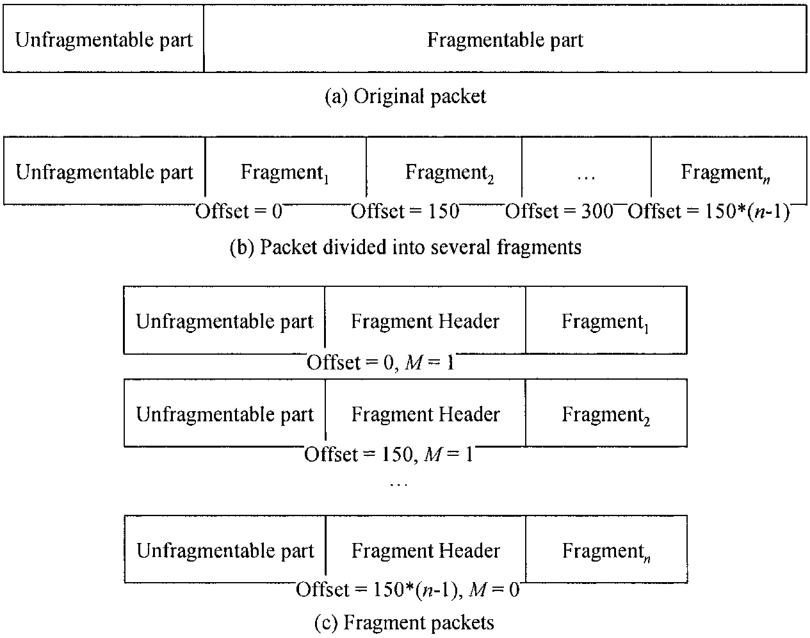


*Figure 2-8.* Fragment Header format.



(a) Original packet

(b) Packet divided into several fragments

(c) Fragment packets

*Figure 2-9.* Fragmentations.

- The Fragment Header shown in Fig. 2-8 contains information as follows:
  Next Header value.
  Fragment Offset calculated relatively from the start of the fragmentable part in 8-byte unit.
  *M* flag is set to 1 except the last fragmentation. The *M* flag of last fragmentation is set to 0 to indicate that there is no more fragmentation.
  Identification generated by a source node.
- On portion of fragmentable part
  Fragmentations except the last one are multiple of 8-bytes long.

At the receiving node, the reassembly is required before passing packets to the upper layer. Fragmented packets with the same source address and destination address and identification will be reassembled. If any of fragmented packets is missing, the reassembly may not be properly performed.

In the reassembly process, Fragment Header is removed from the fragmented packet. Another change should occur in the Next Header field of the last Extension Header of unfragmentable part, and its value is obtained from the Next Header field of Fragment Header in the first fragment. Besides, the Payload Length of the reassembled packet should be recomputed using Eq. (2).

$$\begin{aligned} pl_{orig} &= L_{unfragmentable\ part} + \sum L_{divided\ fragmentable\ part} \\ &= L_{IPv6\ Header} + \sum_{except\ Fragment\ Header} L_{Extension\ Header} + \sum L_{data\ payload\ portion} \\ &= (pl_{first} - fl_{first} - 8) + (8 \times fo_{last} + fl_{last}) \end{aligned} \quad (2)$$

where $pl_{orig}$ is the payload length value of the reassembled packet, $pl_{first}$ is the payload length value of the first fragmented packet, $fl_{first}$ is the length of the fragment following Fragment Header of the first fragmented packet, $fo_{last}$ is the Fragment Offset value of Fragment Header of the last fragmented packet, $fl_{last}$ is the length of the fragment following Fragment Header of the last fragment packet, and $L_{unfragmentable\ part}$ is the length of unfragmentable part.

The format of the reassembled packet from fragmented packets should be identical to the original packet before fragmentation in Fig. 2-9 (a).

### 2.3.4 No Next Header

If the Next Header field value of an IPv6 Header or Extension Header is set to 59, it indicates that there is no following header.

## 2.4 PACKET SIZE AND PATH MTU

Each link connected to IPv6 is required to support at least 1280-byte MTU. If there is any link which may not support this MTU requirement, link-specific fragmentation must be provided by a layer below IPv6 protocol stack. IPv6 enabled node should handle packets whose size is as big as link's MTU. If a packet is too large to send, an intermediate node will build and transmit ICMP error message back to the source. If the size of the packet is smaller than link MTU, the packet will be delivered to the destination.

The maximum packet size to be transmitted across the delivery path without any fragmentation is called path MTU (PMTU), and it is equal to the minimum value of link MTUs of all the links in the path. The Path MTU Discovery protocol is defined in RFC 1981,[5] which is a standard protocol for a node to find the PMTU of an arbitrary path to a destination.

Discovering a PMTU greater than 1280 bytes is strongly recommended, but it is possible to simply restrict MTU of no larger than 1280-byte or fix PMTU as 1280-byte as a default and not to implement PMTU discovery protocol.

When a node builds a packet which is larger than PMTU, it is required to fragment before starting transmission. The Fragment Header explained above is inserted into each fragmented packet.

When an IPv6 host builds a packet which is eventually destined to IPv4 node,[7] the sending node may receive ICMP error message, such as 'too big message' notifying that MTU is less than 1280. In the case, the source node does not have to drop the size of the following packets to smaller than 1280-byte. The IPv6 node rather puts Fragment Header in the following packets, which allows intermediate IPv6-to-IPv4 translating router to fragment packets with suitable identification value as well as to translate IPv6 packets into IPv4 packets. The reason why IPv6 solves the problem in this way is very simple; fragmentation is only allowed at source nodes in IPv6 while intermediate nodes can perform fragmentation only in IPv4. The payload of IPv4 fragment packet would be reduced to 1232 bytes.[8]

## 2.5 FLOW LABEL

To handle various qualities of service requests, such as the best effort service or real-time service, IPv6 defines 20-bit Flow Label field in the IPv6 Header. This field is still under standardization. Hosts or routers that do not understand Flow Label field should set this field to 0 when they build packets. Further, any node which does not understand Flow Label field will just ignore or pass the packet unchanged when a received packet has a certain flow level value.[3]

A flow is the sequence of packets between a specific source and a destination node with some special requirements on handling packets regardless whether they are destined to single point or multi-points. The intermediate routers will handle packets according to specified flow labels. Resource Reservation Protocol (RSVP) and Diff-Serv are considered as candidate methods for the flow label.

[7] For communications between IPv6-only and IPv4-only node, translation mechanism should be operated between the different IP-based nodes. Normally, an ingress or egress router runs it in favor of its site or domain.

[8] 1232 bytes are calculated from 1280−(40+8), where 40 is the length of the IPv6 Header, and 8 is the length of the Fragment Header. The payload would be even smaller if there is additional Extension Header.

A flow is identified by the association with source address and non-zero flow label value, which is chosen randomly between 1 to FFFFF in hexadecimal by the source node. There may be multiple active flows between the same communication parties. Besides, a packet whose flow label is set to zero does not belong to any flow.

Traffics belonged to the same flow must be generated from the same source address with the same flow label, and they should be delivered to the same destination. If any packet whose flow has some value except zero includes Hop-by-Hop Options Header or Routing Header, all subsequent packets with the same flow label should be generated with the same Hop-by-Hop Options Header or Routing Header, respectively.

The lifetime of a flow label should be predefined along a flow's delivery path. Before the maximum lifetime of any flow label is expired, the flow label should not be reused.

Nodes may reboot due to various reasons. In that case, the node should be very careful not to assign the flow label which has been already assigned to the other flow and still has valid lifetime. To record flow labels on a stable storage device may prevent some inevitable accidents.

## 2.6 TRAFFIC CLASS

The Traffic Class field substitutes for ToS field of IPv4, and field length is 8-bit as same as ToS field. The Traffic Class is set by a sending node to apply different priorities or classes to traffics. Intermediate routers use the Traffic Class to identify and distinguish between different priorities or classes of traffics. This Traffic Class field is still under development.

## REFERENCES

1. S. Deering and R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460 (December 1998).
2. Microsoft Corporation, IPv6 Deployment Strategies (December 2002).
3. J. Rajahalme, A. Conta, B. Carpenter, and S. Deering, IPv6 Flow Label Specification, RFC 3697 (March 2004).
4. J. Postel, Assigned Numbers, RFC 790 (September 1981).
5. J. McCann, S. Deering, and J. Mogul, Path MTU Discovery for IP version 6, RFC 1981 (August 1996).

# APPENDIX A: ASSIGNED INTERNET PROTOCOL NUMBERS

Assigned numbers defined in RFC 790 are arranged in Table 2-2.[4] This number is used in Protocol field of IP version 4 or Next Header field of IP version 6 to identify the Next Header format or the next level protocol.

*Table 2-2.* Assigned numbers.

| Protocol number in decimal | Description | Protocol number in decimal | Description |
|---|---|---|---|
| 0 | Reserved | 18 | Multiplexing |
| 1 | ICMP | 19 | DCN |
| 2 | Unassigned | 20 | TAC Monitoring |
| 3 | Gateway-to-Gateway | 21 ~ 62 | Unassigned |
| 4 | CMCC Gateway Monitoring Message | 63 | any local nctwork |
| 5 | ST | 64 | SATNET and Backroom EXPAK |
| 6 | TCP | 65 | MIT Subnet Support |
| 7 | UCL | 66 ~ 68 | Unassigned |
| 8 | Unassigned | 69 | SATNET Monitoring |
| 9 | Secure | 70 | Unassigned |
| 10 | BBN RCC Monitoring | 71 | Internet Packet Core Utility |
| 11 | NVP | 72 ~ 75 | Unassigned |
| 12 | PUP | 76 | Backroom SATNET Monitoring |
| 13 | Pluribus | 77 | Unassigned |
| 14 | Telenet | 78 | WIDEBAND Monitoring |
| 15 | XNET | 79 | WIDEBAND EXPAK |
| 16 | Chaos | 80~254 | Unassigned |
| 17 | User Datagram | 255 | Reserved |

Chapter 3

# ADDRESS ARCHITECTURE

## 3.1 INTRODUCTION

IPv4 protocol uses 32-bit IP address. With 32-bit, we can make approximately four billions of numbers, but we can not fully utilize the 4 billion address space mainly because IP address space has been assigned by the class. Class is identified by the leftmost 3 bits of IPv4 address. Several novel mechanisms such as CIDR and Network Address Translator (NAT) are devised to better utilize IPv4 address space.[1] However, significant portions of assigned addresses are currently unused. NAT has shortcomings disrupting end-to-end communications.

There are also high demands for IP address assignment from many countries, companies and individuals. As the evolution of mobile communications, there is high chance that we need tremendous amount of IP address. For example, 4G mobile communications assume that all mobile terminals including current cellular terminals are IP enabled. Evolution of p-to-p communications also increases the demand for globally unique IP addresses. Thus, 32-bit address space may not enough to satisfy these demands.

IPv6 uses 128-bit length identifier to distinguish a host from others in Internet. Numerically, the length of IPv6 identifier is four times longer than the length of IPv4 identifier. Thus, there might be worries about the size of routing table. However, IPv6 address space is designed to be hierarchically managed.

In IPv4, there are three address types depending on the packet transmission scope or the number of receivers, such as unicast, multicast and broadcast addresses. Besides, special addresses, such as network address,[9] direct broadcast address, limited broadcast address, unspecified address, and loop-back address are defined.[1] Among these special addresses, unspecified address and loop-back address are still employed in IPv6.

IPv6 address can be classified into three classes depending on whether IPv6 address identifies only one interface, or whether packets are delivered to all group members; unicast address, anycast address, and multicast address.[2] At first, unicast address is an identifier for a single interface. The anycast address is an identifier for multiple interfaces, not limited to only one node. When a forwarded packet has an anycast address value in the destination address field, it is delivered to the nearest node among group members identified by that address. The metric to measure 'nearness' depends on the routing protocol, usually distance value. Same as anycast address, the multicast address is an identifier for multiple interfaces belonging to different nodes. However, packets destined for multicast address are delivered to every single member in a group. Anycast address is taken from the unicast address space, thus, has the similar format with unicast address. Multicast address type in IPv6 is more practically and efficiently used than that of IPv4. The broadcast address is obsolete in IPv6, and multicast address takes the equivalent role.

IPv6 address space is also divided into several types depending on its usage scope, such as link-local address, site-local address and global-unique address.[3] The first type, link-local address is a unicast address used to reach neighbors in a link scope. Every interface must be assigned with at least one link-local address. Inside the link-local scope, link-local address is used to identify each other. Site-local address is a unicast address used to reach neighbors in a site scope, but it is now deprecated and unused even in experimental sites. Global-unique address is globally unique to distinguish one host from others in the network. Multiple global-unique addresses may be assigned to one interface. Each IPv6-enabled interface is assigned both link-local and global-unique addresses.

The IPv6 address architecture seems to be more complex than that of IPv4, but it can be elaborately and efficiently managed.

[9] IPv4 address is composed of network ID and host ID. Depending on the leftmost 3 bits or address class, the length of network ID is determined.
Network address: specific network ID and all 0s in host ID
Direct broadcast address: specific network ID and all 1s in host ID
Limited broadcast address: all 1s in network ID and all 1s in host ID
Unspecified address : all 0s
Loopback address: 127 in network ID and any number in host ID

## 3.2 EXPRESSION OF IPv6 ADDRESS

To express 128-bit length IPv6 address in text strings, there are three formats; usual format, abbreviated format, and mixed format. Like ‘address/prefix’ notation in IPv4, these three formats can be written with prefix notation.

Normally, IPv6 address is represented using hexadecimal numbers. Thus, there exist 8 fields for the usual address format as follows; *x:x:x:x:x:x:x:x*, where *x* denotes four hexadecimal digits. For example, 111:2222:3333:0:0:0: ABCD:5678 is possible. It is not necessary to write leading 0 or leading successive 0s at each field of addresses.

*Table 3-1.* Example of compressed address format.

| Naïve format | | Compressed format |
|---|---|---|
| 1111:2222:3333: :0:0:0:1234:5678 | | 1111:2222:3333::1234:5678 |
| FF01:0:0:0:0:0:0:1 | May be rewritten as: | FF01::1 |
| 0:0:0:0:0:0:0:1 | | ::1 |
| 0:0:0:0:0:0:0:0 | | :: |

*Table 3-2.* Example of mixed address format.

| Naïve format | | Mixed format |
|---|---|---|
| 1111:2222:3333: :0:0:0:1234:5678 | | 1111:2222:3333::1234:5678 |
| FE80:0:0:0:0:0:0:129.141.52.38 | may be rewritten as: | FE80::129.141.52.38 |
| 0:0:0:0:0:0:203.253.21.3 | | ::203.253.21.3 |
| 0:0:0:0:0:FFFF:203.253.21.3 | | ::FFFF:203.253.21.3 |

*Table 3-3.* Example of wrongly compressed address.

| Legal representation | Illegal (wrong) representation |
|---|---|
| 1234:0000:0000:ABC0:0000:0000:0000:0000/64 | 1234::ABC0/64 |
| 1234::ABC0:0:0:0:0/64 | 1234:0:0:ABC/64 |
| 1234:0:0:ABC0::/64 | 1234::ABC/64 |
| 1234::ABC0/64 is equal to 1234:0000:0000:0000:0000:0000:0000:ABC0 because we can not drop trailing zeros. | |

*Table 3-4.* Reserved prefix according to address scope or specific usage.

| Address type | Prefix in Binary | Prefix in Hexadecimal |
|---|---|---|
| Unspecified | 00…0 (128bits) | :: |
| Loop-back | 00…1 (128bits) | :: 1/128 |
| Multicast | 11111111 | FF00::/8 |
| Link-local unicast | 1111111010 | FE80::/10 |
| Site-local unicast | 1111111011 | FEC0::/10 |
| Global unicast | Everything else | |

Sometimes, an address may have all 0s in any *x* in the usual address format, *x*:*x*:*x*:*x*:*x*:*x*:*x*:*x*, and it should be a tedious job to fill them up. IPv6 supports compressed address format. Successive 0s are abridged with symbol '::' which indicates that one or more 0s are hidden. The double colon should be used only once in an address. Some examples are specified in Table 3-1.

When we think of mixed networks with IPv6 and IPv4 or interconnection between these two different internet protocols, it should be very convenient to mingle these two address notations. An IPv4 node easily gets IPv6 connectivity using its own IPv4 address without any further upgrade or address assignment from IPv6 domain.

The mixed IPv6 address is built as follows: *x*:*x*:*x*:*x*:*x*:*x*:*a.b.c.d*, where *x* denotes four hexadecimal digits and *a.b.c.d* is IPv4 address. The mixed addresses are exampled in Table 3-2. As mentioned above, this address type enables a node in a public IPv4 network to get IPv6 connectivity without any support from a router or no change of network topology it belongs to. The compressed address format may also be applied for the mixed address format.

Besides, IPv6 address can be expressed with a prefix, similar way of IPv4 address prefix. The notation for an address is ipv6-address/prefix-length, where ipv6-address is any address type defined above, and prefix-length is a decimal value representing how many leftmost bits of IPv6 address will become the prefix. For instance, an IPv6 address, 1234:0000:0000:ABC0: 1234:5678: 9ABC:1234/64 has 64-bit length prefix, 1234:0000:0000:ABC0. Table 3-3 shows legal and illegal examples of address/prefix notation.

There are reserved prefixes depending on the scope or specific usage; multicast, link-local, and global unicast. Unspecified and loop-back addresses are also defined in IPv6. These various address formats are shown in Table 3-4.

## 3.3 UNICAST ADDRESS

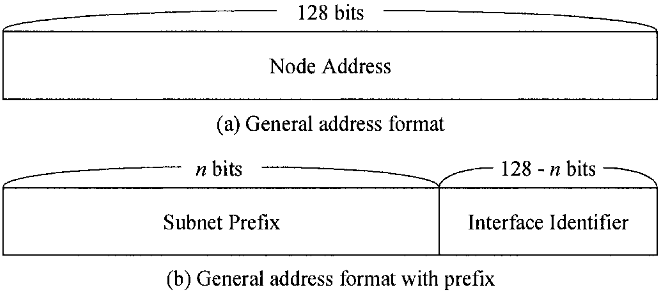
Unicast address identifies only one node in networks. Communication based on unicast addresses is one to one communication. When a node communicates with a single corresponding party, unicast address is used as an identifier of each party in communication.

Fig.3-1 shows unicast address format. IPv6 address is 128-bit long, and it has internal structure, which enables hierarchical address management and delegation. When IP address does not have internal structure, it is formatted as Fig. 3-1(a), and when subnet prefix on link exists, address format changes, as in Fig. 3-1(b). It is quite simple, but *n*-bit length subnet prefix should be pre-defined and managed within the link. More sophisticated address formats are described in the following section.

IPv6 has several unicast address types. They may be grouped into two types; address with specific scope and address with specific purpose. At first, depending on the specific scope, addresses are categorized into three types again: global unique address, link-local address and site-local address. Each address type is explained in the following subsections. Depending on the specific purpose, IPv4-compatible IPv6 address and IPv4-mapped IPv6 address are defined in IPv6. These two IPv4-embedded IPv6 addresses are valuably used for the initial stage of IPv4/IPv6 transition.

## 3.3.1 Unspecified address

Reserved address for unknown host is called unspecified address and its format is 0:0:0:0:0:0:0:0 or ::. This address can not be assigned to any physical interface. When a node plugs in and starts packet exchange with neighboring nodes before any address is assigned, the unspecified address is used as a source address which implies that source node's address is unknown.



(a) General address format

(b) General address format with prefix

*Figure 3-1.* General unicast address format.

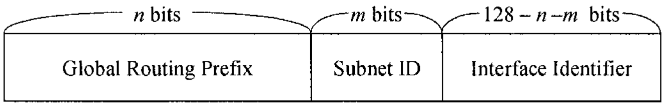


*Figure 3-2.* Global unicast address format with subnet ID.

This address indicates the absence of an address. This address should not be used in the Destination Address field of a packet or IPv6 Routing Headers. Routers should not forward any IPv6 packet having unspecified address as a source address to other links.

### 3.3.2 Loop-back address

The unicast address, 0:0:0:0:0:0:0:1 or ::1 is called loop-back address. A node uses this address type to send packets to its own interface. This address can not be assigned to any physical interface and can not be used in Source Address field of a packet. Packets destined to the loop-back address should not be forwarded to an outside node, and packets from other nodes with loop-back address should be silently discarded.

### 3.3.3 Global unicast address

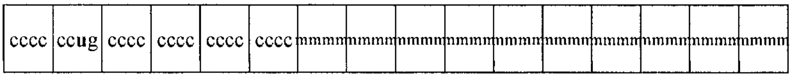
When a unicast address identifies a node uniquely in the entire network, the unicast address is called global unicast address. The global unicast address is divided into several fields: $n$-bit global routing prefix, $m$-bit subnet ID and interface identifier, as shown in Fig. 3-2.[4-7]

- Global routing prefix is assigned to a site, and commonly this value is hierarchically structured. The prefix is hierarchically managed and assigned by RIR, LIR and ISPs. When the leftmost 3-bit is set to non-zero, the length of interface identifier is fixed as 64 bits.
- Subnet ID is a link identifier inside a site. It is hierarchically handled by site administrators.
- Interface identifier is used to identify an interface attached to a node.

If the leftmost three bits of global routing prefix value is not 000, in binary notation, the length of interface identifier is constrained to 64 bits. The modified EUI-64 format which is specified in the following section is used for interface identifiers. In the general case, [10] IESG and IAB recommend to use /35 or /48 as global routing prefix.

If the leftmost three bits of global routing prefix value is 000, there is no restriction on the size of an interface identifier. Well-known example for this case is IPv4-embedded IPv6 addresses, where 80 bits of prefix are filled with successive 0s.

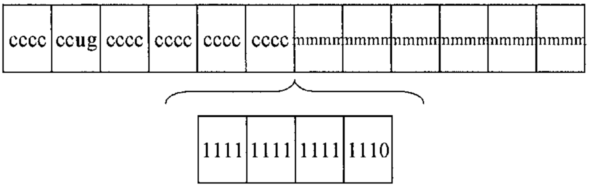
[10] In allocating address, IESG and IAB recommend to build the boundary between the public and the private topologies.



| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cccc | ccug | cccc | cccc | cccc | cccc | mmmm | mmmm | mmmm | mmmm | mmmm | mmmm | mmmm | mmmm | mmmm | mmmm |

u: universal/local bit g: individual/group bit
c: bits of the company_id m: bits of manufacturer-selected extension id

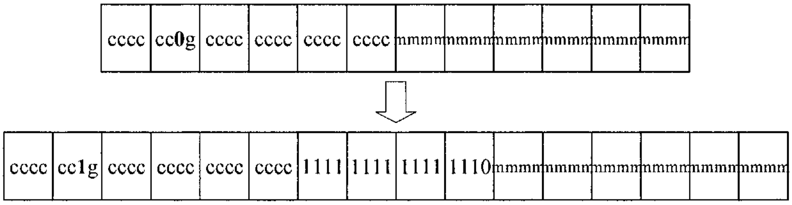
*Figure 3-3.* EUI-64 format.



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| cccc | ccug | cccc | cccc | cccc | cccc | mmmm | mmmm | mmmm | mmmm | mmmm | mmmm |

| | | | |
|---|---|---|---|
| 1111 | 1111 | 1111 | 1110 |

*Figure 3-4.* Example: inserting FFFE into 48-bit MAC address.



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| cccc | cc0g | cccc | cccc | cccc | cccc | mmmm | mmmm | mmmm | mmmm | mmmm | mmmm |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cccc | cc1g | cccc | cccc | cccc | cccc | 1111 | 1111 | 1111 | 1110 | mmmm | mmmm | mmmm | mmmm | mmmm | mmmm |

*Figure 3-5.* Modified EUI-64 format.

### 3.3.3.1 Interface identifier

Interface identifier is the identity of interface on a link and usually called link-layer identifier. It should be unique inside a subnet, and it is not recommended to assign the same interface identifier to physically different nodes on a link. For all global unicast addresses except those starting 000 in the binary notation, interface identifier should be 64-bits long and further built with the modified EUI-64 identifier.

The IEEE EUI-64 has 64-bits length, and the format is shown in Fig. 3-3.

- *c* bit is used to specify the company identifier.
- *u* bit is used to indicate universal or local.
- *g* bit is used to indicate individual or group.
- *m* bit is used to specify manufacturer-selected extension identifier.

Globally unique EUI-64 identifier should have 0 in the *u* bit. The only change to build interface identifier from EUI-64 identifier is inverting *u* bit from 0 to 1.

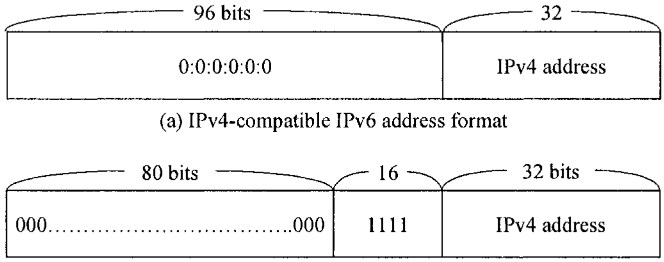
MAC address registered at each network interface card has 48-bit length. With the 48-bit MAC address, we can yield modified-EUI-64 identifier, as follows:

1. Insert two octets in the middle of 48-bit MAC address. Inserted value is FFFE in hexadecimal notation as shown in Fig. 3-4. Symbols in this figure have same meaning as shown in Fig. 3-3.
2. Change *u* bit value into 1, as shown in Fig. 3-5. Initially, the *u* bit is set to 0.

## 3.3.4 IPv6 address with embedded IPv4 address

When address format is ::a.b.c.d or ::F:a.b.c.d, where F is hexadecimal number and a.b.c.d is IPv4 address, it specifies a specific IPv6 address with embedded IPv4 address. The former type is called IPv4-compatible IPv6 address, and its format is shown in Fig. 3-6, where the first 96 bits are filled up with 0, and the following 32 bits are composed of IPv4 address. The latter type is called IPv4-mapped IPv6 address.

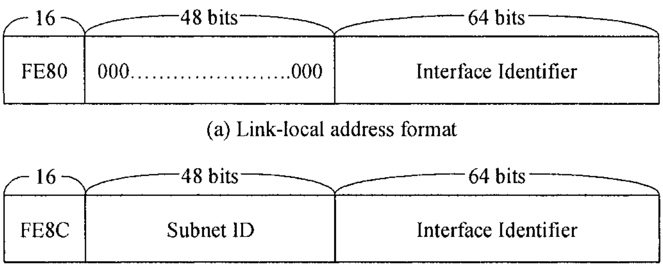
In IPv6 transition mechanisms, IPv4-compatible IPv6 address is used to support a dynamic tunnel between IPv6-enabled hosts over IPv4 network. The node assigned with IPv4-compatible IPv6 address becomes tunnel end points. It is the easiest way to provide IPv6 connectivity to isolated IPv6 hosts over IPv4 network without any change or support from the infrastructure.



(a) IPv4-compatible IPv6 address format

(b) IPv4-mapped IPv6 address format

*Figure 3-6.* IPv6 address embedded IPv4 address.



(a) Link-local address format

(b) Site-local address format

*Figure 3-7.* Local scoped address.

### 3.3.5 Local-scope unicast address

For the local-scope unicast address, link-local address and site-local address are defined in IPv6. The link-local address is only valid within a single link while the site-local address is only valid within single site. Each address format is depicted in Fig. 3-7.

The link-local address is designed for addressing a node on a single link, especially in address autoconfiguration and neighbor discovery process. Site-local address is originally designed for addressing a node on a single site. Currently, this address format is abandoned and unused.[8]

## 3.4 ANYCAST ADDRESS

Anycast address is used in one to many communications. When anycast address is used in the Destination Address field in a packet, then the packet will be forwarded and routed to the nearest node among anycast group members identified by that address. The final receiving node of the packet is determined by nearness, which is learned using ordinary routing protocols.

Anycast address is taken from the unicast address space, which confuses us in distinguishing anycast address from unicast address. When a unicast address, which was initially assigned to one interface, is assigned to multiple interfaces, the address type should be changed from unicast into anycast, and this status change should be announced. Similarly, when an address type changed from anycast to unicast, nodes which are assigned the address should be configured to know the change of the address type.

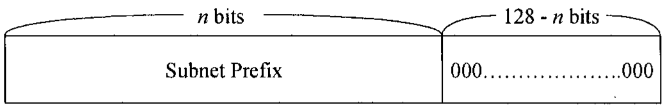


*Figure 3-8.* Anycast address format.

Anycast address is composed of subnet prefix and successive 0s, as shown in Fig. 3-8. The *n*-bit subnet prefix identifies a topological area where all group members belong to the same anycast address.

For any assigned anycast address, there is the longest subnet prefix, *P*. Inside of area identified by *P*, anycast address should be kept up as a separate route entry in a router, while only one aggregated route entry for anycast addresses with prefix *P* will be notified to the outside.[9]

Most common use of anycast address is to identify a set of routers providing internet service. For example, anycast address may be contained in the Routing Header of a packet explicitly. Then, one of anycast members on the way to the packet's destination will get the packet and forward it to next destination. If the next destination address is also anycast address, the nearest router among the second anycast members will continue packet delivery. Anycast address which is assigned to the set of routers in IPv6 domain effectively helps to utilize packet routing. Besides, when IPv4 domain leaks routing information into IPv6 domain, a router leaking route information may be configured with an anycast address which is assigned to the set of routers in IPv6 domain.

At present, anycast address is not widespread due to complications and hazards. Any packet whose source address field is anycast address can not be originated or forwarded. Anycast address should not be assigned to IPv6 hosts, but be assigned to IPv6 routers.

## 3.5 MULTICAST ADDRESS

While unicast address used in one to one communication, multicast address is used in one to many communication. When a packet is generated with multicast address as a destination address, then the packet will be delivered to all of multicast group members.

Both anycast address and multicast address are for multiple interfaces belonging to different nodes; however, packets destined to a multicast address are delivered to every member in a multicast group. Multicast address serves the purpose for broadcast address. In Fig. 3-9, we can see multicast address format.

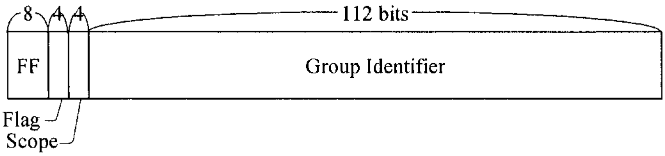
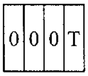


*Figure 3-9.* Multicast address format.



T=0, permanently-assigned (well-known multicast address)
T=1, non-permanently-assigned
(transient, or experimental multicast address)

*Figure 3-10.* Flag field of multicast address.

In IPv4, class D address space, where addresses start with 111 at the leftmost 3 bits, is reserved for multicast, and the reserved multicast addresses are used as identifiers for only pre-arranged or pre-agreed multicast groups. Multicast is more effectively utilized in IPv6. There are many reserved multicast addresses, and they operate as the substructure of IPv6. In Table 3-7, several reserved multicast address are shown. More multicast address will be reserved for the general use in the near future.

There are some rules to use multicast address as follows:

- Multicast address must not be used as the source address in IPv6 packet or as the intermediate destination address in the Routing Header.
- Routers must not forward any multicast packets beyond the bounds of the scope field in the multicast address of the packet.

## 3.5.1 Multicast address format

When the leftmost 8 bit is set to FF in hexadecimal, it indicates IPv6 multicast address. The multicast address is composed of 8 bits set to FF, flag, scope fields, and group identifier, as shown in Fig. 3-9.

- Flag field: it is 4-bit length field, where the first 3 bits are reserved, and must be initialized to 0. The last bit is called *T* bit. When the *T* bit set to 0, the multicast address is well-known address and assigned by IANA. When the *T* bit is set to 1, the multicast address is temporary. The flag field in multicast address is specified in Fig. 3-10.

- Scope field: 4-bit length field is used to limit the scope of the multicast group and specified in Table 3-5.
  When the Scope field is set to 0001 in binary, multicast address covers only an interface-local scope, and it is only used to check loop-back.
  When the Scope field is set to 0010 in binary, multicast address covers link scope.
  When the Scope field is set to 0101 in binary, multicast address covers site scope.
  When the Scope filed is set to 0100 in binary, multicast address covers the smallest region which needs the administrative configuration.
  When the Scope filed is set to 1000 in binary, multicast address covers the scope of multiple sites under a single organization.
  Unassigned scope is available for administrators to define additional multicast scope or is reserved for the future use.
- Group identifier: 112-bit identifier classifies multicast groups, whether it is permanent or temporary, within the given scope.
- All of group ID is defined explicitly with scope value. It is not allowed to use any group ID defined in Table 3-6 with other scope value or *T* flag set to 0. Further, reserved multicast addresses should not be assigned to any multicast group.

*Table 3-5.* Scope value of multicast address.

| Scope | Description | Scope | Description |
|---|---|---|---|
| 0000 | Reserved | 1000 | Organization-local scope |
| 0001 | Interface-local scope | 1001 | Unassigned |
| 0010 | Link-local scope | 1010 | ″ |
| 0011 | Reserved | 1011 | ″ |
| 0100 | Admin-local scope | 1100 | ″ |
| 0101 | Site-local scope | 1101 | ″ |
| 0110 | Unassigned | 1110 | Global scope |
| 0111 | ″ | 1111 | Reserved |

*Table 3-6.* Reserved multicast address.

| Multicast address type | Description |
|---|---|
| Reserved multicast address | FF00~FF0F:0:0:0:0:0:0:0 |
| All-nodes multicast address | FF01:0:0:0:0:0:0:1<br>FF01:0:0:0:0:0:0:2 |
| All-routers multicast address | FF01:0:0:0:0:0:0:2<br>FF02:0:0:0:0:0:0:2<br>FF05:0:0:0:0:0:0:2 |
| Solicited-node multicast address | FF02:0:0:0:0:1:FFXX:XXXX |

## 3.5.2 Reserved multicast address

There are several pre-defined well-known multicast addresses. In Table 3-7, several reserved multicast addresses are shown. More multicast addresses will be reserved for the general use in the near future.

- All-nodes multicast address: this address is a link scoped address to reach all nodes in the link. Any packet destined to the all-nodes multicast address should not be transferred across a link.
- All-routers multicast address: this address is a link scoped address to reach all routers in the link. Any packet destined to the all-routers multicast address should not be transferred across a link.
- Solicited-node multicast address: this address is a link scoped address to reach the target address. When an IPv6 node plugs in an IPv6 link, it builds link-local address and starts Duplication Address Detection (DAD) for address verification.[10, 11] DAD is explained in Chapter 6, in detail. Solicited-node multicast address is used in exchanged packets for DAD. Solicited-node multicast address is formed as FF02::1:255.x.y.z where x.y.z. is the rightmost 24 bits from 64-bit interface identifier. This address type ranges from FF02:0:0:0:0:1:FF00:0000 to FF02:0:0:0:0:1:FFFF:FFFF. Fig. 3-11 shows the way to build solicited-node multicast address. In addition to that, IPv4 multicast address for solicited-node multicast address is shown Fig. 3-12.

Both all-nodes multicast address and all-routers multicast address are used to identify hosts within interface-local scope (0001 in binary) or link-local scope (0010 in binary).

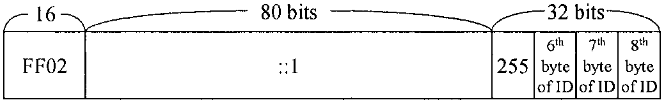


*Figure 3-11.* Solicited-node multicast address format.

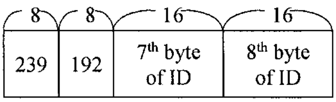


*Figure 3-12.* IPv4 multicast address format for solicited-node multicast address.

## 3.6 REQUIRED ADDRESSES

As explained in previous sections, all interfaces must hold at least one link-local address and may be assigned multiple IPv6 addresses whose type may be one of unicast, anycast, or multicast.

In detail, a host is required to be assigned and configured or to understand the following addresses:[4]

- Link-local address for each interface
- Unicast and anycast addresses that have been configured for the node's interfaces (manually or automatically)
- Loop-back address
- All-nodes multicast addresses
- Solicited-node multicast addresses corresponding to its unicast and anycast addresses
- Multicast addresses of all other groups to which the node belongs

A router is also required to be set up or recognize all addresses which are required for a host. Additionally, a router has to support the subnet-router anycast addresses for all interfaces where it is configured to behave as a router, all other anycast addresses and the all-routers multicast addresses.

## REFERENCES

1. B. Forouzan, TCP/IP Protocol Suite (McGrawHill, 2000).
2. S. Deering and R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460 (December 1998).
6. R. Hinden, M. O'Dell, and S. Deering, An IPv6 Aggregatable Global Unicast Address Format, RFC 2374 (July 1998).
5. R. Hinden, Proposed TLA and NLA Assignment Rules, RFC 2450 (December 1998).
7. R. Hinden, S. Deering, R. Fink, and T. Hain, Initial IPv6 Sub-TLA ID Assignments, RFC 2928 (September 2000).
3. R. Hinden and S. Deering, Internet Protocol Version 6 (IPv6) Addressing Architecture, RFC 3513 (April 2003).
4. R. Hinden, S. Deering, and E. Nordmark, IPv6 Global Unicast Address Format, RFC 3587 (August 2003).
8. C. Huitema and B. Carpenter, Deprecating Site Local Addresses, work in progress (March 2004).
9. D. Johnson and S. Deering, Reserved IPv6 Subnet Anycast Addresses, RFC 2526 (March 1999).
10. S. Thomson and T. Narten, IPv6 Stateless Address Autoconfiguration, RFC 2462 (December 1998).
11. S. Thomson, T. Narten, and T. Jinmei, IPv6 Stateless Address Autoconfiguration, work in progress (June 2004).

Chapter 4

# INTERNET CONTROL MESSAGE PROTOCOL FOR IPv6 (ICMPv6)

## 4.1 INTRODUCTION

Internet Control Message Protocol (ICMP) operates as a communication manager between routers and a host or between hosts. ICMP messages provide feedback for problems occurred on the routing path to the destination. For example, a router may not forward a packet due to various reasons such as the packet generated from a source node is too big to go through an intermediate router, the packet has some fatal errors, the router knows more optimized path to the destination, or the router does not have enough buffer capacity to handle the packet. Even a destination node may generate ICMP messages due to any of previously stated reasons. Once ICMP message is generated, it is handed down to IP layer which in turn encapsulates ICMP message with IP packet. That packet is transferred to the destination node using general IP routing mechanisms.[1]

IPv6 requires ICMP as IPv4 does, but several changes are made for IPv6. New protocol, ICMPv6 is defined in RFC 2463.[2,3] ICMPv6 is also mainly used to report errors encountered in processing packets, and performs diagnostic functions. This protocol plays very important role in Neighbor Discovery protocol[4] and Path MTU Discovery protocol.[5]

The 8-bit Message Type field of ICMPv6 message identifies each message type. Depending on the first bit, ICMPv6 messages are classified into two types: error messages and information messages. When the first bit is set to 0, the ICMP message belongs to error message.[11] When the first bit is set to 1, the ICMP message belongs to information message.[12] RFC 2463 defines several message types as follows.

- Message type value for error messages:
  1: Destination Unreachable
  2: Packet Too Big
  3: Time Exceeded
  4: Parameter Problem
- Message type value for information messages:
  128: Echo Request
  129: Echo Reply

ICMPv6 message follows IPv6 Header and one or more IPv6 Extension Headers. When a packet is an ICMPv6 message, the Next Header value of the nearest preceding header is set to 58. General ICMPv6 message format is shown in Fig. 4-1.

- The Type field of ICMP message indicates the type of the message whether it is error message or information message.
- The Code field is related with Type field and specifies the additional information of message.
- The Checksum field is used to determine whether ICMPv6 Header and IPv6 Header are corrupted. The function of this field has not been changed from ICMPv4.
- The data contained in the Message Body depends on the Type and Code value of ICMPv6 Header.

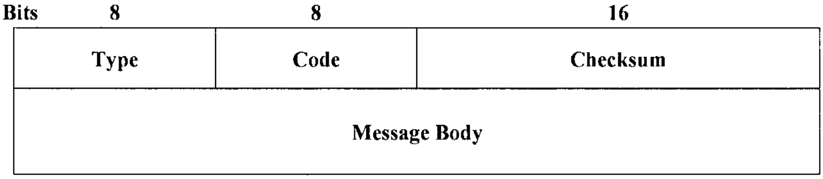


*Figure 4-1.* General ICMP message format.

[11] Message type value for error message lies between 0 and 127.

[12] Message type value for information message lies between 128 and 255.

## 4.2 RULES TO DETERMINE SOURCE ADDRESS FOR MESSAGE

ICMPv6 message is encapsulated with IPv6 packet. Thus, a node that wants to send an ICMPv6 message to another node which caused problems should determine the source and destination IPv6 addresses of the message. If the node is assigned with multiple unicast IPv6 addresses, it must select one as a source IPv6 address. There are several cases in selecting a source address as follows:

- If an ICMPv6 message is a response to a message destined to one of its unicast addresses, then the destination address of the original message is used as a source address of ICMP message.
- If an ICMPv6 message is a response to a message destined to multicast address or anycast group where the node is a member, then the source address of the response message should be one of its unicast addresses belonging to the interface where the original message is received.
- If an ICMPv6 message is a response to a message destined to a unicast address that does not belong to the node, then the node should select adequate one among its unicast addresses as the source address of the response message.
- Otherwise, node's routing table should be checked to determine which interface should be used to send an ICMPv6 message to the destination, and the node is required to select adequate one among its unicast addresses belonging to its interface as the source address of the message.

*Table 4-1.* ICMPv6 error message.

| Message type | Type field | Code field | Description |
|---|---|---|---|
| Destination Unreachable | 1 | 0 | No route to destination |
| | | 1 | Communication with the destination is administratively prohibited. |
| | | 2 | Not assigned |
| | | 3 | Address unreachable |
| | | 4 | Port unreachable |
| Packet Too Big | 2 | | No Code field is defined. |
| Time Exceeded | 3 | 0 | Hop limit exceeds in transit |
| | | 1 | Fragment reassembly time exceeds |
| Parameter Problem | 4 | 0 | Erroneous header field encounters |
| | | 1 | Unrecognized next header type encounters |
| | | 2 | Unrecognized IPv6 option encounters |

*Table 4-2.* ICMPv6 information message.

| Message Type | Type field | Description |
|---|---|---|
| Echo Request<br>Echo Reply | 128<br>129 | Both Echo Request and Echo Reply message are used in Ping command |
| Multicast Listener Query<br>Multicast Listener Report<br>Multicast Listener Done | 130<br>131<br>132 | Three messages are used in Multicast Listener Discovery. These will be employed in multicast group management protocol. |
| Router Solicitation | 133 | When a new node enters into a link, it may send out Router Solicitation message to request a router to send back Router Advertisement message immediately. |
| Router Advertisement | 134 | A router advertises its presence into its attached link periodically or in response to Router Solicitation message. This message will contain prefix(es) information used for address configuration and on-link determination. |
| Neighbor Solicitation | 135 | When a new node enters into a link and starts address duplication check, it first sends out Neighbor Solicitation message to find address duplication. This message is also used to determine the link-layer address of neighbor and to check reachable status of specific node. |
| Neighbor Advertisement | 136 | Neighbor Advertisement message is sent out in response to Neighbor Solicitation message. Unsolicited Neighbor Advertisement message is also used to announce a link-layer address change from a node. |
| Redirect | 137 | Redirect message is used to inform hosts of a better first hop to a destination. Usually a router sends out this message to a source node. |
| Home Agent Address Discovery Request | 144 | When a mobile node wants to learn home agent address dynamically, it uses a Home Agent Address Discovery Request message, which invokes the dynamic home agent address discovery mechanism. |
| Home Agent Address Discovery Reply | 145 | A home agent sends this message in response to Home Agent Address Discovery Request message from a mobile node. |
| Mobile Prefix Solicitation | 146 | When a mobile node is away from home link, it sends ICMP Mobile Prefix Solicitation message to solicit a Mobile Prefix Advertisement message. |
| Mobile Prefix Advertisement | 147 | A home agent sends this message periodically or in response to solicitation from mobile nodes. |

## 4.3 MESSAGE PROCESSING

When a node builds an ICMPv6 message, and when a node receives an ICMPv6 message specified in Table 4-1 and Table 4-2, the node processes the message with following rules. These rules are required to be implemented on every node:

- If an ICMPv6 error message with unknown type is received, it must be passed to the upper layer. However, ICMPv6 information message with unknown type should be discarded and should not be passed to the upper layer.
- Every ICMPv6 error message whose type value is less than 128 should include as much of the original IPv6 packet which invoked an error as possible, but resulting ICMP message can not exceed IPv6 MTU.
- When received ICMP message is required to be passed to the upper layer, Next Header type to determine an adequate upper transport layer should be checked to handle ICMPv6 message.
- An ICMPv6 error message should not be generated from a node if one of following conditions is met:
  A node receives an ICMPv6 error message.
  A node receives an IPv6 packet with multicast address in the Destination Address field.[13]
  A node receives an IPv6 packet with link-layer multicast address in the Destination Address field.
  A node receives an IPv6 packet with link-layer broadcast address in the Destination Address field.
  A node receives an IPv6 packet with non-unique address in the Destination Address field, e.g. IPv6 unspecified address.
- A node should limit the sending rate of ICMPv6 error messages to reduce the excessive network resource consumption due to incessant or unnecessary error messages on the network. To limit the rate of transmission of error messages, timer-based or bandwidth-based control mechanism may be employed.

[13] There are two exceptions: Packet Too Big message can be generated with multicast address in the Destination Address field, which allows path MTU discovery using IPv6 multicast address. Parameter Problem message can be also generated to report unrecognized IPv6 option.

## 4.4 MESSAGE FORMATS

### 4.4.1 Error messages

Four error messages are defined in ICMPv6; Destination Unreachable, Packet Too Big, Time Exceeded, and Parameter Problem messages. Each error message is distinguished by the Type Field in ICMPv6 message. Depending on type and code, each message type is subdivided, as shown in Table 4-1.

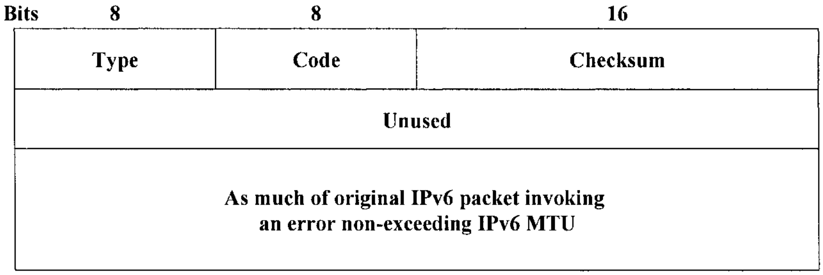
#### 4.4.1.1 Destination Unreachable

A Destination Unreachable message is generated by a router when an IP packet can not be delivered to the destination due to some specific reasons other than network congestions.[14] For example, when communication with the destination is administratively prohibited, or destination address is unreachable, the Destination Unreachable message is generated and transmitted by an intermediate router along the destination. Once ICMPv6 Destination Unreachable message is received, the IP layer of the receiving node should notify it to the upper layer protocol.

Code field in ICMP Header specifies a particular reason why a packet can not be delivered to the destination.

- If a packet can not be delivered due to the lack of a matching entry for the destination address in the forwarding routing table, the Code field is set to 0 in an error message. Only when a router does not have a default route to the destination in its routing table, it generates this error message, where Type field is set to 1 and Code field is set to 0.
- If a packet can not be delivered due to administrative prohibition, the Code field is set to 1 in an error message. When any firewall filter is found, this message can be generated.
- If a packet can not be delivered due to any other reason, such as inability to resolve the IPv6 destination address into a corresponding link address or a link-specific problem, then the Code field is set to 3 in an error message to notify address unreachable.
- Besides, if there is no proper or even alternative transport protocol to accept a packet, then the destination node should send back a Destination Unreachable message, where the Code field is set to 4.

[14] An ICMPv6 message should not be generated when a packet is dropped due to congestion.



| Bits 8 | 8 | 16 |
|---|---|---|
| Type | Code | Checksum |
| Unused | | |
| As much of original IPv6 packet invoking an error non-exceeding IPv6 MTU | | |

*Figure 4-2.* Destination Unreachable message format.

The ICMP message format is shown in Fig. 4-2.

- In IP Header:

  The source address of this error message is chosen by the source address selection rules.

  The destination address of ICMPv6 message is determined from the Source Address field of the erroneous packet.

- In ICMP message:

  The Type value of Destination Unreachable message is 1.

  The Code field is used to specify detailed information why the original packet may not be delivered. The code value is listed in Table 4-1.

  Unused field is reserved for all code values, and it must be initialized to zero by the sender and ignored by the receiver.

#### 4.4.1.2 Packet Too Big

A Packet Too Big message is generated by a router when a packet can not be forwarded due to the packet size. If the packet is larger than the MTU of the outgoing link, this message is generated by an intermediate router on the path to the destination. This ICMPv6 error message is used as the part of the path MTU discovery process. Once a node receives the Packet Too Big message, it must be notified to the upper layer protocol.

The packet format is shown in Fig. 4-3.

- In IP Header:

  The source address of this error message is chosen by the source address selection rules.

  The destination address of ICMPv6 message is determined from the Source Address field of the invoking packet.

- In ICMP message:

  The Type field of Destination Unreachable message is set to 2.

Unlikely Destination Unreachable message, the Code field is unused in this error message, and this field must be initialized to zero by the sender and ignored by the receiver.

The MTU field specifies the maximum path MTU.

There is an exception in sending rules when a node sends a Packet Too Big message, as explained in the Message Processing section. This message is generated in response to a packet with an IPv6 multicast destination address, a link-layer multicast, or link-layer broadcast destination address.

### 4.4.1.3 Time Exceeded

If a router receives a packet with a hop limit of zero, or a router decrements packet's hop limit down to zero, it must discard the packet and send an ICMPv6 Time Exceeded message with Code value 0 to the packet originator. This error message indicates either a routing loop or too small initial hop limit value. A Time Exceeded message must be passed to the upper layer protocol.

Time Exceeded message has same format as Destination Unreachable message in Fig. 4-2.

- In IP Header:

  The source address of this error message is chosen by the source address selection rules.

  Destination address of ICMPv6 message is determined from the source address field of the invoking packet.

- In ICMP message:

  The Type field of Destination Unreachable message is set to 3.

  The Code field specifies more detailed information which is listed in Table 4-1. The Code field is set to 0 if packet's hop limit exceeds in transit. It will be set to 1 when fragment reassembly time exceeds.

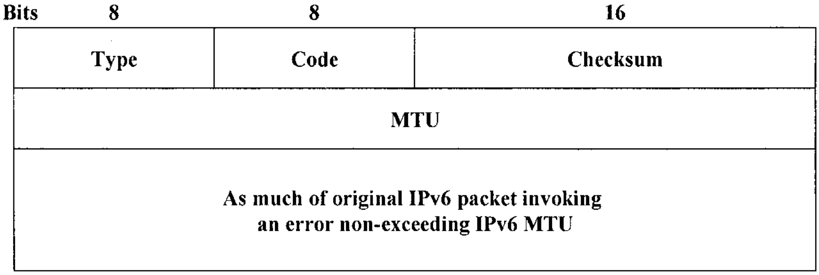

Bits
8
8
16
Type
Code
Checksum
MTU
As much of original IPv6 packet invoking
an error non-exceeding IPv6 MTU

*Figure 4-3.* Packet Too Big message format.

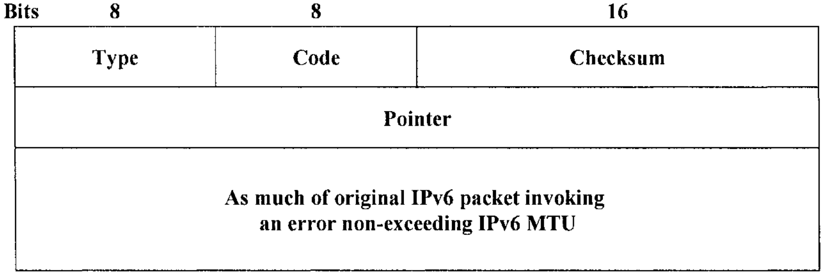

Bits
8
8
16
Type
Code
Checksum
Pointer
As much of original IPv6 packet invoking
an error non-exceeding IPv6 MTU

*Figure 4-4.* Parameter Problem message format.

### 4.4.1.4 Parameter Problem

When any IPv6 intermediate node on the path to the destination finds any problem from a field in the IPv6 Header or Extension Headers in processing a packet, it can not complete the processing of the packet. Then, it must discard the packet and should send a Parameter Problem message to the source node.

This ICMP message contains the erroneous packet and pointer which indicates location of the problem in the packet. If the location of an error is beyond the maximum size of Data field in ICMPv6 message, the pointer will point beyond the end of the ICMPv6 message.

The packet format is shown in Fig. 4-4. A node receiving this error message must pass it to the upper layer protocol.

- In IP Header:

  The source address of this error message is chosen by the source address selection rules.

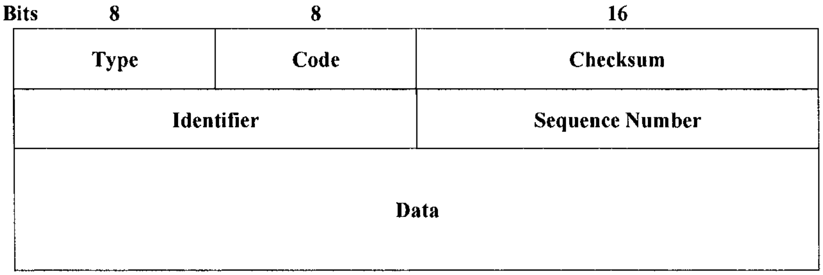
  The destination address is determined from the source address field of the invoking packet.

- In ICMP message:

  The Type field of Destination Unreachable message is set to 4.

  The Code field is set to 0 to inform the source of erroneous header field, 1 to inform the source of unrecognized Next Header type, and 2 to inform the source of the unrecognized IPv6 option.

  The Pointer field indicates the place of original packet's header where the error is perceived.



| Bits 8 | 8 | 16 |
|---|---|---|
| Type | Code | Checksum |
| Identifier | | Sequence Number |
| Data | | |

*Figure 4-5.* Echo Request message format.

For example, an ICMPv6 message whose Type field value is set to 4, Code field value is set to 1, and Pointer field contains 40 indicates that the IPv6 Extension Header following the IPv6 Header of the original packet had an unrecognized Next Header field value.

## 4.4.2 Information message

Only two types of information message are defined in RFC 2453, such as Echo Request message and Echo Reply message. Other information message types can be found in Path MTU Discovery protocol as well as Neighbor Discovery protocol.

The Echo Request and Reply messages are used for one of the mostly common TCP/IP utilities, 'ping' (Packet INternet Gropher), which determines whether a specified host is available on the network and can communicate with the peer. Once a host on the network receives Echo Request message, it should respond with Echo Reply message.

### 4.4.2.1 Echo Request

The Echo Request message format is shown in Fig. 4-5.

- In IP Header:

  The source address of this ICMPv6 message is chosen by the source address selection rules.

  The destination address of an ICMPv6 message can be any available IPv6 address.

- In ICMP message:

  The Type field of Echo Request message is set to 128.

  The Code field is unused in this message type, and this field must be initialized to zero by the sender and ignored by the receiver.

New fields, such as Identifier and Sequence Number fields are defined. They are used to identify an Echo Reply message matching to an Echo Request message. However, their value may be set to zero.

Data field will be filled up with zero or arbitrary data in byte-unit.

All nodes are required to implement a function to generate Echo Reply messages responding to Echo Request messages. Echo Request messages may be passed to upper layer protocol when a node receives ICMP messages. Thus a node should implement an application-layer interface to handle Echo Request and Echo Reply messages.

### 4.4.2.2 Echo Reply

In response to Echo Request message, a node should send back Echo Reply message. The Echo Reply message format is as same as the Echo Request message format, as shown in Fig. 4-5.

- In IP Header:

  The source address of this Echo Reply message should be same as the destination unicast address of the invoking Echo Request message.

  The destination address of ICMPv6 message is copied from the source address of the Echo Request message.
- In ICMP message:

  The Type field for Echo Request message is set to 129.

  The Code field is unused in this message type, and this field must be initialized to zero by the sender and ignored by the receiver.

  The Identifier and Sequence Number are copied from invoking Echo Reply message. Later, the node invoking ICMP message exchange will match the Identifier and sequence to the original ones

  The data copied from ICMPv6 Echo Request message must be put into the Data field of ICMPv6 Echo Reply message without any change.

Echo Reply messages must be passed to the upper layer protocol which originated an Echo Request message or may be passed to other upper layer protocol that did not originate the Echo Request message.

Even if a node receives an Echo Request message which is destined to a multicast address, it should generate an Echo Reply message in response. In this case, a unicast address assigned to this node will be used as a source address of the reply message. The source address selection rules can be adopted.

## REFERENCES

1. D. Comer, Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture (Prentice Hall, 1994).
2. A. Conta and S. Deering, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC 2463 (December 1998).
3. A. Conta and S. Deering, Internet Control Message Protocol (ICMPv6) for the Internet 4. Protocol Version 6 (IPv6) Specification, work in progress (June 2004).
4. T. Narten, E. Nordmark, and W. Simpson, Neighbor Discovery for IP version 6, RFC 2461 (December 1998).
5. J. McCann, S. Deering, and J. Mogul, Path MTU Discovery for IP version 6, RFC 1981 (August 1996).

Chapter 5

# NEIGHBOR DISCOVERY

## 5.1 INTRODUCTION

Neighbor Discovery (ND) protocol defined in RFC 2461 offers a number of advantages to IPv6.[1] This protocol integrates several services which are already used in IPv4, such as Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), router discovery, and redirect service. It also has newly added functions such as parameter discovery, address autoconfiguration, next-hop determination, neighbor unreachability detection, and duplication address detection, which are not covered in IPv4. In Table 5-1, services provided by ND protocol are specified in detail. We explain only four services provided by ND in this chapter: address resolution, neighbor unreachability detection, router and prefix discovery, and redirect. Duplication address detection will be discussed in the following chapter. New services may be added in ND in the near future.

- Address resolution is the process by which a node determines link-layer address from a given IP address. Sending nodes should perform address resolution to send unicast packets if they do not have knowledge about the link-layer address corresponding to the unicast address.
- Neighbor Unreachability Detection (NUD) enables a node to confirm that packets sent to the neighbor will be correctly forwarded to the destination. NUD may provide significant improvements in robustness of the packet delivery process. It is especially useful for unreliable networks because a part of the routing path in the network can be failed.
- Duplication Address Detection (DAD)[2-4] allows a node to find out whether the specific address it try to use is already allocated to other

node in a link. This mechanism is explained in detail in the following chapter.

- Once DAD for node's link-local address is successfully done, router discovery process may follow to learn prefix information. The node may solicit advertisement from an on-link router, or it may get periodical advertisement messages from routers. There may be multiple routers on-link, and nodes may get more than one advertisement message from different routers. Then, the nodes will record addresses of all routers for the emergency.
- When a router informs a node of a better first-hop node to reach a destination, it may perform redirect function. This function is identical to the redirect in IPv4.

All messages defined in ND protocol are contained in ICMPv6 packets.[5, 6] ICMPv6 Header and ND message header will appear following IPv6 Header and Extension Headers. There are five new ICMP message types for ND: a pair of Router Solicitation and Router Advertisement messages, a pair of Neighbor Solicitation and Neighbor Advertisement messages, and a Redirect message. These messages are listed in Table 5-2.

*Table 5-1.* Services from Neighbor Discovery protocol.

| Services | Description |
|---|---|
| Router Discovery | An algorithm to locate routers at an attached link |
| Prefix Discovery | An algorithm to discover the set of address prefixes that define which nodes are on-link for an attached link |
| Parameter Discovery | An algorithm to learn link parameters like link MTU or Internet parameters like hop limit used to determine to forward an outgoing packet |
| Address Autoconfiguration | An algorithm to configure an address for an interface automatically |
| Address Resolution | An algorithm to determine the link-layer address of on-link destination with given destination's IP address |
| Next-Hop Determination | An algorithm to determine the next hop to send any traffic for a destination. The next hop can be an intermediate router or the destination itself. |
| Neighbor Unreachability Detection | An algorithm to determine whether a specific neighbor is no longer reachable. If unreachable neighbor is a router, non-router node will try to find the other and select new router as a default. |
| Duplication Address Detection | An algorithm to determine a tentative address is already used by other node |
| Redirect | An algorithm to inform a host of a better first-hop node to reach a specific destination. It is usually used by routers. |

## 5.2 CONCEPTUAL MODEL OF A HOST

Databases maintained in hosts should be firstly considered to understand Neighbor Discovery protocol. Hosts are required to maintain information for each interface, such as Neighbor Cache, Destination Cache, Prefix List, and Default Router List.

- Neighbor Cache: Neighbor Cache records information about individual neighbors to which traffic has been recently sent. The most important information is neighbors' on-link unicast IP address. Each record contains link-layer address, a flag indicating whether the neighbor is a router, a pointer to any queued packets waiting for the address resolution, reachability information, the number of unanswered probes, and the time the next NUD will happen. Information from the last three fields is used for NUD. Neighbor's reachability information can be categorized into five states; incomplete, reachable, stale, delay, and probe.
- Destination Cache: Destination Cache records destination addresses to which traffics have been sent recently. This record maps a destination IP address from a packet into the IP address of next-hop neighbor.
- Prefix List: Prefix List records prefixes used on the link. This information is built from Router Advertisement messages.
- Default Router List: Default Router List records the default router to which packets will be sent.

*Table 5-2.* Message types for Neighbor Discovery protocol.

| Message type | Description |
|---|---|
| Router Solicitation | When a new node enters into a link, it may send out Router Solicitation message to request a router to send back Router Advertisement message immediately. |
| Router Advertisement | A router advertises its presence into its attached link periodically or in response to Router Solicitation message. This message type contains prefix (or prefixes) information used for the address configuration and on-link determination. |
| Neighbor Solicitation | When a new node enters into a link and starts Duplication Address Detection (DAD), it first sends out Neighbor Solicitation message to check the address duplication. This message is also used to determine the link-layer address of neighbor and to check reachability status of the specific node. |
| Neighbor Advertisement | Neighbor Advertisement message is sent out in response to Neighbor Solicitation message. Unsolicited Neighbor Advertisement message is also used to announce a link-layer address change of a node. |
| Redirect | Redirect message is used to inform hosts of a better first hop to a destination. Usually a router sends out this message to a source node. |

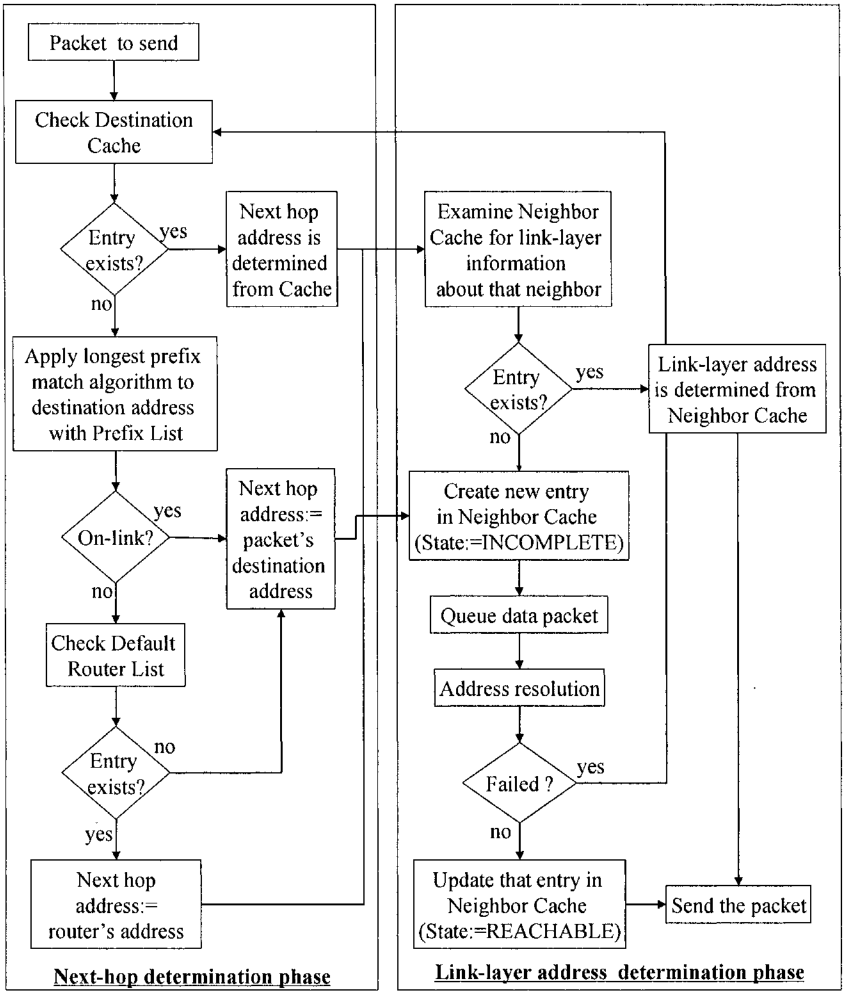

Packet to send
Check Destination Cache
Entry exists?
yes
no
Next hop address is determined from Cache
Apply longest prefix match algorithm to destination address with Prefix List
On-link?
yes
no
Next hop address:= packet's destination address
Check Default Router List
Entry exists?
no
yes
Next hop address:= router's address
Next-hop determination phase
Examine Neighbor Cache for link-layer information about that neighbor
Entry exists?
yes
no
Link-layer address is determined from Neighbor Cache
Create new entry in Neighbor Cache (State:=INCOMPLETE)
Queue data packet
Address resolution
Failed ?
yes
no
Update that entry in Neighbor Cache (State:=REACHABLE)
Send the packet
Link-layer address determination phase

*Figure 5-1.* Sending algorithm.

For example, when a node tries to send packets to the destination, combined information from the Destination Cache, the Prefix List, and the Default Router List is required to find the IP address of the appropriate next hop node. This process is called next-hop determination.

### 5.2.1 Sending algorithm

To transmit a packet to a destination, two determination phases must be performed: next hop determination and link-layer address determination. Next hop determination algorithm will determine the next hop to send any packet to the destination. The next hop may be an intermediate router or the destination itself.

Next hop determination is not performed for every outgoing packet due to the efficiency and availability of network resources. Instead, when a node has a packet to send, it firstly consults its Destination Cache. If any record for the destination is found in the cache, next-hop determination is unnecessary to be performed.

Once a next hop address is determined, the sending node must determine link-layer address, as shown in right of Fig. 5-1.

1. The node will consult Neighbor Cache whether a proper record exists. If no proper one is found, the sender creates new record with 'incomplete' state and starts address resolution process.
2. Once the sending node receives Neighbor Advertisement message, link-layer address for the corresponding neighbor is eventually known. Then, the sender will update Neighbor Cache and start to send packets.

## 5.3 SERVICES FROM NEIGHBOR DISCOVERY PROTOCOL

### 5.3.1 Router discovery

Once a node configures its link-local address, it starts router discovery phase to learn prefix information and configuration parameters related to the address autoconfiguration. Router discovery may be used to locate neighboring routers on the same link. Through the router discovery process, a host learns the range of addresses which can be reached directly without any intermediate router.

A pair of Router Solicitation and Router Advertisement message is exchanged between a node and a router. A host may send Router Solicitation message and start router discovery. It may get Router Advertisement message periodically generated by on-link routers.

#### 5.3.1.1 Router's aspect

Router Advertisement message may contain information which is necessary for receiving nodes to designate the advertising router as a default router, such as list of prefixes usable on a link which allow hosts to perform address autoconfiguration, flags associated with prefixes that specify whether hosts will use stateful or/and stateless address configuration, and some Internet parameters, such as the hop limit and link MTUs. The Router Advertisement message also contains Prefix Information option to notify receiving on-link nodes of prefix information.

When a router sends Router Advertisement message in response to the valid solicitation from on-link node, unicast address can be used in the destination address field of Router Advertisement message. However, all-nodes multicast address is usually used. A router on link will send unsolicited Router Advertisement messages periodically or randomly.

#### 5.3.1.2 Host's aspect

Once a host receives Router Advertisement message from on-link router, it builds and maintains a Prefix List using advertised on-link prefixes. When multiple routers are present on a link, hosts may receive different Router Advertisement messages or a collected Router Advertisement message which contains multiple prefixes used on the link. In addition, hosts may get router information through the stateful autoconfiguration process. Hosts are required to compare a newly received message with earlier one and update a specific parameter or option value. The process that a host handles received Router Advertisement message is shown in Fig. 5-2.

A host does not have to add all on-link routers into its Default Router List; rather it may prefer not to store all of them. It is compulsory for a host to have at least two router addresses. After processing the fixed portion of Advertisement messages, the host handles Option fields.

Instead of waiting for Router Advertisement messages, hosts may solicit Advertisement messages to locate routers or to learn prefixes. The message format is shown in Fig. 5-3. The target address in Options field should not contain unspecified address. Hosts should not send Router Advertisement messages. They must silently discard any received Router Solicitation messages from non-routers.

### 5.3.2 Address resolution

Address resolution is the process which a node determines link-layer address from a given IP address. When there is a unicast packet to send, but there is no knowledge about link-layer address corresponding to the unicast address, a sending node performs address resolution. However, it should not be performed for multicast addresses. Address resolution mechanism is only made up of Neighbor Solicitation and Neighbor Advertisement messages.

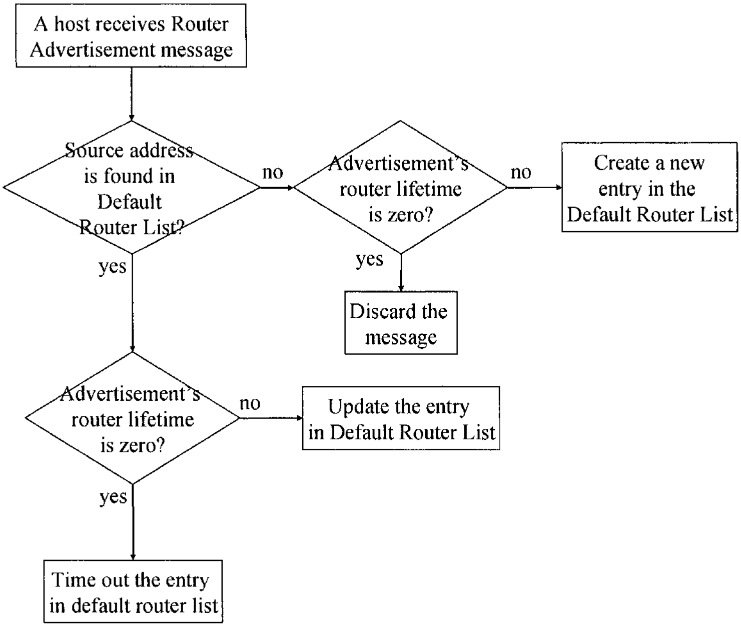

A host receives Router Advertisement message
Source address is found in Default Router List?
no
Advertisement's router lifetime is zero?
no
Create a new entry in the Default Router List
yes
Discard the message
yes
Advertisement's router lifetime is zero?
no
Update the entry in Default Router List
yes
Time out the entry in default router list

*Figure 5-2.* Processing Router Advertisement message.

**IP packet**
**Destination address**
All-routers multicast address
**Source address**
IP address assigned to the sending node or unspecified address

**ICMP message**
**Type**
133
**Options**
...

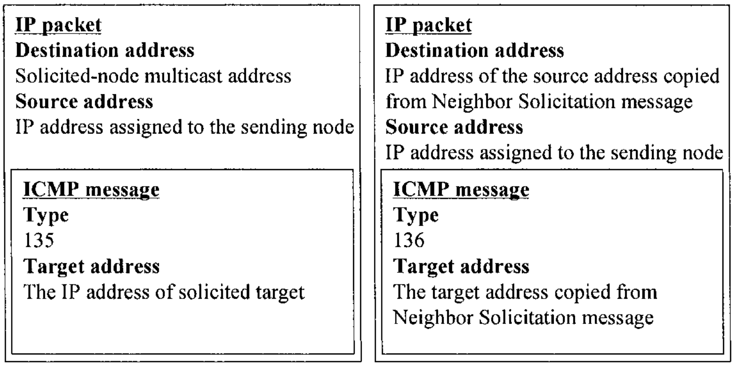
(a) Router Solicitation message

**IP packet**
**Destination address**
All-nodes multicast address
**Source address**
Link-local address assigned to router

**ICMP message**
**Type**
134
**Options**
Prefix Information
List of valid on-link prefixes

(b) Router Advertisement message

*Figure 5-3.* Exchanged messages for router discovery.



**IP packet**
**Destination address**
Solicited-node multicast address
**Source address**
IP address assigned to the sending node

**ICMP message**
**Type**
135
**Target address**
The IP address of solicited target

(a) Neighbor Solicitation Message

**IP packet**
**Destination address**
IP address of the source address copied from Neighbor Solicitation message
**Source address**
IP address assigned to the sending node

**ICMP message**
**Type**
136
**Target address**
The target address copied from Neighbor Solicitation message

(b) Neighbor Advertisement Message

*Figure 5-4.* Exchanged messages for the address resolution.

The sending node firstly creates a new entry with incomplete state in a Neighbor Cache and sends Neighbor Solicitation message to its neighbors. This Solicitation message is sent to the solicited multicast address corresponding to the target address, which is identified by FF02:0:0:0:0:1:FFXX:XXXX, where sequential Xs are lower 24 bits of 64 bits Identifier ID. This address type is explained in Chapter 3. The packet formats used in the address resolution process are shown in Fig. 5-4.

A node may start a new address resolution process for another node even if the previous process has not been completed. When the node waits for the previous address resolution to complete, it will queue the subsequent address resolution request packet until the previous address resolution is over. A queue is required to have at least one packet for a session. The queue is handled according to first in first out (FIFO) policy. When it is overflowed, newly arrived packet will replace the oldest entry. Responded entry will be transmitted regardless of its position in the queue.

If a node receives a Neighbor Solicitation message form its neighbor, it should verify that message as follows:

- The target address in the Neighbor Solicitation message should be a valid unicast or anycast address assigned to the receiving interface.
- If the target address is a tentative one, then it must be under DAD process. This Neighbor Solicitation message should be handled according to the address configuration process.

After the message is verified, the receiving node checks whether there is a Source Link-Layer Address option if the target address is not the unspecified address. The receiving node is required to create or update Neighbor Cache for the source address of the Solicitation message and sends back a Neighbor Advertisement message to the soliciting node. The response message format to the Neighbor Solicitation message is shown in Fig. 5-4 (b). The target address is copied from the Neighbor Solicitation message.

When a node receives a valid Neighbor Advertisement message, it searches for the target entry in its Neighbor Cache. If no entry is present, the node discards that message. Otherwise, the node will take appropriate action on that target entry of its Neighbor Cache. In some cases, a node's link-layer address can be changed. A node will send unsolicited Advertisement message to notify neighbors of that change.

To inform that the message is unsolicited one, the solicited flag of unsolicited message must be set to zero. If the sending node is a router, the router flag must be set to one. New link-layer address is contained in the Target Link-layer Address option.

### 5.3.3 Neighbor unreachability detection

Communication failures in a network may occur due to various reasons at any time. If the destination fails, there is no way to make successful communication. However, if an intermediate node along the path to the destinations fails, recovery mechanism will be helpful to reconstruct a path between the source and the destination. NUD may be used for all paths between nodes, for instance, host to host, host to router, router to host, and router to router. However, the last case may not be commonly used because routing protocols can solve the unreachability problem by itself. NUD detects the failure of a neighbor or the intermediate router to a destination. When a node wants to confirm that packets sent to the neighbor are being correctly forwarded to the destination and being processed properly, it sends Neighbor Solicitation unicast message for the reachability confirmation from the next hop.

Neighbor unreachability is handled depending on whether the destination is a neighbor or not. If the destination is an on-link node, address resolution should be performed again. If the destination is an off-link node, the default router should be used as an intermediate destination instead of former neighboring router. NUD should be performed only for the unicast address. NUD should not be performed if there is no traffic to send to a neighbor. Thus, after a node performs NUD, it will send packets to the neighbor.

The recovery process is working with the next-hop determination. Let's go back to the Fig. 5-1. If there are data to send, but no link-layer address is known, a node needs to perform address resolution process for a neighbor's address. If the address resolution fails, a node invokes the next-hop determination procedure again, and alternate default routers may be tried. If a reachability confirmation is received from the neighbor, then the path is recovered.

### 5.3.4 Redirect function

When a router finds a better first-hop for a host to a destination, it sends Redirect packets to inform a host of it. Then, traffics from the host can be redirected using a better route. Redirect message format is shown in Fig. 5-9. The Target Address field in Redirect message contains the better next hop to the destination address and the Destination Address field contains real destination address copied from a packet which triggered redirect. If the target address is equal to the destination address, then this message notifies that the destination node is in the neighborhood of the host. No response message to the Redirect message is required.

A router should know the link-local address of its neighboring routers to identify routers using the link-local address in the target address of Redirect message. A host should not originate a Redirect message. When a host receives a valid Redirect message, it should update the corresponding entry in its Destination Cache to redirect subsequent traffics. If no entry for the destination exists, a host should create a new one.

## 5.4 MESSAGES FORMATS

### 5.4.1 Router Solicitation

A host sends out Router Solicitation message to find a router in its attached link, and this message triggers a router to generate Router Advertisement message. ICMP message format for Router Solicitation message is shown in Fig. 5-5.

In IP Header:

- The source address of Router Solicitation message becomes an IP address assigned to the sending interface. If this message is generated by a node under DAD process, source address field is filled with unspecified address.
- The destination address of Router Solicitation message is usually the all-routers multicast address.

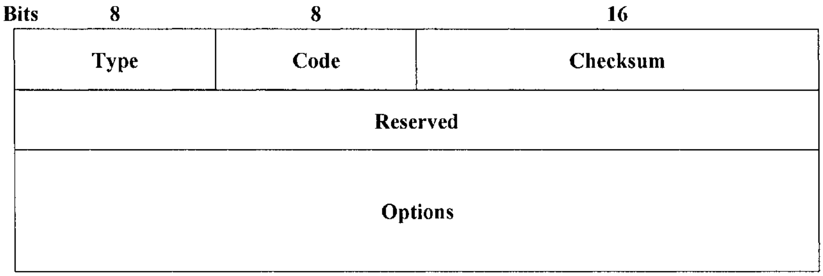


*Figure 5-5.* Router Solicitation message format.

- If any security association for the IP Authentication Header exists between a sender and a destination address, then the sender should include Authentication Header.
- Hop limit is set to 255.
  In ICMP message:
- The Type field of Router Solicitation message is set to 133.
- The Code field is set to zero.
- The Reserved field is unused and must be initialized to zero by the sender. If this field has some value other than zero, then the receiver must ignore it.
- In the Option field, source link-layer address can be contained, if an interface is assigned with an address. Unspecified address can not be contained in the Option field. New option type can be defined, and receivers must ignore any option that they do not understand and keep up processing the message.

## 5.4.2 Router Advertisement

A router sends out Router Advertisement message in response to Router Solicitation message or periodically. ICMP message format for Router Advertisement message is shown in Fig. 5-6.

- In IP Header:
  The source address of Router Advertisement message should be a link-local address assigned to the sending interface where Router Solicitation message is received.
  The destination address of Router Advertisement message is usually source address of the received Router Solicitation message or all-nodes multicast address.

If any security association for the IP Authentication Header exists between a sender and a destination address, then the sender should include Authentication Header.

Hop limit is set to 255.

- In ICMP message:

  The Type field of ICMP Router Advertisement message is set to 134.

  The Code field is set to zero.

  The Current Hop Limit field has 8-bit unsigned integer value. For outgoing IP packets, the Current Hop Limit field should be set to the hop count field of the IP Header as a default. When the Current Hop Limit field is set to zero, it specifies 'unspecified' by the forwarding router.

  Two special bits are defined in Router Advertisement message; *M* and *O*. *M* bit is used to specify 'managed address configuration'. When *M* bit is set, hosts use both the stateful address autoconfiguration protocol and stateless address autoconfiguration protocol for address autoconfiguration.

  *O* bit is used to specify 'other stateful configuration'. When *O* bit is set, hosts use the stateful address autoconfiguration protocol for non-address information autoconfiguration. The usage of these *M* and *O* bits is described in Chapter 6.

  The Reserved field is composed of 6 bits. It must be initialized to zero by the sender. If this field has some value other than zero, then the receiver must ignore it.

| Bits 8 | 8 | | | 16 |
|---|---|---|---|---|
| **Type** | **Code** | | | **Checksum** |
| **Current Hop Limit** | M | O | **Reserved** | **Router Lifetime** |
| **Reachable Time** | | | | |
| **Retransmission Timer** | | | | |
| **Options** | | | | |

*Figure 5-6.* Router Advertisement message format.

The Router Lifetime field associated with the default router in second unit contains 16-bit unsigned integer value. If the Router Lifetime field is set to 0, it indicates that the router can not be a default router, thus should not be set as a default router. The router lifetime influences receiving nodes to select the router as a default router.

The Reachable Time field contains 32-bit unsigned integer. This field is used by the neighbor unreachability detection algorithm. The time in Reachable Time field is represented in milliseconds. If a node receives any ICMP packet whose Reachable Time field is set to non-zero, then the receiving node assumes that the neighbor is reachable after that specified time. When the router does not want to specify any value, this field is set to 0.

The Retransmission Timer field contains 32-bit unsigned integer. This field is used by the address resolution as well as the neighbor unreachability detection algorithm. Time in the Retransmission Timer field is also represented in milliseconds. When the router does not want to specify any value, this field is set to 0.

In the Option field, Source Link-Layer Address, MTU and Prefix Information options can be contained. Sending node's interface address may be contained in the Source Link-layer Address option. If each link has different MTU value, MTU information should be sent on such links. Prefix Information option contains the on-link prefixes, which may be used for the address autoconfiguration. A router should include all of its on-link prefixes in Router Advertisement message, which allows multihomed hosts to have complete prefix information for the links where they are attached. New option type can be defined, and receivers must ignore any option that they do not understand.

### 5.4.3 Neighbor Solicitation

A node sends Neighbor Solicitation messages to request the link-layer address of a target node while it provides its own link-layer address. This Neighbor Solicitation message will be forwarded as a multicast packet when a node wants to resolve a specific address. It will be forwarded as a unicast packet when a node wants to verify the reachability of a specific neighbor node. ICMP message format for Neighbor Solicitation message is shown in Fig. 5-7.

- In IP Header:

  The source address of Neighbor Solicitation message can be either an unspecified address or unicast IPv6 address assigned to the interface of the sending node. When a node is under DAD, unspecified address is used as a source address.

The destination address of Neighbor Solicitation message can be either the target address or the solicited-node multicast address corresponding to the target address.

If any security association for the IP Authentication Header exists between a sender and a destination address, then the sender should include the Authentication Header.

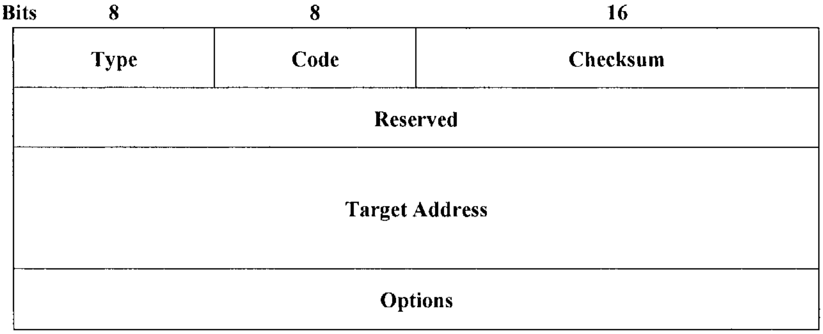
Hop limit is set to 255.

- In ICMP message:

  The Type field of ICMP Neighbor Solicitation message is set to 135.

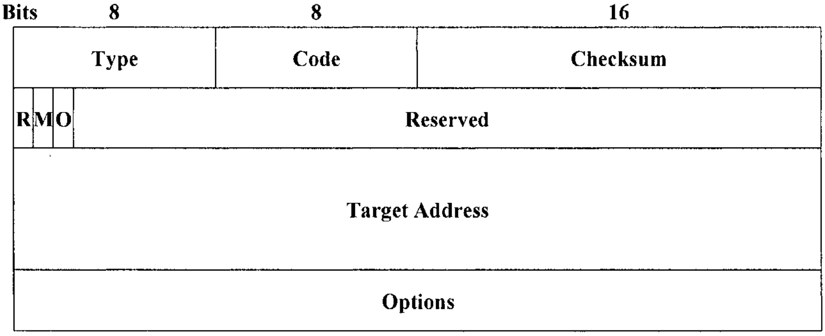
  The Code field is set to 0.

  The Reserved field is unused and must be initialized to zero by the sender. If this field has some value other than zero, then the receiver must ignore it.



| Bits 8 | 8 | 16 |
|---|---|---|
| Type | Code | Checksum |
| Reserved | | |
| Target Address | | |
| Options | | |

*Figure 5-7.* Neighbor Solicitation message format.



| Bits 8 | 8 | 16 |
|---|---|---|
| Type | Code | Checksum |
| R M O Reserved | | |
| Target Address | | |
| Options | | |

*Figure 5-8.* Neighbor Advertisement message format.

The Target Address field contains the IP address of the solicited target, and it should not be a multicast address.

For Neighbor Solicitation message, Source Link-Layer Address option may be used in the Option field. Source link-layer address is the link-layer address for the sender, but it can not be included in Option field when source address is unspecified yet. In other cases, this option should be included no matter what the message is multicast solicitation or unicast solicitation. New option types can be defined, but receivers must ignore any option that they do not understand.

### 5.4.4 Neighbor Advertisement

Nodes send Neighbor Advertisement messages in response to Neighbor Solicitation messages. Even if there is no solicitation, nodes may send Neighbor Advertisement messages to propagate new information quickly. ICMP message format for Neighbor Advertisement message is shown in Fig. 5-8.

- In IP Header:

  The source address of Neighbor Advertisement message is unicast IPv6 address assigned to the interface of the sending node.

  There are three cases for the selection of the destination address. When a Neighbor Advertisement message is sent in response to Neighbor Solicitations, the destination address is copied from the source address of the Solicitation message with the exception of unspecified IP address. If unspecified address is used in the Source Address field of Solicitation message, all-nodes multicast address fills up the Destination Address field of Neighbor Advertisement message. For unsolicited Advertisement message, the all-nodes multicast address is commonly used.

  If any security association for the IP Authentication Header exists between a sender and a destination address, then the sender should include the Authentication Header.

  Hop limit is set to 255.

- In ICMP message:

  The Type field of ICMP Neighbor Advertisement message is set to 136.

  The Code field is set to zero.

  Three special bits are defined in ICMP Router Advertisement message, namely, *R*, *S* and *O*. The *R* bit is a router flag. When the *R* bit is set, it indicates that the sending node is a router. The *R* bit is used to detect neighbor unreachability. The *S* bit is a solicited flag. When the *S* bit is set, it indicates that this packet is sent in response to a Neighbor Solicitation message. The *S* bit is also used to detect neighbor unreachability.

However, this bit should not be set when a message is sent to the multicast destination or when a Neighbor Advertisement message is built without solicitation from any other nodes. The *O* bit is an override flag. When this bit is set, it indicates that link-layer address in the advertisement packet is the latest one and should update each node's cached link-layer address. The *O* bit should not be set in solicited Advertisement message destined to the anycast address.

The Reserved field is composed of 29 bits and must be initialized to zero by the sender. If this field has some value other than zero, then the receiver must ignore it.

The Target Address field is used in two ways. In the case of solicited advertisement, this field is filled with the address from the Neighbor Solicitation message corresponding to this Neighbor Advertisement message. In the case of the unsolicited advertisement, the purpose of Neighbor Advertisement message is informing the change in the link layer address. In that case, this field will be filled with the new link layer address. Multicast address can not be the part of Target Address field.

In the Option field, Target Link-Layer Address option is contained in the Neighbor Advertisement message. When there is multicast or unicast Neighbor Solicitation message, a node responding to this solicitation must include this option in an Advertisement message. New option type can be defined. However, a receiver must ignore any option that it does not understand.

### 5.4.5 Redirect

When a router finds a better first-hop for a node to the destination, it sends Redirect messages to inform a host of it. Then, packets from the host can be redirected to a better route. If target address in Redirect message is equal to the destination address, then this message notifies that the destination is an on-link node. The Redirect message format is shown in Fig. 5-9.

- In IP Header:

  The source address of a Redirect message is a link-local address assigned to the interface where this message is generated and transmitted

  The destination address of a Redirect message is a source address of the packet that triggered the Redirect.

  If any security association for the IP Authentication Header exists between a sender and a destination address, then the sender should include the Authentication Header.

  Hop limit is set to 255.

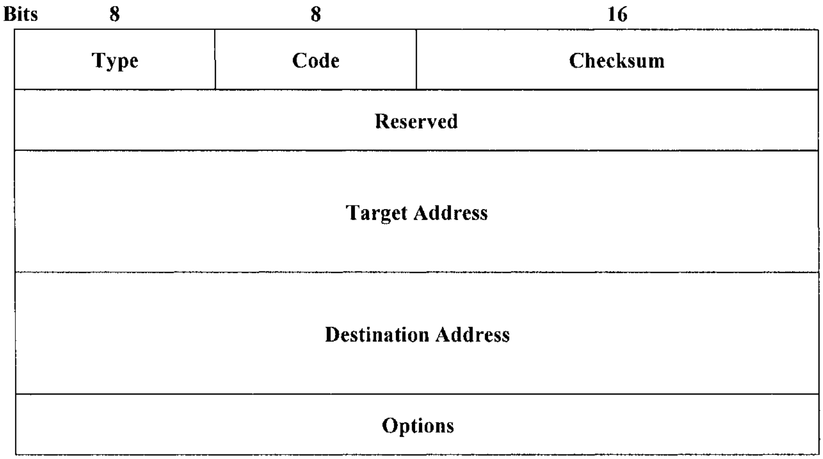


*Figure 5-9.* Redirect message format.

- In ICMP message:
  The Type field is set to 137.
  The Code field is set to 0.
  The Reserved field is unused and must be initialized to zero by the sender. If this field has some value other than zero, then the receiver must ignore it.
  In the Target Address field, a better first hop to the destination is specified. If the target address in this field is the same as the address in the Destination Address field, then this address implies that the destination node is a neighbor.
  There are two options defined in the Redirect message, namely, Target Link-Layer Address and Redirected Header options. The Target Link-Layer Address option contains the link-layer address of the target node. In the Redirected Header option, as much as possible portion of the IP packet that triggered the Redirect message is contained.[15]

[15] Redirect packet can not exceed 1280 bytes.

## 5.5 OPTIONS

Messages defined in Neighbor Discovery protocol have zero or more options, and some of options may appear multiple times in the same message. General option format is shown in Fig. 5-10. As a common feature of option format, there are Type and Length fields. Depending on the type value, the option is classified into five types, as specified in Table 5-3.

- The type field is composed of 8 bits identifier. Type value and name of each type are given in Table 5-4.
- The length field is also composed of 8 bits and contains unsigned integer value. The Length field contains the length of option field including the type and length field in 8-byte unit. When the length field is set to 0, it is invalid value, and a node will silently discard any neighbor discovery message whose length field in the option contains zero.

In the following section, each option type is discussed in detail.

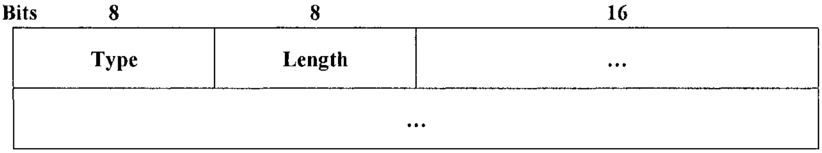


*Figure 5-10.* General Option format.

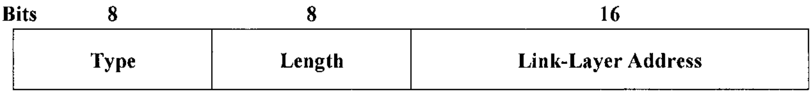


*Figure 5-11.* Option field for source/target link-layer address.

*Table 5-3.* Option types for ND messages.

| Type | Option name |
|---|---|
| 1 | Source link-layer address |
| 2 | Target link-layer |
| 3 | Address prefix information |
| 4 | Redirect header |
| 5 | MTU |

### 5.5.1 Source Link-Layer Address

The Source Link-Layer Address option contains sender's link-layer address. This option is used in Neighbor Solicitation, Router Solicitation, and Router Advertisement messages. When this Source Link-Layer Address option is specified in the other neighbor discovery messages, it should be silently ignored. The option format for the source link-layer address is shown in Fig. 5-11.

- The Type field for the Source Link-Layer Address option is set to 1.
- The Link-Layer address field has a variable length. Contents and format of this field are determined depending on the link layer.

### 5.5.2 Target Link-Layer Address

The Target Link-Layer Address option contains target's link-layer address. It is used in Neighbor Advertisement and Redirect messages. When this Target Link-Layer Address option is specified in the other neighbor discovery messages, it should be silently ignored.

Option format for target link-layer address has the same format as the one for source link-layer address.

- The Type field for the Target Link-Layer Address option is set to 2.
- The Link-layer address field has a variable length. Contents and format of this field are determined depending on the link-layer.

### 5.5.3 Prefix Information

The Prefix Information option contains on-link prefixes and prefixes for address autoconfiguration. This information will be provided for any host in the same link to help address autoconfiguration process. This option is used in the Router Advertisement message. When this Prefix Information option is specified in other neighbor discovery messages, it should be silently ignored. Option format for prefix information is shown in Fig. 5-12.

- The Type field for the Prefix Information option is set to 3.
- The Prefix Length field is composed of 8 bits, and it contains unsigned integer value. The length expresses the number of valid bits in the prefix field and extends to 128.
- Two special bits are defined in Prefix Information option, namely, *L* and *A*.

  The *L* bit represents on-link flag. When the *L* bit is set to one, it indicates that the prefix specified in the Prefix field can be used for on-link determination. Otherwise, the advertisement makes no statement about on-link or off-link properties of the prefix.

The *A* bit represents an autonomous address-configuration flag. When this bit is set to 1, it indicates that the prefix specified in the Prefix field can be used for the autonomous address configuration.

- There are two Reserved fields in the prefix information option, namely, Reserved1 and Reserved2.
  Reserved1 field is composed of 6 bits, while Reserved2 field is composed of 32 bits. Both Reserved fields are unused and must be initialized to zero by the sender. If these fields have any other value than zero, then they must be ignored by the receiver.
- The Valid Lifetime field is composed of 32 bits and has unsigned integer value. This lifetime is expressed in seconds relative to the time when the packet is sent. The advertised prefix is valid for the on-link determination during the valid lifetime. If all bits of this field are set to one, it represents infinity.
- The Preferred Lifetime field is composed of 32 bits and has unsigned integer value. This lifetime is expressed in seconds relative to the time when the packet is sent. The address generated from the prefix by means of stateless address autoconfiguration will be valid only on the preferred lifetime. If all bits of this field are set to one, it represents infinity.
- The Prefix field contains an IP address or a prefix of an IP address.

A router should not include a Prefix Information option in Router Advertisement message for the link-local prefix. Hosts should ignore such Prefix Information option.

### 5.5.4 Redirected Header

The Redirected Header option contains all or part of the packet that is redirected, and this option is used in Redirect message. When this Redirected Header option is specified in other neighbor discovery message, it should be silently ignored. Option format for the Redirected Header is shown in Fig. 5-13.

- The Type field for the Redirected Header option has 4.
- The Length field represents the length of the option. The content of the length field is expressed in 8-byte unit.
- The Reserved field is unused and must be initialized to zero by the sender. If this field has any other value than zero, then it must be ignored by the receiver.
- The last field of Redirected Header Option contains the truncated packet due to the maximum size of the Redirect message, which is 1280 bytes long.

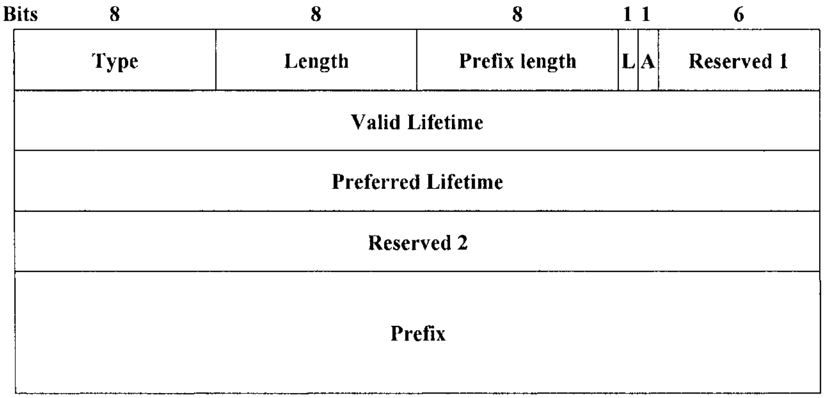


*Figure 5-12.* Option field for prefix information.

### 5.5.5 MTU

If multiple links are connect by a bridge and each link employs different layer 2 technology, then the maximum MTU on each link may be different. Nodes on such a link will be unable to determine the right size of MTU if the bridge does not generate ICMP Packet Too Big message to notify on-link nodes of dynamic PMTU[7] on each segment. In this case, routers use this MTU option to specify the maximum MTU size which can be supported by all links.

The MTU option is used to insure that all nodes on the same link use the same MTU value, especially when the link MTU is unknown. This option is used only in Router Advertisement message. When this MTU option is specified in other neighbor discovery messages, it should be silently ignored. The MTU option format is shown in Fig. 5-14.

- The Type field for the MTU option is set to 5.
- The Reserved field is unused and must be initialized to zero by the sender. If this field has non-zero value, then it must be ignored by the receiver.
- The MTU field contains the recommended MTU size for the link and it is expressed by 32-bit unsigned integer.

| Bits | 8 | 8 | 16 |
|---|---|---|---|
| | Type | Length | |
| | Reserved | | |
| | IP Header + Data | | |

*Figure 5-13.* Option field for Redirected Header.

| Bits | 8 | 8 | 16 |
|---|---|---|---|
| | Type | Code | Reserved |
| | MTU | | |

*Figure 5-14.* Option field for MTU.

## REFERENCES

1. T. Narten, E. Nordmark, and W. Simpson, Neighbor Discovery for IP version 6, RFC 2461 (December 1998).
2. S. Thomson and T. Narten, IPv6 Stateless Address Autoconfiguration, RFC 2462 (December 1998).
3. S. Thomson, T. Narten, and T. Jinmei, IPv6 Stateless Address Autoconfiguration, work in progress (June 2004).
4. N. Moore, Optimistic Duplicate Address Detection for IPv6, work in progress (June 2004).
5. A. Conta and S. Deering, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC 2463 (December 1998).
6. A. Conta and S. Deering, Internet Control Message Protocol (ICMPv6) for the Internet 4. Protocol Version 6 (IPv6) Specification, work in progress (June 2004).
7. J. McCann, S. Deering, and J. Mogul, Path MTU Discovery for IP version 6, RFC 1981 (August 1996).

Chapter 6

# ADDRESS AUTOCONFIGURATION

## 6.1 INTRODUCTION

When a node plugs in and wants to be a member of a network, information such as IP address and router information is needed for the configuration of that node. It can be configured manually or automatically with such information. In IPv4, manual configuration is common. This creates heavy burden on network administrators as well as users. However, IPv6 provides automatic address configuration feature to IPv6-enabled nodes, which allows assigning unique addresses and getting network information when they plug in.[1]

Autoconfiguration is composed of a sequence of processes which create a link-local address, verify and guarantee the uniqueness of assigned addresses, determine which information should automatically be configured, and decide whether stateless, stateful or both address configuration mechanism would be adopted. Each step issued above may be automatically performed in IPv6. Since autoconfiguration is one of major features of IPv6, it is hard to consider IPv6 and autoconfiguration separately.

This chapter explains how to build link-local and global addresses, and how to guarantee uniqueness on the address assignment. Both stateless and stateful address autoconfiguration are explained.

## 6.2 STATELESS AND STATEFUL AUTOCONFIGURATIONS

IPv6 defines two autoconfiguration mechanisms: stateless and stateful address autoconfigurations. When stateful autoconfiguration mechanism is employed for the address configuration, a specific server stores and manages whole address information for the managed domain. Thus, it is possible to manage address resources efficiently.

Almost no manual configuration on hosts is required in stateless autoconfiguration, and only minimal configuration is needed on routers.[2, 3] No additional servers are required for this mechanism. Simply, a host builds its own address using local information and information learned from advertisement messages from routers, which contain prefixes as well as several other parameters. The host may solicit messages instead of waiting for it.

On the contrary, a host obtains whole address information and parameters from a server in stateful address autoconfiguration.[4] The server keeps address database which lists assigned and un-assigned addresses and maps assigned address to a corresponding host.

Stateless and stateful autoconfiguration mechanisms can replace each other, or both mechanisms may be used together. For instance, a host may be assigned IP address using stateless autoconfiguration and learn other information through stateful autoconfiguration. Algorithm for address assignment is shown in Fig. 6-1.

In both mechanisms, IPv6 address and lifetime are tied up. IP address assigned to an interface may be regarded as leased one for fixed time duration. If the lifetime expires, the binding becomes invalid, and the address may be reassigned to another interface. For the graceful address expiration, two address states such as 'preferred' and 'deprecated' are defined for a valid address, as shown in Fig. 6-2.

Initially, a node is assigned with preferred IP address and uses it for a given lifetime. Any communication using the preferred IP address is not restricted. Later on, the address state becomes deprecated which implies that the assigned address becomes invalid. An address is valid if it is in preferred or deprecated state. However, it is strongly encouraged not to use the deprecated address for new communications. New communications should use the preferred address.

### 6.2.1 Algorithm for autoconfiguration

When an IPv6-node plugs in, it starts autoconfiguration process as follows:

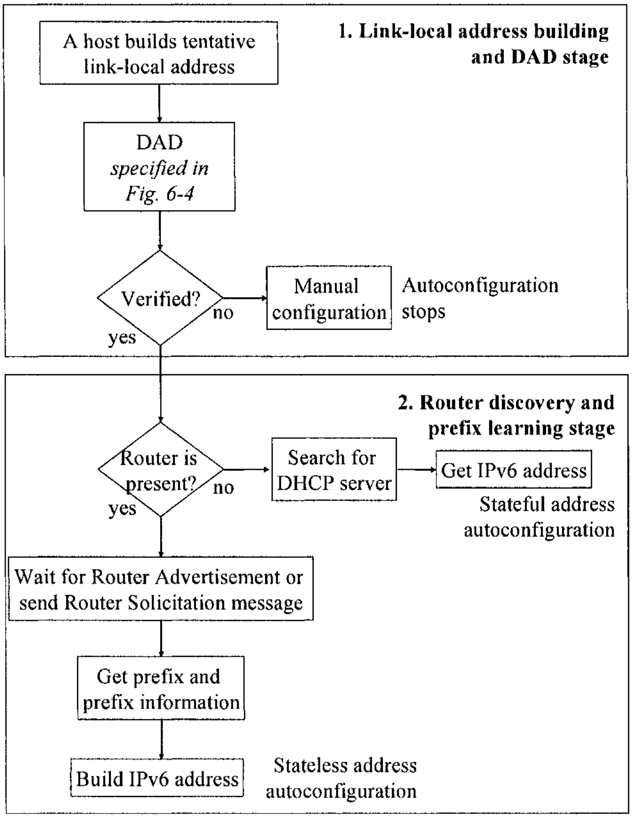

A host builds tentative link-local address
1. Link-local address building and DAD stage
DAD specified in Fig. 6-4
Verified?
no
yes
Manual configuration
Autoconfiguration stops
2. Router discovery and prefix learning stage
Router is present?
no
yes
Search for DHCP server
Get IPv6 address
Stateful address autoconfiguration
Wait for Router Advertisement or send Router Solicitation message
Get prefix and prefix information
Build IPv6 address
Stateless address autoconfiguration

*Figure 6-1.* Algorithm for address autoconfiguration.

1. An IPv6 generates 64-bit length interface identifier that distinguishes its interface from others on the subnet. The 64-bit interface identifier is induced from 48-bit MAC address recorded at each interface card, as explained in Chapter 3. Then, the node is able to build its link-local address by appending an interface identifier to the well-known link-local prefix, which is FE80::/10. The link-local address format is shown in Fig. 3-7 (a).
2. The node now joins several multicast groups, such as all-nodes multicast group and solicited-node multicast group. All-nodes multicast address is identified by FF01::1 and FF01::2, and solicited-node multicast address is identified by FF02:0:0:0:0:0:1:FFXX:XXXX, where sequential Xs are lower 24 bits of 64 bits interface identifier. The multicast address format

is shown in Fig. 3-9, and the solicited-node multicast address format is shown in Fig. 3-11. Besides, reserved multicast addresses are listed in Table 3-6.

3. The state of a link-local address generated in the second step is 'tentative'. The tentative address should go through duplication check process, which is called DAD, to verify that this address is not used by other nodes on the link. After the tentative address is determined as unique one, it can be assigned to the interface. Manual configuration should be performed if the address is already used by the other node. A pair of Neighbor Solicitation and Neighbor Advertisement messages is used for DAD. Messages exchanged between nodes to carry out DAD are shown in Fig. 6-3.
4. Once DAD is successfully performed, the node is looking for a router. This process is called router discovery, as explained in Chapter 5. If a router is present on the link, this router will notify the host of prefix and prefix option information used in the link. A pair of Router Solicitation and Router Advertisement messages is used in this stage. Messages exchanged between an IPv6 node and on-link router for router discovery are shown in Fig. 5-3.
5. If no routers are found, the node will get no response message and should attempt stateful autoconfiguration. The stateful autoconfiguration is performed by sequential message exchanges between DHCP server and DHCP clients. This mechanism is explained in Chapter 7.
6. Now, the node is able to generate its global unicast IP address by appending interface identifier to the prefix information learned from on-link router or get IP address from a DHCP server. To speed up the autoconfiguration process, both DAD and router discovery can be performed simultaneously.

The stateless autoconfiguration mechanism will be adopted when IP resources do not need to be managed strictly. It may work quite well unless the failure probability of DAD is high. When network resources are required to be managed tightly, stateful autoconfiguration mechanism can be adopted. It is because a server (or servers) controls the whole address assignment in stateful autoconfiguration. Both mechanisms can be applied for a network concurrently.

To guarantee the uniqueness of an address, a node is required to perform the DAD algorithm on each address. The DAD algorithm should be performed regardless of whether the address is obtained from the stateful or stateless address autoconfiguration.

## 6.2.2 Details in address configuration

Once a link-local address is verified for the uniqueness, global address can be formed by appending node's interface identifier to the prefix learned from Router Advertisement message. The advertised prefix length is normally 64-bit length. A router either periodically or intermittently broadcasts Router Advertisement message.

When a router sends Router Advertisement message, the destination address of the message is all-nodes multicast address.[16] If a host wants to get this advertisement message quickly, it sends Router Solicitation message with all-routers multicast address[17] in the Destination Address field. If no routers are present on the link, hosts are required to attempt to use stateful address autoconfiguration to get IP addresses as well as other necessary information for the configuration of it.

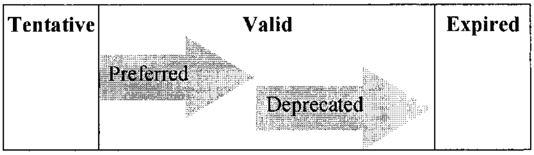


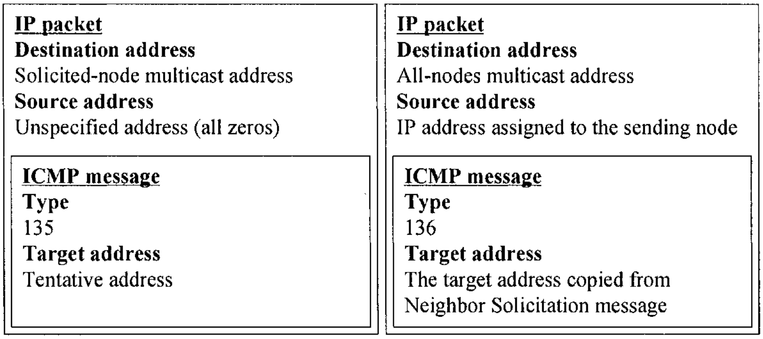
*Figure 6-2.* Address status.

*Table 6-1. M* and *O* bits in Router Advertisement message.

| *M* bit (ManagedFlag) | | *O* bit (OtherConfigFlag) | |
|---|---|---|---|
| False ⇒ True | True ⇒ False | False ⇒ True | True ⇒ Falsc |
| A host should invoke stateful address autoconfiguration. Address and other information should be learned via stateful address autoconfiguration. | No change, i.e., A host keeps running stateful address autoconfiguration. | A host should invoke stateful address autoconfiguration to request information. If *M* bit is set as false, address information is not requested. | No change. A host keeps running stateful address autoconfiguration. |

[16] FF01::1 and FF01::2

[17] FF01::2, FF02::2, and FF05::2



**IP packet**
**Destination address**
Solicited-node multicast address
**Source address**
Unspecified address (all zeros)

**ICMP message**
**Type**
135
**Target address**
Tentative address

(a) Neighbor Solicitation

**IP packet**
**Destination address**
All-nodes multicast address
**Source address**
IP address assigned to the sending node

**ICMP message**
**Type**
136
**Target address**
The target address copied from Neighbor Solicitation message

(b) Neighbor Advertisement

*Figure 6-3.* Message exchanges for DAD between nodes.

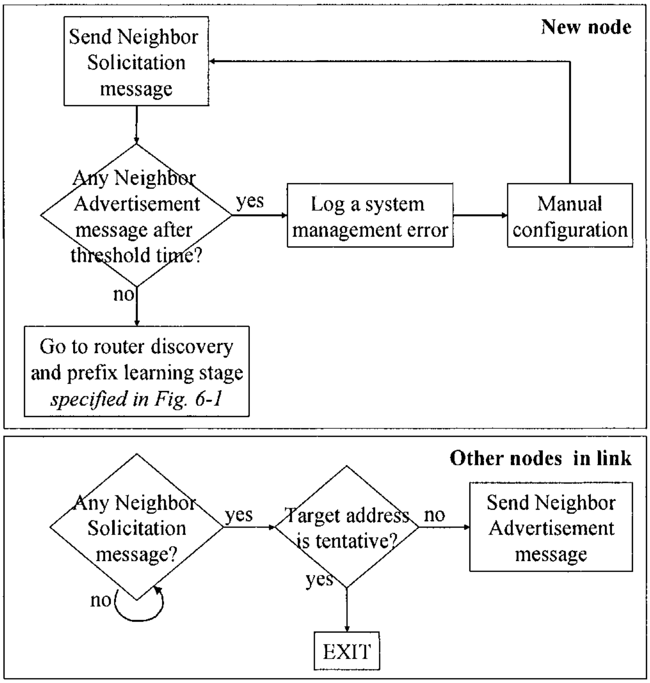

New node
Send Neighbor Solicitation message
Any Neighbor Advertisement message after threshold time?
yes
Log a system management error
Manual configuration
no
Go to router discovery and prefix learning stage *specified in Fig. 6-1*
Other nodes in link
Any Neighbor Solicitation message?
no
yes
Target address is tentative?
no
Send Neighbor Advertisement message
yes
EXIT

*Figure 6-4.* DAD algorithm.

When a host receives valid Router Advertisement message, it copies *M* bit and *O* bit value into ManagedFlag and OtherConfigFlag, respectively.

Changes in *M* or *O* bit cause the host to invoke stateful address autoconfiguration. Specific actions according to each change are displayed in Table 6-1. Once stateful address autoconfiguration is invoked, it must not be invoked again.

## 6.3 DUPLICATED ADDRESS DETECTION (DAD)

The DAD is mandatory job before assigning an address to an interface. The DAD is performed over unicast address and should be applied to any address irrespective of whether it is obtained from stateful or stateless address configuration. The DAD process is specified in Fig. 6-4. A pair of Neighbor Solicitation and Neighbor Advertisement messages is used in the DAD. If any duplication is discovered in this process, then autoconfiguration stops, and manual configuration should start.

The address status under DAD is tentative, and it becomes available on the interface once DAD is successfully performed. The address known as 'duplicated' should not be assigned to an interface, further the tentative address should not be assigned to an interface.

Neighbor Solicitation and Neighbor Advertisement messages contain tentative address for the duplication check in the Target Address field. For message formats of Neighbor Solicitation and Neighbor Advertisement, refer to Fig. 5-7 and 8. If the address in the Target Address field is used by a receiving node, the receiving node sends Neighbor Advertisement message back to the soliciting node. Once the soliciting node receives Neighbor Advertisement messages in response to Neighbor Solicitation message which it originates, it knows that the address is already occupied by the other node. Then, the node should select another address using the manual configuration or other mechanisms.

## 6.4 OPTI-DAD

Whenever an IPv6 node plugs in or it changes its attached link, DAD should be performed. Any node can not initiate communications with peers till the DAD is over. However, when the duplication rate is reasonably low, it should be a waste of time to wait for the completion of DAD process.

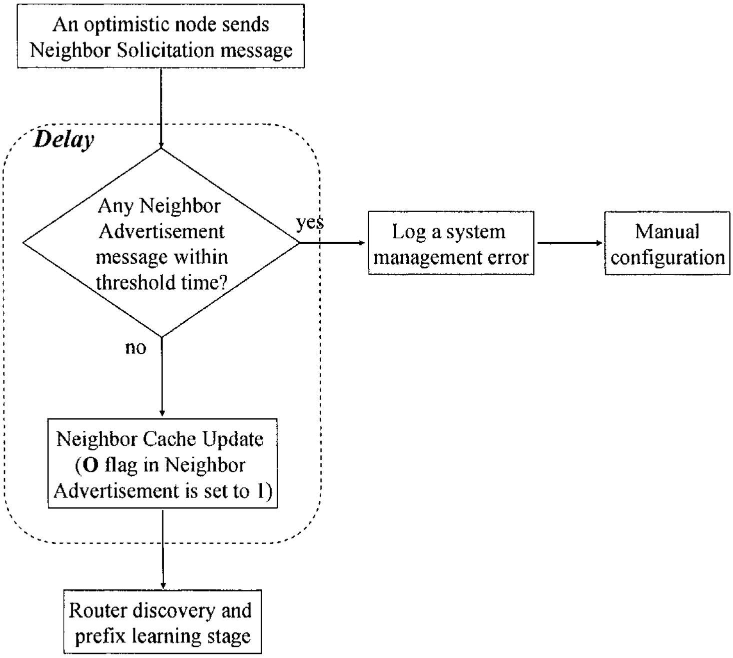

An optimistic node sends Neighbor Solicitation message
Delay
Any Neighbor Advertisement message within threshold time?
yes
Log a system management error
Manual configuration
no
Neighbor Cache Update (**O** flag in Neighbor Advertisement is set to 1)
Router discovery and prefix learning stage

*Figure 6-5.* Considerable delay in DAD.

The optimistic duplication address detection (opti-DAD) is proposed to reduce latencies caused by DAD process.[6] The opti-DAD is compatible with existing protocols such as stateless autoconfiguration and Neighbor Discovery protocols.[5] Besides, opti-DAD enabled node interoperates with other unmodified nodes. If on-link router does not understand opti-DAD, then original DAD will be run.

Currently, new address type and node type are defined for opti-DAD as follows.

- Optimistic address belongs to available address, but DAD is not fully completed yet. Once DAD for the optimistic address is done, the address status will be changed to preferred or deprecated, depending on the address lifetime. An optimistic node is able to start communications using optimistic address before DAD. When neighboring nodes do not understand optimistic address, this address type is regarded as deprecated address.
- Optimistic node is a node which understands opti-DAD and starts communications with others before DAD completion.

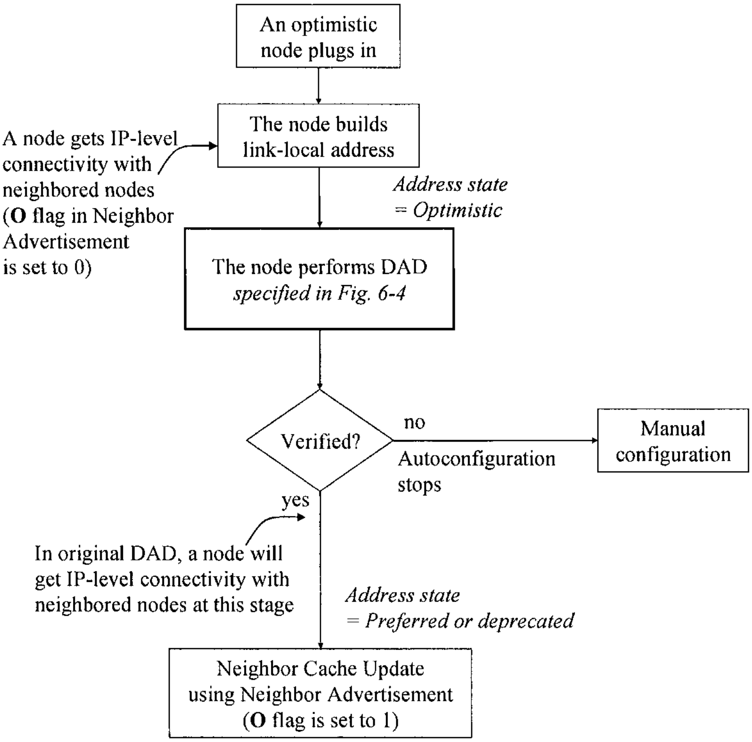

An optimistic node plugs in
The node builds link-local address
A node gets IP-level connectivity with neighbored nodes (**O** flag in Neighbor Advertisement is set to 0)
*Address state = Optimistic*
The node performs DAD *specified in Fig. 6-4*
Verified?
no
Autoconfiguration stops
Manual configuration
yes
In original DAD, a node will get IP-level connectivity with neighbored nodes at this stage
*Address state = Preferred or deprecated*
Neighbor Cache Update using Neighbor Advertisement (**O** flag is set to 1)

*Figure 6-6.* Modified autoconfiguration algorithm for opti-DAD.

## 6.4.1 Consideration of delay in DAD

Regardless of the address autoconfiguration methods, e.g. stateless or stateful, the DAD should be performed. Any address under tentative state can not be used in the Source or Destination Address field of packets until the address state is changed to available. As shown in Fig. 6-5, a node is able to communicate with others only after DAD. If a node starts communications only after link-local address formation, delay indicated with dotted line in Fig. 6-5 will be eliminated.

When the duplication occurs after an optimistic node starts communications with neighboring nodes, all sessions should be stopped, and opti-DAD should be backed off. Then, manual configuration will be tried

### 6.4.2 Modifications for opti-DAD

Modified autoconfiguration algorithm for opti-DAD is shown in Fig. 6-6. To support opti-DAD, on-link router should understand opti-DAD. When an optimistic node plugs in and builds link-local address, the node will get IP connectivity on link. The generated address is typed as optimistic. Now, the node is able to initiate any communications with neighboring nodes while it performs DAD process at the same time. If duplication occurs, every data exchange related to the optimistic node should be stopped, and the normal address configuration process should start. Message types which may be generated by the optimistic node before DAD completion are exampled in the following section.

To support opti-DAD, on-link router should understand opti-DAD and redirect traffic for the optimistic node. Besides, Router Advertisement message should contain Source Link-Layer Address option (SLLAO) to support opti-DAD. If a router on link does not support opti-DAD or no SLLAO is found in Router Advertisement messages, opti-DAD should be stopped, and original DAD process will be run. The SLLAO format is shown in Fig. 5-11.

The optimistic node learns router's link layer address from the SLLAO of Router Advertisement message. A node which does not understand the SLLAO should not configure a new address as optimistic. Later, when the optimistic node wants to send any packet to its neighboring nodes, it forwards the packet to router's link-local address. The router will perform redirection.

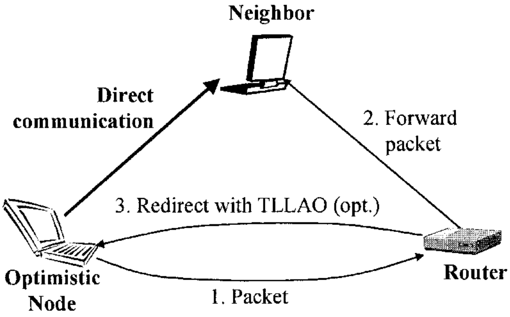


*Figure 6-7.* Example: redirection for opti-DAD.

### 6.4.3 Example

In ND protocol, five ICMP messages, such as Neighbor Solicitation, Neighbor Advertisement, Router Solicitation, Router Advertisement and Redirect, are defined. Among them, Neighbor Solicitation, Neighbor Advertisement and Router Solicitation messages may be originated by an optimistic node before the address verification. Besides, normal data packets may be transmitted by the optimistic node.

- Unsolicited Neighbor Advertisement message:
  When an optimistic node sends unsolicited Neighbor Advertisement message, the *O* flag (override flag) in the message should be set to 0 not to allow on-link neighboring nodes to update their neighbor caches for the optimistic node. The Neighbor Advertisement message format is shown in Fig. 5-8.
- Neighbor Solicitation:
  When an optimistic node generates Neighbor Solicitation message before DAD, unspecified address should be used in the Source Address field. Optimistic address should not be used. Besides, Neighbor Solicitation message should not contain SLLAO to avoid the disruption of Neighbor Cache.
- Router Solicitation:
  When an optimistic node sends Router Solicitation message with the source address set to optimistic address, the SLLAO should not be contained in the message to avoid the disruption of Neighbor Cache.
- Redirection for address resolution:
  If an optimistic node can not contact neighboring nodes because they are not in Neighbor Cache, then the node sends packets to on-link router for redirection. For redirection, destination address will be set to router's link-local address learned from SLLAO of Router Advertisement message. Packets are forwarded to the router, and the router forwards them to the proper destination node. Now, the router knows that there is the better path between the optimistic and the destination node. The router provides the optimistic node with ICMP Redirect message, which may contain Target Link-Layer Address option (TLLAO). In the TLLAO, destination node's link-layer address is contained. The Redirect message causes optimistic node's Neighbor Cache to be updated. Now, the optimistic node may initiate direct communications with the destination node. The redirection is exampled in Fig. 6-7.

## REFERENCES

1. S. Deering and R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460 (December 1998).
2. S. Thomson and T. Narten, IPv6 Stateless Address Autoconfiguration, RFC 2462 (December 1998).
3. S. Thomson, T. Narten, and T. Jinmei, IPv6 Stateless Address Autoconfiguration, work in progress (June 2004).
4. R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, Dynamic Host Configuration Protocol for IPv6 (DHCPv6), RFC 3315 (July 2003).
5. T. Narten, E. Nordmark, and W. Simpson, Neighbor Discovery for IP version 6, RFC 2461 (December 1998).
6. N. Moore, Optimistic Duplicate Address Detection for IPv6, work in progress (June 2004).

Chapter 7

# DYNAMIC HOST CONFIGURATION PROTOCOL (DHCPv6)

## 7.1 INTRODUCTION

Dynamic Host Configuration Protocol (DHCP), which has been extensively used in IPv4 network, helps hosts to configure IP addresses as well as some additional information. It may save the network management cost by minimizing the involvement of network administrator in configuration of newly introduced hosts. It is used as a mechanism to provide portability to the mobile terminal. It also helps Mobile IPv4 when there is no foreign agent. If there are no foreign agents in Mobile IPv4,[2] Care-of-Address (CoA) should be provided by DHCP servers. Thus, it is widely adopted in various applications such as wireless LAN, campus network, and offices of the global company.

DHCP servers pass network configuration parameters to hosts on the network and further support automatic reusability of addresses in networks. DHCP allows managing address resources and related information in a concentrated manner, resulting in reducing network management costs.

DHCPv6[1] is classified as a stateful configuration protocol, which is explained with stateless address autoconfiguration protocol in Chapter 6.[3, 4] Hosts do not specially need DHCP to get IP address in IPv6 networks because any host is able to get globally unique IP address via stateless address autoconfiguration. However, in some networks, strict address management or dynamic assignment feature using DHCP servers becomes more important than benefits which can be obtained if stateless configuration mechanism is employed. In such cases, network administrators may adopt

DHCPv6 in a network for the configuration of addresses as well as additional parameters.

## 7.2 TERMINOLOGY

IPv6 supports multicast addresses to any IPv6 enabled device, and the following multicast addresses are reserved for DHCP.

- All-DHCP-Relay-Agents-and-Servers address (FF02::1:2)
- All-DHCP-Servers address (FF05::1:3)

All-DHCP-Relay-Agents-and-Servers address is a link-scoped multicast address. This address type is used when a host (i.e. a DHCP client) wants to communicate with on-link neighboring relay agents or DHCP servers. All relay agents and servers are members of this multicast group. All-DHCP-Servers address is a site-scoped multicast address and used when a relay agent wants to communicate with DHCP servers. All servers in a site are members of this multicast group.

In addition to reserved multicast addresses, new message types are defined for communications between DHCP servers and clients. Thirteen message types are specified in Table 7-1, and new type may be added in the future.

In DHCPv6, three types of node are defined as follows:

- Relay agent: Relay agent is an intermediate node between DHCP client and server. Relay agent and DHCP clients should be present on the same link. When DHCP server and client are not on the same link, relay agent will relay messages between them.
- DHCP server: DHCP server provides IP addresses with network configuration parameters to DHCP clients.
- DHCP client: DHCP client is a normal node to request DHCP server to assign a new IP address.

Before DHCP solicitation, clients are required to build Identity Association (IA). Besides each DHCP node is associated with DHCP Unique Identifier (DUID).

- IA: IA is a collection of addresses assigned to one of client's interfaces. It is used by the DHCP server and the client to identify and manage IPv6 addresses.
- IA Identification (IAID): IAID is an identifier for an IA, and it is assigned by a client uniquely over all IAs.
- DUID: DUID is a DHCP unique identifier for a DHCP node. DUID is globally unique and assigned to each node.

*Table 7-1.* DHCP message types.

| Message type | Type value | Description |
|---|---|---|
| Solicit | 1 | Used when a client locates DHCP servers |
| Advertise | 2 | Used when a server notifies clients that it is available for DHCP service corresponding to a Solicit message |
| Request | 3 | Used when a client requires configuration parameters, such as IP address and other information |
| Confirm | 4 | A client sends Confirm message to any available server to check whether IP address assigned to its interface is still valid in the link where the client is attached |
| Renew | 5 | A client sends this message to the server which has provided the client with IP address and other configuration parameters. This message is used when a client wants to extend the lifetime of its assigned IP address and to update other configuration parameters. |
| Rebind | 6 | A client sends Rebind message to any available server to extend the lifetime of its assigned IP address and to update other configuration parameters. A Rebind message is sent unless a client gets any Reply to Renew message. |
| Reply | 7 | A server sends Reply message in response to Solicit, Request, Renew and Rebind messages from clients. |
| Release | 8 | A client sends Release message to the server which has provided IP address to notify that the client wants to release one or more assigned IP addresses. |
| Decline | 9 | A client sends Decline message to a server to notify that the assigned address from the server is used by the other node in the link where the client is attached. |
| Reconfigure | 10 | Used when a server notifies clients of changes in configuration parameters. Clients in the link initiate Renew/Reply or Information-Request/Reply transaction with the server to update stale information. |
| Information-Request | 11 | Used when a client requests configuration parameters without IP address. |
| Relay-Forward | 12 | A relay agent sends Relay-Forward message to relay messages from clients to servers. |
| Relay-Reply | 13 | A server sends Relay-Forward message containing a message that is delivered to a client. |

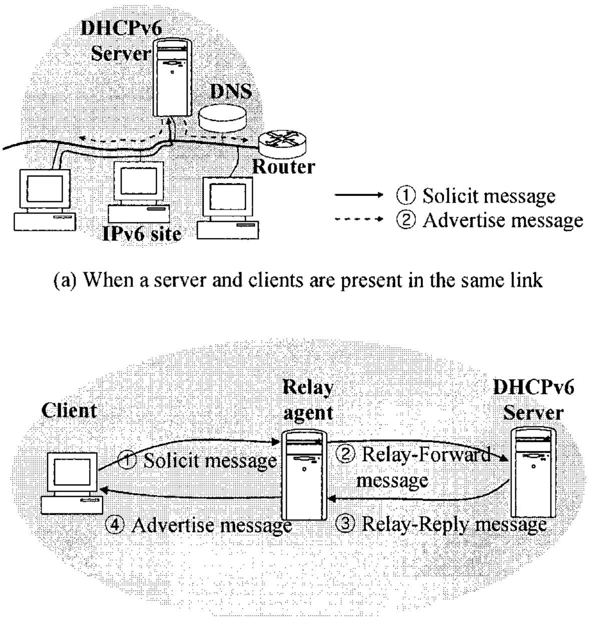
## 7.3 DHCP SERVER SOLICITATION

Address assignment from DHCP server is performed through two steps: DHDP server solicitation and configuration exchanges. DHCP server solicitation is for a client to search on-link DHCP servers. Once the client locates DHCP server (or servers), it requests the server to assign address using configuration exchange which is explained in the following section.

To locate a DHCP server on-link, a client will form and send Solicit message, where IA is contained in IA option. As explained in the previous section, IA is a collection of addresses assigned to one of client's interfaces. Each IA is composed with IAID and related configuration information. IA is used by DHCP server and the client to identify and manage IPv6 addresses. The Solicit message is destined to All-DHCP-Relay-Agents-and-Servers multicast address.

Once a server receives valid Solicit message agreed to administrative policy, the server sends back an Advertise message to notify soliciting clients of its availability. If the server is not allowed to answer to soliciting clients, the server should silently discard the Solicit message. In Advertise message from the server, IA copied from the Solicit message should be contained. After all, the client is able to initiate a configuration exchange explained in the following section.

The process is shown in Fig. 7-1(a). When a relay agent is present between a server and clients, server solicitation is processed via a relay agent as shown in Fig. 7-1 (b).



(a) When a server and clients are present in the same link

(b) When a server is not directly connected to clients in the link

*Figure 7-1.* DHCP server solicitation.

Upon a client receives several valid Advertise messages, the client may choose multiple Advertise messages based on the highest server preference value. Among messages having the same server preference value, the client may select a message containing interesting information.

After the client selects a server, the client stores all information from Advertisement message, such as server preference value, and advertised addresses. Later, if the client needs to switch the server to alternative one, the client will choose it by referring the preference value.

## 7.4 DHCP CLIENT-INITIATED CONFIGURATION EXCHANGE

Once a client locates a server or servers as specified in the previous section, it initiates a message exchange with a server or servers to obtain or update configuration information. The client starts the configuration exchange as a part of configuration process of the operating system. The application layer triggers the client-initiated configuration exchange to request stateful address autoconfiguration or to extend the lifetime of assigned address using Request messages. As a response to a Request message, the server will send a Reply message destined to the soliciting node's unicast address, as shown in Fig. 7-2.

During the common life cycle of an address, Request, Renew, Rebind, Release, Decline, and Reply messages are exchanged between DHCP clients and the server. When a client moves to a new link, Confirm and Reply messages are exchanged to validate assigned addresses. When other configuration information except addresses is needed, Information-Request and Reply messages are exchanged. Whenever a server sends a Reply message, it must be generated with a unicast address targeted to the soliciting node.

Upon a client receives a valid Reply message in response to Solicit, Request, Confirm, Renew, Rebind or Information-Request message, the client extracts configuration information contained in the Reply message and reports Status Code or information contained in the Status Code option to the upper application layer. There are six types of message exchange as follows.

- Request and Reply message exchange: DHCP client and server exchange Request and Reply messages for the address assignment after DHCP server discovery.
- Confirm and Reply message exchange: DHCP client and server exchange Confirm and Reply messages to verify assigned addresses.

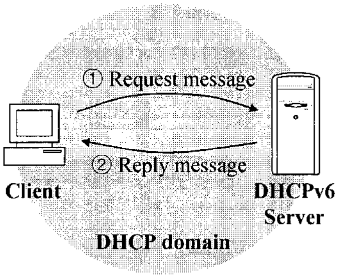


*Figure 7-2.* DHCP client-initiated configuration exchange (Request-Reply message exchange).

- Renew and Reply message exchange: DHCP client and server exchange Renew and Reply messages to extend the lifetime of assigned addresses.
- Rebind and Reply message exchange: DHCP client sends Rebind message when it does not receive Reply message for Renew message.
- Release and Reply message exchange: DHCP client and server exchange Release and Reply messages to release addresses assigned by the server.
- Decline and Reply message exchange: DHCP client sends Decline message to the server to notify that the assigned address is already taken by other nodes on the link.

Detailed description of each message exchange is given in the following subsections.

### 7.4.1 Request and Reply message exchange

DHCP server and client exchange Request and Reply messages for the address assignment. Client initiated configuration message exchange is shown in Fig. 7-2.

1. A client sends Request message to obtain address for IA and other configuration information. The requesting client will include one or more IA options in Request message.
2. When a server receives a valid Request message, it first creates a binding for the requesting client according to the administrative policy and configuration information.
3. Then, the server records the IAs and other information requested by the client and returns addresses with other information to the client in IA options in Reply message.
4. Reply message is generated as follows:
   The Message Type field is set to 7.
   The Transaction ID of Reply message is copied from the corresponding Request message.

Besides, Server Identifier option containing server's DUID and the Client Identifier option copied from Request message must be included in Reply message.

### 7.4.2 Confirm and Reply message exchange

There are various occasions that clients need to check whether the assigned address is still valid. Clients may move to new locations even if they are non-mobile devices. It they are mobile devices, movements between subnetworks will be more frequent. In these cases, prefixes learned from the old link may not be appropriate any more at the new link. If DHCP client is a mobile node with wireless interfaces, address re-verification may be required when the node moves between access points. When clients reboot or return from the sleep mode, validity checking of addresses is also needed. In such situations, the client must initiate Confirm and Reply message exchange.

1. In Confirm message, the client includes IA assigned to its interface and address information associated with IA.
2. When a server receives Confirm message, the server checks whether the addresses in Confirm message are appropriate for the link where the client is attached.
3. If all addresses in Confirm message are verified as appropriate for the link where the client is attached, the server returns Reply message with the address status as 'SUCCESS' to the client.
4. Unless any address passes the test, the server returns the address status as 'NOT-ON-LINK' to the client.
5. Reply message is generated as follows:
   The Message Type field is set to 7.
   The Transaction ID of Reply message is copied from the corresponding Confirm message.
   Server Identifier option containing server's DUID and the Client Identifier option copied from Confirm message must be included in Reply message.
   In addition, the server includes Status Code option to indicate the status of Confirm message in Reply message.

### 7.4.3 Renew and Reply message exchange

To extend the lifetime for address (or addresses) associated with an IA, a client sends Renew message to the server that assigned IP address to the client.

1. The Renew message contains an IA and associated IA option for assigned addresses.
2. Upon the server receives Renew message that contains an IA option from the client, it searches client's binding and certifies that information in IA of Renew message is identical to information stored for that client.
3. If the server can not find an appropriate client entry for the IA, it returns Reply message with IA, which does not contain address, and Status Code option set to 'NO-BINDING'.
4. If the server detects that any one of addresses is not valid any longer, it will set the lifetime of the invalid address to 0 in Reply message.
5. If the server finds that the extending of address lifetime for the client is adequate, then it sends back the IA with new lifetimes to the client.
6. Reply message is generated as follows:
   The Message Type field is set to 7.
   The Transaction ID of Reply message is copied from the corresponding Renew message.
   The Server Identifier option containing server's DUID and the Client Identifier option copied from Renew message must be included in Reply message.

### 7.4.4 Rebind and Reply message exchange

When a client does not get Reply message corresponding to Renew message, it sends Rebind message to any available server to extend the lifetime of its assigned IP address and to update other configuration parameters.

1. Once a server receives Rebind message containing an IA option from a client, it checks client's binding and certifies that information in the IA is identical to information stored for that client.
2. If the server detects any one of addresses is not appropriate any longer, it will set the lifetime of the invalid address to 0 in Reply message.
3. Otherwise, if the server finds addresses in the IA for the client are adequate, then it should return the IA to the client with new lifetimes.
4. Reply message is generated as follows:
   The Message Type field is set to 7.
   The Transaction ID of Reply message is copied from the corresponding Rebind message.
   The Server Identifier option containing server's DUID and the Client Identifier option copied from Rebind message must be included in Reply message.

To obtain configuration information without the address assignment, the client sends Information-Request message to the server.

1. Once the server receives Information-Request message, it locates appropriate configuration parameters for the client, based on the server administration policy.
2. Reply message is generated as follows:
   The Message Type field is set to 7.
   The Transaction ID of Reply message is copied from the corresponding Information-Request message.
   Server Identifier option containing server's DUID and the Client Identifier option copied from Information-Request message must be included in Reply message.

## 7.4.5 Release and Reply message exchange

A client sends Release message to DHCP server to notify that it wants to release one or more assigned IP addresses from the server. Upon the server receives a valid Release message, it verifies IAs and the addresses associated with them.

1. If IAs in Release message are bound to the client, and the address are associated with IAs and assigned by the server, then the server deletes the address from IAs and makes the address available for other clients.
2. After all the addresses are processed, Reply message is generated as follows:
   The Message Type field is set to 7.
   The Status Code option is set to 'SUCCESS'.
   The Server Identifier option containing server's DUID and the Client Identifier option copied from Release message must be included in Reply message.
3. If any IA from Release message is not present in the binding information of the server, the server adds an IA option using the IAID from Release message and sets Status Code option as 'NO-BINDING'.
   No other options are attached in the IA option.
   The server may determine to hold a record for the address and IAs whose lifetimes are expired. The server assigns it to the same client when the client requests an address later.

## 7.4.6 Decline and Reply message exchange

A client sends Decline message to a server to notify that the assigned address from the server is used by other nodes on the link where the client is attached.

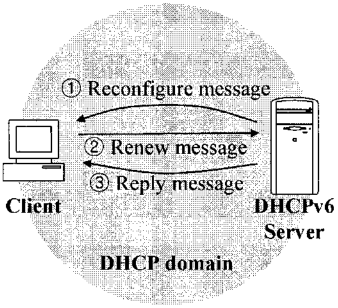


*Figure 7-3.* DHCP Server-initiated configuration exchange (Request-Reply message exchange).

1. Once the server receives valid Decline message, it verifies IAs and the addresses associated with IAs.
2. If the server finds IAs in Decline message from its binding information, the addresses have been assigned by the server to IAs, and the server deletes the addresses from IAs.
3. The server marks addresses from Decline message not to be assigned to other clients.
4. After all the addresses are processed, Reply message is generated as follows:
   The Message Type field is set to 7.
   The Status Code option is set to 'SUCCESS'.
   The Server Identifier option containing server's DUID and the Client Identifier option copied from Decline message must be included in Reply message.
5. If any IA from Decline message is not present in binding information of the server, the server adds an IA option using the IAID from Decline message and set Status Code option as 'NO-BINDING'. No other options will be attached in the IA option.

## 7.5 DHCP SERVER-INITIATED CONFIGURATION EXCHANGE

When a server wants to notify clients of some changes of configuration parameters or other information about links, DHCP server initiates configuration exchange. For example, when some links of DHCP domain are to be renumbered with some changes in the location of directory servers, the server initiates configuration exchange to make clients to get new configuration information. Fig. 7-3 shows server-initiated configuration exchange.

The server sends Reconfigure message to a client to initiate Renew and Reply or Information-Request and Reply message exchanges as explained below. Every message in the server-initiated configuration exchange should be transmitted with unicast address and may be sent at any time.

Once a client receives Reconfigure messages from a DHCP server which has assigned the address and offered configuration information, it logs these reconfiguration events and sometimes notifies application layer programs of the changes because the results of the reconfiguration may affect applications.

To send Reconfigure message, the server builds messages as follows:

- The Message Type field is set to Reconfigure.
- The Transaction-id field is set to 0.
- The Server Identifier option containing its DUID and the Client Identifier option containing client's DUID are also included in the Reconfigure message.
- The server may include an Option Request option to notify the client of updated or newly added information.
- The server must include Reconfigure Message option for the receiving client to select response message types; Renew or Information-Request message.

The server unicasts a separate Reconfigure message to each client, even if the server invokes multiple clients to initiate reconfiguration because Reconfigure messages should be generated with unicast addresses.

Once a client receives a valid Reconfigure message, the client responds with Renew or an Information-Request message as indicated by the Reconfigure Message option.

## 7.5.1 Renew and Reply message exchange

A client may respond to Reconfigure message with Renew message same as in the client-initiated configuration. However, at this time, the client includes the Option Request option and IA options copied from the Reconfigure message.

Once a server receives valid Renew message, it sends Reply message back to the client same as in client-initiated configuration. The Reply message may include options containing IAs and new values for other configuration parameters even if the client does not request.

### 7.5.2 Information-Request and Reply message exchange

The client may respond to Reconfigure message with Information-Request message same as in the client-initiated configuration. However, at this time, the client includes a Server Identifier option with the Identifier copied from the Reconfigure message.

Once a server receives valid Renew message, it sends Reply message to the client same as in the client-initiated configuration and may include options containing new values for other configuration parameters even if the client does not request.

As the client receives a valid Reply message from a DHCP server, the client processes options and sets configuration parameters. The client records and updates the lifetimes of addresses associated with IAs in Reply message.

## 7.6 RELAY AGENTS

A relay agent may have a list of destination server addresses and be configured to use it. Unless the relay agent is explicitly configured to use it, the All-DHCP-Servers multicast address (FF05::1:3) must be used as a default.

A relay agent relays both messages from clients and Relay-Forward messages from other relay agents to DHCP servers. When a relay agent receives a valid message from a client, the relay agent constructs a new Relay-Forward message as follows:

1. The Source address is copied from the header of IP datagram.
2. The Relay Message option is also copied from the received DHCP message.
3. The relay agent adds other options to the Relay-Forward message. This relaying process is shown in Fig. 7-1 (b).
4. When a server receives relayed messages from a client via a relay agent, the server uses a Relay-Reply message in returning a response to the client. When a server does not know an exact address of a client but needs to send Reconfigure message to the client, the server may also use a Relay-Reply message. A response to the client should be relayed through the same relay agent that relayed original DHCP message triggering this response message.

## 7.7 DHCP UNIQUE IDENTIFIER (DUID)

Each client and server has a DHCP Unique Identifier (DUID). DUID is globally unique. DHCP servers use DUID to distinguish between clients to select proper configuration parameters for each client and to associate IA with clients. Clients also use DUID to identify servers.

DUID is built with Type and Length fields, where the length of Type field and Length field is two bytes and variable, respectively. Maximum length of DUID can be only 128 bytes except Type field. At present, three types are defined for DUID, as shown in Table 7-2.

DUID is carried in the Option field of DHCP message because it has variable length. However, it is not required for every DHCP message to contain DUID. DUID must be treated as opaque value. Additional DUID types may be defined in the future.

### 7.7.1 DUID-LLT

When Type value is set to 1, DUID is built based on the link-layer address with time, and it is called DUID based on link-layer address with time (DUID-LLT), as shown in Fig. 7-4.

- The Type field is set to 1 to indicate DUID-LLT.
- The Hardware Type field should contain a valid hardware type value which is assigned by IANA as described in RFC 826.[4]
- The Time field specifies the time when DUID is generated and it is expressed in seconds since the midnight of universal time coordinated January 1, 2000 modulo 232.
- The variable Link-Layer Address field contains link-layer address generated from the client's interface identifier when DUID is formed.

When clients or servers use DUID-LLT, it should be stored in the stable storage. Furthermore, it is recommended to use it continuously even if the interface used to generate DUID is removed from the DHCP device. Collisions may happen in building DUID. DHCP client must have administrative policy to replace the duplicated DUID with new one.

*Table 7-2.* DUID types.

| Type value | Description |
|---|---|
| 1 | Built based on link-layer address with time |
| 2 | Built based on unique ID assigned by vendors |
| 3 | Built based on link-layer address |

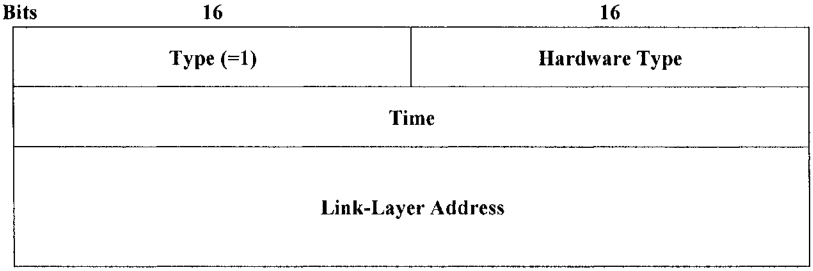


*Figure 7-4.* DUID based on link-layer address with time.

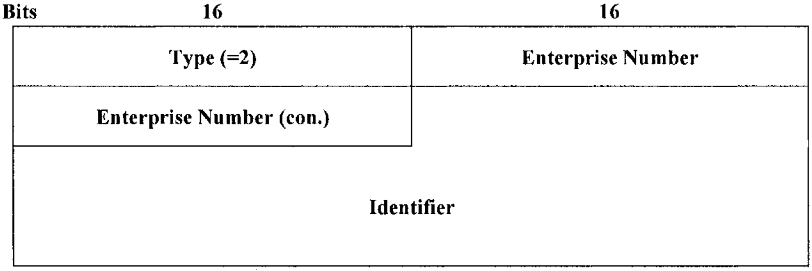


*Figure 7-5.* DUID based on enterprise number with identifier.

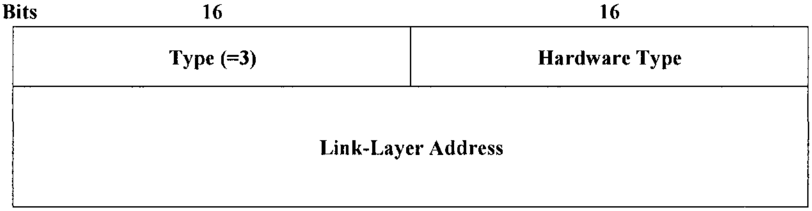


*Figure 7-6.* DUID based on link-layer address.

## 7.7.2 DUID-EN

When the Type value is 2, DUID is built based on the unique identifier assigned by vendors, and it is called as DUID based on unique ID assigned by vendors (DUID-EN), as shown in Fig. 7-5.

- The Type field is set to 2 to indicate DUID-EN.
- Vendors register private enterprise numbers to IANA, and IANA maintains this numbers.
- The Identifier field contains a unique identifier assigned by the vendor. The method to build identifiers is independently defined by each vendor. Each identifier should be given to the device uniquely at the time of manufacturing and stored in non-volatile or non-erasable storage.

### 7.7.3 DUID-LL

When Type value is set to 3, DUID is built based on the link-layer address, and it is called as DUID based on link-layer address (DUID-LL), as shown in Fig. 7-6.

- The Type field is set to 3 to indicate DUID-LL.
- The Hardware Type field should contain a valid hardware type value which is assigned by IANA as described in RFC 826.
- The Link-Layer Address field should contain link-layer address generated from permanently connected interface of DHCP device.

When DHCO devices does not have permanent network interface, DUID-LL should not be used.

## 7.8 IDENTITY ASSOCIATION (IA)

Identity Association (IA) is a collection of addresses assigned to a client. IA is used for a server and a client to identify and manage addresses. Each IA is composed of IA Identification (IAID) and related configuration information. If the client has more than one interface, it will be assigned several IAs. IA is classified by IAID, which is labeled by the client uniquely over all IAs belonging to the client.

Every interface of the client is associated with one distinct IA. This association between the interface and IA should be exclusive, thus assigned IA can not be used concurrently. To get configuration information from DHCP server, the client uses specific IA assigned to its interface.

IAID is required to be consistent by storing IAID information at non-volatile or using some algorithm to help keeping consistency as long as the client configuration is not changed. Addresses in IA have preferred lifetime and valid lifetime learned from DHCP server by IA option.

## 7.9 MANAGEMENT OF TEMPORARY ADDRESSES

A client may request a server to assign a temporary address, which is a global-scope address with random interface identifier for a short period of time and would be deprecated later. The temporary address assignment by the server will be handled same as other IPv6 address assignment except that DHCP server does not notify clients of detailed address information, such as the lifetime, a method to use temporary addresses. Once clients request temporary addresses, DHCP server will assign them via the Identity Association for Temporary Addresses (IA_TA) option.

## 7.10 MESSAGE FORMATS

### 7.10.1 Message formats for client and server

DHCP messages exchanged between servers and clients have the same fixed header and variable option field formats, as shown in Fig. 7-7.

- The Type field identifies DHCP message type. Available Type value is specified in Table 7-1.
- Transaction-ID is used in a message exchange.

### 7.10.2 Message formats for relay agent and server

Relay agents exchange messages with DHCP servers to relay messages from clients or other relay agents and help clients that are not directly connected to servers in the same link. Two kinds of message type are defined: Relay-Forward and Relay-Reply messages. The message format is shown in Fig. 7-8.

- The Hop-Count field contains the number of relay agents that have relayed this message. It is increased by 1, whenever the message is forwarded to a relay agent on its way.
- In Link-Address field, a server's global address is used to identify the link on which a client is attached.
- The address of a client or a relay agent is contained in the Peer-Address field to specify an intermediate relay agent or a source client. Thus, this field will be changed at every relay.

Once DHCP server receives Relay-Forward message from a relay agent, it sends Relay-Reply message back to the relay agent which relayed client's DHCP message.

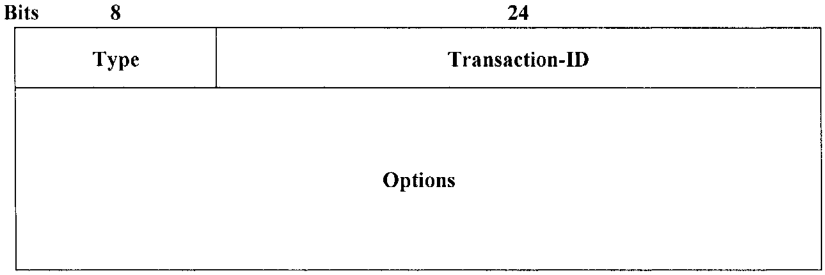


*Figure 7-7.* DHCP message format from Type 1 to 11.

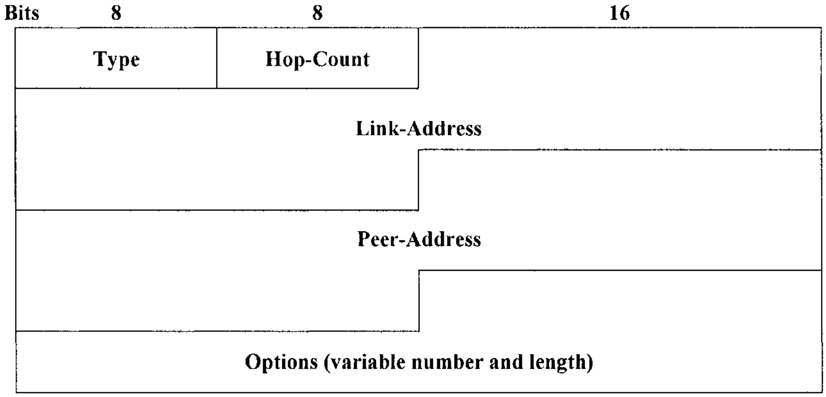


*Figure 7-8.* DHCP message format from type 12 to 13.

- The hop Count, Link-Address and Peer-Address fields of Relay-Reply message are copied from Relay-Forward message.

## 7.11 OPTIONS

Options are used to carry additional information and parameters in DHCP messages. Every option shares a common base format, as shown in Fig. 7-9. Currently, 19 options are defined, as specified in Table 7-3, and new options may be defined in the future.

The general format of DHCP options is shown in Fig. 7-9.

- The Option Code field specifies option type, and code value is described in Table 7-3.
- The Option Length field contains the total length of the Option Data field in bytes.
- Data for each option is included in the Option Data field. The format of this data depends on the option type.

### 7.11.1 Client Identifier and Server Identifier options

The Client Identifier option and the Server Identifier option are used to carry a DUID identifying a client from other DHCP servers. The Client Identifier option format is depicted in Fig. 7-10.

- When the Type field is set to 1, it indicates Client Identifier option
- When the Type field is set to 2, it indicates Server Identifier option.
- In Option Length field, the length of DUID is specified in bytes.
- If the Option Code is set to CLIENTID, DUID for a client is contained in Option Data field.
- Otherwise, if the Option Code is set to SERVERID, DUID for a server is contained in the Option Data field.

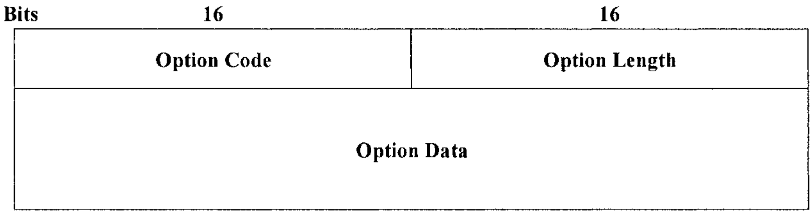


*Figure 7-9.* General option format.

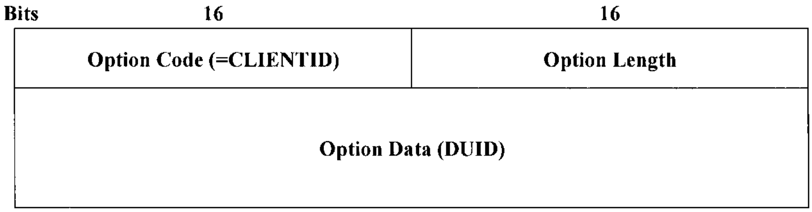


*Figure 7-10.* Client Identifier option format.

### 7.11.2 IA_NA option

The Identity Association for Non-temporary Addresses (IA_NA) option is used to carry an IA_NA with parameters. Non-temporary addresses associated with the IA_NA are also contained in this option. All of the addresses in the IA_NA option are used by the client as non-temporary addresses. The option format is described in Fig. 7-11.

- When the Type field is set to 3, it indicates IA_NA option.
- The Option Length contains the length of IA_NA options field plus 12.
- The IAID is the unique identifier for IA_NA, and it should be unique among the identifiers for all client's IA_NAs.
- The T1 field of IA_NA option specifies the time when the client contacts with DHCP server again which assigned addresses to IA_NA of the client to extend the address lifetime.
- The T2 field specifies the time when the client contacts any available server to extend the lifetime of the addresses assigned to IA_NA.
- In the IA Non-temporary Address Options (IA_NA_Options) field, IA Address option should be included. The IA Address option specifies IPv6 addresses associated with IA_NA option explained above.
- The IA Address option format is shown in Fig. 7-13.
  When the Type field is set to 5, it indicates IA Address option.
  The Option Length for IA address option is the length of IA address options field plus 24.
  In the Preferred-Lifetime field, preferred lifetime for the address contained in the option is specified in seconds.
  In the Valid-Lifetime field, valid lifetime for the address contained in the option is specified in seconds.

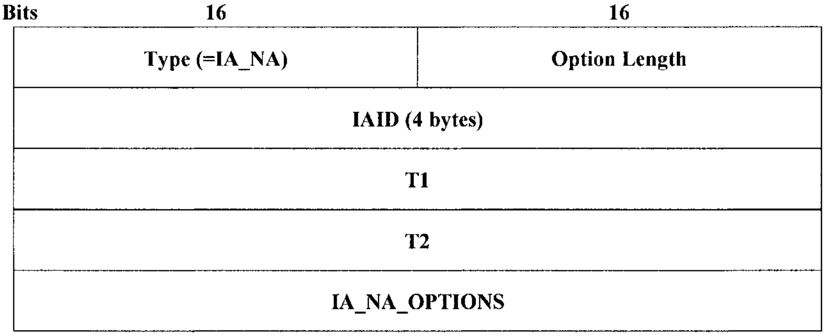


*Figure 7-11.* Identity Association for Non-temporary Addresses option format.

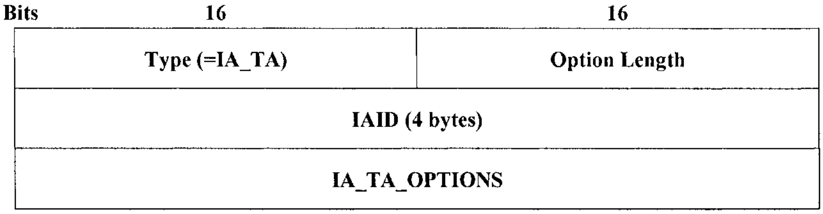


*Figure 7-12.* Identity Association for Temporary Addresses option format.

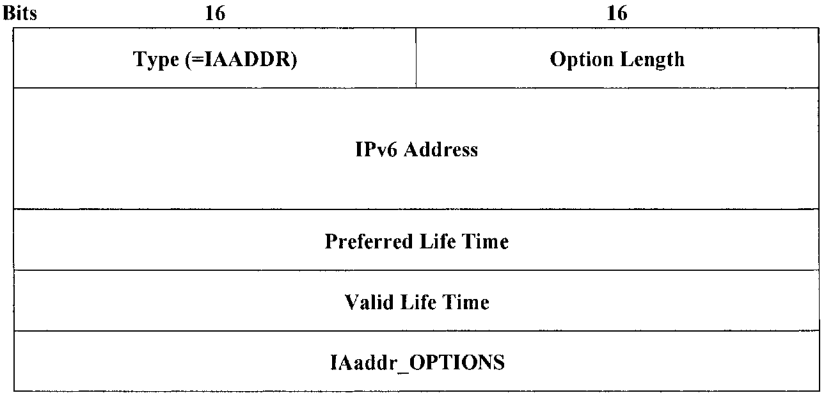


*Figure 7-13.* IA Address option format.

## 7.11.3 IA_TA option

The Identity Association for the Temporary Addresses (IA_TA) option is used to carry an IA_TA with parameters. Temporary addresses associated with the IA_TA are also contained in this option. All of the addresses in the IA_TA option are used by the client as temporary addresses. The option format is described in Fig. 7-12.

- When the Type field is set to 4, it indicates IA_TA option.
- The Option Length field contains the length of IA_TA options field plus 4.
- The IAID is the unique identifier for IA_TA. It should be unique among the identifiers for all of the client's IA_TAs.

- In the IA_TA_Options (IA Temporary Address Options) field, IA Address option should be included, as similar to IA_TA option.

### 7.11.4 Option Request option

The Option Request option is used to identify a list of options in a message between a client and a server. The Option Request option format is shown in Fig. 7-14.

- When the Type field is set to 6, it indicates Option Request option.
- The Option Length contains a value which is double the number of request options.
- In the Requested Option Code field, option code requested by a client is contained.

### 7.11.5 Preference option

DHCP servers use the Preference option to help clients to select specific servers. The option format is shown in Fig. 7-15.

- When the Type field is set to 7, it indicates Preference option.
- The Option Length is fixed as 1.
- The preference value for a server is specified in the Preference Value field.

### 7.11.6 Elapsed Time option

The Elapsed Time option is used to indicate how long a client has been trying to complete a DHCP message exchange. The option format is shown in Fig. 7-16.

- When the Type field is set to 8, it indicates Elapsed Time option.
- The Elapsed Time is measured from the time when the client sent the first message in the message exchange.
- The Option Length is fixed as 2.

This Elapsed Time option should be included in a message to indicate how long a client has been waiting for the completion of the DHCP message exchange.

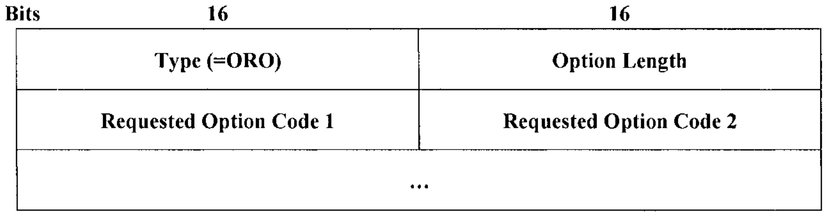


*Figure 7-14.* Option Request option format.

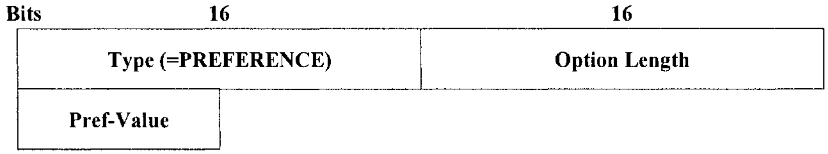


*Figure 7-15.* Preference option format.

*Table 7-3.* DHCP options.

| Option code | Name | Description |
|---|---|---|
| 1 | OPTION_CLIENTID | Client Identifier option |
| 2 | OPTION_SERVERID | Server Identifier option |
| 3 | OPTION_IA_NA | Identity Association for Non-Temporary Address option |
| 4 | OPTION_IA_TA | Identity Association for Temporary Address option |
| 5 | OPTION_IAADDR | IA address option |
| 6 | OPTION_ORO | Option request option |
| 7 | OPTION_PREFERENCE | Preference option |
| 8 | OPTION_ELAPSED_TIME | Elapsed time option |
| 9 | OPTION_RELAY_MSG | Relay message option |
| 11 | OPTION_AUTH | Authentication option |
| 12 | OPTION_UNICAST | Server unicast option |
| 13 | OPTION_STATUS_CODE | Status code option |
| 14 | OPTION_RAPID_COMMIT | Rapid commit option |
| 15 | OPTION_USER_CLASS | User class option |
| 16 | OPTION_VENDOR_CLASS | Vendor class option |
| 17 | OPTION_VENDOR_OPTS | Vendor-specific information option |
| 18 | OPTION_INTERFACE_ID | Interface-Id option |
| 19 | OPTION_RECONF_MSG | Reconfigure message option |
| 20 | OPTION_RECONF_ACCEPT | Reconfigure accept option |

| Bits | 16 | 16 |
|---|---|---|
| | Type (=ELAPSED_TIME) | Option Length |
| | Elapsed Time | |

*Figure 7-16.* Elapsed Time option format.

| Bits | 16 | 16 |
|---|---|---|
| | Type (=RELAY_MSG) | Option Length |
| | DHCP-Relay-Message | |

*Figure 7-17.* Relay Message option format.

## 7.11.7 Relay Message option

The Relay Message option is used in a Relay-Forward or Relay-Reply message to carry a DHCP message. The Relay Message option format is shown in Fig. 7-17.

- When the Type field is set to 9, it indicates Relay Message option.
- The Option Length specifies the length of DHCP relay message.
- In the DHCP Relay Message field of Relay message, the original DHCP message and other condensed information are contained.

## 7.11.8 Authentication option

The Authentication option is used to authenticate the identity and contents of DHCP message and supports multiple authentication protocols. Any DHCP message should not attach more than one Authentication option. The Authentication option format is shown in Fig. 7-18.

- When the Type field is set to 11, it indicates Authentication option.
- The Option Length contains the length of authentication information field plus 11.

- The Protocol field specifies authentication protocol employed in the Authentication option to generate Authentication Information.
- The Algorithm field states a specific algorithm used in the authentication protocol.
- The Replay Detection Method (RDM) field identifies a specific type of replay detection in the Authentication option. The replay detection information for the RDM is described in the Replay Detection field.
- The last field, Authentication Information field is used to define authentication information.

### 7.11.9 Server Unicast option

The Server Unicast option is used to indicate that clients are allowed to send unicast messages to a DHCP server. The Server Unicast option format is shown in Fig. 7-19.

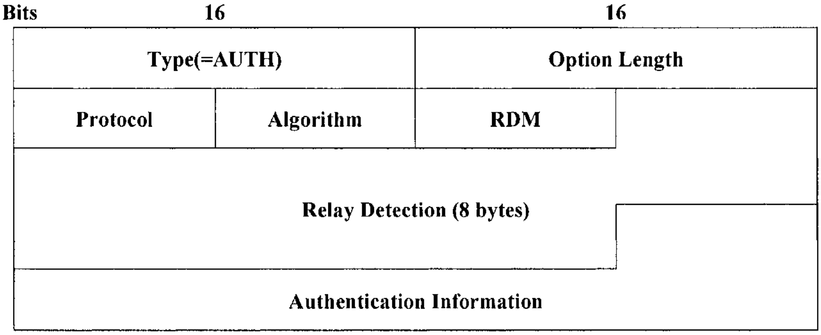


*Figure 7-18.* Authentication option format.

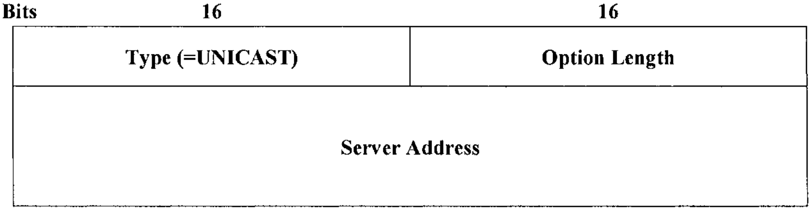


*Figure 7-19.* Server Unicast option format.

| Bits | 16 | 16 |
|---|---|---|
| | Type (=STATUS_CODE) | Option Length |
| | Status Code | |
| | Status Message | |

*Figure 7-20.* Status option format.

- When the Type field is set to 12, it indicates Server Unicast option.
- The Option Length field is set to16.
- In the Server Address field, server's unicast IP address is specified.

### 7.11.10 Status option

The Status option returns a status indication related to the DHCP message. This option is used in the Option field of DHCP message or in the Option field of another option. The Status option format is shown in Fig. 7-20.

- When the Type field is set to 13, it indicates Status option.
- The Option Length contains the length of status message plus 2.
- A specific numeric code for status is used in the Status Code field.

### 7.11.11 Rapid Commit option

The Rapid Commit option signals the use of two message exchanges for the address assignment. The Rapid Commit option format is shown in Fig. 7-21.

- When the Type field is set to 14, it indicates Rapid Commit option.
- The Option Length field is set to 0.
- When a client performs Solicit-Reply message exchange, the client may include this option in the Solicit message.

### 7.11.12 User Class option

The User Class option is used to identify the type or category of user or applications it represents. The User Class option format is shown in Fig. 7-22.

- When the Type field is set to 15, it indicates User Class option.
- The Option Length specifies the length of the User Class Data field.

- The User Class Data field is used to specify user class in which the client joins.

### 7.11.13 Vendor Class option

The Vendor Class option is used to identify the vendor which manufactured the hardware on which the client is running. Only DHCP clients use this option. The Vendor Class option format is shown in Fig. 7-23.

- If the Type field is set to 16, it indicates Vendor Class option.
- The Option Length contains the length of Vendor Class Data field plus 4.

| Bits | 16 | 16 |
|---|---|---|
| | **Type (=RAPID_COMMIT)** | **0** |

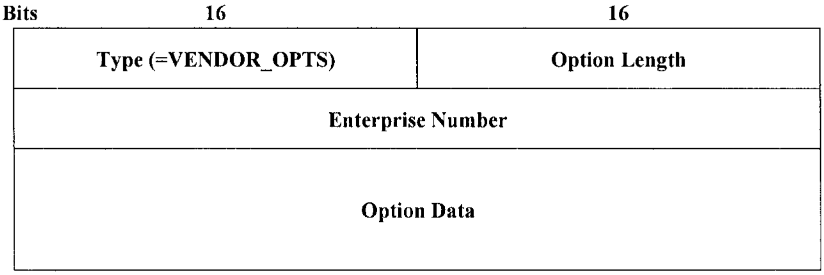
*Figure 7-21.* Rapid Commit option format.

| Bits | 16 | 16 |
|---|---|---|
| | **Type (=USER_CLASS)** | **Option Length** |
| | **User Class Data** | |

*Figure 7-22.* User Class option format.

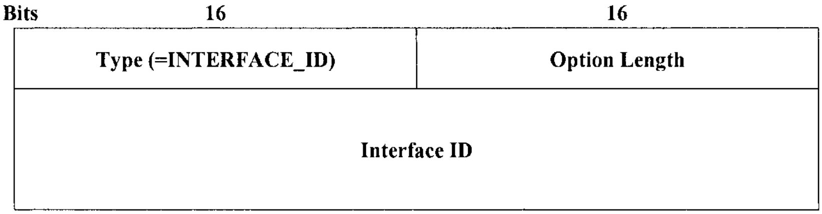
| Bits | 16 | 16 |
|---|---|---|
| | **Type (=VENDOR_CLASS)** | **Option Length** |
| | **Enterprise Number** | |
| | **Vendor Class Data** | |

*Figure 7-23.* Vendor Class option format.



| Bits 16 | 16 |
|---|---|
| Type (=VENDOR_OPTS) | Option Length |
| Enterprise Number | |
| Option Data | |

*Figure 7-24.* Vendor-Specific Information option format.



| Bits 16 | 16 |
|---|---|
| Type (=INTERFACE_ID) | Option Length |
| Interface ID | |

*Figure 7-25.* Interface-ID option format.

### 7.11.14 Vendor-Specific Information option

The Vendor-Specific Information option is used to exchange vendor-specific information, and it is used by clients and servers. The Vendor-Specific Information option format is shown in Fig. 7-24.

- If the Type field is set to 17, it indicates Vendor-Specific Information option.
- The Option Length contains the length of Option Data field plus 4.
- In the Enterprise-Number field, the enterprise number identifying specific vendor is contained.

### 7.11.15 Interface-ID option

The Interface-ID option is used by a relay agent to identify the interface on which messages from a client were received. The Interface-ID option format is shown in Fig. 7-25.

- If the Type field is set to 18, it indicates Interface-ID option.

| Bits | 16 | 16 |
|---|---|---|
| | Type (=RECONF_MSG) | Option Length |
| | Msg-Type | |

*Figure 7-26.* Reconfigure Message option format.

| Bits | 16 | 16 |
|---|---|---|
| | Type (=RECONF_ACCEPT) | 0 |

*Figure 7-27.* Reconfigure-Accept option format.

- The Option Length field specifies the length of the interface-ID field.
- In the Interface-ID, an opaque value with arbitrary length is generated by the relay agent to identify interfaces belonging to the relay agent.

## 7.11.16 Reconfigure Message option

The Reconfigure Message option is included in a Reconfigure message to indicate whether a client is required to respond with Renew message or an Information-Request message. The Reconfigure Message option format is shown in Fig. 7-26.

- If the Type field is set to 19, it indicates Reconfigure Message option.
- The Option Length field is fixed to 1.
- For the Message Type field, two numbers are defined: 5 for Renew message and 11 for Information Request message.

## 7.11.17 Reconfigure Accept option

A client uses this option to notify a server whether it is willing to accept Reconfigure messages. DHCP server uses this option in the same manner. The Reconfigure Accept option format is shown in Fig. 7-27.

- The Type field which is set to 20 indicates Reconfigure Accept option.
- The Option Length field is fixed as 0.

## REFERENCES

1. R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, Dynamic Host Configuration Protocol for IPv6 (DHCPv6), RFC 3315 (July 2003).
2. C. Perkins, IP Mobility Support for IPv4, RFC 3344 (June 2002).
2. S. Thomson and T. Narten, IPv6 Stateless Address Autoconfiguration, RFC 2462 (December 1998).
3. S. Thomson, T. Narten, and T. Jinmei, IPv6 Stateless Address Autoconfiguration, work in progress (June 2004).
4. D. Plummer, An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48-bit Ethernet Address for Transmission on Ethernet Hardware, RFC 826 (November 1982).

Chapter 8

# INTERCONNECTION BETWEEN IPv4 AND IPv6

## 8.1 INTRODUCTION

It is hard to predict when IPv4 address will be exhausted and when we are able to stop using IPv4. However, we definitely can say that we can not move on IPv6 in a day. Instead, IPv4 and IPv6 will exist together for a significant amount of time through transition mechanisms.

In the early stage of evolution to IPv6, we need to consider the environment where isolated IPv6 domains communicate each other in IPv4 network surroundings. To route packets in this environment, we need to consider how well to forward IPv6 data from a source to a destination nodes through IPv4 network.[1]

Various interconnection mechanisms provide interoperability between IPv4 and IPv6 entities throughout the IPv4/IPv6 mixed network environment. They may be classified into two groups; tunneling mechanisms and translation mechanisms as shown in Fig. 8-1. Tunneling mechanisms help isolated IPv6 nodes or IPv6 sites to communicate over IPv4 networks, and translation mechanisms allow IPv4 and IPv6 nodes to communicate.[2] Gradual deploying of IPv6 while providing uninterrupted IPv4 services is expected to happen. In this chapter, these interconnection mechanisms are explained in detail. Necessary terminology will be explained in the next section.

## 8.2 TERMINOLOGY

Following node types should be understood before starting analysis of interconnection mechanisms.

- IPv4-only node: a host or a router supporting only IPv4. This node does not understand IPv6.
- IPv6/IPv4 node: a host or a router to support both IPv6 and IPv4. This node is called dual stack node.
- IPv6-only node: a host or router to support only IPv6. This node does not understand IPv4.
- IPv4 node: a host or a router to support IPv4. Both IPv6/IPv4 and IPv4-only nodes are included.
- IPv6 node: a host or a router to support IPv6. Both IPv6/IPv4 and IPv6-only nodes are included.
- 6to4 host: a host allocated with at least one 6to4 address. This host gets IPv6 connectivity via 6to4 IPv6 router.
- 6to4 router: an IPv6 router supporting 6to4 pseudo-interface to its site. Generally, a border router between IPv6 site and public IPv4 networks becomes 6to4 router.
- 6to4 site: an IPv6 site using 6to4 address for IPv6 connection.
- Relay router: a special 6to4 router configured to connect native IPv6 sites with one or more 6to4 sites.

## 8.3 DUAL STACK

IPv6 nodes may be compatible with IPv4 nodes by implementing both IP protocol stacks and running an appropriate protocol depending on the capability of communicating peer. IPv6 nodes running both IPv4 and IPv6 are called IPv6/IPv4 or dual stack systems.[3] For example, when a dual stack node tries to communicate with IPv4-only node, IPv4 protocol should be run. A dual stack system allows interoperation between IPv4-based nodes and IPv6-based nodes, and applications' gradual transition from IPv4 to IPv6. In a dual stack system, any application based on only one internet protocol stack (e.g. IPv6) may be coexisted and used with other applications based on the other internet protocol stack (e.g. IPv4).[4]

Although a node is implemented with IPv4 and IPv6 protocols, one of them can be disabled for some reasons. It may be operated in one of three modes as follows:[5]

- IPv4 stack enabled, and IPv6 stack disabled
- IPv6 stack enabled, and IPv4 stack disabled
- Both IPv4 and IPv6 stacks enabled

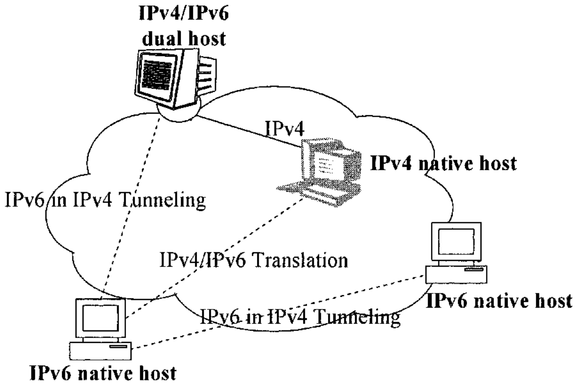


*Figure 8-1.* Mixed scenario with IPv4 and IPv6 networks.

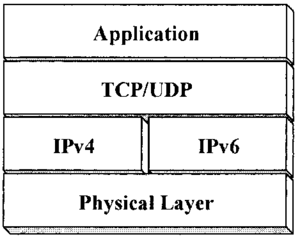


*Figure 8-2.* Dual stack.

Dual stack node may disable one of IP stacks and run the other one. For example, if IPv6 stack is disabled in a dual stack node, then it operates as IPv4 node, and so on.

Dual stack enabled system will be applied to Tunnel End Point (TEP) in interconnection mechanisms. Dual stack nodes may support automatic tunnel, configured tunnel, or both. There are three instances of tunnel assistance on one IPv4/IPv6 host, such as running only automatic tunnel, running only configured tunnel, and running both tunnel mechanisms.

When both IP protocols are enabled in a dual stack system, IPv4 and IPv6 addresses should be configured. For the address assignment to a dual stack system, DHCP or other mechanisms can be employed to assign IPv4 address, and DHCPv6 or stateless address autoconfiguration mechanism can be employed to assign IPv6 address to the node. If a node is assigned globally unique IPv4 address, and it decides to use IPv4-compatible IPv6 address, no address assignment protocol is necessary, and automatic tunnel technique may be accommodated.

## 8.4 IPv6 IMPLEMENTATION OVER IPv4 TUNNEL

A tunneling mechanism which interconnects different IP-based networks provides simple solution for the interoperation with minimal changes in surroundings. This mechanism effectively supports interconnection of IPv6 and IPv4. A tunnel may be generated over IPv4 networks to connect isolated IPv6 hosts or sites over IPv4 networks, or it may be generated over IPv6 networks to connect isolated IPv4 hosts or sites. The latter will be employed at the final stage of transition scenarios from IPv4 to IPv6.[6]

As shown in Fig. 8-1, a tunnel has isolated IPv6 end systems or routers communicating over the current IPv4 infrastructure without requiring any upgrade. Tunneling mechanism will be regarded as the primary issue and the most practical strategy for ISPs and enterprises to develop and deploy IPv6. IPv6 tunnel does not give any influence to current infrastructure and IPv4 services. Thus, it benefits for service providers to furnish simple end-to-end IPv6 service. Besides, it is very useful to connect isolated IPv6 hosts, isolated IPv6 domains, or isolated IPv6 networks, such as 6bone, over IPv4.

Various tunneling mechanisms have been developed and implemented.[5,7] Depending on the node type of tunnel end points, tunneling mechanisms can be classified into four classes, as shown in Table 8-1. In Table 8-2, tunneling mechanisms are explained depending on whether automatic tunnel is supported, and IPv4 address is used to build IPv6 address.

*Table 8-1.* Classification of tunnel mechanism.

| | Tunnel source | Tunnel destination | Description |
|---|---|---|---|
| 1 | Router | Router | IPv6/IPv4 router tunnels IPv6 packets through IPv4 infrastructure. Tunnel exists in one segment packets passed on end-to-end route. |
| 2 | Host | Router | IPv6/IPv4 host tunnels IPv6 packet to intermediate reachable IPv6/IPv4 router through IPv4 infrastructure. In this case, tunnel exists in the first segment packets passed on end-to-end route. |
| 3 | Host | Host | IPv6/IPv4 host tunnels IPv6 packet to destination, IPv6 or IPv6/IPv4 host through IPv4 infrastructure. Tunnel exists in entire end-to-end route. |
| 4 | Router | Host | IPv6/IPv4 router tunnels IPv6 packets destination, IPv6 or IPv6/IPv4 host. Tunnel exists in the final segment packets passed on end-to-end route. |

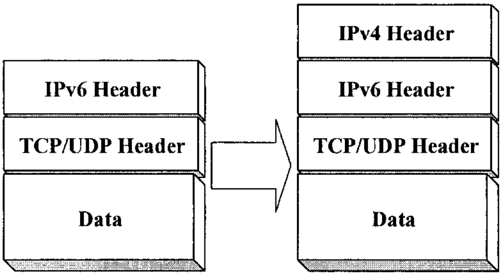


*Figure 8-3*. Encapsulation IPv6 packet into IPv4 packet.

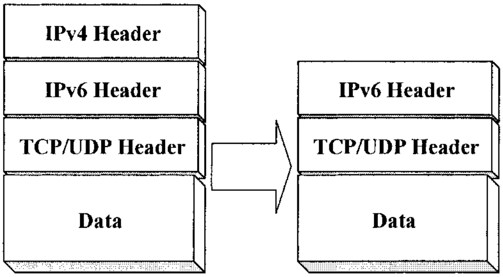


*Figure 8-4*. Decapsulation IPv6 packet from IPv4 packet.

A way to encapsulate and decapsulate packets is shown in Figs. 8-3 and 8-4, respectively.

General procedure to manage tunnels is as follows:[8]

1. Tunnel entrance node (encapsulator) builds IPv4 Header, encapsulates IPv6 packet with the IPv4 Header, and forwards the encapsulated packet to the IPv4 network.

   Whenever it forwards IPv6 packet onto the tunnel, encapsulator may keep soft state per each tunnel to hold some variables (e.g., tunnel MTU)
2. Tunnel exit node (decapsulator) receives packets, reassembles if necessary, and decapsulates IPv4 Header to recover IPv6 packet.

Every tunnel end point should support both IPv4 and IPv6. Thus, tunnel end points should be dual stack nodes as shown in Fig. 8-2. In addition, border routers also must support both IPv4 and IPv6 at the same time.

Eight tunneling mechanisms are discussed in the following subsections. Especially, characteristics and appropriate usage of each mechanism will be discussed.

- IPv6 configured tunnel
- Automatic tunnel with IPv4-compatible IPv6 address
- 6over4 tunnel
- 6to4 Tunnel
- ISATAP
- DSTM
- Tunnel Broker
- Teredo

IPv6 configured tunnel needs some manual configurations. Tunnel source and destination addresses are required to setup in the configured tunnel. However, automatic tunneling mechanisms do not need them. With IPv4-compatible IPv6 address, tunnels between end hosts can be automatically formed. 6over4 mechanism depends on neighbor discovery protocols, and determines tunnel end point by mapping IPv6 multicast address into IPv4 multicast address. 6over4 tunnel is very useful over Ethernet or virtual link layer, but IPv4 multicast based routing should be provided. 6to4 tunnel and ISATAP tunnel can be employed in small sites. They are very similar because the entire bits of IPv4 address are included in building IPv6 address. Tunnel broker service belongs to semi-automatic tunnel. Automatic tunnel setup can be made with IPv4-comptible IPv6 address.[13]

### 8.4.1 IPv6 configured tunnel

In configured tunnel, tunnel end point is determined by the configuration information of an encapsulating node. Thus, for each tunnel, encapsulation node should keep tunnel information, especially address of the tunnel end point. IPv6 configured tunnel is usually created for permanent link connections between two IPv6 domains over IPv4 backbone. Further, this tunnel mechanism is mainly employed for secure communications between two routers. It is also used for communications between a host and an edge router. Tunnel end points should be dual stack enabled nodes. Following procedure is required to configure tunnel end points:[9]

1. Configuring IPv4 and IPv6 addresses to interfaces of dual stack enabled routers.
2. Designating a tunnel entrance and an exit point (source and destination) with IPv4 addresses. To connect isolated sites, ISP may provide appropriate IPv6 address prefix. ISP may provide IPv4 address to setup tunnel end points.

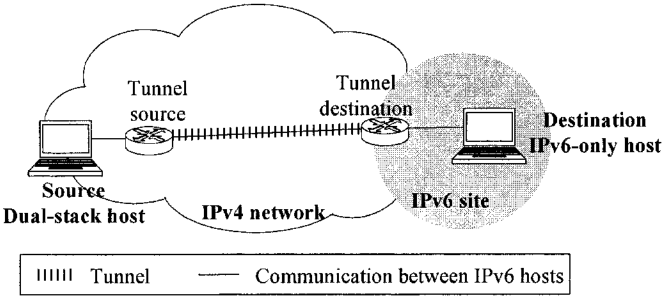


*Figure 8-5.* Configured tunnel.

| 96 bits | 32 |
|---|---|
| 0:0:0:0:0:0 | IPv4 address |

*Figure 8-6.* IPv4 compatible IPv6 address format.

When IPv6 packets are transmitted, the tunnel entrance node encapsulates it with IPv4 Header, and forwards it to the tunnel exit node. Tunnel information configured at the tunnel entrance will determine a tunnel destination. If the tunnel entrance node has multiple tunnels for the destination, the node will make a choice depending on routing information.

In common, configured tunnel exists between routers, and each tunnel is independently managed. The more tunnels mean the more management overheads.

If protocol translation techniques are stable and predefined on a local network, a tunnel entrance node may apply NAT when the final destination node is IPv4 node.

## 8.4.2 Automatic tunnel with IPv4-compatible IPv6 address

Tunnel setup can be done automatically when IPv4-compatible IPv6 addresses are used at source and destination nodes, which is commonly called automatic tunnel. Automatic tunnels can lie between edge routers or between edge routers and end hosts. Tunnel end points should be dual stack enabled. This mechanism allows for IPv4/IPv6 nodes to communicate each other over IPv4 networks using IPv6 without preconfigured tunnels.

In automatic tunnel mechanisms, the IPv4 addresses of tunnel end points (tunnel source and tunnel destination) are automatically determined from the rightmost 32 bits of IPv6 address. IPv4-compatible IPv6 address is a special address format which is composed of 96 bits filled with successive 0s and 32-bit IPv4 address as shown in Fig. 8-6.

There is a specific algorithm to acquire an IPv4 compatible IPv6 address using IPv4 based address configuration protocols.

1. IPv6/IPv4 node uses standard IPv4 protocols to acquire IPv4 address as follows:[2, 5]
   DHCP (Dynamic Host Configuration Protocol)
   BOOTP (Bootstrap Protocol)
   RARP (Reverse Address Resolution Protocol)
   Manual configuration
2. A node uses this address for its IPv4 interface.
3. To build IPv6 address, the node concatenates successive 96-bits of 0 to 32-bit IPv4 address acquired from the first step.

Automatic tunnel technique can be easily used at the earlier stage in IPv6 transition scenarios. It has a benefit such that tunnel setup is very simple over IPv4 network, but scalability in large networks may be considerably deteriorated because each host should have an IPv4 address to build a tunnel.

## 8.4.3 6over4 tunnel

6over4 tunnel mechanism allows isolated IPv6 hosts to be connected with other IPv6 hosts in the same link using IPv4 multicast address.[10] 6over4 tunnel is like a virtual link between 6over4 hosts over IPv4 networks.

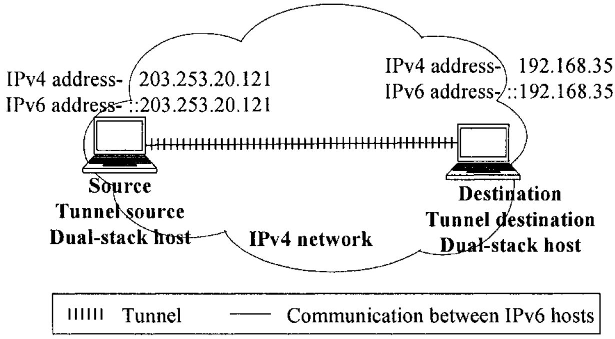

IPv4 address- 203.253.20.121
IPv6 address- ::203.253.20.121
IPv4 address- 192.168.35.32
IPv6 address- ::192.168.35.32
Source
Tunnel source
Dual-stack host
IPv4 network
Destination
Tunnel destination
Dual-stack host
IIIII Tunnel
—— Communication between IPv6 hosts

*Figure 8-7.* Automatic tunnel with IPv4 compatible IPv6 address.

To use this 6over4 tunnel, a site or a domain is required to have connectivity to public IPv4 networks and basically support IPv4 multicast to form IPv6 link-local addresses.[11, 12] IPv4 multicast address is used to perform IPv6 neighbor discovery. Once neighbor discovery process is done, 6over4 hosts get local IPv6 connectivity.

Hosts should be dual stack systems and allocated with at least one IPv4 address. In some cases, IPv4 subnets in the same local multicast scope can be part of private IPv4 networks. If no IPv6-enabled router is found, a 6over4 host distinguishes itself from others using link-local address. Further, if a router which is IPv6 enabled and connected to IPv6 network either logically or physically is found, 6over4 hosts will get global IPv6 connectivity.

For the address assignment, stateless or stateful autoconfiguration mechanism can be adopted. Usually, an IPv6 interface is assigned with more than one addresses; link-local address, all-nodes multicast address, solicited-node multicast address, a (or several) global IPv6 address, and etc. An interface must join the all-nodes multicast address and the solicited-node multicast address using its tentative address before DAD, as explained in Chapter 5.

To build 6over4 tunnel, a host in IPv4 network will follow next steps:

1. When an IPv6 node plugs in, it configures link-local address. Fig. 3-7 (a) shows link-local address format where the modified EUI-64 format is used for 64-bit interface identifier. If an IPv4 address is embedded into a link-local address, the leading 32-bit of interface identifier is set to 0s.
2. The node performs DAD to check duplication on unicast address before assigning it to an interface. Neighbor Solicitation and Neighbor Advertisement messages are encapsulated using IPv4 Header and forwarded to the targeted destination. For example, the destination address of IPv6 Neighbor Solicitation message is the solicited-node multicast address, and this message is encapsulated by IPv4 Header with multicast address. The solicited-node multicast address format and IPv4 multicast address format for solicited-node are shown in Fig. 3-11 and Fig. 3-12, respectively.
3. Hereafter, if there is any IPv6 enabled router in a subnet, router discovery process will be performed. Thus, each 6over4 host will be able to get specific prefix from the router and form unique IPv6 address.
4. Furthermore, if the router is connected to public IPv4 Internet or has connectivity to IPv6, 6over4 hosts can communicate with any other host outside of the subnet. Unless there is a router to provide prefix, hosts in the subnet identify themselves with link-local address.

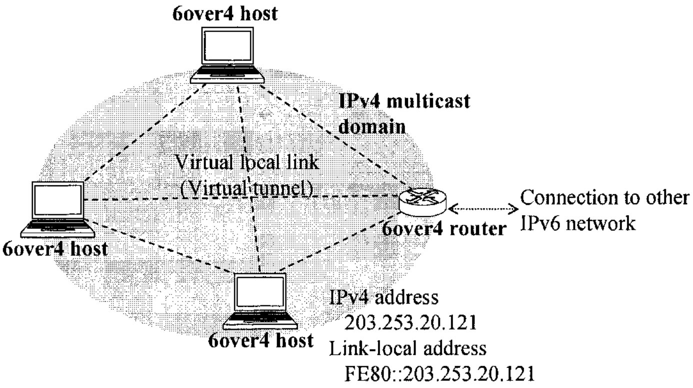

6over4 host
IPv4 multicast domain
Virtual local link (Virtual tunnel)
Connection to other IPv6 network
6over4 router
6over4 host
IPv4 address 203.253.20.121
6over4 host
Link-local address FE80::203.253.20.121

*Figure 8-8.* 6over4 tunnel.

| **Encapsulator IPv4 Header** |
|---|
| **Destination address**<br>239.192.20.121<br>**Source address**<br>0.0.0.0 |
| **IPv6 Header** |
| **Destination address**<br>FF02::1:255.253.20.121<br>**Source address**<br>0::0 |

*Figure 8-9.* Example: encapsulated packet format for 6over4 tunnel.

### 8.4.3.1 Address Format

Link-local address format is shown in Fig.3-7 (a). Typically, interface identifier is 64-bit long and based on EUI-64 identifiers. Sometimes, embedded IPv4 address with successive 0s in the head can be determined as an interface identifier. Chapter 3 explains address architecture and format in detail.

To build solicited node multicast address, following steps must be done.

1. The first 16-bit is set to FF02 in hexadecimal.
2. The next 80-bit is set to sequential 0s with following 1.

3. The last 32-bit is set to 255.X.Y.Z, where X, Y and Z are 6th, 7th and 8th bytes from the interface identifier of link-local address, as depicted in Fig. 3-11.

Corresponding to the solicited node multicast address, IPv4 multicast address for the solicited node is also defined, as shown in Fig. 3-12. For IPv4 multicast address for solicited node, 239.192.Y.Z is defined, where Y and Z are 7th and 8th bytes from interface identifier of link-local address.

Besides, IPv4 all-nodes multicast address, 239.192.0.1, and IPv4 all-routers multicast address, 239.192.0.2, are defined.

### 8.4.3.2 Example

In Fig. 8-8, a host is assigned IPv4 address 203.253.20.121. This 6over4 host generates link-local address and performs DAD. At first, this host builds IPv6 Neighbor Solicit message with solicited-node multicast address and encapsulates it with IPv4 Header. Finally, this message is transmitted using multicast in its subnet. Encapsulated Neighbor Solicitation packet is shown in Fig. 8-9.

6over4 tunnel mechanism supports IPv6 connectivity without IPv4-compatible IPv6 address or configured tunnel. This mechanism can be effectively employed in IPv4 network or an IPv4 network mixed with IPv6 network, especially in the transition stage from IPv4 to IPv6. However, this scheme has a restriction that it should be used only in the same link. The interface of IPv6 enabled router and host should be able to manage 6over4 mechanism.

6over4 technique needs neither IPv4-compatible IPv6 address, nor does tunnel setup information. Thus, any site at the initial stage of transition easily employs IPv6 through 6over4 edge router and provides connection to other IPv6 networks or IPv6 backbones.[13]

## 8.4.4 6to4 tunnel

6to4 tunnel is the most favorite mechanism for IPv6 site administrators to provide IPv6 communication services to hosts without any explicit tunnel setup.[14] This mechanism enables hosts in the IPv6 site to communicate each other and further provides connections to native IPv6 networks via relay routers. 6to4 is intended as an interim transition mechanism when IPv4 and IPv6 are present together, not as a permanent solution.[11, 12]

When 6to4 hosts in different sites need to communicate, packets will pass through 6to4 routers, which is tunnel end points. Usually, one or more border routers of sites are designated as 6to4 routers. Every packet destined to exterior IPv6 sites or domains should pass through the 6to4 router. Fig. 8-11 shows a simple scenario to build 6to4 sites.

A 6to4 router supports IPv6 connectivity for its site without IPv4 compatible IPv6 address, but with the unique IPv4 address which is assigned to one of interfaces belonging to the 6to4 router. This 6to4 mechanism provides connectivity to sites, rather than individual hosts. No additional routing information for 6to4 domain will be generated in IPv4 routing table, and only one entry will appear in the native IPv6 routing table. With such minimal configuration on a border router in 6to4 site, connections between isolated IPv6 sites or domains are easily obtained.[13, 15] However, weakness on security of site to site tunnels should be considered and still under research.[16]

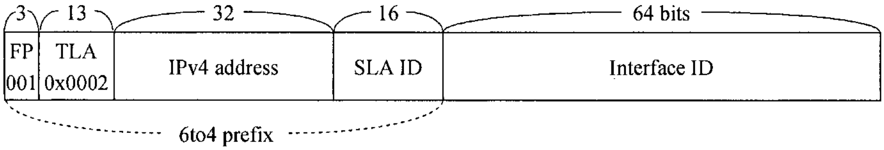


*Figure 8-10.* 6to4 address format.

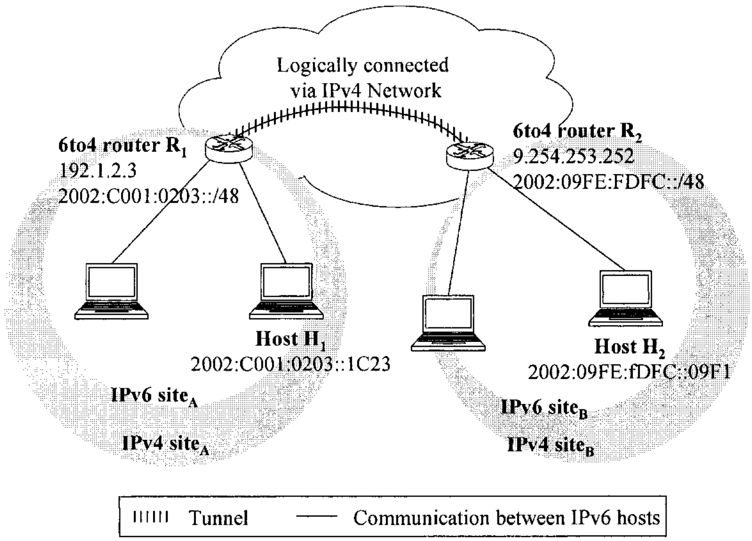


*Figure 8-11.* Simple scenario to build 6to4 sites.

To build 6to4 sites, following minimal changes should be applied to a router and hosts:

1. A 6to4 router selects one of its well-known addresses for 6to4 prefix and broadcasts prefix information to its site.
2. Hosts inside 6to4 sites learn prefix information from 6to4 routers, build 6to4 address, and set 6to4 address as a default IPv6 address.

New DNS records for 6to4 site are created. For example, if the site A has IPv4 address, 192.1.2.3, as shown in Fig. 8-11, new IPv6 DNS records with prefix, 2002:c001:0203::/48 is defined. Similarly, for the site B whose IPv4 address is 9.254.253.252, new IPv6 DNS records with prefix, 2002:09FE: FDFC::/48 will be defined.

When a packet is leaving its site and forwarded to external IPv4 network via 6to4 router, the packet should be encapsulated using IPv4 Header, as shown in Fig. 8-12. Forwarding router will be an encapsulator, and receiving router will be a decapsulator. Encapsulation and decapsulation processes are as same as Figs. 8-3 and 8-4.

In addition to the connection between 6to4 sites, a 6to4 host may need connectivity to native IPv6 domains. To achieve connection between these two different domains, relay router is necessary. Relay router is selected from one of general IPv6 routers except that it understands 6to4 address format and provides forwarding service. For 6to4 service, relay router possesses both 6to4 and native IPv6 addresses. At least one 6to4 relay router is required between a 6to4 sites and an IPv6 networks.

Three routing regions will be defined as follows:

1. Interior 6to4 routing domain of each 6to4 site
2. Exterior 6to4 routing domain interconnecting 6to4 routers and relay routers
3. Exterior 6to4 routing domain of each isolated IPv6 site

For the first case, simple scenario stated above can be employed. For the second case, routing protocol, Border Gateway Protocol (BGP) may be used to obtain routes to native IPv6. Alternatively, one relay router may be appointed as a default router to forward messages to native IPv6 domain. For the last case, a relay router should advertise a route to 2002::/16 to the native IPv6 domain.

#### 8.4.4.1 Address

To provide connectivity between isolated IPv6 sites and native IPv6 domains, and to help address configuration inside the sites, 6to4 mechanism defines new address type, as shown in Fig. 8-10.

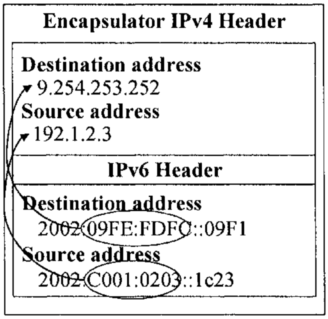


*Figure 8-12.* Encapsulated packet format.

1. 6to4 prefix is built with representative IPv4 address, and a node holding the representative address becomes the entrance of the site. With one globally unique IPv4 address, unique IPv6 prefix is given to the site.
2. The first 16-bit is filled with 2002 in hexadecimal.
3. SLA ID is set to sequential zeros.
4. Interface identifier format follows the same format explained in Chapter 3.

### 8.4.4.2 Example

Let's get back to Fig. 8-11. Basically, IPv4 sites A and B are connected to IPv4 networks, and both sites are able to forward and receive IPv4 packets. The router $R_1$ selects one of its IPv4 addresses, which is 192.1.2.3, for 6to4 service. $R_1$ builds IPv6 prefix as 2002:c001:0203::/48 and broadcast prefix information to the site A. Hosts in the site A receives that prefix information message and set it as a default. Now hosts in site A are able to build global unique IPv6 addresses. For example, host $H_1$ builds its IP address as 2002:C001:0203::1C23, where 1C23 is $H_1$'s identification in hexadecimal. In the same way, the router $R_2$ builds IPv6 prefix for the site$_B$ and broadcasts prefix information to the site. The host $H_2$ generates its IP address as 2002:09F3:FDFC::09F1.

When the host $H_1$ wants to communicate with any host in the site$_A$, packets generated by $H_1$ will be processed in the same way as the general IPv6 packets. If $H_1$ wants to communicate with any host in exterior site, for example, host $H_2$, processing mechanism is a little different:

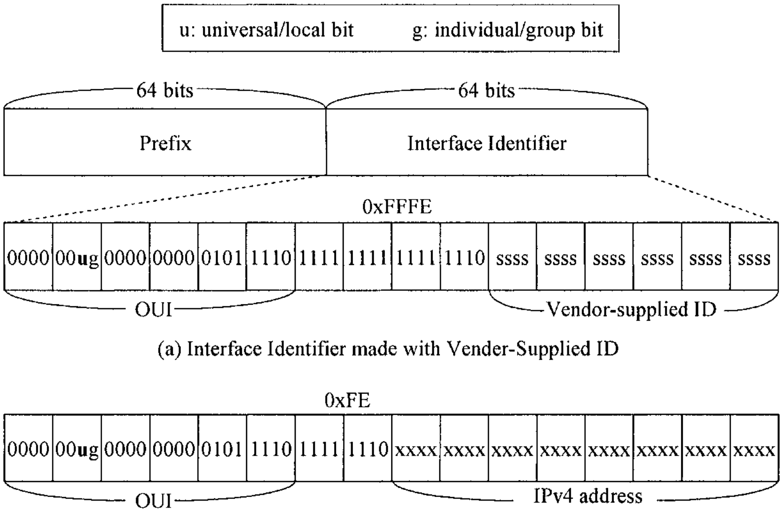
1. $H_1$ will query DNS resource record for a host on site B.
2. Then, $H_1$ will get IP address with the prefix, 2002:09FE:FDFC::/48.
3. Packets generated by $H_1$ will be forwarded to 6to4 router, $R_1$.

4. 6to4 router, $R_1$ recognizes the packets as 6to4 type and will encapsulate the IPv6 packet as shown in Fig. 8-12. The destination IP address of the encapsulated packet is extracted from the Destination Address field of IPv6 Header.
5. Finally, $R_2$ decapsulates the packet and forwards it to the destination node, $H_2$.

### 8.4.5 ISATAP

Intra-site automatic tunnel addressing protocol (ISATAP) is one of automatic tunneling mechanisms. As same as 6to4 mechanism, no specific configuration is necessary for IPv6 connectivity.[17]

IPv4 protocol is regarded as a link-layer protocol to IPv6 protocol, and isolated hosts will be virtually connected as they are directly neighbored. An ISATAP router allocates IPv6 address to ISATAP clients in ISATAP domain. Any unicast address prefix defined in Chapter 3 can be used in ISATAP.


u: universal/local bit
g: individual/group bit
64 bits
64 bits
Prefix
Interface Identifier
0xFFFE
0000 00ug 0000 0000 0101 1110 1111 1111 1111 1110 ssss ssss ssss ssss ssss ssss
OUI
Vendor-supplied ID
0xFE
0000 00ug 0000 0000 0101 1110 1111 1110 xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx
OUI
IPv4 address

(a) Interface Identifier made with Vender-Supplied ID

(b) Interface Identifier made with IPv4 Address

*Figure 8-13.* Address format for ISATAP.

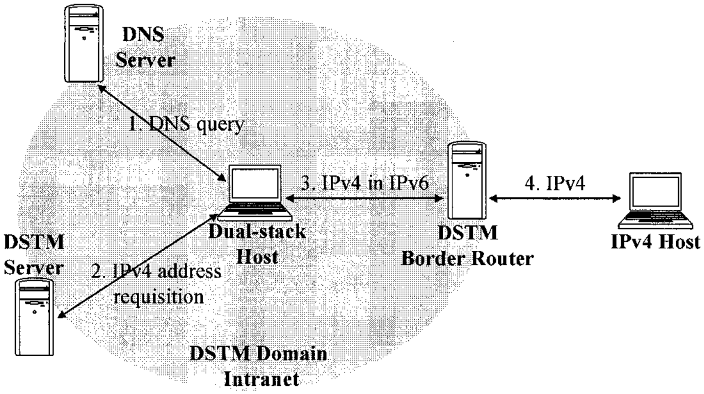


*Figure 8-14.* DSTM.

As shown in Fig. 8-13, interface identifier for ISATAP address is composed of 24-bit IANA OUI field and 40-bit extension identifier. The OUI value is defined as 0x00005E. When the first 16 bits of extension identifier are filled with 0xFFFE, the next 24 bits will contain vendor-supplied ID in the interface identifier. Otherwise, if the first 8 bits are filled with 0xFE, then the next 32 bits will contain IPv4 address assigned to a network interface card.

Encapsulation and decapsulation processes are very same as other tunneling mechanisms. IPv6 host itself becomes encapsulator, and encapsulated packet will be decapsulated at an ISATAP router and forwarded to IPv4 network. When an ISATAP router decapsulates packets, ingress filtering should be performed for both IPv4 and IPv6 addresses before forwarding packets.

### 8.4.6 DSTM

The Dual Stack Transition Mechanism (DSTM) carries IPv4 packets in IPv6 packets within IPv6 intranet. This mechanism will provide IPv4 connectivity for early IPv6 adopters.[18] DSTM is composed of DSTM server, DSTM border router and DSTM hosts. This mechanism may be largely used at the final stage of the transition scenario to IPv6.

The DSTM server will allocate IPv4 address to DSTM clients and guarantees uniqueness of address allocation during address lifetime. Besides, the DSTM server will designate a tunnel end point which is a decapsulator, or called as DSTM border router. To determine an appropriate tunnel end point for each client, load balancing and scalability will be firstly considered.

Once a DSTM border router receives an encapsulated packet, it decapsulates and transmits it to the final destination. To process packets from IPv4 client residing outside of DSTM domain, the border router is required to cache a table matching IPv4 and IPv6 addresses of DSTM client to IPv4 and IPv6 addresses of the foreign node.

The DSTM client is a dual stack host that runs DSTM client software. DSTM mechanism works simply well, as shown in Fig. 8-14.

1. When a dual stack host in IPv6 DSTM domain needs to communicate with IPv4 host outside of domain, the host will request temporary IPv4 address to DSTM server.
2. Once the host (DSTM client) is assigned temporary IPv4 addresses from DSTM server, it builds IPv4 packet, where source address is DSTM client address and destination address is the IPv4 address of destination node. The client may query for outer host's IP address to DNS within DSTM domain.
3. Then, the DSTM encapsulates IPv4 packet using IPv6 packet and forwards it to the designated DSTM border router.

Once a DSTM border router receives any IPv4 message encapsulated by IPv6 packet, the router decapsulates and forwards it to the destination. For the IPv4 address assignment, DSTM servers are defined instead of DHCPv4 servers because basically IPv4 routing is not available within DSTM domain intranet. It is not discouraged to use DHCPv4, but additional IPv4 connectivity should be supported in DSTM domain.[18]

## 8.4.7 Tunnel broker

Tunnel broker is a virtual ISP, providing IPv6 services to isolated IPv6 hosts or sites in IPv4 networks, where the isolated hosts will run both IPv4 and IPv6 protocols and the isolated sites will have dual stack routers at the entry of sites.[19] As IPv6 backbone is deployed, this mechanism should be available and serviceable for IPv4 users to be connected to the IPv6 backbone with no change. Contrary to the configured tunnel mechanism that requires setup and management for each tunnel, tunnel broker service automatically manages tunnel request and setup, and supports interconnection between isolated hosts and IPv6 backbone. [11, 12, 15]

### 8.4.7.1 Components

Tunnel broker service is illustrated in Fig. 8-15, which is composed of a tunnel broker, a tunnel server, and a tunnel client.

- Tunnel broker is a dual stack router which is connected to IPv6 backbone. It is located in IPv4 networks and deals with tunnel creation, modification and deletion on behalf of the isolated IPv6 end users in IPv4 networks.

- Tunnel server is also a dual stack router which is connected to the public IPv4 network and IPv6 backbone. As a tunnel server receives tunnel configuration message from a tunnel broker, it creates, modifies, or deletes a tunnel. The tunnel server becomes a tunnel end point for a tunnel client.
- Tunnel client is an isolated dual stack IPv6 node on IPv4 networks. The client can be a router to represent a site. When a host wants to be connected to IPv6 networks, the host may choose tunnel broker service.

For the scalability of tunnel service, a tunnel broker must have several tunnel servers sharing network loads to prevent congestions. When a tunnel broker receives any request from a client, it first locates an adequate tunnel server for the client and sends configuration orders to the tunnel server. Adequateness is determined by considering distance between a client and a server, and the number of clients which a server is providing services with. The tunnel broker may also register a host on DNS, dynamically, if it is necessary. Detailed process is discussed below.

A client and a tunnel broker should have preconfigured (or automatically established) security configuration to preclude unauthorized users. User identity and credential information must be offered at first. After proper security association, the client should provide information to the tunnel broker such as IPv4 address of the client, host name used in the DNS (not mandatory), and client function (i.e., a standalone host or a router).

If a tunnel client is an IPv6 router to provide IPv6 connectivity to several hosts, the client is supposed to provide more information to a tunnel broker such as the required number of IPv6 address, or necessity of prefix delegation.

### 8.4.7.2 Tunnel creation

A service provider transmits scripts to configure tunnels from isolated hosts to IPv6 networks and allocates IPv6 address to the isolated end system. Furthermore, the service provider may allocate network prefix to a router, and the router will assign IPv6 address to hosts which want to get IPv6 connectivity.

Tunnel creation process is as follows:

1. A dual stack host exchanges security association with a tunnel broker.
2. The dual stack node transmits configuration information to the tunnel broker. For example, there are user identification, password, IPv4 address, available time, DNS information, etc.

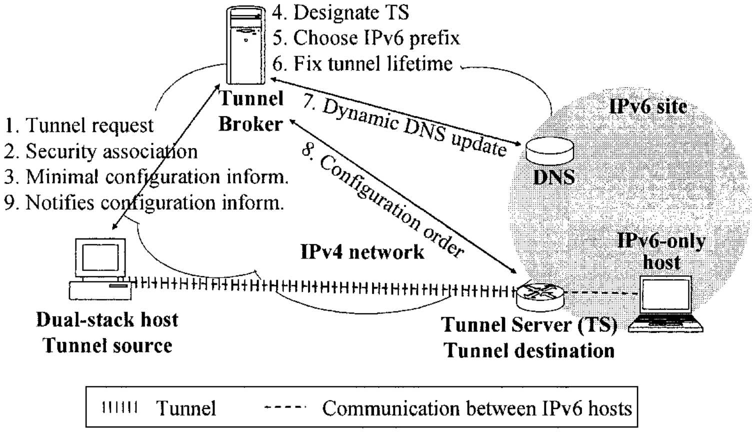


*Figure 8-15.* Tunnel broker service.

3. Once the host is verified, the tunnel broker locates an adequate tunnel server. Depending on the distance between tunnel server and host, traffic and number of connectivity, tunnel broker will locate adequate one.
   Tunnel broker designates IPv6 prefix for a host, or it may provide stateful address.
   Tunnel broker determines tunnel lifetime.
   If the host requests to register IP address to DNS, tunnel broker will register host name and IP address to DNS. Dynamic update mechanism can be adopted.
4. Tunnel broker notifies the tunnel server of new configuration information. The designated tunnel server will be a tunnel end point.
5. Tunnel broker notifies the host of tunnel creation.

As tunnel creation process is over, static tunnel between a host (router) and tunnel server becomes active, and tunnel client gets an access to IPv6 networks. This mechanism has strong advantages in providing easiness and convenience for users, but security fragility should be considered when the isolated server sends configuration information to clients.

## 8.4.8 Teredo

IPv4 Transition mechanisms explained until now work with IPv6-enabled hosts identified by public IPv4 addresses. However, these mechanisms do not support IPv6 connectivity to isolated hosts behind NAT. Private addresses can not be used in public networks like 6to4 mechanism. Coupling NAT and tunnel broker mechanism is not scalable. It may produce unexpected results, such as the quality deterioration because path between NAT clients and IPv6 destination hosts via tunnel server may be a detour. Even worse, tunnel server may be a bottleneck on the routing path.

New transition mechanism is required for private users behind NAT. The Teredo is proposed to provide IPv6 connectivity to isolated hosts behind NAT. The Teredo provides end to end automatic tunnel to Teredo clients and offers paths to IPv6-only hosts.[20]

### 8.4.8.1 Components

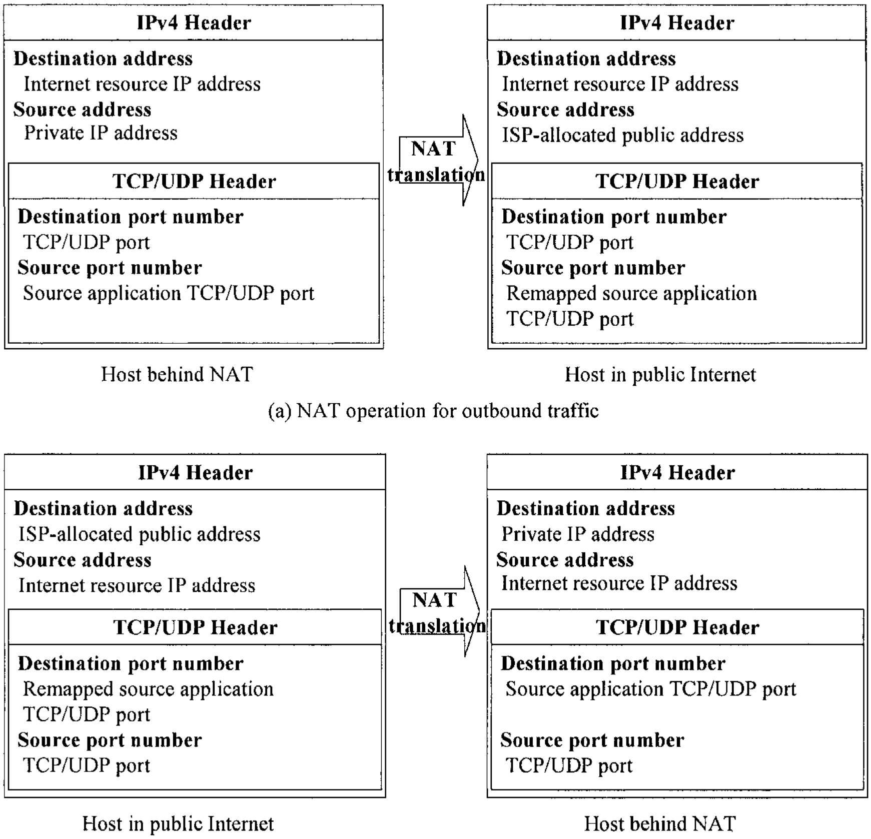
Teredo is composed of three components: Teredo server, Teredo relay, and Teredo client.

- Teredo server provides its IPv4 address to Teredo client. Once a Teredo client learns server's IPv4 address, it will automatically build server's IPv6 address where IPv4 address is embedded.
- Teredo relay helps Teredo clients connect to IPv6-only hosts. Communications with IPv6-only hosts in backbone is not supported at a stroke. Instead, Teredo server will mediate between Teredo clients and Teredo relays and select appropriate Teredo relay. The appropriateness is determined by the distance between a client and a relay as well as the number of clients which a relay is providing service for.
- Teredo client is a node residing behind NAT and wants to have IPv6 connectivity. For the address configuration, a Teredo client learns prefix information from Teredo server. Especially, the Teredo client is required to have server's IPv4 address before the qualification process and to maintain mapped address and port number associated with Teredo service port. Date and time since a client has interaction with a Teredo server, Teredo IPv6 address, and list of recent Teredo peers should be kept in the client.

For Teredo service, new prefix, 3FFE:831F::/32 is defined. This prefix belongs to the Microsoft's reserved address space. When packets pass through NAT to the outside, both address and port number will be mapped into external address and external port number, respectively. In Teredo, outbound traffics from a Teredo client are encapsulated IPv6 packets, but NAT does not care about IPv6 Header and remaining parts. For inbound traffics, the similar procedure should be processed by NAT. The destination Teredo UDP port number is reserved as 3544.

- Teredo IPv6 service prefix: 3FFE:831F::/32
- Teredo IPv6 client prefix: global 64-bit prefix composed of Teredo IPv6 service prefix and Teredo server address

- Teredo node identifier: 64-bit identifier composed of flag indicating NAT type, obfuscated Teredo mapped address and obfuscated Teredo mapped port number
- Teredo IPv6 address: 128-bit IPv6 address acquired by appending Teredo node identifier to Teredo IPv6 client prefix. Fig. 8-17 depicts Teredo IPv6 address format
- Teredo UDP port: 3544
- Teredo mapped address: global IPv4 address translated by NAT
- Teredo mapped port: global port number translated by NAT



(a) NAT operation for outbound traffic

(b) NAT operation for inbound traffic

*Figure 8-16.* NAT operation.

### 8.4.8.2 NAT

NAT provides IP mapping service between private address and public address, which allows nodes in private domain to communicate with exterior nodes.[21] NAT also performs port number translation. Usually, NAT works with UDP because UDP does not keep session information. When NAT sees UDP packet coming, NAT may provide mapping service even if no information is present. Fig. 8-16 shows common NAT operation for outbound and inbound traffics, respectively.

Basically, for outbound traffics, internal UDP port number and IP address are mapped into external UDP port number and IP address. For inbound traffics, mapping is also performed by NAT. The mapped information is stored in the NAT translation table. Depending on NAT type, traffics from unknown exterior nodes may be discarded at NAT.

NAT can be classified into three types: cone NAT, restricted NAT and symmetric NAT.

- In cone NAT approach, no restriction is given to inbound or outbound traffics. NAT provides mapping internal UDP port number and IP address into external UDP port number and IP address, or vice versa.
- In restricted NAT approach, only pre-configured inbound traffics are allowed to enter into NAT. Mapping between an internal IP address and port number and external address and port number for specific foreign IP address, and port number should pre-exist in NAT. Even inbound traffics whose destination address and port number are equal to any entry in NAT translation table are silently discarded if they are from unknown external addresses or port numbers.
- If NAT translates an internal IP address and port number differently depending on the destination, it is symmetric NAT. Teredo is not compatible with symmetric NATs.

### 8.4.8.3 Address

Teredo address format is shown in Fig. 8-17.

- Teredo prefix is 3FFE: 831F::/32.
- Next 32 bits are filled with Teredo server's IPv4 address, which is pre-configured in a Teredo client.
- When a Teredo client finds that it is behind cone NAT, the Flag field is set to 0x8000, which indicates cone NAT. Otherwise, the Flag field is set to 0x0000.

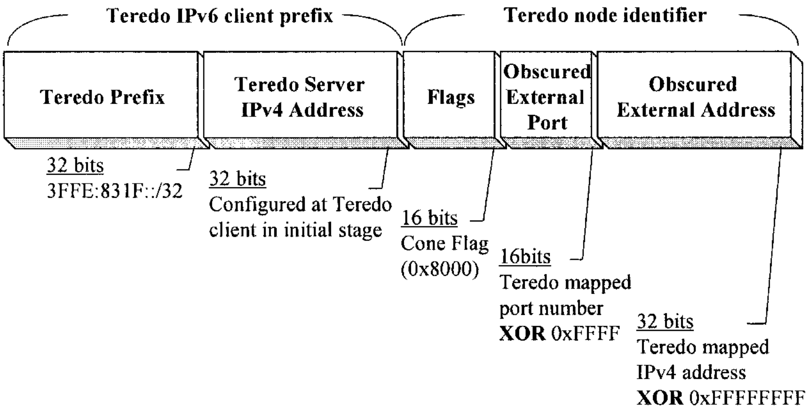

Teredo IPv6 client prefix
Teredo node identifier
Teredo Prefix
Teredo Server IPv4 Address
Flags
Obscured External Port
Obscured External Address
32 bits
3FFE:831F::/32
32 bits
Configured at Teredo client in initial stage
16 bits
Cone Flag (0x8000)
16bits
Teredo mapped port number XOR 0xFFFF
32 bits
Teredo mapped IPv4 address XOR 0xFFFFFFFF

*Figure 8-17.* Teredo address format.

- Obscured external port number and obscured external address are learned from the Indication of Router Advertisement message from a Teredo server. Obscured port number is obtained by performing XOR external port number with 0xFFFF. Obscured external address is obtained by performing XOR external address with 0xFFFFFFFF.

### 8.4.8.4 Teredo packets

In Teredo, data packets and Bubble packets are defined. Bubble packet is not used for data exchange but used for the qualification process to create or maintain IP address and port number of an exterior node on NAT mapping table.

Teredo data packets are delivered on UDP packet within IPv4 packet, and it is shown in Fig. 8-18:

- In IPv4 Header, source and destination IPv4 addresses are contained. When a Teredo client builds a packet, the Source Address field of IPv4 Header is filled with client's private IPv4 address. Later on, NAT maps the private address in the Source Address field into public address.
- In UDP Header, source and destination port numbers are contained, and source port number is translated into public one by NAT.
- NAT does not touch IPv6 part. Source and destination IPv6 addresses of a packet belong to original source and final destination, respectively. In address fields of IPv6 Header, at least one Teredo address should be contained.

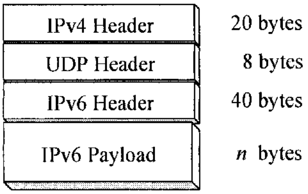


*Figure 8-18.* Data packet format.

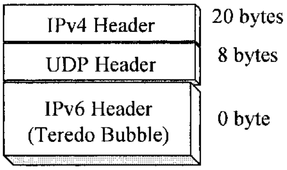
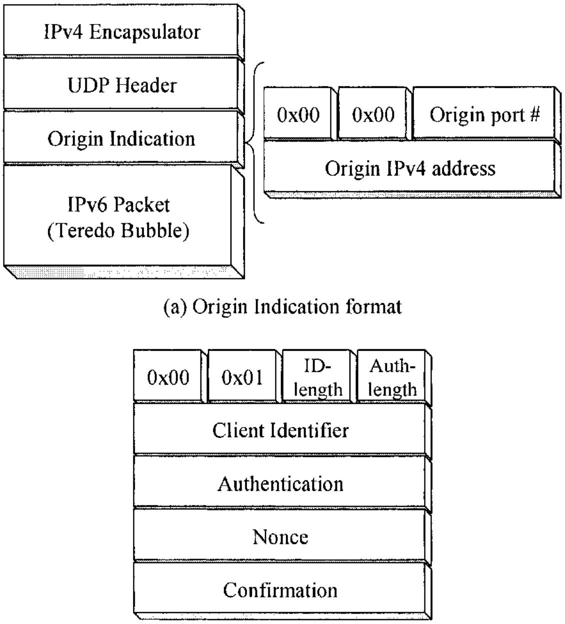


*Figure 8-19.* Teredo Bubble packet format.



(a) Origin Indication format

(b) Authentication Indication format

*Figure 8-20.* Teredo Bubble packet with Indication.

Teredo Bubble packet is defined to create or maintain NAT mapping. The purpose of Bubble packet is to create new entry for outer node's entry in NAT mapping table, thus Bubble packet does not contain any IPv6 payload. Minimal overhead is expected when Bubble packet is used for the initial communications setup through NATs. The packet format is shown in Fig. 8-19. However, no Bubble packet exchange is required when initial communications is launched between Teredo clients behind cone NAT.

Besides, Origin Indication is used in Bubble packet to indicate an IPv4 address and port number of Teredo client which are translated by NAT. Usually, when a Teredo server sends Router Advertisement message in response to Router Solicitation message, Origin Indication is added to the message to notify translated address information of Teredo client. Packet with Origin Indication is shown in Fig. 8-20 (b).

Bubble packet may include the other indication type, the Authentication Indication that contains client identifier, authentication value, 8-byte nonce value, and 1-byte confirmation. The length of this Authentication Indication is variable, and the format is shown in Fig. 8-20 (b). When authentication and Origin Indications are contained together in the same packet, authenticator always precedes Origin Indication.

### 8.4.8.5 Qualification process

To obtain IPv6 connectivity, Teredo client must experience qualification process. The qualification process is composed of repeated sequential exchanges of Router Solicitation and Advertisement messages:

1. At first, Teredo client starts to send Router Solicitation message.

   The source IPv6 address of Router Solicitation message is link local address, where the Flag field is filled with 0x8000, and destination address is set to Teredo server's link-local address.

   For instance, there are Teredo client$_A$ and Teredo server$_B$ and Teredo server$_C$ in network, as shown in Fig. 8-21. If the preferred Teredo server of client$_A$ is assumed to be server$_B$, client$_A$ will send Router Solicitation message to server$_B$. The packet format of Router Solicitation message generated from client$_A$ is shown in Fig. 8-22 (a).
2. The Router Solicitation message from client$_A$ is modified by NAT and will be transmitted to the final destination, server$_B$, which sees client$_A$'s external IPv4 address.
3. The client$_A$ waits for Router Advertisement message from server$_B$. Unless messages are coming from Teredo server$_B$ before timer expires, client$_A$ repeats to send Router Solicitation message. Maximum number of repetition is 3, and time out value is 4 seconds according to RFC 2461.

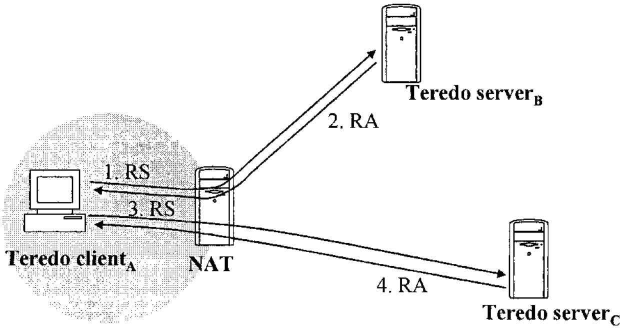
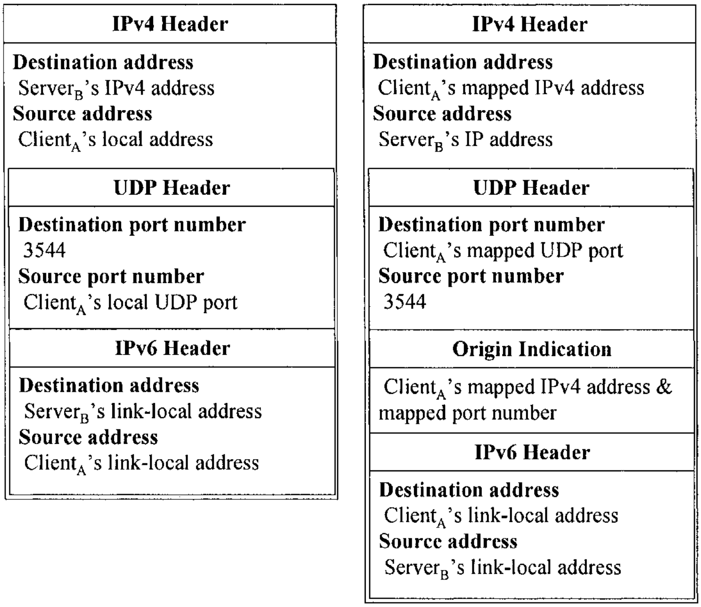


*Figure 8-21.* Initial configuration of Teredo client.

4. Once Teredo server, server$_B$ receives Router Solicitation message, where the Flag field of IPv6 source address is set to 0x8000, server$_B$ notices that the client$_A$ is under the initial qualification process. The server$_B$ builds Router Advertisement message with its alternate IPv4 address. The message format for Router Advertisement message is shown in Fig. 8-22 (b).
5. If this message goes through NAT, and if it is proved to be valid, then client$_A$ knows that it is behind cone NAT and the qualification process is over.

   As client$_A$ gets Router Advertisement messages from Teredo server$_B$ within the time specified above, it checks that the message contains Origin Indication and valid message. From the prefix information option of IPv6 Header in the advertisement message, client$_A$ learns valid Teredo IPv6 server prefix, where the first 32-bit is global Teredo IPv6 service prefix, and the following 32 bits should be server's IPv4 address. Teredo mapped address and mapped port number are also learned from the Origin Indication. Now Teredo client$_A$ will be able to generate its Teredo IPv6 address.
6. If Teredo client$_A$ fails to get Router Advertisement messages from step 5, it sends Router Solicitation message again, but this time the Flag field is set to 0x0000.
7. When Teredo server$_B$ receives this Router Solicitation message, it builds Router Advertisement message, where source IPv6 address is as same as the destination IPv6 address of Router Solicitation message.
8. If a Teredo client$_A$ receives Router Advertisement message, it recognizes it resides behind restricted NAT.
9. Next process is identical to cone NAT except that Teredo client$_A$ should check whether it is behind symmetric NAT.



**IPv4 Header**

**Destination address**
Server$_B$'s IPv4 address
**Source address**
Client$_A$'s local address

**UDP Header**

**Destination port number**
3544
**Source port number**
Client$_A$'s local UDP port

**IPv6 Header**

**Destination address**
Server$_B$'s link-local address
**Source address**
Client$_A$'s link-local address

(a) Router Solicitation message from Client$_A$ to Server$_B$

**IPv4 Header**

**Destination address**
Client$_A$'s mapped IPv4 address
**Source address**
Server$_B$'s IP address

**UDP Header**

**Destination port number**
Client$_A$'s mapped UDP port
**Source port number**
3544

**Origin Indication**

Client$_A$'s mapped IPv4 address & mapped port number

**IPv6 Header**

**Destination address**
Client$_A$'s link-local address
**Source address**
Server$_B$'s link-local address

(b) Router Advertisement message from Server$_B$ to Client$_A$

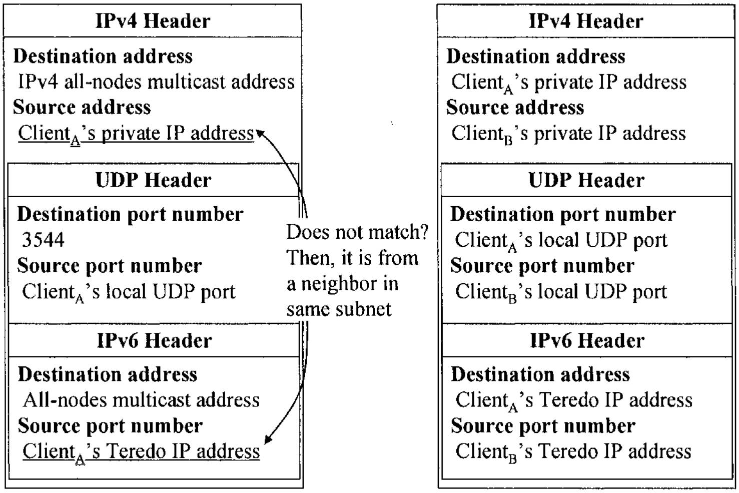
*Figure 8-22.* Exchanged packet format for initial configuration.

10. Client$_A$ sends a Router Solicitation message to another or secondary tunnel server$_C$. If client$_A$ gets Router Advertisement messages whose indication information is different from the other advertisement message from the former server$_B$, then the client recognizes that it resides behind symmetric NAT. In the case of symmetric NAT, no Teredo service is provided.

Once the qualification process is over, Teredo client configures its IPv6 address and is able to communicate with other Teredo clients or IPv6-only hosts. Communications is processed differently, depending on the type of correspondent node, which is divided into three groups such as Teredo clients in the same site, Teredo clients in a different site, and IPv6-only host.

### 8.4.8.6 Initial communications between Teredo clients in the same site

For initial communications between Teredo clients in the same site, Neighbor Discovery protocol is used to find neighbors.


IPv4 Header
Destination address
IPv4 all-nodes multicast address
Source address
Client$_A$'s private IP address
UDP Header
Destination port number
3544
Source port number
Client$_A$'s local UDP port
IPv6 Header
Destination address
All-nodes multicast address
Source port number
Client$_A$'s Teredo IP address
Does not match? Then, it is from a neighbor in same subnet
IPv4 Header
Destination address
Client$_A$'s private IP address
Source address
Client$_B$'s private IP address
UDP Header
Destination port number
Client$_A$'s local UDP port
Source port number
Client$_B$'s local UDP port
IPv6 Header
Destination address
Client$_A$'s Teredo IP address
Source port number
Client$_B$'s Teredo IP address

(a) Request message from client$_A$ (b) Reply message from client$_B$

*Figure 8-23.* Exchanged packet format for initial communications between Teredo clients in the same site.

1. When a Teredo client$_A$ wants to know its neighbors, it builds a Neighbor Solicitation message as shown in Fig. 8-23 (a).
2. After the neighboring Teredo client$_B$ receives this solicitation message, it becomes aware of difference in the source address of IPv4 Header and IPv6 Header, and finally determines that the message is from neighboring Teredo client. If a packet is from exterior sites, IPv4 destination address and mapped IPv4 address induced from destination IPv6 address should be identical.
3. Teredo client$_B$ builds response packet as Fig. 8-23 (b). Now, Teredo client$_A$ knows client$_B$'s IPv6 Teredo address, and it is able to make communications.

### 8.4.8.7 Initial communications between Teredo clients in different sites

For initial communications with Teredo client behind cone NAT, no Bubble packet exchange is required. External IPv4 address and port number for correspondent node can be extracted from the last 48 bits of Teredo node identifier.

On the other hand, to communicate with Teredo client behind restricted NAT, Bubble packet exchange is necessary before the real data transmission.

Fig. 8-24 shows an example where Teredo client$_A$ wants to correspond with Teredo client$_B$, and both clients reside in restricted NAT. If no entry for address information for client$_A$ is present at NAT which is located in front of client$_B$, traffics from client$_A$ will be discarded silently. To initiate communications with client$_B$, following process is necessary:

1. Teredo client$_A$ sends Bubble packet to client$_B$, but this Bubble will be discarded when NAT which is located in front of client$_B$ does not have information for client$_A$.
2. Then, client$_A$ sends Bubble packet to client$_B$'s Teredo server again. In the Bubble packet, destination address of IPv4 packet is set to client$_B$'s Teredo server (Teredo server$_C$), and IPv6 destination address is set to client$_B$'s Teredo address. Teredo server$_C$'s address is extracted from client$_B$'s IPv6 address.
3. Teredo server$_C$ receives Bubble packet from client$_A$, sees IPv6 destination address, and forward this packet to client$_B$, where source address of IPv4 Header is set to server$_C$'s address. This packet is not blocked, because an entry for Teredo server$_C$ is found in NAT communications due to the previous communications between client$_B$ and server$_C$.
4. The final destination, client$_B$ receives the Bubble packet from server$_C$ and builds and transmits Bubble packet to Teredo client$_A$. This packet is not blocked due to the same reason why packets from Teredo server$_C$ may pass NAT. Now, either one of clients behind restricted NAT are able to initiate communications.

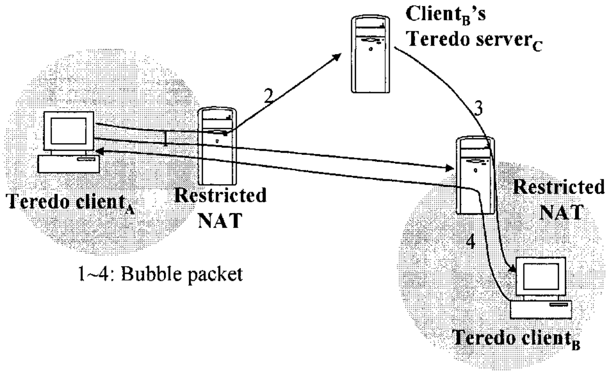
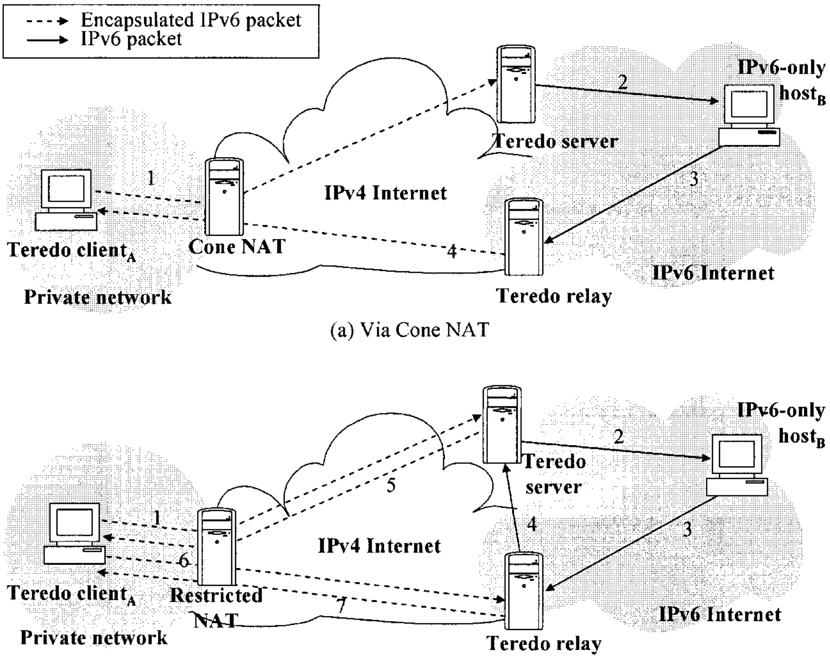


*Figure 8-24.* Initial communications between Teredo clients in different site via restricted NAT.



(a) Via Cone NAT

(b) Via Restricted NAT

*Figure 8-25.* Initial communications between Teredo client and IPv6-only host.

### 8.4.8.8 Communication between Teredo client and IPv6-only hosts

If a Teredo client wants to communicate IPv6-only hosts in native IPv6 network, such as 6bone, Teredo relay operates as a tunnel end point in behalf of Teredo client, as shown in Fig. 8-25. Depending on whether the Teredo client resides behind cone NAT or restricted NAT, the form of initial communications becomes different.

When the Teredo client is behind cone NAT, initial communication process is handled as shown in Fig. 8-25 (a).

1. Teredo client$_A$ must find out the nearest Teredo relay of an IPv6-only host$_B$, at first. It sends ICMPv6 Echo Request message, where IPv6 destination address is IPv6-only host's global address and IPv4 destination address is Teredo server's IPv4 address.
2. When Teredo server receives this ICMPv6 Echo Request message, it becomes aware that the destination address in IPv6 Header identifies the

other node. The server decapsulates IPv4 and UDP Headers and forwards it to host$_B$.

3. Once IPv6-only host receives ICMPv6 Echo Request message, it builds and transmits a Reply message. Concurrent routing algorithm will guide this Teredo-addressed packet into the nearest Teredo relay.
4. Then, Teredo relay encapsulates the received ICMPv6 Echo Reply message with IPv4 and UDP Headers. IPv4 address and UDP number belonging to Teredo relay are used in the encapsulation process.
5. Now, Teredo client$_A$ knows the adequate Teredo relay for IPv6-only host, and it is able to send an initial packet to the IPv6-only host via Teredo relay. Tunnel end points are Teredo client and Teredo relay.
6. When the Teredo relay receives a packet from Teredo client$_A$, Teredo relay decapsulates and forwards it to the IPv6-only host.

If a Teredo client is behind a restricted NAT, initial communication process becomes complicated, as shown in Fig. 8-25 (b). Initial traffics from a Teredo relay may be discarded by the restricted NAT if no entry is found in it. Therefore, traffics destined to Teredo client should take a roundabout way via Teredo client's server.

1. ~ 3. Same as above.
4. The Teredo relay may know that the destination client$_A$ is behind a restricted NAT by noticing that the Flag field of Teredo address is set to all zeros. If there is no entry for the Teredo relay in the restricted NAT, any packet transmitted from the Teredo relay will be silently discarded. The Teredo relay generates Bubble packet destined to the client$_A$ via client$_A$'s Teredo server. Address of client$_A$'s Teredo server is extracted from client$_A$'s address.
5. Once client$_A$'s Teredo server receives Bubble packet from the Teredo relay, it forwards this message to client$_A$. Teredo relay's IPv4 address and port number will be contained in Indication.
6. Now, Teredo client$_A$ knows the intermediate and nearest Teredo relay for IPv6-only host and sends Bubble packet to the Teredo relay, which allows traffics from Teredo relay to go through the restricted NAT.
7. Teredo relay forwards the ICMPv6 Echo Reply message that has been suspended since step 4.
8. Now, Teredo client$_A$ is able to send an initial packet to the IPv6-only host via Teredo relay. Tunnel end points are Teredo client and Teredo relay.
9. When the Teredo relay receives a packet from Teredo client, Teredo relay decapsulates and forwards it to the IPv6-only host$_B$.

*Table 8-2.* Summary: detailed classification of tunneling mechanisms.[2]

| | Primary usage | Merits | Demerits | Requirements |
|---|---|---|---|---|
| IPv6 configured tunnel | Stable and secure communication link | No DNS required to provide IPv6 service | End-to-end tunnel,<br>Management overhead,<br>NAT | IPv6 address registered at ISP,<br>Dual stack router |
| Automatic tunnel with IPv4-compatible IPv6 address | Single host or small site,<br>Infrequent communication | Simple tunnel setup | Limited communication between only IPv4 compatible sites | IPv6 prefix (0::/96),<br>IPv4 address |
| 6to4 tunnel | Connection of isolated IPv6 domains | Tunnel setup with minimal management overhead | 6to4 router and relay router are critical point,<br>Bad scalability | IPv6 prefix (2002::/16),<br>Dual stack router |
| ISATAP tunnel | Campus site,<br>Connection of IPv6 sites with no router | No configuration is necessary | - | Dual stack router |
| 6over4 tunnel | Campus site,<br>Connection of IPv6 sites with no router | Simple tunnel setup,<br>IPv4 compatible IPv6 address is not necessary | No tunnel support without IPv4 multicast routing | Connection to public IPv4 network,<br>IPv4 Multicast routing should be supported |
| DSTM | Campus site | Connection of isolated IPv4 hosts on IPv6 site | Bottleneck at DSTM border router | DSTM server,<br>DNS translation |
| Tunnel broker | Isolated IPv6 end-host | Tunnel setup and management from ISP | Limited security | Script building and transmission |
| Teredo | Connection of isolated hosts behind NAT | Simple tunnel setup with smallest management overhead | Limited security in NAT,<br>No tunnel support for Symmetric NAT | Teredo Agent |

## 8.5 TRANSLATION MECHANISM

IPv6 connectivity service between isolated IPv6 hosts or domains via IPv4 network can be provided using various transition mechanisms stated in the previous section. However, IPv6 connectivity service between IPv6 host and IPv4 host can be settled by translation mechanisms.

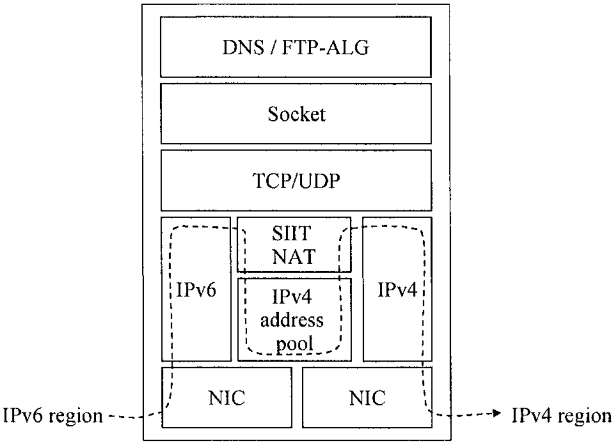


*Figure 8-26.* NAT-PT architecture.

The stateless IP/ICMP translation (SIIT) protocol provides connectivity service between IPv6-only and IPv4-only hosts via address translation. SIIT does not keep state information for each session and further does not specify or set limit how long IPv4 or IPv6 address is assigned to host.[22]

On the contrary, network address translation-port translation (NAT-PT) uses IPv4 address pool for address assignment to IPv6 hosts dynamically whenever a session is initiated across IPv4/IPv6 domain. For transparent routing, NAT-PT keeps mapping tables which bind IPv6 address to IPv4 address.[23] If an application program in a host refers and contains IP address in a message, then the address translation in the application layer should be performed simultaneously with IP layer address translation.

The Application Level Gateway (ALG) may be employed in NAT-PT. For instance, there is DNS ALG or FTP ALG. NAT-PT will provide complete solution for communications between IPv6-only and IPv4-only hosts with ALG and SIIT.

Fig. 8-26 shows NAT-PT architecture and its specific process is as follows:

1. When a packet is received from IPv6 domain, NAT-PT ingress filtering is applied to the IPv6 address to check whether the address is valid.
2. Once the address is verified, NAT-PT lookups its IPv4 address pool and assigns IPv4 address with valid lifetime.

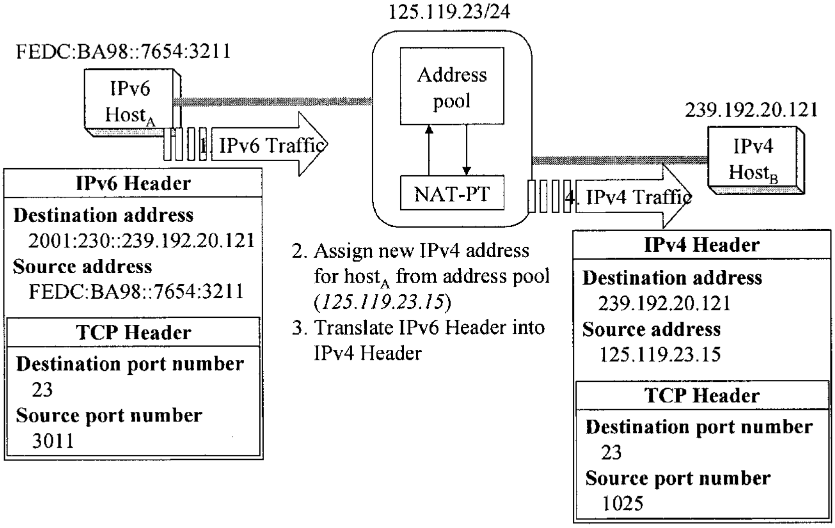


*Figure 8-27.* Basic NAT-PT operation for outbound traffic.

*Table 8-3.* Example: mapping table of NAT-PT.

| | Source IPv6 address | Mapped IPv4 address | Parameters |
|---|---|---|---|
| | ... | ... | ... |
| Outbound | FEDC:BA98::7654:3211 | 125.119.23.15 | Port number 3011 is mapped to 1025 |
| | ... | ... | ... |

3. For outbound traffic, IPv6 host may query IPv4 host. DNS ALG in NAT-PT will act in behalf of IPv6 host. DNS ALG will query IPv4 host, and query type will be replaced from ‘AAAA’ or ‘A6’ to ‘A’, where ‘AAAA’ identifies domain name for IPv6 address and ‘A’ identifies domain names for IPv4 addresses.
4. Once DNS ALG gets DNS information from DNS server residing IPv4 domain, it changes DNS record type to AAAA or A6 and modifies IPv4 to IPv6 address and finally returns reply to the IPv6 host.

For example, when a host$_A$ whose IPv6 address is FEDC:BA98::7654:3211 wants to make communications with IPv4 host$_B$ whose address 239.129.20.121, NAT-PT will assign a new IPv4 address to host$_A$, 125.119.23.15, as shown in Fig. 8-27. For the address translation, NAT-PT keeps state of each session and detailed information in mapping table, as shown in Table 8-3. NAT-PT builds IPv4 packet, where IPv4 Header is translated from IPv6 Header and Data field is also copied from

Data field of IPv6 packet. In addition, for inbound traffic, IPv4 address of IPv4 host will be modified in the same way as outbound traffic. An inbound query from IPv4-based DNS will be processed by DNS ALG.

## REFERENCES

1. F. Baker, E. Lear E, and R. Droms, Procedures for Renumbering an IPv6 Network without a Flag Day, work in progress (February 2004).
2. Microsoft, IPv6 Deployment Strategies, Microsoft Corporation (December 2002).
3. J. Bound, Dual Stack Transition Mechanism, work in progress (April 2004).
4. S. Roy, A. Durand, and J. Paugh, Issues with Dual Stack IPv6 on by Default, work in progress (May 2004).
5. E. Nordmark and R. Gilligan, Basic Transition Mechanisms for IPv6 Hosts and Routers, work in progress (June 2004).
6. A. Conta and S. Deering, Generic Packet Tunneling in IPv6 Specification, RFC 2473 (December 1998).
7. R. Gilligan and E. Nordmark, Transition Mechanism for IPv6 Hosts and Routers, RFC 2893 (August 2000).
8. D. Haskin and R. Callon, Routing Aspects of IPv6 Transition, RFC 2185 (September 1997).
9. A. Durand and F. Parent, Requirements for assisted tunneling, work in progress (June 2004).
10. B. Carpenter and C. Jung, Transmission of IPv6 Packets over IPv4 Domains without Explicit Tunnels, RFC 2529 (March 1999).
11. C. Huitema, R. Austein, S. Satapati, and R. Van der Pol, Unmanaged Networks IPv6 Transition Scenarios, RFC 3750 (April 2004).
12. C. Huitema, R. Austein, S. Satapati, and R. Van der Pol, Evaluation of IPv6 Transition Mechanisms for Unmanaged Networks, work in progress (June 2004).
13. B. Carpenter and K. Moore, Connection of IPv6 Domains via IPv4 Clouds without Explicit Tunnels, RFC 3056 (February 2001).
14. M. Lind, V. Ksinant, S. Park, A. Baudot, and P. Savola, Scenarios and Analysis for Introducing IPv6 into ISP Networks, work in progress (June 2004).
15. J. Bound, IPv6 Enterprise Network Scenarios, work in progress (June 2004).
16. P. Savola and C. Patel, Security Considerations for 6to4, work in progress (June 2004).
17. F. Templin, T. Gleeson, M. Talwar M, and D. Thaler, Intra-Site Automatic Tunnel Addressing Protocol (ISATAP), work in progress (May 26, 2004).
18. J. Bound, Dual Stack Dominant Transition Mechanism (DSTM), work in progress (January 2005).
19. A. Durand, P. Fasano, I. Guardini, and D. Lento, IPv6 Tunnel Broker, RFC 3053 (January 2001).
20. Microsoft, Teredo Overview, Microsoft Corporation (January 2003).
21. P. Strisuresh, M. Holdrege, IP Network Address Translator (NAT) Terminology and Considerations, RFC 2663 (August 1999).
22. E. Nordmark, Stateless IP/ICMP Translation Algorithm (SIIT), RFC 2765 (February 2000).
23. P. Tsirtsis and G. Srisuresh, Network Address Translation - Protocol Translation (NAT-PT), RFC 2766 (February 2000).

Chapter 9

# DOMAIN NAME SYSTEM (DNS)

## 9.1 INTRODUCTION

Domain name system (DNS) is a distributed database that offers mapping service from domain name into IP address.[1, 2] DNS helps users contact web sites or resources in Internet with simple domain names, instead of long numeric IP addresses. Since operating systems can not understand domain names, DNS translates them into IP addresses which are logical addresses of devices. DNS also offers other services such as the reverse mapping from IP addresses to domain names.[3]

Whenever a client application needs to communicate with another party, but it does not have any knowledge about logical addresses except domain names, it queries one of the nearest name servers and gets appropriate answers in response to that query.

DNS has been developed as a systematical delegation model and distributed domain database with hierarchy. Hierarchical name space supports flexible structure with additional extensions. Distributed database architecture augments management efficiency.

The identifier of IP version 4 and version 6 is 32-bit long and 128-bit long, respectively. Current DNS service can not be applied to IPv6 addresses because the length of IPv6 address is four times longer than that of present IPv4 address. Extensions or modifications to incorporate with IPv6 addresses are required to current DNS systems.

New record type, ‘AAAA’ is defined to support 128-bit IPv6 address in DNS.[4] The AAAA record is just an extension of present ‘A’ record type.

Thus, most DNS entities will handle the new record type without much trouble.

As DNS plays a decisively important role for services in IPv4 based network, it will be applied to IPv6 based networks or even mixed networks with the same or even higher prominent importance. Effective management of name spaces, prefix delegation, address aggregation and renumbering on the mixed networks are still under development as hot issues[5, 6] for more elegant DNS services.

## 9.2 TERMINOLOGY

The following terminologies should be understood before starting analysis of DNS mechanism.

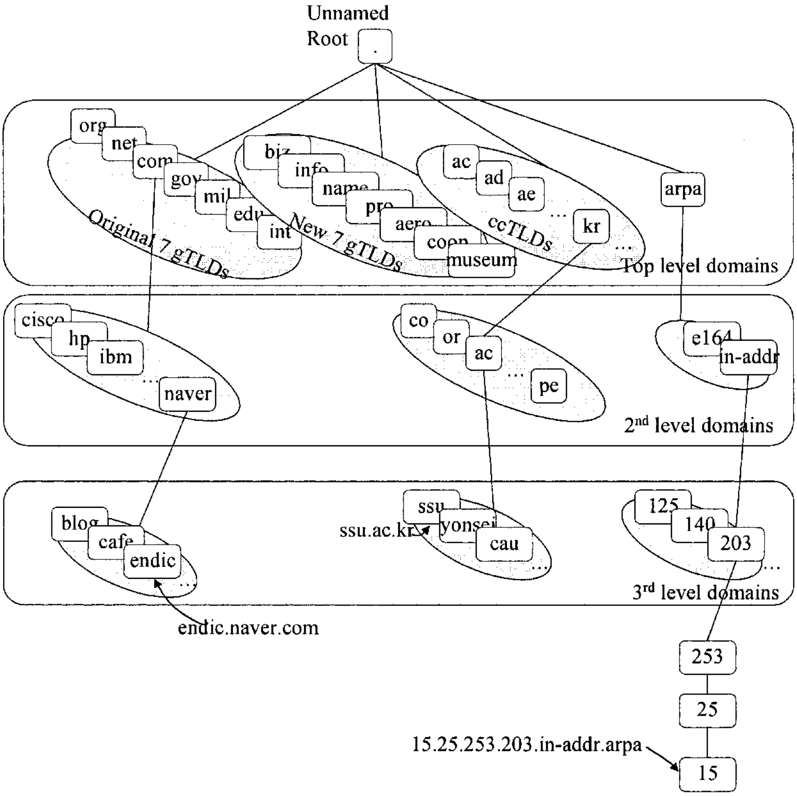


*Figure 9-1.* Hierarchical organization of DNS.

DNS name server keeps domain name information and provides mapping service in behalf of users.

- Domain name is a sequential list from the current node's level up to the unnamed root, as shown in Fig. 9-1, where every node in DNS is given with a label. Each label is limited to 63 characters. Domain name ended with dot is called an absolute domain name or fully qualified domain name (F.Q.D.N.), for example, www.hotmail.com.
- DNS name resolution is composed of several sequential query and response exchanges between name servers. DNS has been designed as distributed systems because every name server can not hold all domain name information in the world. Each name server keeps only partial domain name information, and it makes questions to another when it can not resolve queries from users.

DNS has three major components, such as domain name space with resource records, name server, and resolver. The relation between these three components is shown in Fig. 9-2.

- The domain name space is the set of whole domain names described in Fig. 9-1. The distribution of domain name space and assignment of the domain name are administrated by global registries which manages the distribution of globally unique domain names. Each domain name is associated with specific type, such as, A, AAAA, PTR, CNAME, HINFO, and MX. When a query message is generated, it always describes the desired type of resource information.
- Name server is a server program which keeps information about the zones which they belong to. A name server may cache to keep some domain information of other zones.
- Resolver is a program that sends query to the local name server and gets an answer in response to the query on behalf of clients. The resolver should be configured with at least one local name server and also have a cache to keep some domain information.

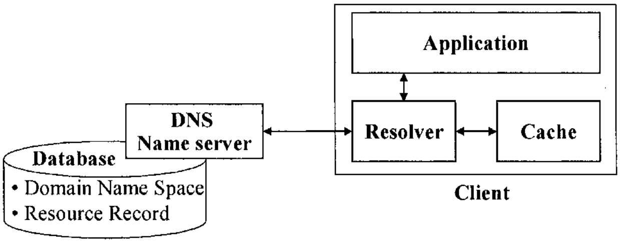
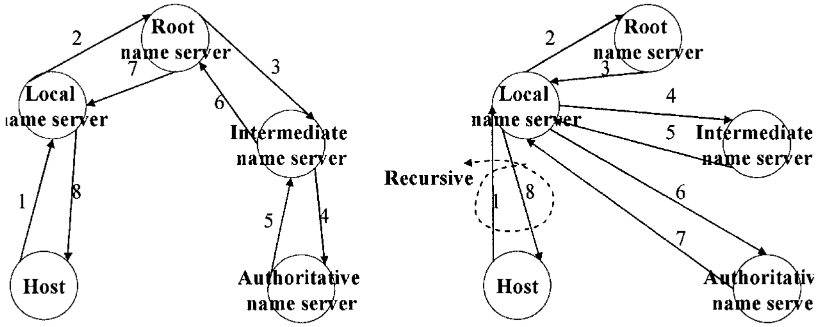


*Figure 9-2.* Simple domain name system architecture.



(a) Recursive name lookup

(b) Name lookup with mixed approaches

*Figure 9-3.* Name lookups by recursive and iterative.

A query from a client is handled by recursive, iterative or mixed way.[7]

- In recursive approach, a requested name server looks up a query for a client until it finds an appropriate answer.
- In iterative approach, once a requested name server finds the referral for a query, it stops further processing and gives the referral as an answer. Then, the client keeps querying to another server whose address is learned from the referral.
- In common, a query is handled recursively between clients and local name servers while the remaining lookup is handled iteratively. Both mechanisms are simplified in Fig. 9-3.

Depending on address delegation, management, and administration, domain and zone are defined in DNS. These two terms have somewhat similar concept, and sometimes they are used indiscriminately. However, domain and zone are obviously different from the administrator's point of view.

- Domain is the set of all domain names under the delegated domain.
- Zone is a subtree in DNS tree, which is independently administrated. After the delegation, at least one name server should be provided to a zone. A zone can be divided into smaller zones again. Once a zone is split, delegation follows, and the zone in the upper level in DNS tree does not contain domain information about split zones except referral, which is a kind of pointer to indicate zones in the lower level.

Depending on the functions for the name resolution, DNS server is categorized into three groups, such as local name server, root name server, and authoritative name server, as follows:

- Local name server: when a client queries domain name or IP address, firstly the query message is sent to a local name server. When the local name server can not find an adequate response for the query, it forwards query to the nearest root name server. To reduce loads on intermediate and root name servers, the local name server will process name resolution using iterative mechanism as shown in Fig. 9-3 (b).
- Root name server: once a root name server receives a query, and if it has records for the requested query, it sends reply messages to the local name server. In most cases, the root name server does not keep records for hostnames. Instead, the root name server gives IP address of another name server (referral) which may have information for the query. Another name server may be an intermediate name server on Internet or authoritative name server.
- Authoritative name server: a name server is authoritative for a host if it stores address information corresponding to the hostname. It responds to the query for the hostname.

In addition to categorizing name server depending on the function, we can classify name servers into two groups, such as primary and secondary name servers. When multiple name servers are offered, the primary name server has all necessary information and takes primary responsibility while secondary name servers are surplus ones. The secondary name servers obtain all information from the primary name server. This process is called zone transfer. Once some changes are made in a zone, they will be distributed to all name servers belonging to the zone. Main purposes to have multiple name servers in a zone (or zones) are load balancing. It is strongly recommended to deploy primary name servers and secondary name servers at different places.

## 9.3 DNS ARCHITECTURE

The whole picture of DNS structure is described as a 'genealogical table' without spouse originating from a single ancestor. A parent may have several children, and each child may become a parent, in turn, who has his own children.

From the analogy above, some DNS terms explained at the previous section are matched as follows:

- The genealogical table is called domain.
- Single family is called zone.
- Each family member (a node in Fig. 9-4) is given with its own name, which is called a label in DNS. Parents yield their name to children, and children yield their name to grandchildren, and so on. The domain name

is built with own name and all sequentially inherited names. Thus, domain name expresses the relative relationship with parents. Each member is given with a unique name among his siblings. It is allowed to have the same name in another family.

For example, www.icann.org, where the rightmost 'org' is a top level domain and 'icann' is the second level domain. Each character string separated by dots is called a label in DNS.

- When a child builds its new family, this new family represents subdomain.
- Each parent must assume responsibility for his family members, and it should not be handed over to another family or ancestor. This responsibility relationship in DNS is called delegation.

The zone has a slightly different meaning from the domain. The domain is the set of all domain names under the delegated domain while the zone refers to that domain, but it does not contain any delegated domain names below that zone.

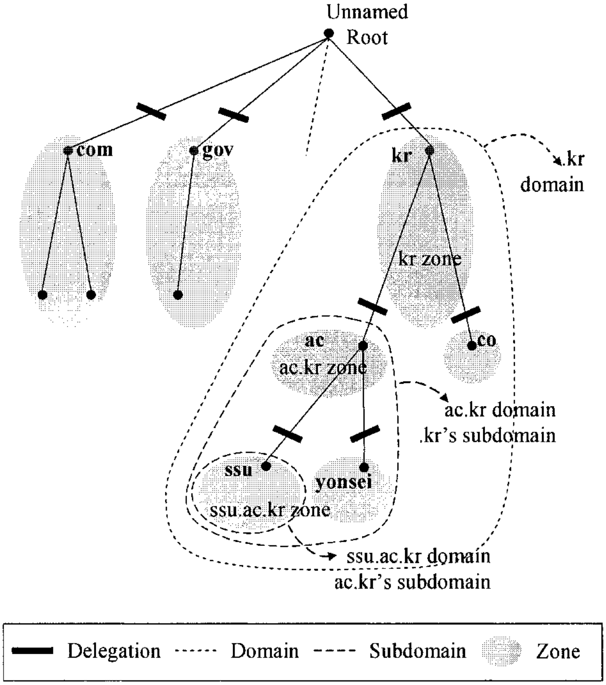


*Figure 9-4.* Example: hierarchical organization of the DNS.

Fig. 9-4 shows concepts of domain, domain name, subdomain, delegation and zone. In Fig. 9-4, the .kr domain has two subdomains: ac.kr and co.kr, using abbreviation of academy and company, respectively. Inside the ac.kr domain, 2 zones are found: ssu.ac.kr and yonsei.ac.kr, for the universities Soongsil and Yonsei. Responsibility will be delegated to subdomains, ssu.ac.kr and yonsei.ac.kr for its each zone by the upper domain, ac.kr.

Currently, there are 13 root name servers in the world, except mirroring or secondary servers. Fig. 9-5 depicts current DNS root name servers with location. Each root name server is given to one–character name, such as A, B, or C.

## 9.4 DOMAIN NAME SPACE

The DNS does not have many rules in building names. Based on the hierarchical model and inheritance as in Fig. 9-1, any label without meaning is possibly given to a node in DNS tree. Current name space can be considered in the horizontal side and vertical side.

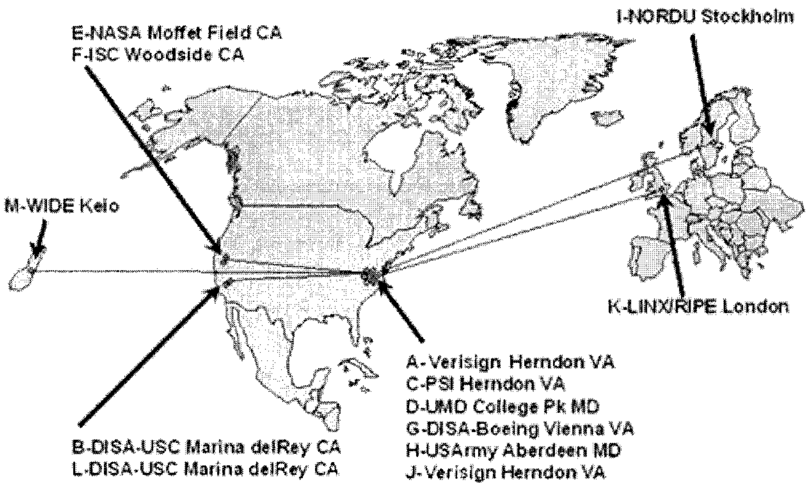


*Figure 9-5.* DNS root name servers – name, location (Feb. 1998, http://www.wia.org).

### 9.4.1 Horizontal aspect of DNS

When we consider the horizontal side of name spaces, the name space is categorized into three groups: generic top level domains, country top level domains and .arpa domain.

For the generic top level domains (gTLDs), seven gTLDs were created with three letters, such as .com, .net, .org, .gov, .mil, .edu, and .int. The first three domains (.com, .net, .org) have no restriction to be registered, while the other four are limited with some purpose, such as military or education. In 2001 and 2002, seven more gTLDs were created, such as, .biz, .info, .name, .pro, .aero, .coop, and .museum. These 14 domains are specified in Table 9-1.

The country top level domains, (ccTLDs) are made up of two letters, such as .uk, .kr, and .au. The ccTLDs have been delegated to over 240 domains, as detailed in Appendix A.

*Table 9-1.* Generic Top Level Domains (gTLDs).

| Domain | Description |
|---|---|
| Original gTLDs created in 1980s | |
| .com | For commercial organizations. |
| .net | Formerly, for network infrastructure. |
| .org | Formerly, for noncommercial organizations. |
| .gov | For government organizations. |
| .mil | For military organizations. |
| .edu | For educational organizations. |
| .int | For international organizations. |
| In 2001 and 2002 ICANN[18] includes following new domain names into gTLDs to speed up Internet expansion and reserve more domain name spaces. | |
| .biz | Registration is available. The Registry to operate this domain is NeuLevel, http://www.nic.biz. |
| .info | Registration is available. The Registry to operate this domain is Afilias, http://www.nic.name. |
| .name | Registration is available. The Registry to operate this domain is Global Name Registry, http://www.nic.info. |
| .pro | Still under negotiation. http://www.nic.pro. |
| .aero | Registration is available. This domain is sponsored by Societe Internationale de Telecommunications Aeronautiques SC (SITA), http://www.nic.aero. |
| .coop | Registration is available. The domain is sponsored by the National Cooperative Business Association (NCBA), http://www.nic.coop. |
| .museum | Registration is available. The domain is sponsored by Museum Domain Management Association (MuseDoma), http://www.nic.museum. |

[18] ICANN, Internet Corporation for Assigned Names and Numbers, is an organization to manage Internet domain name system.

Besides, one special top level domain, .arpa is used for the inverse mapping from IP addresses to domain names. The .arpa domain is administered by the Internet technical community under the Internet Architecture Board (IAB).[8] The new DNS mapping service, Telephone Number (ENUM),[19] which maps telephone numbers into domain names, is based on .arpa domain.[9]

### 9.4.2 Vertical aspect of DNS

If we consider DNS in the vertical side, it is divided into several levels, such as the top level, the second level, the third level, and so on. At the top level domain, there are 14 generic top level domains, around 240 country top level domains, and .arpa domain.

The domain name is made up of labels starting at the current node up to the unnamed root, as shown in Fig. 9-1. Each label is separated with single dot (.), such as ssu.ac.kr. This naming convention is opposite to the naming of UNIX file system, which starts from the root to the file name, and uses slash (/) to distinguish paths and file name. All domains in DNS should be named uniquely, but duplicate label may be used in different paths.

## 9.5 NAME RESOLUTION

A resolver offers an interface between application programs and DNS in client's side. The resolver searches the location where the requested resource record is administered, forwards query messages to the appropriate name server, and finally returns response message to the questioning client.

IP address of the root name server should be preconfigured at the resolver to search the entire domain database. To prevent frequent query for same domain name, the resolver may have a cache to keep queried domain information. Resource records usually have some specific lifetime which is determined by the corresponding administrative name server. After the valid lifetime, expired resource records should be deleted from the cache.

Not all hosts are required to implement resolver. Most of hosts are running as a 'stub resolver' that provides interfaces to application programs

[19] ENUM supports connectivity between applications based on completely different communication infrastructures, with contemporary Domain Name Systems. Simply, this mechanism has PSTN users communicate to other parties in no matter what environment they belong, and access to resources on Internet.

and sends query message to dedicated local name server, which is acting as a real resolver (designated resolver) when application programs request domain name information. When operating as stub resolvers, hosts do not have to learn all information or some changes about root name servers. Currently, designated resolvers are provided by ISPs.

A name server which is simply called DNS server maintains information for a zone which is a partial section of total domain name space. The name server does not keep all domain names in the Internet. Actually, there is no name server in the world to have all domain information. Instead, delegations and referrals enable to find appropriate information. For example, the root name servers do not have all information about domain, but contains referrals to the next level name servers.

Name resolution procedure is shown in Fig. 9-6.

1. A client sends a query message, sunny.ssu.ac.kr to a local name server.
2. When the local name server does not know the answer for the query message from the client, it forwards the query message to one of root name servers.
3. The root name server then gives referral (IP address of .kr name server) to the local name server.

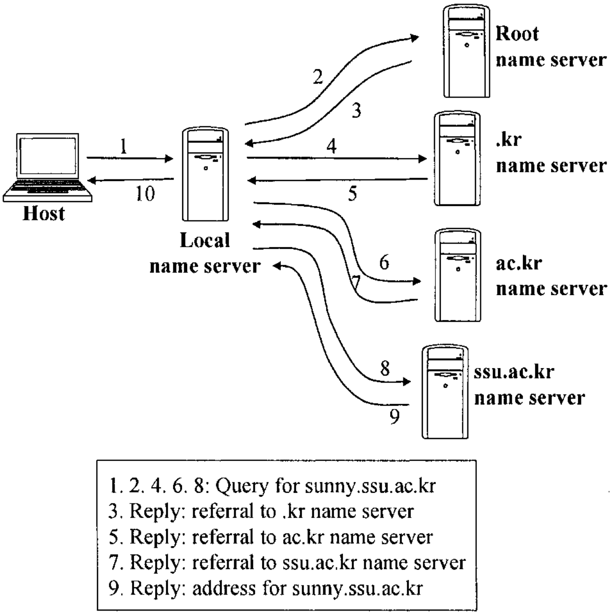

Root
name server
.kr
name server
ac.kr
name server
ssu.ac.kr
name server
Host
Local
name server
1
2
3
4
5
6
7
8
9
10
1. 2. 4. 6. 8: Query for sunny.ssu.ac.kr
3. Reply: referral to .kr name server
5. Reply: referral to ac.kr name server
7. Reply: referral to ssu.ac.kr name server
9. Reply: address for sunny.ssu.ac.kr

*Figure 9-6.* Name resolution - query chain.

4. The local name server sends the query message to the .kr name server.
5. The .kr name server gives referral (IP address of ac.kr name server) to the local name server.
6. The local name server sends the query message to the ac.kr name server.
7. The ac.kr name server gives referral (IP address of ssu.ac.kr name server) to the local name server.
8. The local name server sends the query message to the ssu.ac.kr name server.
9. Finally, the administrative name server for sunny.ssu.ac.kr gives IP address of sunny.ssu.ac.kr to the local name server.
10. Once the local name server gets IP address for sunny.ssu.ac.kr, it forwards IP address to the host.

From the client point of view, only local name server is involved for the name resolution because iterative mechanism is employed between the local name server and remaining all other name servers.

## 9.6 PACKET FORMAT

Query and Response messages exchanged between hosts and name servers are accomplished via DNS protocol. Both TCP and UDP protocols can be employed to exchange messages. Fig. 9-7 shows DNS message format contained in IP packet.[1, 10]

The DNS message is divided into 5 parts, such as DNS Header, Question, Answer, Authority, and Additional, as shown in Fig. 9-8.

- The Header is always present in a message. Detailed header format is explained in Fig. 9-9. The Header specifies message type whether the message is a query, an inverse query, or a response. Information for remaining parts of the message is also included in the Header.
- In the Question part, detailed contents for the query are contained, such as query type, the query class, and the query domain name.
- The Answer part contains resource record(s) for the query.
- The Authority part contains resource record(s) which points to the authoritative name server managing the questioned domain.
- The Additional part will contain resource record related to the query.

### 9.6.1 DNS Header

The DNS Header is composed of 6 fields, such as ID, flags, QDCOUNT, ANCOUNT, NSCOUNT, and ARCOUNT fields.

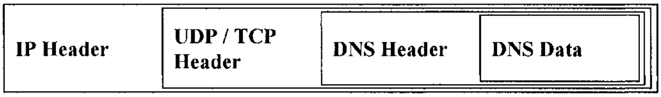


*Figure 9-7.* DNS message contained in IP packet.

| | |
|---|---|
| ID | Flags |
| QDCOUNT | ANCOUNT |
| NSCOUNT | ARCOUNT |
| Question | |
| Answer | |
| Authority | |
| Additional | |

*Figure 9-8.* DNS message format.

- The ID field is 16-bit long and assigned by an application program which triggers DNS query process. This field is copied into all response messages.
- Several flags are defined for the DNS message; QR, Opcode, AA, TC, RD, RA and RCODE flags.

  The QR flag is one-bit long and specifies whether the message is a query or a response.

  The Opcode flag is four-bit long and notifies what kind of query is contained in the message. Code values for Opcode flag are shown in Fig. 9-9.

  The AA (authoritative answer) flag is valid only in a response message. This flag notifies that the responding name server is an administrative name server for the queried domain name.

  The TC (truncation) flag is used to notify that the message is truncated. The message will be truncated when the length is greater than 512 characters.

  The RD (recursion desired) flag is set to request the recursive query processing.

The RA (recursion available) flag is set to notify whether the recursive query processing is available in the name server.

The RCODE (response code) flag is 4-bit long and used only in response messages. The values for RCODE are shown in detail in Fig. 9-9.

- The QDCOUNT, ANCOUNT, NSCOUNT, and ARCOUNT fields contain unsigned 16-bit integer value.

  The QDCOUNT specifies the number of records in the Question part of the query message.

  The ANCOUNT specifies the number of resource records in the Answer part of the response message.

  The NSCOUNT specifies the number of resource records in the Authority part of the response message.

  The ARCOUNT specifies the number of resource records in the Additional part of the response message.

## 9.6.2 Query message

The Question part in the Data portion is composed of Name, Type, and Class fields:

- The Name field specifies a domain name under name lookup.
- The Type field specifies the requested resource record type. 20 record types have been defined. Some of types are obsolete, and some values are listed in Table 9-2.
- The Class field is usually set to 1, which means internet address.

## 9.6.3 Reply message

The final three parts in DNS messages are Answer, Authority, and Additional. Three parts contain resource records for the requested query and share the same format, which is composed of Name, Type, Class, TTL, Resource Data Length, and Resource Data fields:

- The Name, Type, and Class fields have the same features as Query message.

*Table 9-2.* Resource record type for DNS

| Record type | Type value | Description |
|---|---|---|
| A | 1 | Resource record for IPv4 address. |
| NS | 2 | Resource record for name server. |
| CNAME | 5 | Resource record for canonical name. |
| PTR | 12 | Resource record for pointer record. |
| MX | 15 | Resource record for mail exchange. |

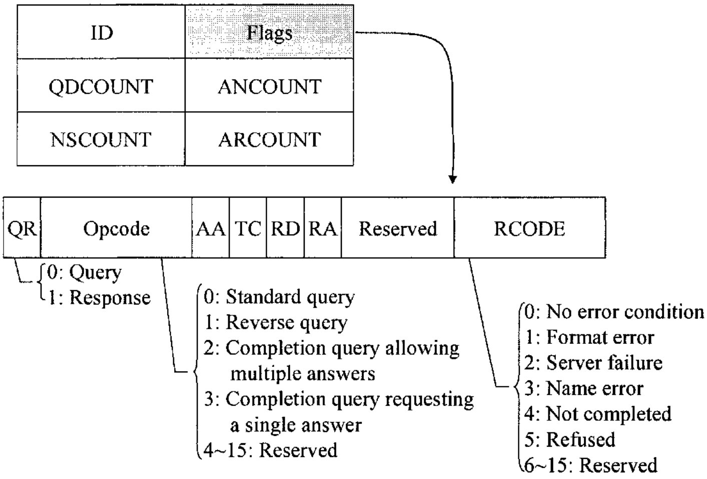


*Figure 9-9.* DNS Header format.

- The TTL (Time To Live) field specifies the valid lifetime in seconds for the resource record in the Resource Data field.
- The Resource Data Length field specifies the length of the content in the Resource Data field.
- The Resource Data field contains answers corresponding to the query. For example, when the Type field is set to 1, this field contains IP address.

## 9.7 DNS EXTENSION

The length of IPv6 address is four times longer than that of IPv4 address. Thus, name to address or address to name mapping service via DNS can not be provided for IPv6 without modification or extension. To support 128-bit IPv6 address, extended record type, AAAA record is defined.[4] AAAA record is just an extension of present A record type. Thus, most DNS devices can handle the new record type without much trouble. The A6 record type is also proposed for the effective management of name space, address aggregation and renumbering.[5] However, A6 record is used in only experimental networks because the real deployment of A6 record causes many problems because of the chains of A6 record.[5, 11, 12]

In IPv4, reverse mapping is provided via .arpa zone. For reverse mapping in IPv6, .int[4] and .arpa[5] have been reserved, where .int's root is ip6.int and .arpa's root is .ip6.arpa. Any IPv6 address registered at ip6.int domain has ip6.int in suffix, and the address is encoded by the reversed order like records for IPv4 address in reverse zone. For example, reverse lookup domain name for FF02:0:1:2:3:4:567:89ab is b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2. 0.0.0.1.0.0.0.0.0.0.0.2.0.F.F.ip6.int. Similarly, any IPv6 address registered at ip6.arpa domain has ip6.arpa in suffix.

In addition to original encoding scheme for reverse domain, RFC 2673 defines new DNS label format to increase efficiency in reverse mapping zone. This new label, bit-label is applied to .arpa domain. A serious problem occurs when DNS server gets queries for bit-label formatted name while it does not understand bit-labeled name. DNS server may think some errors are in the query and discard it. Currently, DNSEXT working group and NGTRANS working group in IETF have decided not to use bit-label format. Sequential hexadecimal number seems to be enough to express delegation of current DNS reverse tree. IETF approved to use ip6.arpa domain for reverse lookup, resulting in obsolete ip6.int.[11] Therefore, ip6.int will vanish gradually from reverse zone.

Common domain names that are already used in IPv4 are also used in IPv6 without any change because domain name is not a name structure for the specific network but a unique name structure used in Internet.

## 9.8 REQUIREMENT FOR DNS SUPPORT IN TRANSITION

When an administrative name server is requested to lookup its database, it may find resource records, such as A and AAAA together, as explained in Table 9-2. The A record is for IPv4 address, and the AAAA record is for IPv6 address. As the administrative name server builds reply message, it may put only one record type, A or AAAA, not both into the message. If both A and AAAA resource records are contained in one reply message, the order of records in the Response message may also indicate preference.[13] Resolvers or client applications choose one record type based on the preference even if both A and AAAA resource records are obtained. Then, the client application will start communications using the preferred IP protocol.

Once the resolver gets a reply from the administrative name server, it may apply filtering to the message as follows:

- Returns only A record to applications.
- Returns only AAAA record to applications.

- Returns both records to applications.

When the resolver returns only A record to applications, the application program will start IPv4 communication with another party. In the case of AAAA record, the application program will start IPv6 communication with another party. If a client gets both A and AAAA records, it may select record type it prefers.

It is recommended to register AAAA record in DNS, only when following conditions are satisfied:

- When IPv6 address is assigned to the node's interface.
- When IPv6 address is configured at the node's interface.
- When the interface is present on the link in IPv6 networks.

When an IPv6 node is isolated from IPv6 network, for example, when IPv6 node is not on the IPv6 network or IPv6 backbone, the third condition is not satisfied. In those cases, the address configured on such IPv6 node should not be registered in DNS.

In the current public Internet, there are several DNS zones which are only accessed by IPv6 networks. Mostly, they are experimental, but they are expected to be used habitually once Root name server operates for IPv6 networks. Following conditions should be satisfied to keep continuation of name spaces:

- All recursive DNS server should be IPv4-only or dual stack.
- Single DNS zone should own at least one DNS server to access IPv4.

There are several types of unicast addresses assigned to one device. Local-scoped addresses should not be registered on DNS including both forwarding and reverse trees.[14]

Registering 6to4 addresses in the forward zone in DNS does not bring additional problems. However, special care is required when we register 6to4 addresses in the reverse zone.[15] Delegation of 2.0.0.2.ip6.arpa is proposed,[16] but it may cause some problems. For example, scalability may be deteriorated because 64-bit prefix of 6to4 address contains IPv4 address. Cooperation from ISPs is necessary for such delegation. Besides, 6to4 prefix is temporary. Even if address information is deleted from the reverse zone, it may last for some time in the cache. If there is any query for the deleted address, name server may answer with wrong information.

Some translation mechanisms are required when IPv6-only node queries domain names registered only for IPv4 address. The other case, when IPv4-only node queries for the domain name registered only for IPv6 address also requires these mechanisms. Terms, such as IPv6 node, IPv6-only node and IPv4-only node, are defined in Chapter 8.

## 9.9 EXAMPLE: DNSv6 USING WINDOWS SERVER 2003

Microsoft shows that IPv6 network can be built up with entities such as IPv6 hosts running Windows operating systems, default router, and DNS server. Using Windows 2003 Server, we can configure test bed for IPv6 network with DNSv6, as shown in Fig. 9-10.[17] We need only five desktops to configure a complete IPv6 network. Each function of five systems is specified in Table 9-3.

To identify each node on link, link-local address will be used. Link-local address is generated by appending node's interface identifier to the well-known link-local prefix.

To communicate with other party on the other network segments, default router which understands IPv6 protocol should be arranged. However, if single segment is considered, default router is not necessary. Using Router Advertisement messages from on-link router, each node on link is able to build its globally unique IPv6 address, as explained in Chapter 6. DHCPv6 is not considered in this scenario.

DNS server provides mapping service from host name into IPv6 address. Once IPv6 host knows the address of DNS server, it will generate Query message for the name resolution. Then, DNS server will return AAAA resource record corresponding to the queried domain name. Domain name creation and modification can be done dynamically.[18]

We need to manually configure IPv4 address, subnet mask, default gateway's address, and DNS's address on each device. DHCP and WINS (Windows Internet Name Service) Server are not considered.

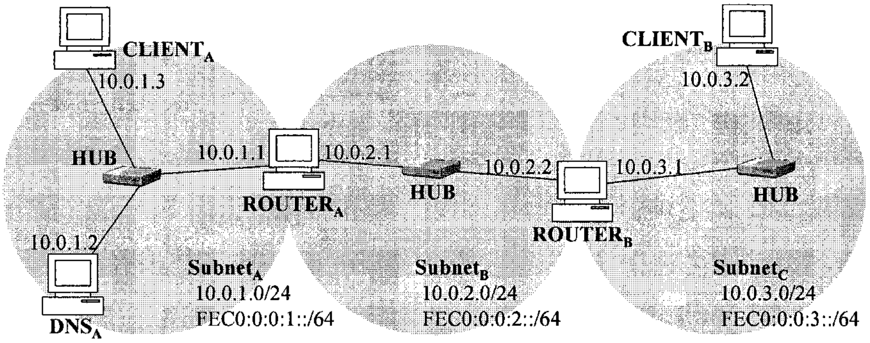


*Figure 9-10.* Test bed for DNSv6 using Windows 2003 Server.

*Table 9-3.* Components on IPv6 testbed.

| System components | |
|---|---|
| $DNS_A$ | DNS server running Windows 2003. |
| $CLIENT_A$ | Client running Windows XP professional. |
| $CLIENT_B$ | Client running Windows XP professional. |
| $ROUTER_A$ | Running Windows 2003. Router. |
| $ROUTER_B$ | Running Windows 2003. Router. |
| **Segment** | |
| $Subnet_A$ | Private IPv4 network ID is 10.0.1.0/24.<br>IPv6 site-local subnet ID is FEC0:0:0:1::/64. |
| $Subnet_B$ | Private IPv4 network ID is 10.0.1.0/24.<br>IPv6 site-local subnet ID is FEC0:0:0:1::/64. |
| $Subnet_C$ | Private IPv4 network ID is 10.0.1.0/24.<br>IPv6 site-local subnet ID is FEC0:0:0:1::/64. |

When the test bed domain is assumed as testlab.microsoft.com, DNS server, $DNS_A$, on the segment will offer DNS service. For DNS service, we need to configure $DNS_A$ as follows:

1. Install Windows 2003 Server.
2. Install or enable IPv6 protocol.
3. Install DNS server service, and set up forward lookup zone as testlab.microsoft.com.
4. Configure following information:
   IP address: 10.0.1.2.
   Subnet mask: 255.255.255.0.
   Default gateway: 10.0.1.1.

## REFERENCES

1. P. Mockapetris, Domain Names-Concepts and Facilities, RFC 1034 (November 1987).
2. P. Mockapetris, Domain Names-Implementation and Specification, RFC 1035 (November 1987.)
3. M. Daniele, B. Haberman, S. Routhier, and J. Schoenwaelder, Textual Conventions for Internet Network Addresses, RFC2851 (June 2000).
4. S. Thomson and C. Huitema, DNS Extensions to support IP version 6, RFC 1886 (December 1995).
5. M. Crawford and C. Huitema, DNS Extensions to Support IPv6 Address Aggregation and Renumbering, RFC 2874 (July 2000).
6. S. Miyakawa and R. Droms, Requirements for IPv6 Prefix Delegation, RFC 3769 (June 2004).
7. P. Albiz and C. Liu, DNS and BIND, 4th ed. (O'REILLY, April 2001).
8. ICANN, http://www.icann.org.

9. ITU-T, http://www.itu.int/ITU-T/inr/enum/.
10. W. Stevens, TCP/IP Illustrated, Volume 1: The Protocols (Addison Wesley, 1994).
11. R. Bush, A. Durand, and B. Fink, Representing Internet Protocol version 6(IPv6) Addresses in the Domain Name System (DNS), RFC 3363 (August 2002).
12. M. Crawford, Binary Labels in the Domain Name System, RFC 2673 (August 1999).
13. R. Gilligan and E. Nordmark, Transition Mechanism for IPv6 Hosts and Routers, RFC 2893 (August 2000).
14. R. Hinden and S. Deering, Internet Protocol Version 6 (IPv6) Addressing Architecture, RFC 3513 (April 2003).
15. K. Moore, 6to4 and DNS, work in progress (October 2002).
16. G. Huston, 6to4 Reverse DNS Delegation, work in progress (October 2004).
17. http://www.microsoft/technet.
18. P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136 (April 1997).
19. IANA, http://www.icana.org.

# APPENDIX A: COUNTRY-CODE TOP-LEVEL DOMAINS (CCTLDS)

Country-code top level domains (ccTLDs) are listed in Table 9-3. Information about the delegation in current ccTLDs is found in the IANA ccTLD database, especially IANA whois service (http://whois.iana.org). The ccTLD database keeps simple information for country code top-level domains, such as an organization to sponsor ccTLD, and technical and administrative contact points.[19]

*Table 9-4.* ccTLDs

| Domain | Country | Domain | Country | Domain | Country |
|---|---|---|---|---|---|
| .ac | Ascension Island | .ad | Andorra | .ae | United Arab Emirates |
| .af | Afghanistan | .ag | Antigua and Barbuda | .ai | Anguilla |
| .al | Albania | .am | Armenia | .an | Netherlands Antilles |
| .ao | Angola | .ar | Antarctica | .aq | Argentina |
| .as | American Samoa | .at | Austria | .au | Australia |
| .aw | Aruba | .ax | Aland Islands | .az | Azerbaijan |
| .ba | Bosnia and Herzegovina | .bb | Barbados | .bd | Bangladesh |
| .be | Belgium | .bf | Burkina Faso | .bg | Bulgaria |
| .bh | Bahrain | .bi | Burund | .bj | Benin |
| .bm | Bermuda | .bn | Brunei Darussalam | .bo | Bolivia |

*Table 9-4.* Cont.

| Domain | Country | Domain | Country | Domain | Country |
|---|---|---|---|---|---|
| .br | Brazil | .bs | Bahamas | .bt | Bhutan |
| .bv | Bouvet Island | .bw | Botswana | .by | Belarus |
| .bz | Belize | .ca | Canada | .cc | Cocos (Keeling) Islands |
| .cd | Congo, The Democratic Republic of the | .cf | Central African Republic | .cg | Congo, Republic of |
| .ch | Switzerland | .ci | Cote d'Ivoire | .ck | Cook Islands |
| .cl | Chile | .cm | Cameroon | .cn | China |
| .co | Colombia | .cr | Costa Rica | .cs | Serbia and Montenegro |
| .cu | Cuba | .cv | Cape Verde | .cx | Christmas Island |
| .cy | Cyprus | .cz | Czech Republic | .de | Germany |
| .dj | Djibouti | .dk | Denmark | .dm | Dominica |
| .do | Dominican Republic | .dz | Algeria | .ec | Ecuador |
| .ee | Estonia | .eg | Egypt | .eh | Western Sahara |
| .er | Eritrea | .es | Spain | .et | Ethiopia |
| .fi | Finland | .fj | Fiji | .fk | Falkland Islands (Malvinas) |
| .fm | Micronesia, Federal State of | .fo | Faroe Islands | .fr | France |
| .ga | Gabon | .gb | United Kingdom | .gd | Grenada |
| .ge | Georgia | .gf | French Guiana | .gg | Guernsey |
| .gh | Ghana | .gi | Gibraltar | .gl | Greenland |
| .gm | Gambia | .gn | Guinea | .gp | Guadeloupe |
| .gq | Equatorial Guinea | .gr | Greece | .gs | South Georgia and the South Sandwich Islands |
| .gt | Guatemala | .gu | Guam | .gw | Guinea-Bissau |
| .gy | Guyana | .hk | Hong Kong | .hm | Heard and McDonald Islands |
| .hn | Honduras | .hr | Croatia/Hrvatska | .ht | Haiti |
| .hu | Hungary | .id | Indonesia | .ie | Ireland |
| .il | Israel | .im | Isle of Man | .in | India |
| .io | British Indian Ocean Territory | .iq | Iraq | .ir | Iran, Islamic Republic of |
| .is | Iceland | .it | Italy | .je | Jersey |
| .jm | Jamaica | .jo | Jordan | .jp | Japan |
| .ke | Kenya | .kg | Kyrgyzstan | .kh | Cambodia |
| .ki | Kiribati | .km | Comoros | .kn | Saint Kitts and Nevis |
| .kp | Korea, Democratic People's Republic | .kr | Korea, Republic of | .kw | Kuwait |

*Table 9-4.* Cont.

| Domain | Country | Domain | Country | Domain | Country |
|---|---|---|---|---|---|
| .ky | Cayman Islands | .kz | Kazakhstan | .la | Lao People's Democratic Republic |
| .lb | Lebanon | .lc | Saint Lucia | .li | Liechtenstein |
| .lk | Sri Lanka | .lr | Liberi | .ls | Lesotho |
| .lt | Lithuania | .lu | Luxembourg | .lv | Latvia |
| .ly | Libyan Arab Jamahiriya | .ma | Morocco | .mc | Monaco |
| .md | Moldova, Republic of | .mg | Madagascar | .mh | Marshall Islands |
| .mk | Macedonia, The Former Yugoslav Republic of | .ml | Mali | .mm | Myanmar |
| .mn | Mongolia | .mo | Macau | .mp | Northern Mariana Islands |
| .mq | Martinique | .mr | Mauritania | .ms | Montserrat |
| .mt | Malta | .mu | Mauritius | .mv | Maldives |
| .mw | Malawi | .mx | Mexico | .my | Malaysia |
| .mz | Mozambique | .na | Namibia | .nc | New Caledonia |
| .ne | Niger | .nf | Norfolk Island | .ng | Nigeria |
| .ni | Nicaragua | .nl | Netherlands | .no | Norway |
| .np | Nepal | .nr | Nauru | .nu | Niue |
| .nz | New Zealand | .om | Oman | .pa | Panama |
| .pe | Peru | .pf | French Polynesia | .pg | Papua New Guinea |
| .ph | Philippines | .pk | Pakistan | .pl | Poland |
| .pm | Saint Pierre and Miquelon | .pn | Pitcairn Island | .pr | Puerto Rico |
| .ps | Palestinian Territory, Occupied | .pt | Portugal | .pw | Palau |
| .py | Paraguay | .qa | Qatar | .re | Reunion Island |
| .ro | Romania | .ru | Russian Federation | .rw | Rwanda |
| .sa | Saudi Arabia | .sb | Solomon Islands | .sc | Seychelles |
| .sd | Sudan | .se | Sweden | .tt | Trinidad and Tobago |
| .tv | Tuvalu | .tw | Taiwan | .tz | Tanzania |
| .ua | Ukraine | .ug | Uganda | .uk | United Kingdom |
| .um | United States Minor Outlying Islands | .us | United States | .uy | Uruguay |

*Table 9-4.* Cont.

| Domain | Country | Domain | Country | Domain | Country |
|---|---|---|---|---|---|
| .uz | Uzbekistan | .va | Holy See (Vatican City State) | .vc | Saint Vincent and the Grenadines |
| .ve | Venezuela | .vg | Virgin Islands, British | .vi | Virgin Islands, U.S. |
| .vn | Vietnam | .vu | Vanuatu | .wf | Wallis and Futuna Islands |
| .ws | Western Samoa | .ye | Yemen | .yt | Mayotte |
| .yu | Yugoslavia | .za | South Africa | .zm | Zambia |
| .zw | Zimbabwe | | | | |

Chapter 10

# MOBILITY SUPPORT FOR IPv6

## 10.1 INTRODUCTION

As the size and weight of terminals and battery become smaller and lighter, the mobile computing and communications becomes real in the every day of life. Mobility service is considered as one of the killer applications in IPv6. Agents usually are in charge of the management of node's mobility as well as address allocations. They help roamed mobile nodes to get message properly.

In IPv4, a mobile node may be identified by its home address, irrespective of the location where the mobile node currently resides. When a mobile node is located away from its home, the mobile node can not communicate with its home address because the prefix of the current link is different from the prefix of the home address. The current access router can not support the routing of packets using the home address. The mobile node should use addresses which are topologically correct. Thus, the mobile node should be allocated a new address in the visiting link. This address is called care-of address (CoA).

In Mobile IPv4 (MIPv4),[1] CoA is allocated by foreign agents which take care of the link where the mobile node currently resides. A CoA obtained in this way is called 'foreign agent CoA'. It is actually IP address assigned to one of interfaces of the foreign agent. CoA may be allocated using DHCP.[2] This CoA is called 'co-located CoA'.

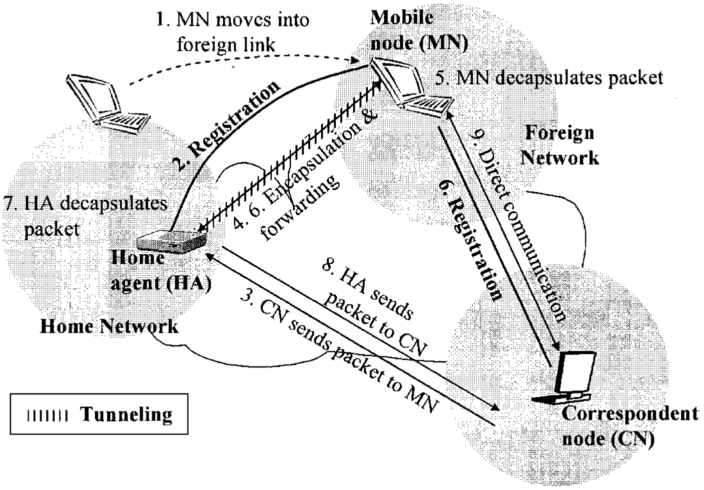


*Figure 10-1.* Outlined operation of MIPv6.

In Mobile IPv6 (MIPv6),[3] CoA may be allocated by stateless or stateful approaches. Stateful approach usually employs DHCP server. There are several ways to form CoA using stateless approach. The most common way is to form CoA using information obtained from the access router. The mobile node may use the prefix advertised from the access router to form the upper 64 bits of CoA and MAC address to form the lower 64 bits of CoA.

The home agent in the MIPv4 protocol provides mapping between care-of address and home address for mobile nodes. The home agent is an entity which resides in the home network of the mobile node. There may be multiple home agents in the home network. Whenever the mobile node gets new CoA, it must notify its home agent of new CoA. This notification process is called the registration. The home agent intercepts packets destined to the mobile node and tunnels them to the foreign agent if it exists. If the foreign agent does not exist, the tunnel must end at the mobile node. Thus, the mobile node must have the decapsulation capability.

MIPv6 adopts a large portion of MIPv4 and further employs many benefits of IPv6. Outlined operation of MIPv6 is shown in Fig. 10-1. The outstanding changes from MIPv4 are as follows:

- Foreign agents are not necessary in MIPv6. Mobile nodes can form CoA using IPv6 stateless autoconfiguration protocols.
- Route optimization is possible in MIPv6. Devising security mechanisms for the route optimization are one of main concerns of MIPv6. Various schemes have been proposed for the authentication of mobile nodes

without needing supports of security infrastructures such as Authentication, Authorization and Accounting (AAA)[4] and public key infrastructures. Return routability (RR) procedure is selected as the mandatory scheme for the authentication of the mobile node to the correspondent nodes. An authentication mechanism using Cryptographically Generated Address (CGA) may be used if the more strong security than RR is needed. Security between the home agent and the mobile node should be provided by IP Security (IPsec).[5] However, some mobile nodes do not have IPsec capability. In some cases, IPsec may not be desirable for small mobile terminals because processing IPsec may overwhelm the processing power of those mobile terminals. Thus, security mechanisms for mobile terminals which do not have IPsec capability are recently considered. Using AAA is recently considered in this case.

- Instead of ARP in MIPv4, MIPv6 uses IPv6 Neighbor Discovery protocol,[6] which allows nodes on link to be independent on link layer protocols.
- Routing Header should be used to send packets from correspondent nodes to mobile nodes. For the MIPv6, Type 2 Routing Header should be used.

## 10.2 TERMINOLOGY AND CONCEPT

Following terminologies and concepts are needed to understand the mobility support mechanism in IPv6.

### 10.2.1 Communication entities

There are three communication entities in MIPv6, such as mobile node, home agent, and correspondent node.

- Mobile node is a node which may change its location where it is attached from the home link to the other links. Visited links are called foreign links.
- Home agent is an entity which exists on the mobile node's home link and has mobile node's current location information. When a mobile node is away from its home network, the home agent intercepts packets destined to the mobile node and tunnels them to the mobile node's registered CoA. If the route optimization is employed, correspondent nodes may have direct communications with mobile nodes without tunneling by home agents. Home agent is usually implemented in the router.

- Correspondent node is a peer node to communicate with the mobile node. It may be mobile or fixed node.

### 10.2.2 Address types

Two types of address are defined, namely home address and care-of address:

- Home address is a unicast address dedicated to a mobile node. Multiple home addresses may be assigned to the mobile node if more than one subnet prefix are defined in the home network.
- Care-of address is a unicast address allocated to a mobile node when it visits a foreign network. A mobile node may be associated with more than one care-of address. The registered address at the mobile node's home agent is called primary care-of address.

### 10.2.3 Handover types

Handover in the mobile communications usually means the change of frequency channels and/or CDMA codes. The necessary action occurs in the layer 1 and 2. However, for the mobile terminals which are equipped with IP, another kind of handover must be considered. CoA changes when the mobile node changes the subnet. This is called the layer 3 handover (L3 handover). L2 and L3 handovers are independent processes. L2 handover in wireless LAN is specified by Institute of Electrical and Electronics Engineers (IEEE). L3 handover is specified by IETF. To reduce overall handover latency, L2 handover and L3 handover have to be incorporated. Fast Handover[7] in MIPv6 as well as Low Latency Handover in MIPv4 tries to improve the overall latency by considering this incorporation between two types of handover.

- L2 handover is a process which a mobile node changes its physical link-layer connection to another.[8] When a mobile node moves to a new Access Point (AP), L2 handover takes place.
- L3 handover usually follows L2 handover. In L3 handover, a mobile node identifies that it moves to new link layer where new subnet prefix is used. The mobile node will change its primary care-of address to new one. As a mobile node moves, change of AP followed by the change of the subnet leads to L3 handover.

For the route optimization, correspondent nodes as well as the home agent are required to keep binding information which is the association of mobile node's home address and primary care-of address at the current position. Binding updates process to home agents and correspondent nodes are explained in Sections 4 and 5.

### 10.2.4 Message types

A message including a Mobility Header after IPv6 Header is called mobility message, which is currently defined in MIPv6.[3] Mobility messages are as follows:

- Home Test Init (HoTI)
- Home Test (HoT)
- Care-of Test Init (CoTI)
- Care-of Test (CoT)
- Binding Update (BU)
- Binding Acknowledgement (BA)
- Binding Refresh Request (BRR)
- Binding Error (BE).

The first four messages are used in the return routability procedure between correspondent nodes and mobile nodes. Return routability procedure is to authenticate the mobile node to the correspondent node before sending binding update messages. The remaining four messages are used in the binding update procedure.

### 10.2.5 Route optimization

Communications between correspondent nodes and mobile nodes is provided via 'bi-directional tunneling' between home agents and mobile nodes or 'route optimization'. When no Binding Entry for the mobile node is found in correspondent nodes, bi-directional tunneling is used between two parties. These two mechanisms are explained in Sections 4 and 5.

### 10.2.6 Databases defined in MIPv6

- Binding Cache: whenever binding updates to home agent and correspondent nodes are completed, a Binding Cache Entry for the mobile node is created or modified in Binding Cache.
- Binding Update List: whenever a mobile node performs binding update to a correspondent node, a new entry is created or updated for the correspondent node in the mobile node's Binding Update List.
- Home Agent List: a router working as a home agent in the link maintains a Home Agent List recording all home agents on that link.

## 10.3 PROTOCOL OVERVIEW OF MIPv6

The MIPv6 operation seems to be very similar to MIPv4 operation, but it provides more effective and optimized way in terms of the performance and security.

When a mobile node resides on its home link, it may be regarded as a wired node on link, and any packet from the node will be handled using general routing protocols. However, it becomes a different story when the mobile node moves to a foreign link.

When the mobile node moves to a foreign link, it should be identified using care-of address as well as home address. The mobile node is able to obtain care-of address using current IPv6 address configuration mechanisms, such as stateless or stateful address configuration protocols. Once the mobile node gets care-of address, it is required to register its current address to one of routers in its home link and requests the router to be a home agent.

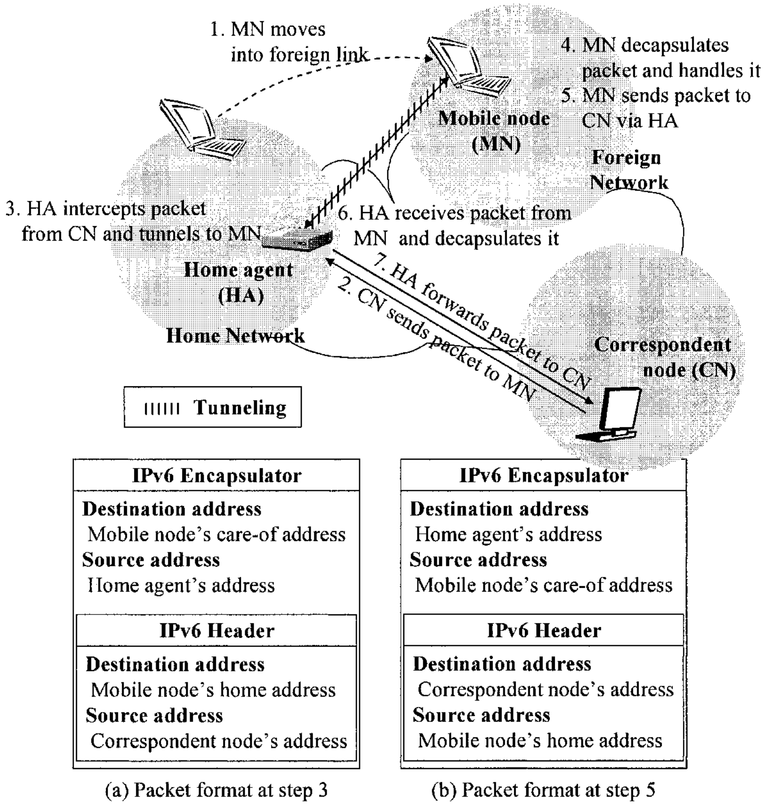
Whenever the mobile node moves to another foreign link, it has to register its current location to the home agent. The home agent will update mobile node's new location. The association between mobile node's home address and care-of address is called binding.

In MIPv6, optimized routing is set to default, but non-optimized one may be used when any one of communication parties does not understand any part of optimization. To use the optimized path, correspondent registrations should be performed before packet exchanges. Home registration is the process which a mobile node notifies its home agent of its movement with current location information. Correspondent registration does the similar thing to the correspondent nodes.

Once the correspondent registration is done successfully, mobile node's care-of address is registered at the correspondent node's Binding Cache Entry. The mobile node also adds the address of correspondent node on its Binding Update List, as shown in Table 10-1. Whenever the mobile node changes its location, it should perform correspondent registration with correspondent nodes listed in the Binding Update List.

### 10.3.1 Communication over non-optimized path

Packets are delivered over non-optimized path if either one of communicating peers does not support route optimization. Even if both communicating peers support route optimization, it may happens if a mobile node does not have valid information on its Binding Update List for a correspondent node or the correspondent node does not know the mobile node's current location information.



| IPv6 Encapsulator |
|---|
| **Destination address**<br>Mobile node's care-of address<br>**Source address**<br>Home agent's address |
| **IPv6 Header** |
| **Destination address**<br>Mobile node's home address<br>**Source address**<br>Correspondent node's address |

(a) Packet format at step 3

| IPv6 Encapsulator |
|---|
| **Destination address**<br>Home agent's address<br>**Source address**<br>Mobile node's care-of address |
| **IPv6 Header** |
| **Destination address**<br>Correspondent node's address<br>**Source address**<br>Mobile node's home address |

(b) Packet format at step 5

*Figure 10-2.* Bidirectional tunneling when no caching entry is present in correspondent nodes.

When a correspondent node wants to send packets to IPv6 nodes, it searches its Binding Cache at first. If no valid information is found, packets are exchanged via home agents, as shown in Fig. 10-2.

In detail, communications when a mobile node is away from its home is handled as follows:

1. A correspondent node sends a packet to the mobile node whose Destination Address field is set to the mobile the node's home address.
2. The home agent for the mobile node intercepts it and encapsulates it with the Destination Address field set to mobile node's care-of address.
3. As the mobile node receives the encapsulated packet, it decapsulates and obtains the original packet.

4. Now, the mobile node realizes that the correspondent node does not cache binding information or route optimization is not supported by the correspondent node. Under all circumstances, the mobile node sends reply packet whose source address is set to the mobile node's home address and starts binding update process which is explained in the next section.
5. The mobile node encapsulates the reply packet, where the Destination Address field is set to the address of the mobile node's home agent and Source Address field is set to the mobile node's care-of address. This encapsulation is called the reverse tunneling.

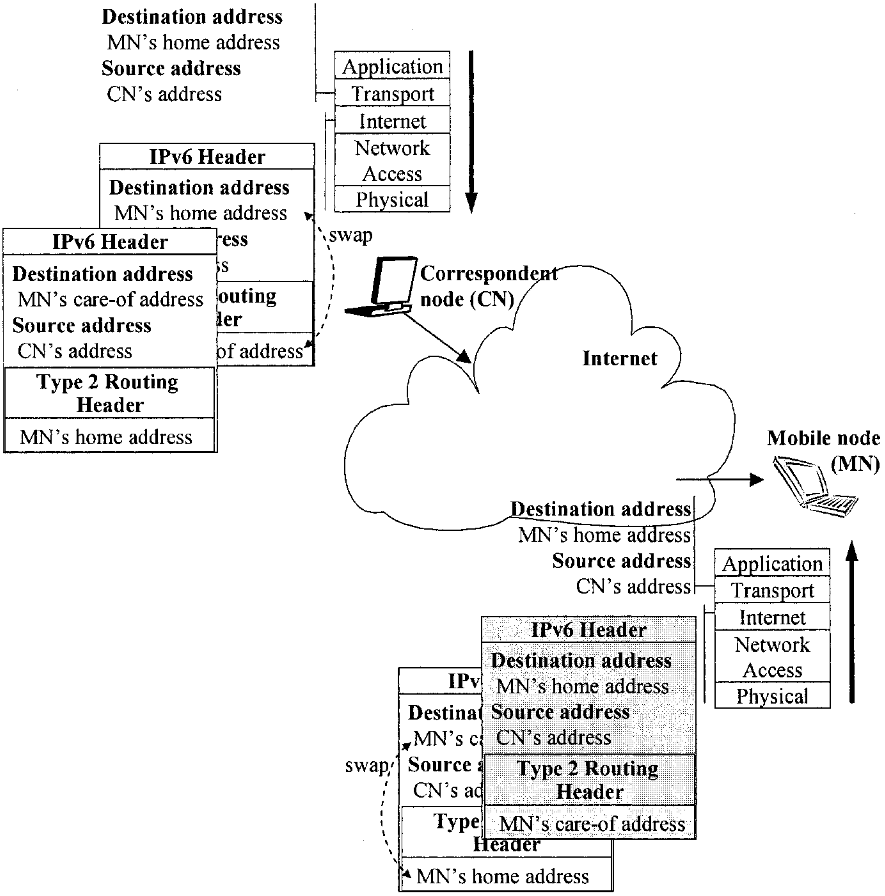


*Figure 10-3.* Packet flow when a correspondent node sends packet to a mobile node with route optimization.

*Table 10-1.* Example for Binding Update List in mobile node.

| Binding update list | |
|---|---|
| ...... | ...... |
| Source address | Mobile node's home address. |
| Destination address | Correspondent node's address. |
| Care-of address | One of current care-of address should be present. |
| Binding state | The binding is successfully done. |
| Life time | Remaining life time should be bigger than zero. |
| ...... | ...... |

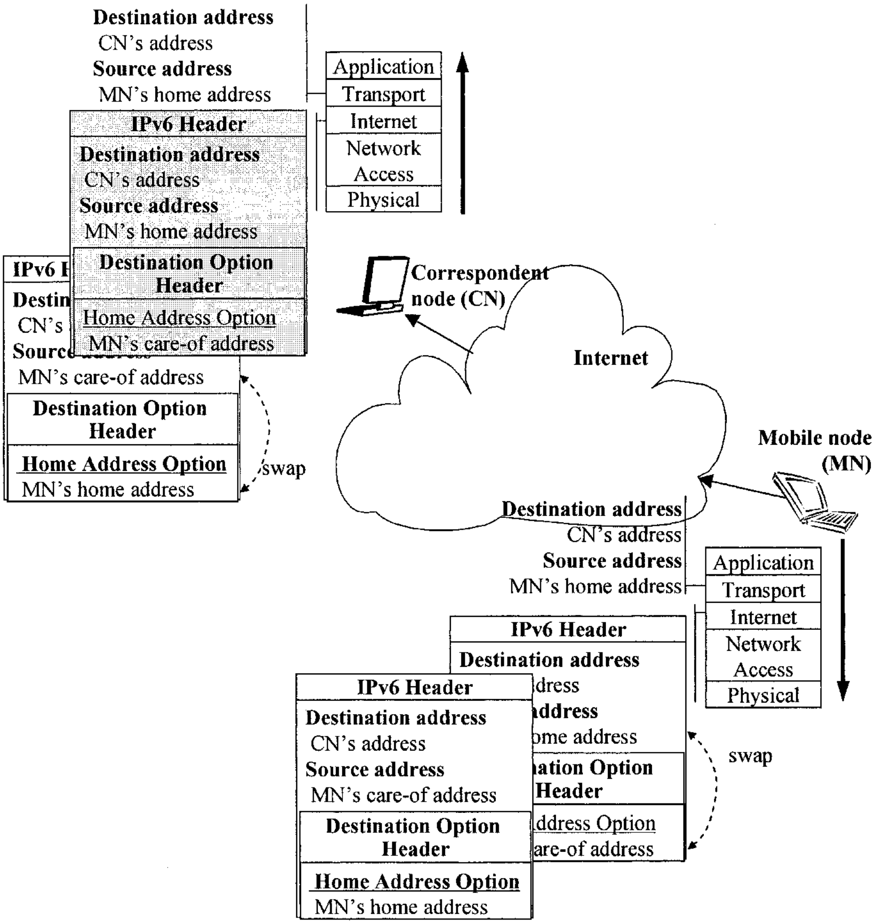


*Figure 10-4.* Packet flow when a mobile node sends packet to a correspondent node with route optimization.

6. The home agent then decapsulates and forwards the packet to correspondent nodes using the normal IP routing mechanism.
7. If the binding update to correspondent nodes will be successfully performed, direct communications will be possible between the mobile node and correspondent nodes. This binding update procedure is specified in the following section. Otherwise, routing over non-optimized path is continuously used between them, as shown in Fig. 10-2.

## 10.3.2 Communication over optimized path

### 10.3.2.1 When a correspondent node starts to send packets to a mobile node on the foreign link

When a correspondent node sends packets to the IPv6 node, it searches its Binding Cache at first. If valid cached address information mapped to the destination node is found, then the correspondent node sends packets to the destination with Type 2 Routing Header which contains mobile node's home address.

In detail, upper layers over the internet layer only knows mobile node's home address as an identifier for the mobile node because address changes are not notified to transport and application layers. Once internet layer receives data from the transport layer, it builds packets as shown in the left of Fig. 10-3:

- The Destination Address field in IPv6 Header is set to the mobile node's care-of address.
- Mobile node's home address is contained in Type 2 Routing Header.

Once the mobile node receives any packet with Type 2 Routing Header, it swaps Destination Address field and the address in the Type 2 Routing Header and hands it to the upper layer protocol. Swapping addresses prevents the session disruption in TCP connections for the mobile node's roaming.[9]

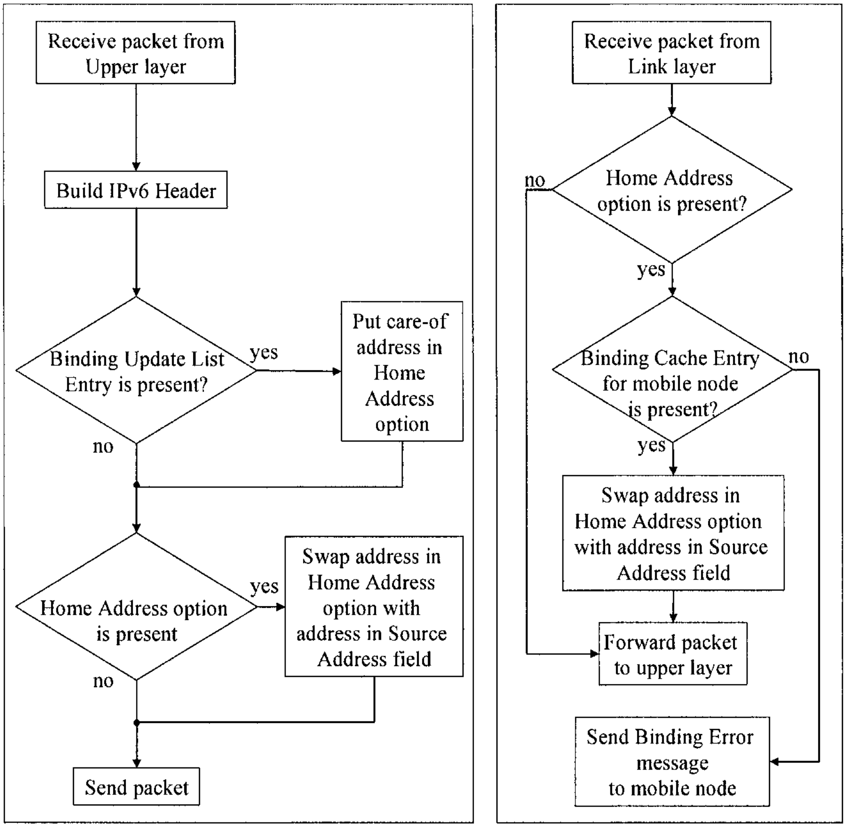
### 10.3.2.2 When a mobile node away from the home sends packets to a correspondent node

Similarly, when a mobile node sends packets directly to correspondent nodes, it adds IPv6 Home Address option to packets. Packet format generated by the mobile node is shown in Fig. 10-4.

- Mobile node's home address is contained in Home Address option to let correspondent nodes to swap the care-of address in Source Address field to the mobile node's home address before giving packets to the upper layer. The Home Address option is specified in Section 7.

- The packet flow between a mobile node and a correspondent node is shown in Fig. 10-4.
- Algorithms to send packets by a mobile node and receive them by a correspondent node are also drawn in Fig. 10-5.

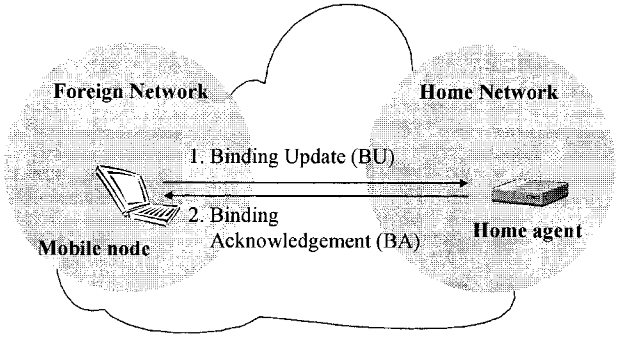
Communications through the direct path between the mobile node and correspondent nodes is desirable in terms of the performance. Latency and burden on the home agents can be reduced.


Receive packet from Upper layer
Build IPv6 Header
Binding Update List Entry is present?
yes
no
Put care-of address in Home Address option
Home Address option is present
yes
no
Swap address in Home Address option with address in Source Address field
Send packet
Receive packet from Link layer
Home Address option is present?
no
yes
Binding Cache Entry for mobile node is present?
no
yes
Swap address in Home Address option with address in Source Address field
Forward packet to upper layer
Send Binding Error message to mobile node

(a) Algorithm when a mobile node sends packet to a correspondent node

(b) Algorithm when a correspondent node receives packet to a mobile node

*Figure 10-5.* Sending algorithm by a mobile node.



| **IPv6 Header** |
|---|
| **Destination address**<br>Home agent's address<br>**Source address**<br>Mobile node's care-of address |
| **Destination Option Header** |
| Home Address Destination option<br>Mobile node's home address |
| **BU Header** |
| Sequence number |

(a) Binding Update message

| **IPv6 Header** |
|---|
| **Destination address**<br>Mobile node's are-of address<br>**Source address**<br>Home agent's address |
| **Type 2 Routing Header** |
| Mobile node's home address |
| **BA Header** |
| Sequence number |

(b) Binding Acknowledgement message

*Figure 10-6.* Binding update to home agents.

## 10.4 BINDING UPDATE TO THE HOME AGENT

Binding Update and Binding Acknowledgement messages are also used for the binding update process to the home agent. However, binding update to the home agents must use IPsec for the authentication and the integrity checking of these messages. Thus, IPsec should be implemented in both of the mobile node and the home agent. Both two communication entities should use Encapsulating Security Payload Header (ESP),[10] and further non-NULL payload authentication algorithm should be used to ensure robust security. Besides, Authentication Header (AH) can be employed. Binding update process is drawn in Fig. 10-6.

### 10.4.1 Registration of primary care-of address

When a home agent receives a Binding Update message, it should verify and authenticate the message. Following tests are necessary for the binding update, as shown in Fig. 10-7:

1. If the home agent function is not implemented in a receiving node, it should reject Binding Update messages and return Binding Acknowledgement messages, where the State field is set to 131 to indicate that home registration is not supported.

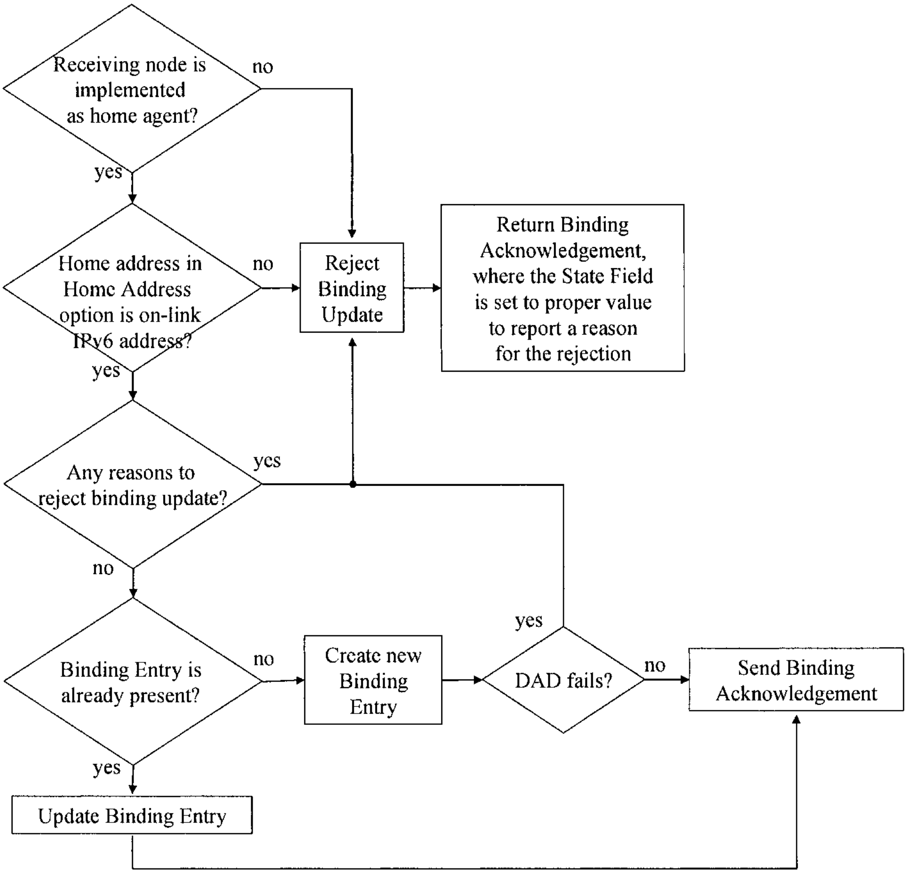

Receiving node is implemented as home agent?
no
yes
Home address in Home Address option is on-link IPv6 address?
no
yes
Reject Binding Update
Return Binding Acknowledgement, where the State Field is set to proper value to report a reason for the rejection
Any reasons to reject binding update?
yes
no
Binding Entry is already present?
no
yes
Create new Binding Entry
yes
DAD fails?
no
Send Binding Acknowledgement
Update Binding Entry

*Figure 10-7.* Home registration algorithm.

2. If the home address in the Home Address option of Binding Update message is not on-link IPv6 address, i.e., the home address does not have home agent's current prefix (or prefixes), then the home agent should reject the Binding Update message and return a Binding Acknowledgement message, where the State field is set to 132 to indicate the invalid home subnet prefix.
3. If any other reasons are found for the rejection of the Binding Update message such as non-sufficient address resource, then the home agent should reject the Binding Update message and return a Binding Acknowledgement message, where the State field is set to the proper value to indicate a reason for the rejection. For the State field, some values are reserved, as shown in Table 10-3.
4. Otherwise, the home agent records the mobile node's new primary care-of address into the Binding Cache and sends back a Binding Acknowledgement message. Once the home agent accepts the Binding Update message, it should create Binding Cache Entry in its Binding Cache. If the cache entry for the mobile node already exists, the home agent will just update Binding Entry. In case of new entry, the home agent should perform Duplication Address Detection (DAD), explained in Chapter 6, to validate whether the home address in Home Address option of Binding Update message is already allocated to other nodes on the home link.
5. If DAD fails, then the home agent should reject the Binding Update message and return the Binding Acknowledgement message, where the State field is set to 134 indicating that duplication address detection failed. If DAD is successfully completed, the home agent builds the Binding Acknowledgement message to the mobile node.

The home agent should return the Binding Acknowledgement message to the mobile node even if *A* (Acknowledge) bit of Binding Update message from the mobile node is set to 0. The Binding Acknowledgement message format is shown in Fig. 10-20. Each field of this message is as follows.

- The Status field should be set to either 1 indicating 'Binding Update is accepted but prefix discovery is necessary', or 0 indicating 'Binding Update is completely accepted'. For instance, if the subnet prefix of the home address in Home Address option is deprecated or will be deprecated, the Status field is set to 1.
- The *K* (Key Management Mobility Capability) bit is set if *K* bit in the Binding Update message from the mobile node was set, or IPsec security associations between the mobile node and the home agent are dynamically performed. Otherwise, the *K* bit should be cleared.
- The sequence number from the Binding Update message is copied to the Sequence Number field.

- The Lifetime field is set to the remaining lifetime of the address registered at the home agent's Binding Cache.

Unless the lifetime of the Binding Cache Entry for the mobile node is expired, the home agent should guarantee the uniqueness of the mobile node's home address, and the binding entry should not be removed from the home agent.

### 10.4.2 De-registration of primary care-of address

When a mobile node returns to its home network, or it realizes that no care-of address is provided to the mobile node in the visited network any longer, it should de-register its registered care-of address in its home agent. The binding update process for de-registration should be also verified and authorized like the registration.

If the home agent has no binding entry for the mobile node, it should reject the Binding Update for de-registration and return the Binding Acknowledgement message, where the State field is set to 133 to indicate that 'not home agent for this mobile node'.

Once the home agent accepts the Binding Update for de-registration, then it deletes existing entry in the Binding Cache for the mobile node and sends Binding Acknowledgement to the mobile node.

Contents of the Binding Acknowledgement message for de-registration in detail are as follows:

- The Status field should be set to 0, indicating 'Binding Update is completely accepted'.
- The *K* bit is set, if *K* bit in the Binding Update message from the mobile node was set, or IPsec security associations between the mobile node and the home agent are dynamically performed. Otherwise, the *K* bit should be cleared. Now, mobile node's home address is regarded and used as its care-of address for the key management.
- The sequence number from the Binding Update message is copied to the Sequence Number field.
- The Lifetime field is set to zero.

Now, the home agent stops intercepting packets destined to the mobile node's home address.

## 10.5 BINDING UPDATE TO CORRESPONDENT NODES

Correspondent registration is a registration between a mobile node and a correspondent node, and it is achieved via two steps; return routability procedure and binding update process. Strictly speaking, correspondent registration procedure is performed after return routability procedure, but here ‘correspondent registration’ is used as a unified terminology for the simplicity.

While authentication infrastructure between a mobile node and its home agent is required for home registration, return routability process is used to verify that the sender is a proper mobile user. Return routability process does not protect communication parties from attackers who are present between them; however, it is very effective to preclude counterfeit Binding Update message from non-right users and to limit attackers who lie on a specific path.

Keyed-hash algorithm is used in MIPv6 to protect integrity and authenticity of the Binding Update messages. The key for the algorithm, called binding management key (Kbm), is obtained by data exchange during the return routability process. Besides, node keys, Nonces, Cookies, Tokens, and cryptographic functions are used for the return routability. Once return routability procedure finishes successfully, binding update process starts. These sequential procedures are shown in Fig. 10-8 and 10-9.

## 10.5.1 Return routability

Four messages, such as Home Test Init (HoTI), Home Test (HoT), Care-of Test Init (CoTI), and Care-of Test (CoT), are used in the return routability procedure. Home Test Init and Care-of Test Init messages are sent simultaneously from a mobile node to a correspondent node. As the response of former two messages, Home Test and Care-of Test messages are generated by the correspondent node and delivered to the mobile node. Processing time to process these four messages is negligible.

### 10.5.1.1 Home Test Init and Care-of Test Init messages

The mobile node sends Home Test Init message to the correspondent node via the home agent. The reason to send this message is to acquire ‘Home Keygen Token’. Home Test Init message contains Home Init Cookie, which is 64-bit random value generated by the mobile node whenever it sends Home Test Init message. The mobile node should remember the Cookie because the correspondent node will return it later.

As sending Home Test Init message, the mobile node sends Care-of Test Init message directly to the correspondent node. The reason to send this message is to acquire Care-of Keygen Token. Care-of Test Init message contains Care-of Init Cookie, which is 64-bit random value generated by the mobile node whenever it sends Care-of Test Init message. The mobile node should remember the Care-of Init Cookie because the correspondent node will return it later.

Home Test Init message and Care-of Test Init message contains Home Init Cookie and Care-of Init Cookie, respectively. Cookies are used to verify response messages from receivers, correspondent nodes. Cookies should be newly generated whenever it is inserted into messages.

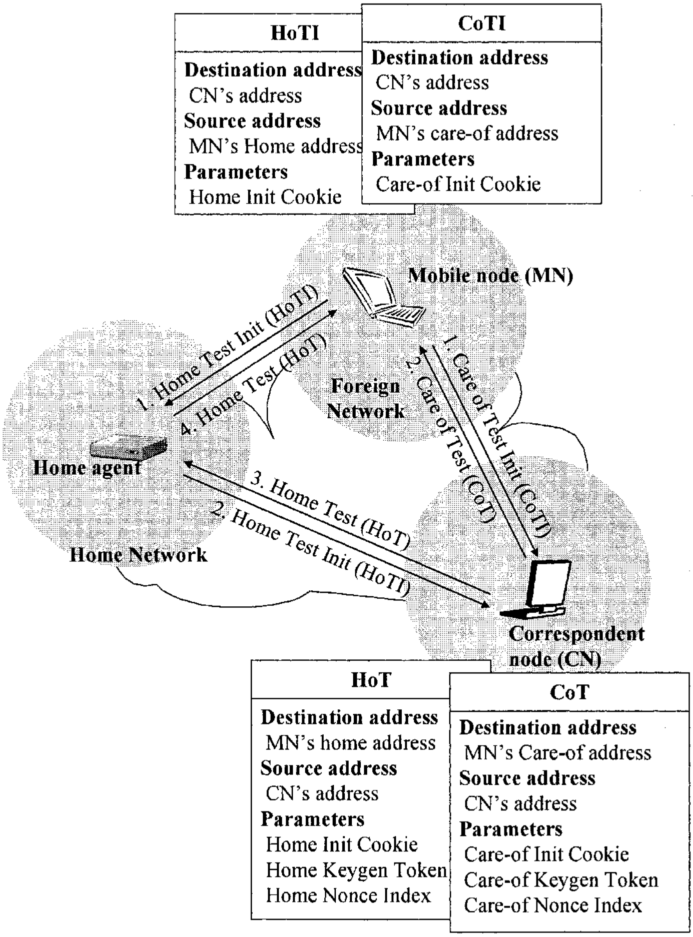


*Figure 10-8.* Return routability procedure.

### 10.5.1.2 Home Test and Care-of Test messages

In response to Home Test Init message, the correspondent node sends back Home Test message to the mobile node via the mobile node's home agent. Three parameters, Home Init Cookie, Home Keygen Token, and Home Nonce Index are carried in this message. The Home Init Cookie will be returned to the mobile node using Home Test message to confirm that Home Test message is originated from the node which received Home Test Init message from the mobile node.

In addition to sending Home Test message, the correspondent node sends back Care-of Test message directly to the mobile node in response to a Care-of Test Init message.

Once the correspondent node receives the Home Test Init message, it will build a 64-bit Home Keygen Token with secret node key (Kcn), home address, and Nonce. 64-bit Care-of Keygen Token is also generated when the correspondent node receives the Care-of Test Init message. Eq. (1) specifies an algorithm to build Keygen Token, where | denotes concatenation.

$$\begin{aligned} Keygen\ Token = {} & first\ 64\ bytes\ from\{HMAC\_SHA1 \\ & (kcn, (address \mid Nonce \mid 1\ byte\ 0\ or\ 1))\} \end{aligned} \tag{1}$$

Home Keygen Token is obtained as follows:

1. At first, the home address, Nonce, and 1-byte 0s are concatenated together in order. The last 0s are used to distinguish it from Care-of Cookies.
2. Now, node key and concatenated number are inputted into HMAC_SHA1(*K*,*m*), which denotes message '*m*' with key *K*.[11]
3. Output from HMAC_SHA1 is called Message Authentication Codes (MACs). The first 64 bytes from MACs is Home Keygen Token.
4. To build Care-of Keygen Token, care-of address, Nonce, and 1 byte-1s are used instead of home address and 1-byte 0s, respectively.

Once the mobile node gets both Home and Care-of Test messages, return routability procedure is over. Now, the mobile node creates 20-byte binding key (Kbm) using Home and Care-of Keygen Tokens earned from return routability procedure. Two Tokens are concatenated together and hashed together by SHA1(),[11] as shown in Eq. (2).

$$Kbm = SHA1(Home\ Keygen\ Token \mid Care - of\ \ Keygen\ Token) \quad (2)$$

### 10.5.1.3 Nonce and node key

The node key (Kcn) is 20-byte random number generated by the correspondent node. This key should not be shared with other nodes. This key helps the correspondent node verify that Keygen Token used to build Kbm by the mobile node originally belongs to the correspondent node. A correspondent node may generate new node key at any time, which removes the necessity of the secure key storage.

The Nonce is a random number generated at regular intervals by individual correspondent node. The recommended length for Nonce is 8 bytes. When a new Nonce is generated, it is indexed. Incremental order by the time when a Nonce is generated may be used. Nonce index helps a correspondent node find specific Nonce used to create a Keygen Token. Home and Care-of Nonce indices can be the same in the Home and Care-of Test messages.

## 10.5.2 Binding update

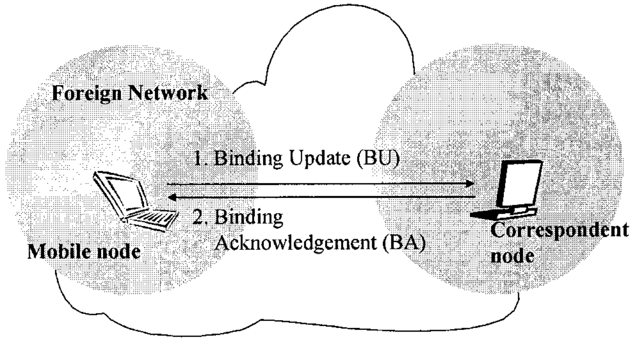
Once return routability is successfully completed, the mobile node builds a binding management key (Kbm). Kbm is created from the Keygen Tokens, as specified in Section 5.1.2, and used to authenticate Binding Update message.

In Binding Update messages, several parameters are included, such as

- Home address within the Home Address Destination option
- Sequence number within the Binding Update Header
- Home and Care-of Nonce indices within the Nonce Indices option
- The first 96 bytes from MACs, which is called Authenticator, as described in Eq. (3), where input key is Kbm, and input message is the concatenation of care-of address, correspondent node's address, and Binding Update message. In Eq. (3), CN and | denotes correspondent node and concatenation, respectively.

$$\begin{aligned} Authenticator = first\ 96\ bytes\ from\{&HMAC\_SHA1\ (Kbm, \\ &(care - of\ \ address \mid CN's\ address \\ &\mid Binding\ message))\} \end{aligned} \quad (3)$$

Once the correspondent node verifies the MAC in the Binding Update message, it creates an entry in Binding Cache for the mobile node.



| | |
|---|---|
| **IPv6 Header** | **IPv6 Header** |
| **Destination address**<br>Correspondent node's address<br>**Source address**<br>Mobile node's care-of address | **Destination address**<br>Mobile node's care-of address<br>**Source address**<br>Correspondent node's address |
| **Destination Option Header** | **Type 2 Routing Header** |
| Home Address Destination option<br>Mobile node's home address | Mobile node's home address |
| **BU Header** | **BA Header** |
| Sequence number | Sequence number |
| **Message Data of BU** | **Message Data of BA** |
| Nonce Indices option<br>Home Nonce Index<br>Care-of Nonce Index<br>Binding Authorization Data option<br>Authenticator | Binding Authorization Data option<br>Authenticator |

(a) Binding Update message

(b) Binding Acknowledgement message

*Figure 10-9.* Binding update after return routability.

When the correspondent node receives a Binding Update message during the correspondent registration process, it should reply with a Binding Acknowledgment message, which contains the same sequence number copied from Binding Update message and the first 96 bytes from MACs, where input key is Kbm, and input message is the concatenation of care-of address, correspondent node's address, and Binding Acknowledgement message.

Binding Update messages are also used when the mobile node wants to delete previously established binding in the correspondent node. In this case, Kbm is generated with only Home Keygen Token as Eq. (4).

$$Kbm = SHA1(Home\ Keygen\ Token) \tag{4}$$

# 10.6 PREFIX MANAGEMENT

## 10.6.1 Prefix solicitation

As the lifetime of the mobile node's home address becomes expired, the mobile node sends a Mobile Prefix Solicitation message to its home agent to update or to get fresh prefix information. The Mobile Prefix Solicitation message is very similar to the Router Solicitation message in Neighbor Discovery protocol.[6] Only difference between them is that Mobile Prefix Solicitation message is originated by the mobile node in a foreign link and delivered to its home agent while Router Solicitation message is routed directly to the router. The important points on building the Mobile Prefix Solicitation message is as follows:

- The Home Address option should be contained to carry the mobile node's home address in the message.
- IPsec should be used.
- The Identifier field in the ICMP Header should be set to a random value.

To prevent mobile node from using invalid home addresses, the Lifetime field value should be set smaller than the remaining lifetime of its home registration.

## 10.6.2 Prefix advertisement

For mobile nodes which are away from home, routers working as mobile node's home agents should advertise prefix information to the mobile node using Mobile Prefix Advertisement message. The mobile node may solicit the advertisement message, instead of waiting for periodical advertisement messages. Mobile Prefix Advertisement and Solicitation messages are explained in Section 7.

Unsolicited Mobile Prefix Advertisement messages from the home agent are generated as follows:

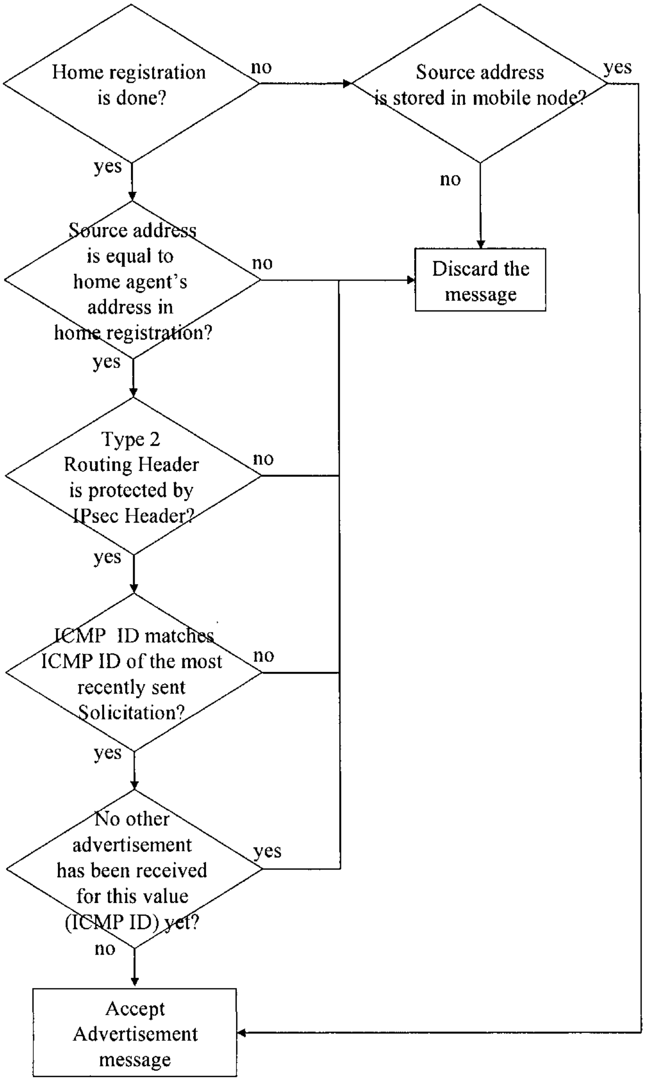

Home registration is done?
no
yes
Source address is stored in mobile node?
yes
no
Source address is equal to home agent's address in home registration?
no
yes
Discard the message
Type 2 Routing Header is protected by IPsec Header?
no
yes
ICMP ID matches ICMP ID of the most recently sent Solicitation?
no
yes
No other advertisement has been received for this value (ICMP ID) yet?
yes
no
Accept Advertisement message

*Figure 10-10.* Algorithm to verify Prefix Advertisement message.

- The Source Address field of IPv6 Header is set to the home agent's IP address. Home agent's IP address is either address which the mobile node used for the home registration or default global home agent address if no binding is present.

- The Destination Address field of IPv6 Header is a unicast address of the mobile node. If the advertisement message is reply to the solicitation message, it is destined to the soliciting mobile node's home address.
- If there are prefix changes in the home link or renumbering, the home agent should send Advertisement messages to all mobile nodes required to change its home address.
- Type 2 Routing Header should be added to contain mobile node' home address.
- IPsec should be used.

Once a mobile node receives Mobile Prefix Advertisement messages, it should verify them with following rules:

- Source Address field of IPv6 Header should be filled with the home agent's IP address to which the mobile node most recently sent Binding Update message for the home registration. If no registration has existed previously, then the mobile node should discard the advertisement message except the case that the mobile node already stores home agent's address. In the other cases, the mobile node should discard the advertisement message.
- The message should have Type 2 Routing Header, which should be protected by an IPsec Header. In the other cases, the mobile node should discard the advertisement message.
- The ICMP Identifier in the Mobile Prefix Advertisement message should be identical to the ICMP identifier of the most recent Mobile Prefix Solicitation message. If no reply message corresponding to the Solicitation has been received, then the Mobile Prefix Advertisement message is again solicited by the mobile node. If the received advertisement message is unsolicited one, then the mobile node should discard it and send a Mobile Prefix Solicitation.
- Otherwise, any Mobile Prefix Advertisement message that does not satisfy above rules should be discarded. The algorithm to verify Mobile Prefix Advertisement message is shown in Fig. 10-10.

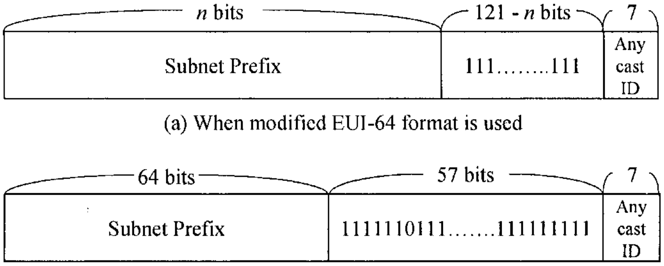
### 10.6.3 Dynamic home agent discovery

Whenever a mobile node moves to other subnets, it should perform the home registration. In the home registration process, the mobile node will send Binding Update messages to its home agent to register new primary care-of address as explained in Section 4.

In some cases, the mobile node may not know its home agent address. For instance, a router on the mobile node's home link and serving as a home agent may disappear from the network, and a new router may substitute for the home agent. Then, the mobile should discover one of suitable home agents on its home link using dynamic home agent discovery.

The mobile node sends ICMP Home Agent Address Discovery Request message to IPv6 home-agents anycast address with its home subnet prefix. If any home agent on the mobile node's home link receives the Request message, it returns an ICMP Home Agent Address Discovery Reply message, which contains the address list about home agents residing on the home link. The Home Agent Address field to carry home agents' addresses in the reply message is built as follows:

- The Home Agent Address field should carry all global IP addresses of home agent in the Home Agent List managed by the home agent.
- The order of addresses to be listed in the field depends on the preference value which is learned from the Home Agent Information option of Router Advertisement messages. Home agent's address with the highest preference goes first.
- Addresses of the home agent having the same preference are listed in random order. If more than one global IP address is found for a home agent, these addresses also should be listed in random order.



(a) When modified EUI-64 format is used

(b) When modified EUI-64 format is not used

*Figure 10-11.* Home-agents anycast address format.

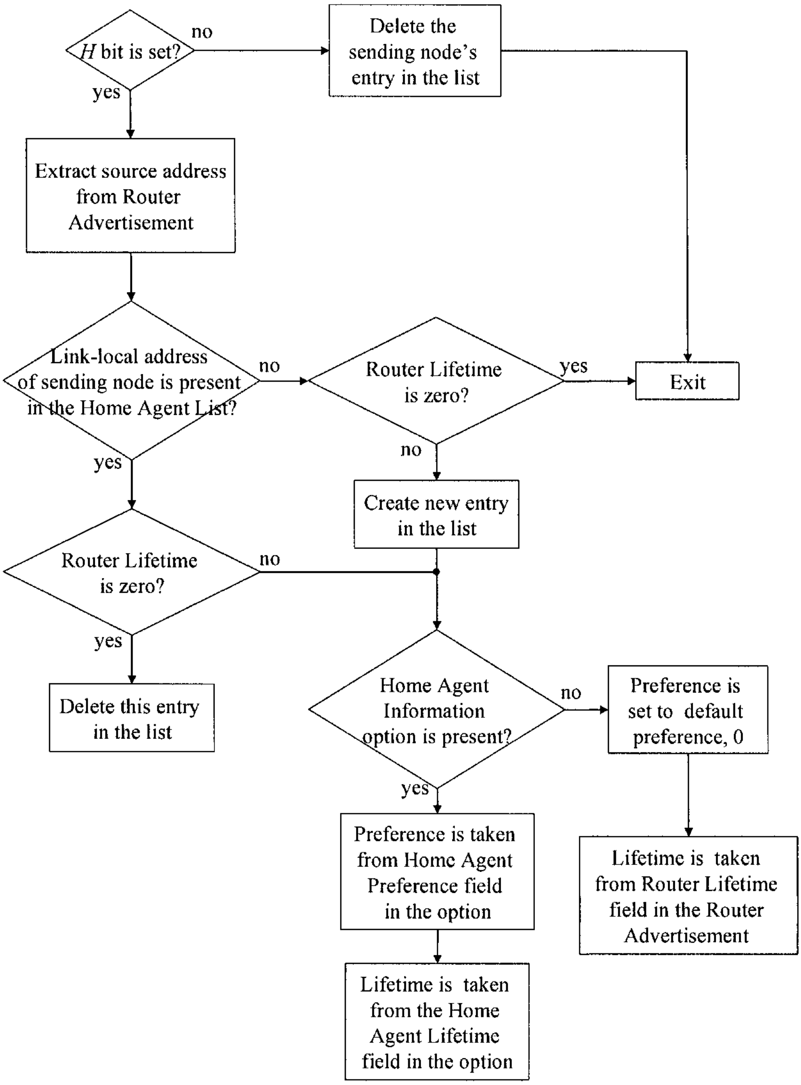

H bit is set?
no
yes
Delete the sending node's entry in the list
Extract source address from Router Advertisement
Link-local address of sending node is present in the Home Agent List?
no
yes
Router Lifetime is zero?
yes
Exit
no
Create new entry in the list
Router Lifetime is zero?
no
yes
Delete this entry in the list
Home Agent Information option is present?
no
Preference is set to default preference, 0
yes
Preference is taken from Home Agent Preference field in the option
Lifetime is taken from Router Lifetime field in the Router Advertisement
Lifetime is taken from the Home Agent Lifetime field in the option

*Figure 10-12.* Algorithm to verify Router Advertisement message by a home agent.

As the mobile node receives Home Agent Address Discovery Reply message, it starts home registration process. The mobile node sends Binding Update message to one of routers in the list from the Reply message, to request the router to operate as a home agent for the mobile node. If no reply message is coming, the mobile node attempts home registration to another router listed in the Home Agent Address Discovery Reply message until the home registration is accepted. Message formats for ICMP Home Agent Address Discovery Request and Reply are explained in Section 7.

### 10.6.4 IPv6 home-agents anycast address

RFC 2526[12] defines reserved subnet anycast address, especially, IPv6 home-agents anycast address. When the subnet prefix length is fixed to 64-bit, remaining 64-bit is composed of 57-bit from modified EUI-64 and 7-bit anycast ID, as shown in Fig. 10-11(a).

When 64-bit subnet prefix is used:

- The universal/local bit in modified EUI-64 should be set to 0 to indicate the local usage. 56 bits following this bit are set to 1.
- The last 7-bit of modified EUI-64 format contains anycast ID, which is set to 126 in decimal, and 7F in hexadecimal.

### 10.6.5 Home Agent List

A router working as a home agent in the home link maintains a Home Agent List recording all home agents on that link. This list supports dynamic home agent address discovery. Through periodic Router Advertisement messages, the home agent learns other home agent information in the same link. This mechanism is very similar to the Default Router List maintained by each node for the Neighbor Discovery. All home agents are required to send Router Advertisement messages with *H* bit[20] (Home Agent bit) set periodically. Each home agent builds Home Agents List.

Once the home agent receives a valid Router Advertisement message, it should verify it with following rules in addition to the Neighbor Discovery mechanism:

1. If the *H* bit in the message is not set, the home agent deletes the sending node's entry in the current Home Agent List. No changes will arise in the home agent.
2. Otherwise, the home agent extracts the source address from IP Header of the message. This address is the link-local address of the sending node.
3. The home agent checks if the extracted address is present in the Home Agent List. If the address is found in the list, and the router lifetime is nonzero, the home agent updates the lifetime and preference according to the Home Agent Information option. If the address is found, and the router lifetime is zero, the home agent deletes the entry from the list.

[20] *H* bit is defined in the Reserved field in the Router Advertisement message. This message format is specified in Section 8.1.

4. If the extracted address is new one, the home agent creates new entry in the Home Agent List and checks Home Agent Information option. Lifetime and preference are taken from the Home Agent Information option if the option is contained in the message. Otherwise, lifetime is learned from Router Lifetime, and preference is set to the default value, 0.

The home agent should keep the entry in its Home Agents List until the lifetime for the entry expires. Once an entry becomes invalid, it should be removed from the Home Agent List. The algorithm to verify Router Advertisement messages by the home agent is described in Fig. 10-12.

## 10.7 MESSAGE TYPES

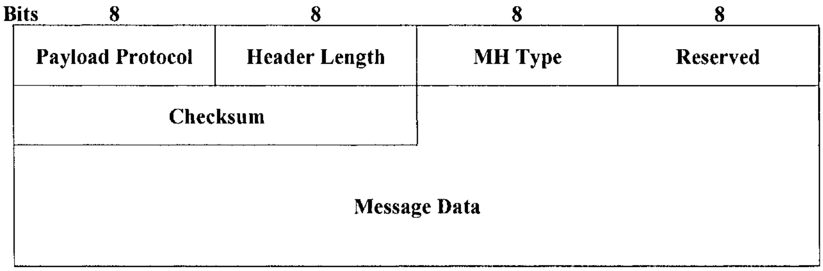
### 10.7.1 Mobility messages

A message contained in the Mobility Header following IPv6 Header is called the mobility message, which is newly defined in MIPv6. New MIPv6 messages are as follows:

- Home Test Init (HoTI) message
- Home Test (HoT) message
- Care-of Test Init (CoTI) message
- Care-of Test (CoT) message
- Binding Update (BU) message
- Binding Acknowledgement (BA) message
- Binding Refresh Request (BRR) message
- Binding Error message (BE) message

The Mobility Header is identified by the next header value, 135 in the preceding header. The header format is shown in Fig. 10-13. Each field is described as follows.

- The 8-bit Payload Protocol field notifies the header format following the Mobility Header. The value for next header is same as IPv6 Next Header field.
- The 8-bit Header Extension Length field specifies the length of the Mobility Header in 8-byte units except the Next Header field.
- The 8-bit MH type is the identifier among variable Mobility Header types. Current MH type values are specified in Table 10-2.
- The next 8-bit field is reserved for the future use and should be initialized to zero by the sender. If this field contains any other value, it should be ignored by the receiver.
- The 16-bit Checksum field contains the checksum of the Mobility Header.



| Bits | 8 | 8 | 8 | 8 |
|---|---|---|---|---|
| | Payload Protocol | Header Length | MH Type | Reserved |
| | Checksum | | | |
| | Message Data | | | |

*Figure 10-13.* The Mobility Header format.

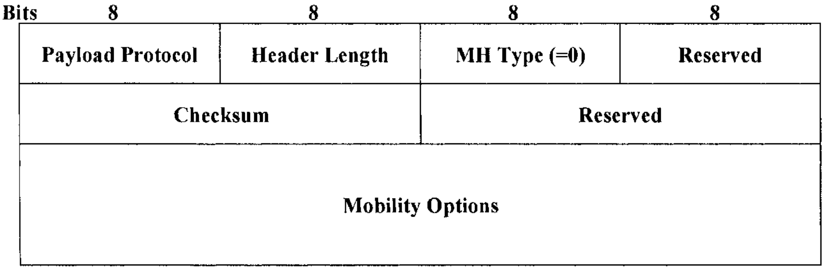
*Table 10-2.* MH type values.

| MH type | Description |
|---|---|
| 0 | Indicates Binding Refresh Request message. |
| 1 | Indicates Home Test Init message. |
| 2 | Indicates Care-of Test Init message. |
| 3 | Indicates Home Test message. |
| 4 | Indicates Care-of Test message. |
| 5 | Indicates Binding Update message. |
| 6 | Indicates Binding Acknowledgement message. |
| 7 | Indicates Binding Error message. |

Specific data according to the indicated MH Type is contained in the Message Data field. Each message type is explained in detail in following subsections. The mobility option specified in Section 7 may be used.

### 10.7.1.1 Binding Refresh Request message

A correspondent node sends Binding Refresh Request (BRR) message to a mobile node to update Binding Cache Entry for the mobile node. Once the mobile node receives this message, it checks its Binding Update List for the sender. If the mobile node wants to keep binding with the correspondent node, it starts return routability procedure. Once return routability procedure ends in success, binding update process is followed. The Lifetime field in the Binding Update message should be set to a new value. The lifetime value should not be bigger than the remaining lifetime of the home registration and the care-of address. As the mobile node sends the Binding Update message to the correspondent node, Binding Update List should be updated.



| Bits | 8 | 8 | 8 | 8 |
|---|---|---|---|---|
| | Payload Protocol | Header Length | MH Type (=0) | Reserved |
| | Checksum | | Reserved | |
| | Mobility Options | | | |

*Figure 10-14.* Binding Refresh Request message format.

If the mobile node wants to remove entry for the mobile node in the Binding Cache of the correspondent node, the mobile node ignores the Binding Refresh Request message and waits for the expiration of lifetime, or it explicitly deletes entry by sending Binding Update message, where lifetime is set to 0, and the care-of address is set to the home address.

When MH Type value is set to 0, it indicates Binding Refresh Request message, and the message format is shown in Fig. 10-14.

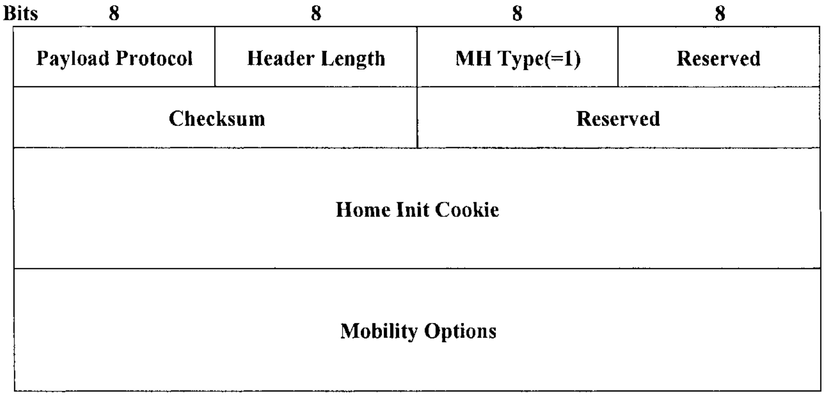
- The Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.
- In the Mobility Options field, there are multiple 8-byte long integer TLV-encoded mobility options. When a receiver gets any message with non-understandable options, it just ignores.

### 10.7.1.2 Home Test Init message

When a mobile node initiates return routability procedure to request Home Keygen Token from a correspondent node, it sends Home Test Init (HoTI) message to the correspondent node.

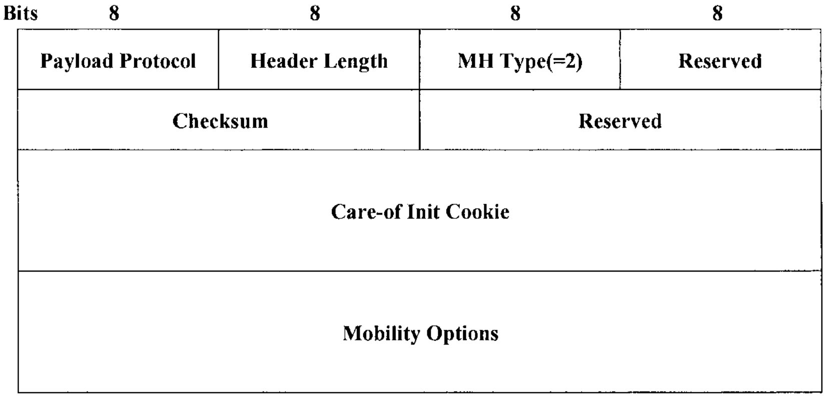
When MH type value is set to 1, it indicates Home Test Init message, and the message format is shown in Fig. 10-15:

- The Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.
- In the Home Init Cookie field, 64-bit random Home Init Cookie is contained and delivered.
- In the Mobility Options field, there are multiple 8-byte long integer TLV-encoded mobility options. When a receiver gets any message with non-understandable options, it just ignores.



| Bits 8 | 8 | 8 | 8 |
|---|---|---|---|
| Payload Protocol | Header Length | MH Type(=1) | Reserved |
| Checksum | | Reserved | |
| Home Init Cookie | | | |
| Mobility Options | | | |

*Figure 10-15.* Home Test Init message format.



| Bits 8 | 8 | 8 | 8 |
|---|---|---|---|
| Payload Protocol | Header Length | MH Type(=2) | Reserved |
| Checksum | | Reserved | |
| Care-of Init Cookie | | | |
| Mobility Options | | | |

*Figure 10-16.* Care-of Test Init message format.

When the mobile node is away from home, this message is tunneled via its home agent and delivered finally to the mobile node. IPsec ESP should be employed on the tunnel between the home agent and the mobile node.

### 10.7.1.3 Care-of Test Init message

When a mobile node initiates return routability procedure to request Care-of Keygen Token from a correspondent node, it sends Care-of Test Init (CoTI) message to the correspondent node.

When MH type value is set to 2, it indicates Care-of Test Init message, and the message format is shown in Fig. 10-16.

- The Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.
- In the Care-of Init Cookie field, 64-bit random Care-of Init Cookie is contained and delivered.

In the Mobility Options field, there are multiple 8-byte long integer TLV-encoded mobility options. When a receiver gets any message with non-understandable options, it just ignores.

### 10.7.1.4 Home Test message

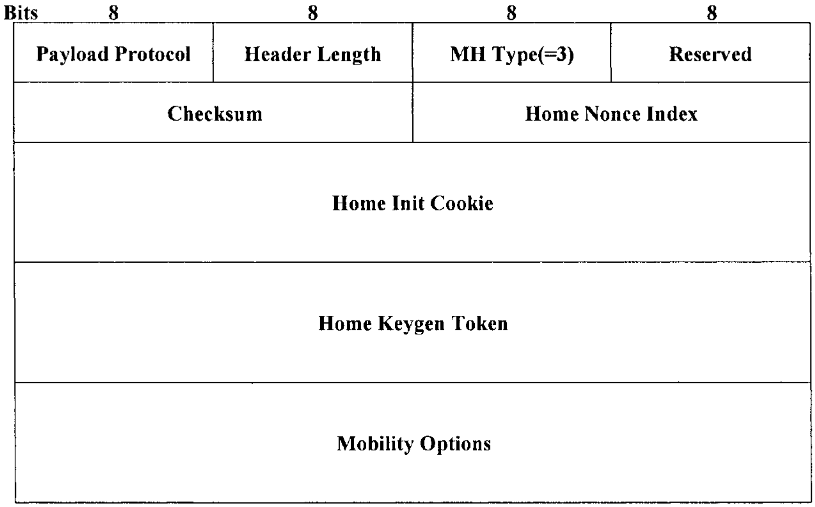
Once a correspondent node receives the Home Test Init message, it responds with Home Test (HoT) message. The message format for Home Test message is shown in Fig. 10-17.

- The MH Type is set to 3.
- The Home Nonce Index field contains Home Nonce Index and will be echoed back via Binding Update message.
- The Home Init Cookie field contains 64-bit Home Init Cookie generated by the correspondent node.
- Home Keygen Token field contains 64-bit Home Keygen Token.
- In the Mobility Options field, there are multiple 8-byte long integer TLV-encoded mobility options. When a receiver gets any message with non-understandable options, it just ignores.

### 10.7.1.5 Care-of Test message

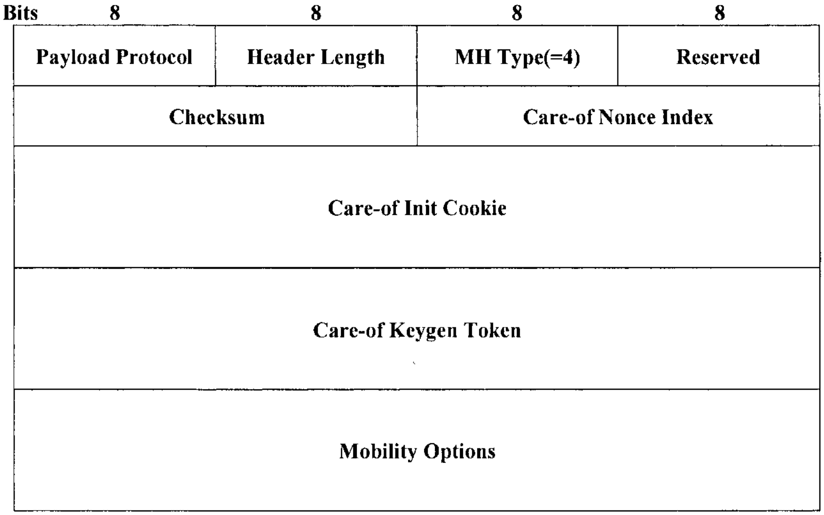
Once a correspondent node receives Care-of Test Init message, it responses with Care-of Test (CoT) message. The message format for Care-of Test message is shown in Fig. 10-18.

- The MH Type is set to 4.
- The Care-of Nonce Index field contains Care-of Nonce Index and will be echoed back via Binding Update message.
- The Care-of Init Cookie field contains 64-bit Care-of Init Cookie generated by the correspondent node.
- Care-of Keygen Token field contains 64-bit Care-of Keygen Token.
- In the Mobility Options field, there are multiple 8-byte long integer TLV-encoded mobility options. When a receiver gets any message with non-understandable options, it just ignores.



| Bits 8 | 8 | 8 | 8 |
|---|---|---|---|
| Payload Protocol | Header Length | MH Type(=3) | Reserved |
| Checksum | | Home Nonce Index | |
| Home Init Cookie | | | |
| Home Keygen Token | | | |
| Mobility Options | | | |

*Figure 10-17.* Home Test message format.



| Bits 8 | 8 | 8 | 8 |
|---|---|---|---|
| Payload Protocol | Header Length | MH Type(=4) | Reserved |
| Checksum | | Care-of Nonce Index | |
| Care-of Init Cookie | | | |
| Care-of Keygen Token | | | |
| Mobility Options | | | |

*Figure 10-18.* Care-of Test message format.

### 10.7.1.6 Binding Update message

Once return routability is successfully completed, a mobile node sends Binding Update message to inform the correspondent node of new care-of address for the mobile node. The mobile node also sends Binding Update message for the home registration, as specified in Section 4.

When MH type value is set to 5, it indicates Binding Update message. The format of mobility data for Binding Update message is shown in Fig. 10-19.

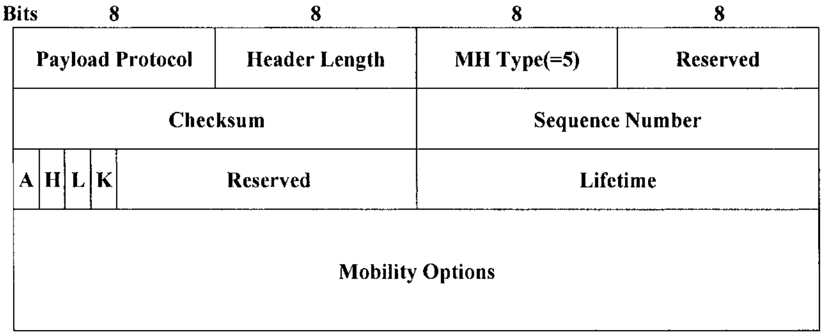
- The Sequence Number field contains 16-bit unsigned integer for the receiving node to index Binding Updates. Besides, the mobile node uses this field later to check Binding Acknowledgement message for the Binding Update.
- Four flags are defined for Binding Update message such as *A*, *H*, *L*, and *K* flags:
  The *A* flag stands for 'Acknowledge', and the mobile node set this flag to request the correspondent node to send reply message back.
  The *H* flag stands for 'Home Registration', and it is used to request receiving node to be the mobile node's home agent. The receiving node of the Binding Update message with *H* bit set should be one of routers sharing the same subnet prefix as the one of mobile node's home addresses in binding state.
  The *L* flag stands for 'Link-Local Address Compatibility', and it is set when the interface identifier of the mobile node's home address is equal to the one of mobile node's link-local addresses.
  The *K* flag stands for 'Key Management Mobility Capability', and it is used to notify that IPsec security associations between the home agent and the mobile node are provided on mobile node's movements. If this *K* bit in the Binding Update message is set to 0, protocols for IPsec security associations between them do not exist on mobile node's movements.
  The *H* and *K* bits should be used in Binding Update messages destined to the home agent, and correspondent nodes should ignore these bits when they receives *H* or *K* bit set in Binding Update messages.
- The Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.
- The Lifetime field contains 16-bit unsigned integer. This field specifies the remaining time unit before the binding is expired. One time unit is 4 seconds.
- If the Lifetime field is set to 0, the mobile node wants to have the correspondent node remove its entry from Binding Cache, and care-of address should be replaced with the mobile node's home address.
- In the Mobility Options field, there are multiple 8-byte long integer TLV-encoded mobility options. When a receiver gets any message with non-understandable options, it just ignores. Several options are valid in a binding update, such as Binding Authorization Data option, Nonce Indices option, and Alternate Care-of Address option. The Binding

Authorization Data option should be sent to a correspondent node with the Binding Update message.

### 10.7.1.7 Binding Acknowledgement message

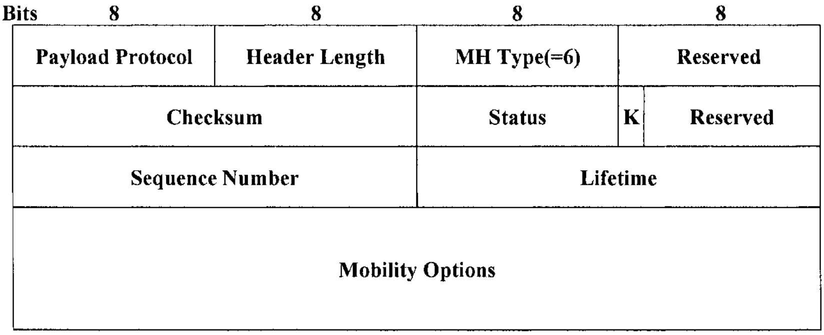
A home agent or a correspondent node sends back a Binding Acknowledgement message to indicate receipt of a Binding Update message. If the Acknowledge (*A*) bit is set in the Binding Update message, a Binding Acknowledgement message should be returned.

When MH Type value is set to 6, it indicates Binding Acknowledgement message. The format of Binding Acknowledgement message is shown in Fig. 10-20.



<table>
<tr><td>Bits</td><td colspan="5">8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td></td><td colspan="5">Payload Protocol</td><td>Header Length</td><td>MH Type(=5)</td><td>Reserved</td></tr>
<tr><td></td><td colspan="6">Checksum</td><td colspan="2">Sequence Number</td></tr>
<tr><td></td><td>A</td><td>H</td><td>L</td><td>K</td><td colspan="2">Reserved</td><td colspan="2">Lifetime</td></tr>
<tr><td></td><td colspan="8">Mobility Options</td></tr>
</table>

*Figure 10-19.* Binding Update message format.



<table>
<tr><td>Bits</td><td>8</td><td>8</td><td>8</td><td colspan="2">8</td></tr>
<tr><td></td><td>Payload Protocol</td><td>Header Length</td><td>MH Type(=6)</td><td colspan="2">Reserved</td></tr>
<tr><td></td><td colspan="2">Checksum</td><td>Status</td><td>K</td><td>Reserved</td></tr>
<tr><td></td><td colspan="2">Sequence Number</td><td colspan="3">Lifetime</td></tr>
<tr><td></td><td colspan="5">Mobility Options</td></tr>
</table>

*Figure 10-20.* Binding Acknowledgement message format.

- The Status field specifies a code to indicate the state of Binding Update message. Values for the Status field are specified in Table 10-3. Once a node accepts the Binding Update message and creates or updates Binding Cache, the Status field in the Binding Acknowledgement message should be set to a value smaller than 128. In other cases, if Binding Update is rejected, the Status Field is set to a value bigger than or equal to 128.

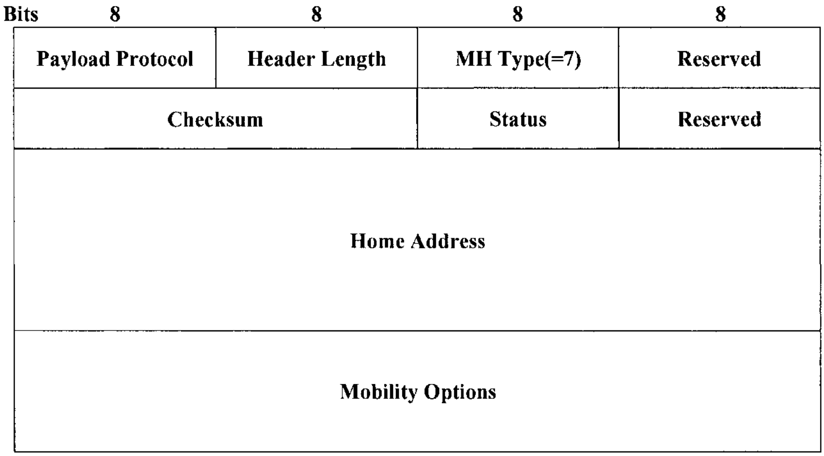


*Figure 10-21.* Binding Error message format.

*Table 10-3.* Status codes for Binding Update message.

| Status type | Descriptions |
|---|---|
| 0 | Binding Update accepted. |
| 1 | Accepted but prefix discovery is necessary. |
| 128 | Reason unspecified. |
| 129 | Administratively prohibited. |
| 130 | Insufficient resources. |
| 131 | Home registration is not supported. |
| 132 | Not home subnet. |
| 133 | Not home agent for this mobile node. |
| 134 | Duplication Address Detection failed. |
| 135 | Sequence number out of window. |
| 136 | Expired home Nonce index. |
| 137 | Expired care-of Nonce index. |
| 138 | Expired Nonces. |
| 139 | Registration type change disallowed. |

- In the Binding Acknowledgement message, *K* flag is defined, which is the same flag as the Binding Update message.
- The Reserved field is set to 0 for the future use, and it should be ignored by the receiver when this field is set to any other value.
- The Sequence Number field contains 16-bit unsigned integer value copied from the Sequence Number field in the Binding Update. This value is used by a mobile node to match Binding Acknowledgement to the Binding Update message.

The Lifetime field contains 16-bit unsigned integer. This field specifies the granted lifetime unit before the binding is expired. One time unit is 4 seconds. The value of this field is undefined if the Status field indicates that the Binding Update was rejected with several reasons specified in Table 10-3.

- In the Mobility Options field, there are multiple 8-byte long integer TLV-encoded mobility options. When a receiver gets any message with non-understandable options, it just ignores. Several options are valid in a Binding Acknowledgement, such as Binding Authorization Data option, and Binding Refresh Advice option. The Binding Authorization Data option should be sent to the mobile node with the Binding Acknowledgement message.

#### 10.7.1.8 Binding Error Message

When an error which is related to the mobility occurs, the correspondent node sends Binding Error (BE) message directly to the address in the IPv6 Source Address field in the erroneous packet. If source address is not a unicast address, Binding Error message should not be returned.

When MH Type value is set to 7, it indicates Binding Error message. The format of message data in mobility data for Binding Error message is shown in Fig. 10-21.

- The Status field specifies codes to indicate reasons why an error occurred. This field contains 8-bit unsigned integer value. Only two types are defined now. If the Status field is set to 1, it says 'Unknown binding for Home Address Destination option,' and if it is set to 2, it says 'Unrecognized MH Type value'.
- The Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.
- The Home Address field contains the home address that is copied from the Home Address Destination option. When mobile node has more than one home address, it will use this information to determine which binding does not exist.

- In the Mobility Options field, there are multiple 8-byte long integer TLV-encoded mobility options. When a receiver gets any message with non-understandable options, it just ignores.

## 10.7.2 Mobility options

Mobility messages specified in Section 7.1 are able to include one or multiple options. Currently, four kinds of option are defined, such as Binding Refresh Advice option, Alternate Care-of Address option, Nonce Indices option, and Binding Authorization Data option. It is not necessary for options to be contained in every Mobility Header. Mobility option is encoded in type-length-value (TLV) format, which follows the conventional IPv6 option format specified in Chapter 2.

### 10.7.2.1 Binding Refresh Advice

The Binding Refresh Advice option can be contained in only Binding Acknowledgement message. A correspondent node will use this option in the Binding Acknowledgement message as a reply to the Binding Update message for the correspondent registration, or a home agent will use this option in the Binding Acknowledgement message as a reply to the Binding Update message for the home registration.

The Binding Refresh Advice option is composed of one-byte Option Type, one-byte Option Length, and two-byte Refresh Interval fields.

- The alignment requirement for Binding Refresh Advice option is 2*n*.
- The Option Type field is set to 2.
- The Option Length field is set to 2.
- The Refresh Interval defined in this option is the remaining time unit before the mobile node should trigger a new home registration process. One time unit is 4 seconds.

### 10.7.2.2 Alternate Care-of Address

Usually, the Source Address field of IPv6 Header in the Binding Update message contains mobile node's care-of address. In some cases, the mobile node can not use its care-of address as a source address, such as when employed security mechanism does not protect the IPv6 Header.

The Alternate Care-of Address option is composed of one-byte Option Type, one-byte Option Length, and 16-byte Alternate Care-of Address fields. The alignment requirement for Alternate Care-of Address option is 8*n*+6.

- The Option Type field is set to 3, and the Option Length field is set to 16.

- The Alternate Care-of Address option can be contained only in the Binding Update message. The Alternate Care-of Address field in this option contains a care-of address for the binding. Thus, it prevents address in the Source Address field of IPv6 packet from being used as the care-of address.

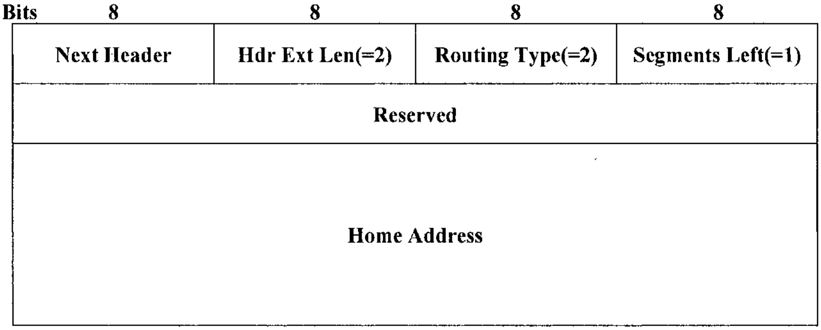
### 10.7.2.3 Nonce Indices

The Nonce Indices option is used in only two cases: a mobile node uses this option in the Binding Update message on correspondent registration, or this option is used with Binding Authorization Data option. During the return routability procedure, the correspondent node builds Home and Care-of Keygen Tokens using random Nonce values.

The Nonce Indices option is composed of one-byte Option Type, one-byte Option Length, two-byte Home Nonce Index, and two-byte Care-of Nonce Index fields. The alignment requirement for the Alternate Care-of Address option is 2*n*.

- The Option Type field is set to 4.
- The Option Length field is set to 4.
- The Home and Care-of Nonce Index fields in Nonce Indices option indicate the correspondent node which Nonce value will be used to generate the Keygen Tokens.

When the mobile node requests the correspondent node to delete its binding information, Care-of Nonce Index field in Nonce Indices option will be dropped.



| Bits 8 | 8 | 8 | 8 |
|---|---|---|---|
| Next Header | Hdr Ext Len(=2) | Routing Type(=2) | Segments Left(=1) |
| Reserved | | | |
| Home Address | | | |

*Figure 10-22.* Type 2 Routing Header format.

#### 10.7.2.4 Binding Authorization Data

The Binding Authorization Data option can only be contained in Binding Update and Binding Acknowledgement messages. This option is composed of one-byte Option Type, one-byte Option Length, and Authenticator fields. The alignment requirement for the Alternate Care-of Address option is 8*n*+2.

- The Option Type field is set to 5.
- In the Option Length field, the length of the authenticator is contained in bytes.
- In the Authenticator field cryptographic value is contained in bytes, as shown in Eq. (3).

From Eq. (3), current primary care-of address is used as the care-of address if the binding update procedure succeeds. When this option is used for de-registration, mobile node's home address will be used instead of care-of address.

### 10.7.3 Home Address option

The Home Address option is contained in the Destination Option Extension Header whose Next Header value is 60. This option is used when a mobile node is away from home and wants to inform the receiving node of its home address. Home Address option is encoded in type-length-value (TLV) format, which follows the conventional IPv6 option format specified in Chapter 2. The Home Address option is strongly discouraged to use when a correspondent node already has an entry about the mobile node on its Binding Cache.

The Home Address option is composed of one-byte Option Type, one-byte Option Length, and 16-byte Home Address fields. The alignment requirement for Home Address option is 8*n*+2.

- Option Type field is set to 201.
- The Option Length field contains the length of the option in bytes except the Option Type and Option Length fields. This field should be set to 16.
- In the Home Address field, mobile node's home address is contained.

The Home Address option should be placed following the Routing Header, before the Fragment Header, and before the AH Header or ESP Header, if any of these headers is present in a packet.

### 10.7.4 Type 2 Routing Header

MIPv6 defines new Type 2 Routing Header to deliver a packet directly to the mobile node's care-of address. When a correspondent node generates IPv6 packet which is destined to the mobile node, it inserts mobile node's care-of address into the IPv6 Destination Address field and mobile node's home address into the Type 2 Routing Header. Once the packet arrives at the care-of address, the mobile node learns its home address from the Type 2 Routing Header and replaces care-of address in the Destination Address field with home address in the Routing Header. Then, the datagram will be delivered to the upper transport layer. This replacement allows packet to avoid firewalls and ingress filtering.[9] The new Routing Header type is only allowed to carry IPv6 address. This replacement is depicted in Fig. 10-3.

The Type 2 Routing Header format is shown in Fig. 10-22.

- The one-byte Next Header field identifies the header type immediately following the Routing Header, and it has the same value as the conventional IPv6 Next Header field.
- The one-byte Hdr Ext Len field indicates the length of Home Address field in byte units, and it is set to 2.
- The one-byte Routing Type field is set to 2 to discriminate from type 0 which is the original Routing Header type.
- The one-byte Segments Left field specifies the number of remaining route segments, and it should be set to 1.
- The 4-byte Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.
- In Home Address field, the mobile node's home address is carried.

Rules to order Type 2 Routing Header follows the ordering rules specified in Chapter 2. The recommended order of Extension Headers is as follows:

IPv6 Header → Hop-by-Hop Options Header → Destination Options Header[21] → Routing Header → Fragment Header → Authentication Header → ESP Header → Destination Options Header[22] (upper-layer header)

When type 0 and Type 2 Routing Headers appear together in a packet, the type 0 Routing Header is always placed first.

## 10.7.5 ICMPv6 message types

Four message types are defined for Mobility service in ICMPv6: ICMP Mobile Prefix Solicitation, ICMP Mobile Prefix Advertisement, ICMP Home Agent Address Discovery Request, and ICMP Home Agent Address Discovery Reply. The first two messages are used in common to manage mobile prefix information, and the next two messages are used in dynamic home agent address discovery.

[21] Options to be processed by the destination specified in the IPv6 Destination Address field

[22] Options to be processed by the final destination of a packet

### 10.7.5.1 ICMP Mobile Prefix Solicitation message

When a mobile node is away from the home link, it sends ICMP Mobile Prefix Solicitation message to solicit Mobile Prefix Advertisement message from a home agent in the mobile node's home link.

- The Source Address field of IPv6 Header is set to the mobile node's care-of address.
- The Destination Address field of IPv6 Header is set to the home agent's address.
- The Hop Limit field has the same initial hop limit value, as any other unicast packet from the mobile node.
- The Destination Option field should be added to the IPv6 Header.
- IPsec Header should be attached to the IPv6 Header.

When type value of ICMP Header is set to 146, it indicates Mobile Prefix Solicitation message, and the message format is shown in Fig. 10-23.

- The Code field is set to 0.
- The Checksum is calculated with the same algorithm for common ICMP checksum.
- The Identifier field is used to match this Mobile Prefix Solicitation message to the next Mobile Prefix Advertisement.
- The Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.

### 10.7.5.2 ICMP Mobile Prefix Advertisement message

A home agent will send a Mobile Prefix Advertisement message to a mobile node which solicited the Advertisement message. This message will provides prefix information about home link while the mobile node is away from its home network. The Mobile Prefix Advertisement message can be triggered by a solicitation from a mobile node, or unsolicited message will be sent.

- The Source Address field of IPv6 Header is set to the home agent's address.
- The Destination Address field of IPv6 Header is set to the mobile node's address if this message is generated by solicitation. Otherwise, the advertisement message can be generated for the mobile node's care-of address only when home registration was complete.
- The type 2 Routing Header should be included to carry the mobile node's home address.

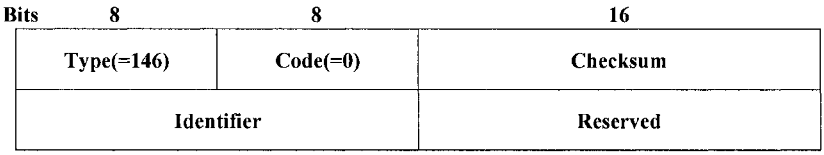


*Figure 10-23.* Mobile Prefix Solicitation message format.

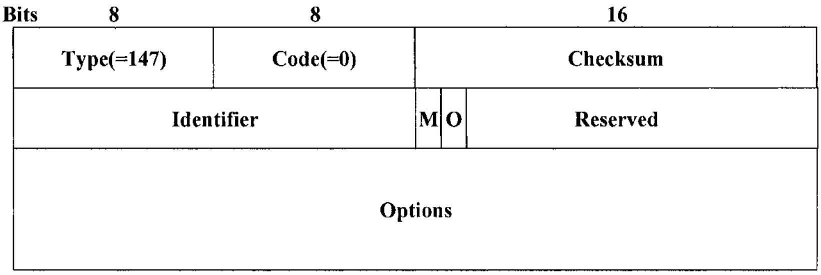


*Figure 10-24.* Mobile Prefix Advertisement message format.

The message format for Mobile Prefix Advertisement is shown in Fig. 10-24.

- When type value of ICMP Header is set to 147, it indicates Mobile Prefix Advertisement message.
- The Code field is set to 0.
- The Checksum is calculated with the same algorithm for common ICMP checksum.
- The identifier from the Solicitation message which invokes this advertisement message should be copied to the Identifier field. In case of an unsolicited advertisement message, the Identifier field should be set to 0.
- *M* and *O* flags are defined in the Mobile Prefix Advertisement message. The *M* flag is 1-bit Managed Address Configuration flag, and the *O* flag is 1-bit Other Stateful Configuration flag. When *M* flag is set, a receiving node is able to use both stateful and stateless address autoconfiguration protocols. When *O* flag is set, the receiving node will use stateful autoconfiguration protocol to obtain information except address. *M* and *O* flags should be cleared when Home Agent DHCPv6 is not supported.
- The Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.

- The Mobile Prefix Advertisement message is able to carry options defined in RFC 2461.[6] Currently, only Prefix Information option is carried in the Mobile Prefix Advertisement message. In each message, prefix (or prefixes) which is allowed for the mobile node to use is carried in one or more Prefix Information options.

### 10.7.5.3 ICMP Home Agent Address Discovery Request message

When a mobile node wants to learn home agent address dynamically, it will use Home Agent Address Discovery Request message, which invokes dynamic home agent address discovery mechanism. The mobile node and its home agent exchange Home Agent Address Discovery Request and Home Agent Address Discovery Reply messages. The destination address of the Home Agent Address Discovery Request message is the MIPv6 home-agents anycast address. The source address is mobile node's current primary care-of address.

The message format for the Home Agent Address Discovery Request is shown in Fig. 10-25.

- When type value of ICMP Header is set to 144, it indicates Home Agent Address Discovery Request message.
- The Code field is set to 0.
- The Checksum is calculated using the same algorithm for common ICMP checksum.
- The Identifier field is used to identify the Home Agent Address Discovery Reply in response to the Home Agent Address Discovery Request message.
- The Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.

### 10.7.5.4 ICMP Home Agent Address Discovery Reply message

Once a home agent receives Home Agent Address Discovery Request message from a mobile node, it should send Home Agent Address Discovery Reply message as a response.

The message format for Home Agent Address Discovery Reply is shown in Fig. 10-26.

- When the type value of ICMP Header is set to 145, it indicates Home Agent Address Discovery Reply message.
- The Code field is set to 0.

<table>
<tr><td>Bits</td><td>8</td><td>8</td><td>16</td></tr>
<tr><td></td><td>Type(=144)</td><td>Code(=0)</td><td>Checksum</td></tr>
<tr><td></td><td colspan="2">Identifier</td><td>Reserved</td></tr>
</table>

*Figure 10-25.* Home Agent Address Discovery Request message format.

<table>
<tr><td>Bits</td><td>8</td><td>8</td><td>16</td></tr>
<tr><td></td><td>Type(=145)</td><td>Code(=0)</td><td>Checksum</td></tr>
<tr><td></td><td colspan="2">Identifier</td><td>Reserved</td></tr>
<tr><td></td><td colspan="3">Home Agent Addresses</td></tr>
</table>

*Figure 10-26.* Home Agent Address Discovery Reply message format.

<table>
<tr><td>Bits</td><td>8</td><td colspan="4">8</td><td>16</td></tr>
<tr><td></td><td>Type</td><td colspan="4">Code</td><td>Checksum</td></tr>
<tr><td></td><td>Current Hop Limit</td><td>M</td><td>O</td><td>H</td><td>Reserved</td><td>Router Lifetime</td></tr>
<tr><td></td><td colspan="6">Reachable Time</td></tr>
<tr><td></td><td colspan="6">Retransmission Timer</td></tr>
<tr><td></td><td colspan="6">Options</td></tr>
</table>

*Figure 10-27.* Modified Router Advertisement message format.

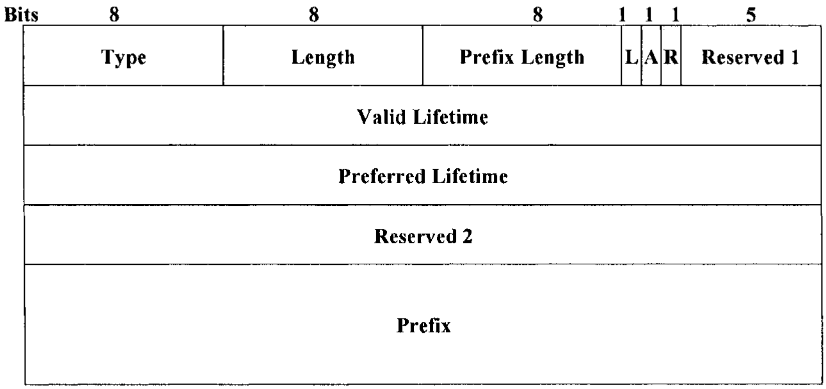


*Figure 10-28.* Modified Prefix Information option format.

- The Checksum is calculated with the same algorithm for common ICMP checksum.
- The identifier from the Request message which invokes this Reply message should be copied to the Identifier field.
- The Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.
- Valid home agents' addresses on the home link are listed in the Home Agent Address field.

## 10.8 CHANGES IN IPv6 NEIGHBOR DISCOVERY PROTOCOL

To support dynamic home agent discovery, MIPv6 modifies some parts of IPv6 Neighbor Discovery Protocol. At first, Router Advertisement message is modified. Two options for the Advertisement message are also modified, and Home Agent Information option is defined.

### 10.8.1 Modified Router Advertisement message

The Router Advertisement message defined in RFC 2461 is modified as follows:

- *H* (Home Agent) bit is defined in the Reserved field of ICMP Header to indicate that the router sending the Advertisement message is also operating as a home agent on the link.

- The Reserved field is reduced from 6-bit field to a 5-bit field. The header format is shown in Fig. 10-27.
- In the Options field, Modified Prefix Information option, Advertisement Interval option and Home Agent Information option may be included.
- Remaining fields have the same feature in the original Router Advertisement message.

### 10.8.2 Modified Prefix Information option

In IPv6 Neighbor Discovery mechanism, only link-local address is used in the Source Address field of IP Header to send Router Advertisement message. To support IP mobility, a new single flag, *R* (Router Address) is defined in the previous Reserved field of Prefix Information option, which allows a router to advertise its prefix information to mobile nodes away from home link. The option format is shown in Fig. 10-28.

- *R* flag indicates that the Prefix field in the option carries a complete IP address assigned to the home agent. When a home agent sends Router Advertisement message, it should include more than one Modified Prefix Information Option with the *R* flag set.
- The Reserved1 field is reduced from 6-bit field to a 5-bit field.
- Remaining fields have the same feature in the original Prefix Information Option.

### 10.8.3 Advertisement Interval option

Advertisement Interval option is defined in the Router Advertisement message to indicate the interval of unsolicited multicast Router Advertisements transmission by a home agent. The option format is shown in Fig. 10-29.

- When the type value is set to 7, it indicates Advertisement Interval option.
- The Length field specifies the length of the option including the Type and Length fields in units of 8 bytes. The Length field should be set to 1.
- The Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.
- The 4-byte Advertisement Interval field specifies the maximum interval between most recently transmitted unsolicited Router Advertisement message and the next one by the home agent. This Interval is expressed in milliseconds.

### 10.8.4 Home Agent Information option

Another new option, Home Agent Information option is defined in the Router Advertisement message to indicate specific information of home agents. The option format is shown in Fig. 10-30.

- When type value is set to 8, it indicates Home Agent Information option.
- The Length field specifies the length of the option including the Type and Length fields in units of 8 bytes. The Length field should be set to 1.
- The Reserved field is set to 0 for the future use, and it should be ignored by a receiver when this field is set to any other value.
- The 4-byte Home Agent Preference field specifies preference for the sending home agent. When the home agent sends Home Agent Address Discovery Reply message to a mobile node, Home Agent List ordered by preference is carried. When Router Advertisement message with *H* bit set is received without this option, the preference value is set to the default value, 0. As the value in this field is bigger, it is more preferable.
- The 4-byte Home Agent Lifetime field specifies the valid lifetime of home agent in seconds. The default value is the lifetime in Router Lifetime of Router Advertisement message, and the maximum value is 18.2 hours.

| Bits | 8 | 8 | 16 |
|---|---|---|---|
| | Type | Length | Reserved |
| | Advertisement Interval | | |

*Figure 10-29.* Advertisement Interval option format.

| Bits | 8 | 8 | 16 |
|---|---|---|---|
| | Type | Length | Reserved |
| | Home Agent Preference | | Home Agent Lifetime |

*Figure 10-30.* Home Agent Information option format.

When a home agent sends Router Advertisement message with *H* bit set, this option may be included. Lifetime in Home Agent Information option should not be equal to the Router Lifetime in Router Advertisement message. A receiving node will learn the lifetime from Router Lifetime in Router Advertisement message if the option is omitted.

## REFERENCES

1. C. Perkins, IP Mobility Support for IPv4, RFC 3344 (June 2002).
2. R. Droms, Dynamic Host Configuration Protocol, RFC 2131 (March 1997).
3. D. Johnson, C. Perkins, and J. Arkko, Mobility Support in IPv6, RFC 3775 (June 2004).
4. D. Mitton, M. St.Johns, S. Barkley, D. Nelson, B. Patil, M. Stevens, and B. Wolff, Authentication, Authorization, and Accounting: Protocol Evaluation, RFC 3127 (June 2001).
5. J. Arkko, V. Devarapalli, and F. Dupont, Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents, RFC 3776 (June 2004).
6. T. Narten, E. Nordmark, and W.Simpson, Neighbor Discovery for IP version 6, RFC 2461 (December 1998).
7. R. Koodli, Fast Handovers for Mobile IPv6, work in progress (January 2004).
8. M. Gast, 802.11 Wireless Networks: The Definitive Guide (O'REILLY, April 2002).
9. S. Chakrabarti and E. Nordmark, Extension to Sockets API for Mobile IPv6, work in progress (April 2004).
10. S. Kent and R. Atkinson, IP Encapsulating Security Payload (ESP), work in progress, RFC 2406 (November 1998).
11. H. Krawczyk, M. Bellare, and R. Canetti, HMAC: Keyed-Hashing for Message Authentication, RFC 2104 (February 1997).
12. D. Johnson and S. Deering, Reserved IPv6 Subnet Anycast Addresses, RFC 2526 (March 1999).

Chapter 11

# ENHANCED HANDOVER SCHEMES FOR MOBILE IPv6

*HMIPv6, FMIPv6, and Early Binding Update*

## 11.1 INTRODUCTION

MIPv6[1] provides seamless communication services to mobile nodes. This protocol realizes transparency to the transport layer even if IP address of the host is changed due to the movement of it. Thus, the socket connection in the TCP can be maintained even if IP address of the mobile node changes. MIPv6 can support the route optimization feature by allowing the direct communication between mobile nodes and correspondent nodes.

However, the seamless communication using the layer 3 mobility exposes a serious drawback for the realtime applications such as Voice over IP (VoIP), realtime broadcasting and the interactive gaming. There are two approaches to solve this problem. One is Fast Handover.[2] It tries to reduce the handover latency by shortening the time to get new CoA when the mobile node changes its subnet. The other approach tries to reduce the frequency of binding update by employing localized movement management. Hierarchical Mobile IPv6 (HMIPv6)[3] is one of the solutions for this approach.

## 11.2 HIERARCHICAL MOBILE IPv6 (HMIPv6)

### 11.2.1 Concept

HMIPv6 requires new entities called Mobility Anchor Points (MAPs) in the visiting network. MAP acts as a local home agent. It is usually implemented on a router. Mobile nodes have to perform binding updates to home agents and correspondent nodes only when it firstly enters into a MAP domain. When mobile nodes move inside the MAP domain, they do not have to perform binding updates to home agent and correspondent nodes. Binding updates to the MAP are only needed.

HMIPv6 is transparent to home agents and correspondent nodes. Thus, no changes are required to home agents or correspondent nodes. If HMIPv6 is not provided in a visited network, a mobile node performs normal MIPv6 protocol and gets connectivity to Internet.

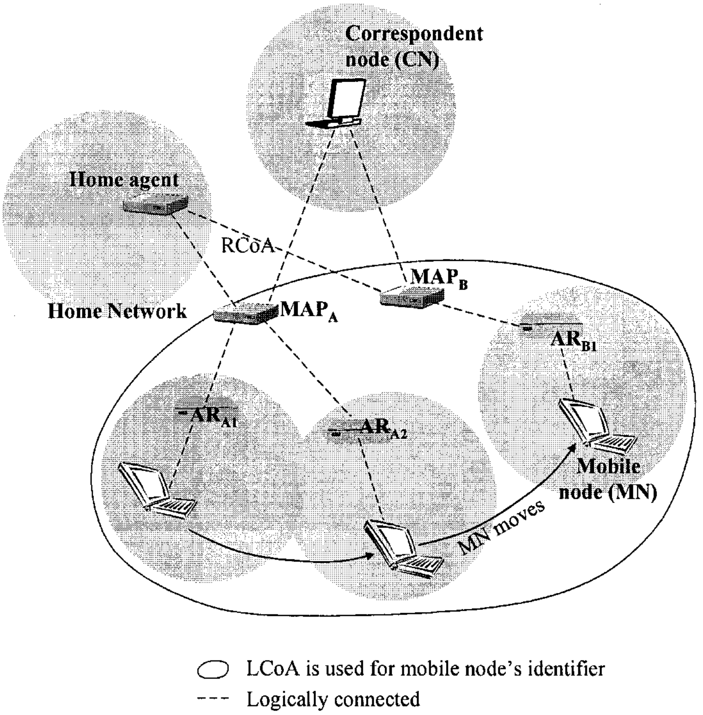


*Figure 11-1.* Hierarchical Mobile IPv6.

Two types of care-of address are defined for HMIPv6 such as on-link care-of address (LCoA) and regional care-of address (RCoA). LCoA is a CoA used as a CoA in the binding update to MAP. RCoA is a CoA in the binding update to home agent and correspondent nodes in case of route optimization. Thus, only RCoA is registered at the mobile node's home agent and correspondent nodes. In other words, there are no communication parties outside of MAP domain holding information about LCoA. Movements within MAP are not informed to outer nodes of MAP. Only movements between MAPs are notified to home agent, which reduces mobile signaling message exchanges between inner MAP domain and outer network.

MAPs keep binding between RCoA and LCoA which has been caused by binding updates to MAP while home agents keep binding between mobile node's home address and the primary care-of address. The primary care-of address registered at the home agent is RCoA.

The mobile node only performs MAP registration rather than home and correspondent registrations when it moves within MAP domain. In such case, no matter how many correspondent nodes communicate with the mobile node, there is only one Binding Update message and Binding Acknowledgement message exchanges between the mobile node and MAP. Only when the mobile node visits different MAP domain, home and correspondent registrations are required.

For example, basic HMIPv6 operation is overviewed in Fig. 11-1, where the $MAP_A$ provides seamless handoff for the mobile node when it moves from $AR_{A1}$ to $AR_{A2}$. Only MAP registration is required in this movement. Thus, only one Local Binding Update message is generated by the mobile node. When the mobile node moves from $MAP_A$ to $MAP_B$, binding updates to home and correspondent nodes should be followed after the MAP registration because mobile node's RCoA is changed.

### 11.2.2 Terminology

In addition to terms defined for MIPv6, new terms are defined as follows:

- Access Router (AR): mobile node's default router.
- Mobility Anchor Point (MAP): MAP is a special router located in a visited network by a mobile node. MAP operates as a local home agent. Multiple MAPs may exist in a visited network.
- Regional care-of address (RCoA): RCoA is an address obtained from the visited network via stateless address configuration mechanism. Prefix information is learned from MAP option of the Router Advertisement message. Only this address type is known to domains outside of MAP.

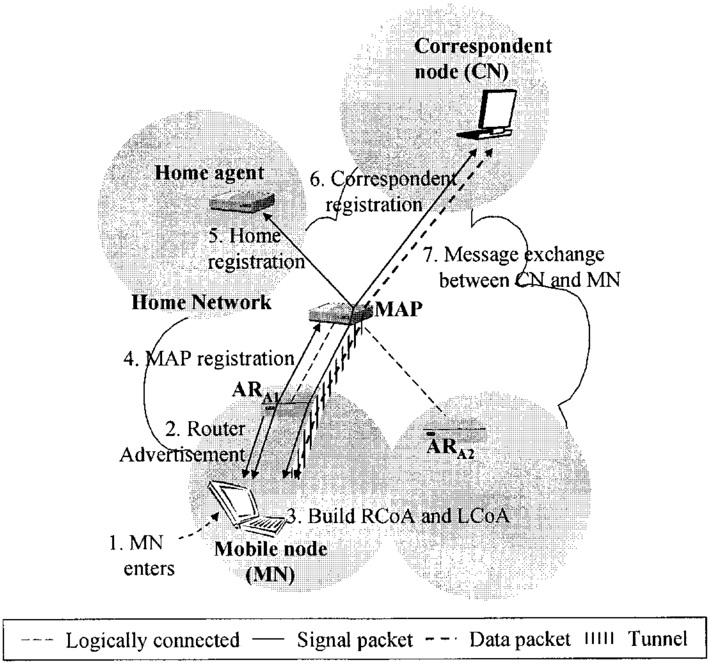
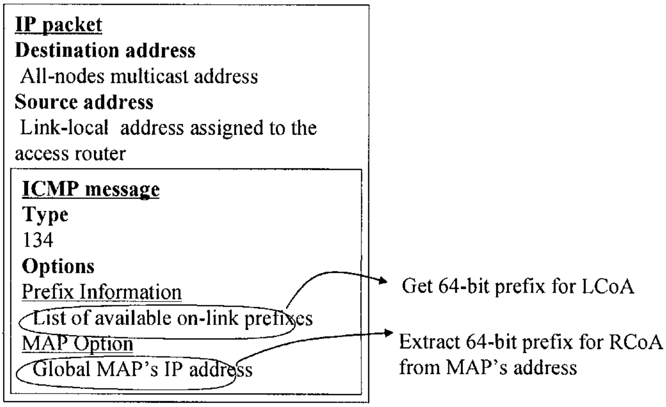


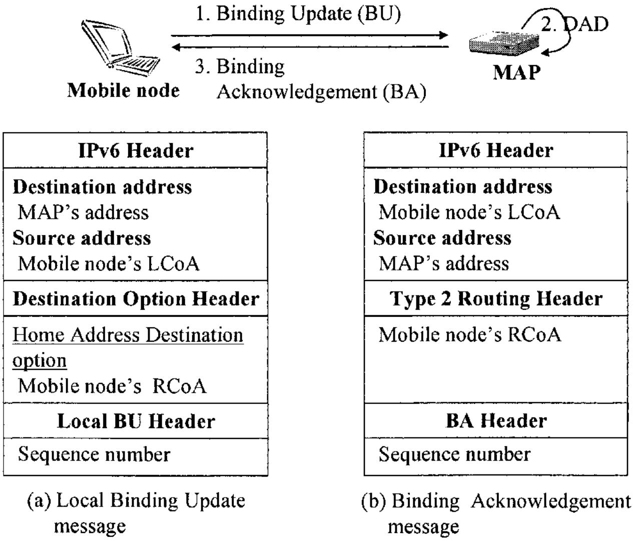
*Figure 11-2.* HMIPv6 example1: when a mobile node enters MAP domain. Messages for home registration may not pass through MAP.


IP packet
**Destination address**
All-nodes multicast address
**Source address**
Link-local address assigned to the access router

ICMP message
**Type**
134
**Options**
Prefix Information
List of available on-link prefixes
MAP Option
Global MAP's IP address

Get 64-bit prefix for LCoA
Extract 64-bit prefix for RCoA from MAP's address

*Figure 11-3.* Router Advertisement message from access router at step 2 in Fig. 11-2.



| **IPv6 Header** |
|---|
| **Destination address**<br>MAP's address<br>**Source address**<br>Mobile node's LCoA |
| **Destination Option Header** |
| Home Address Destination option<br>Mobile node's RCoA |
| **Local BU Header** |
| Sequence number |

(a) Local Binding Update message

| **IPv6 Header** |
|---|
| **Destination address**<br>Mobile node's LCoA<br>**Source address**<br>MAP's address |
| **Type 2 Routing Header** |
| Mobile node's RCoA |
| **BA Header** |
| Sequence number |

(b) Binding Acknowledgement message

*Figure 11-4.* Exchanged messages for MAP Registration at step 4 in Fig. 11-2.

- On-link care-of address (LCoA): LCoA is an address obtained from the visited network via stateless address configuration mechanism. Prefix information is learned from Router Advertisement message from a default router, AR. In original MIPv6, this address type is simply called care-of address.
- Local Binding Update: Local Binding is modified by the Binding Update message to MAP. Once MAP receives this message, it starts DAD[4] for RCoA because RCoA may be already occupied. If there is no conflict, it updates Binding Cache Entry (BCE) for the binding between RCoA and LCoA of the mobile node.
- MAP domain: MAP domain is determined by Router Advertisement messages from access routers, which contain MAP information.

### 11.2.3 Operation

When a mobile node enters into a MAP domain, it will receive Router Advertisements. The mobile node builds its LCoA and RCoA via the stateless autoconfiguration mechanism using information contained in Router Advertisements. They contain prefix information for LCoA in Prefix Information option. 64-bit prefix from Prefix Information option and mobile node's 64-bit interface identifier are concatenated together to build LCoA. MAP address is also contained in Map option of these Router Advertisements. The upper 64 bits of MAP address is used to build RCoA.

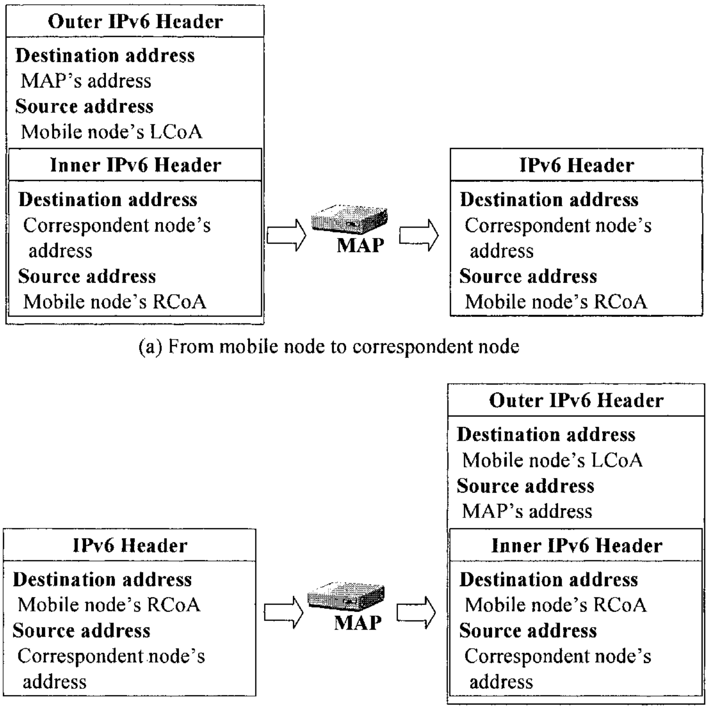
The mobile node performs binding updates to MAP to register its LCoA and RCoA. This process is specified in the following section. Local Binding Update message and Binding Acknowledgement message are exchanged in this process. MAP does not need to know about home address of the mobile node. Once the binding update to MAP is successfully completed, binding updates to home agent and correspondent nodes should be followed. These binding updates are same as specified in MIPv6 standard except that RCoA is used as the care-of address. LCoA is unknown to outside of MAP domain. The example of MAP operation when a mobile node enters into a MAP domain is shown in Fig. 11-2.

Once the binding update to MAP is accomplished, a bi-directional tunnel between MAP and the mobile node is established. Any packet destined to the mobile node's RCoA is captured by MAP. MAP encapsulates it with LCoA as the destination address. Thus, MAP eventually tunnels the packet to the mobile node.

If the mobile node changes its physical location within MAP domain, only binding update to MAP is required. Thus, the old LCoA in BCE will be replaced by a new one. Binding updates to its home agent or correspondent nodes are not necessary. The example when a mobile node moves within MAP domain is described in Fig. 11-6. The MAP domain is determined by Router Advertisement messages from ARs.

## 11.2.4 Binding update to MAP

In addition to binding updates to the home agent and correspondent nodes, binding updates to MAP is defined in HMIPv6. When a mobile node moves into a new MAP domain, it should perform binding update procedures to MAP, home, and correspondent nodes. The binding update to MAP is performed first, and two remaining binding update procedures may be concurrently performed. As shown in Fig. 11-6, when the mobile node moves within the same MAP domain, only binding update to MAP may be required because the existing binding information in correspondent nodes and mobile node's home agent are correct because RCoAs are recorded as CoAs in their BCE. Since binding updates to home agent and correspondent nodes are not required, the binding update latency may be significantly reduced.



**Outer IPv6 Header**
**Destination address** MAP's address
**Source address** Mobile node's LCoA
**Inner IPv6 Header**
**Destination address** Correspondent node's address
**Source address** Mobile node's RCoA

**IPv6 Header**
**Destination address** Correspondent node's address
**Source address** Mobile node's RCoA

(a) From mobile node to correspondent node

**IPv6 Header**
**Destination address** Mobile node's RCoA
**Source address** Correspondent node's address

**Outer IPv6 Header**
**Destination address** Mobile node's LCoA
**Source address** MAP's address
**Inner IPv6 Header**
**Destination address** Mobile node's RCoA
**Source address** Correspondent node's address

(b) From correspondent node to mobile node

*Figure 11-5.* Data packet exchange between a mobile node and correspondent node at step 7 in Fig. 11-2.

The MAP registration process includes a binding update between RCoA and LCoA as well as DAD process for RCoA. Binding Update and Binding Acknowledgement message formats are shown in Fig. 11-4. The detailed procedure is described as follows:

1. When a mobile node visits a network, the node will receive Router Advertisement messages from AR(s). When HMIPv6 is supported in the visited network, the Advertisement message will contain MAP option to allow the mobile to discover MAP address. Available MAP list is kept in the AR and periodically sent using Router Advertisement messages.
2. Once the mobile node receives Router Advertisement messages with the MAP option which contains prefix information for RCoA, it will configure two addresses, RCoA and LCoA by the stateless address configuration mechanism. In detail, the mobile node learns a prefix for LCoA from Prefix Information option and a prefix for RCoA from MAP option in the Router Advertisement message. Then, the mobile node

builds RCoA and LCoA by appending 64-bit prefix to 64-bit interface identifier. The Router Advertisement message format is shown in Fig. 11-3.

3. As the mobile node builds RCoA, the mobile node sends Local Binding Update message to the MAP. Message formats are shown in Fig. 11-4. The Local Binding Update message format is explained in the next section.

   The *A* and *M* flags in the Local Binding Update message should be set to differentiate from the original Binding Update message.

   In the Local Binding Update message to MAP, RCoA is regarded as the home address for the mobile node and contained in the Home Address option. Mobile node's LCoA is used as the source address.
4. Upon MAP receives Local Binding Update message, it binds mobile node's RCoA to its LCoA. Simultaneously, the MAP performs DAD for mobile node's RCoA.

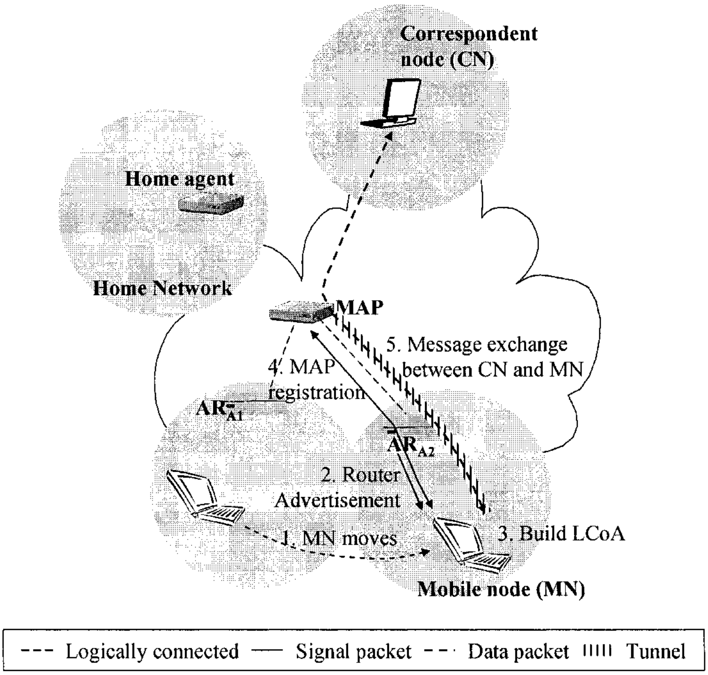


*Figure 11-6.* HMIPv6 example2: when a mobile node moves into new AR within the same MAP domain.

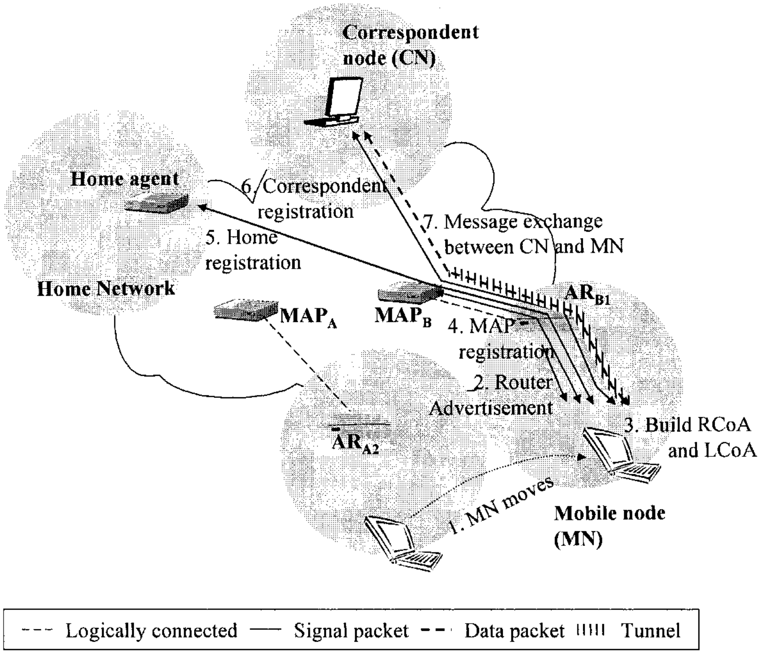


*Figure 11-7.* HMIPv6 Example 3: when a mobile node moves into new MAP domain. Messages for home registration may not pass through MAP.

5. The MAP will return a Binding Acknowledgement message to the mobile node, to indicate the result of the binding update to MAP. The Binding Acknowledgement message displays whether the binding is successfully accomplished. If the binding update fails, an appropriate error code will be returned in the Binding Acknowledgement message.
6. The mobile node must silently discard any acknowledgement packet from MAP without Type 2 Routing Header, which contains mobile node's RCoA.

Once the binding update to the MAP is successfully completed, a bi-directional tunnel between the mobile node and MAP is established. Any communication between the mobile node and correspondent nodes should go through the MAP. The MAP will perform encapsulation for packets from correspondent nodes or mobile node's home agent to the mobile node and also perform decapsulating packets from the mobile node to forward them to correspondent nodes. The format of the packet generated from the mobile node is shown in Fig. 11-5.

- In the outer header, the Source Address field is set to LCoA, and the Destination Address field is set to the MAP's address. MAP's address is learned from the Router Advertisement message from the AR in the visited link.
- In the inner header, the Source Address field is set to RCoA, and Destination Address field is set to the address of the correspondent node.

Correspondent nodes and the home agent only know mobile node's RCoA. They do not know LCoA. Thus, when any of them sends packets to the mobile node, the Destination Address field of packet should be set to the mobile node's RCoA. MAP intercepts them and fetches LCoA corresponding to RCoA by referring the binding cache table. Then, it tunnels packets to the mobile node. For the tunneling, the Destination Address field is filled with LCoA, and Source Address field is filled with the address of MAP. Any delivered packet without Type 2 Routing Header will be discarded in the mobile node.

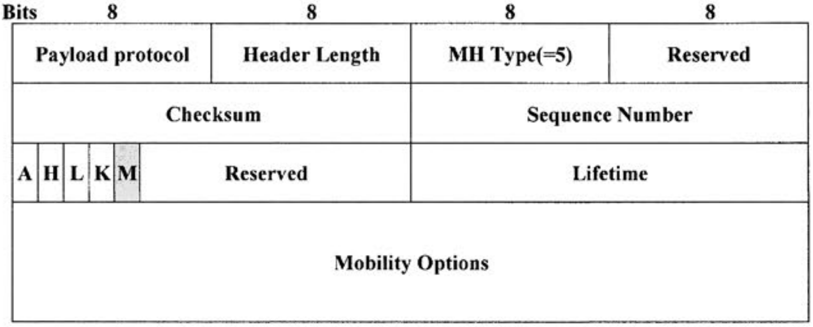
## 11.2.5 Message format

For HMIPv6, Binding Update message of IPv6 is modified. A new option, MAP option is added to Neighbor Discovery protocol.

### 11.2.5.1 Local Binding Update (LBU)

A new flag is added to original Binding Update message to indicate that this message is different from the original Binding Update message and dedicated to MAP, not to home agents or correspondent nodes.

Local Binding Update message is shown in Fig. 11-8. The explanation of each field is given as follows.



| Bits 8 | 8 | 8 | 8 |
|---|---|---|---|
| Payload protocol | Header Length | MH Type(=5) | Reserved |
| Checksum | | Sequence Number | |
| A H L K M Reserved | | Lifetime | |
| Mobility Options | | | |

*Figure 11-8.* Local Binding Update message format.

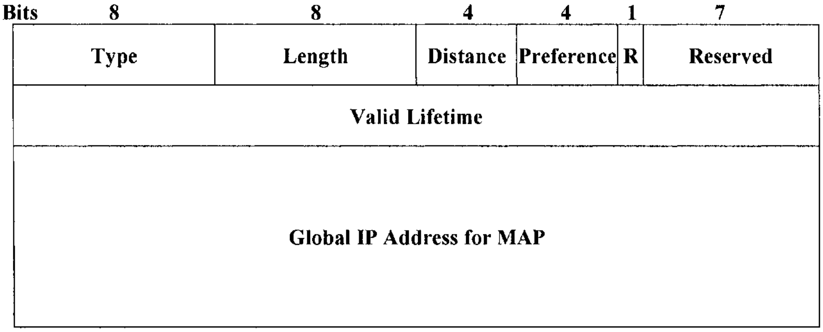


*Figure 11-9.* MAP option format.

- *M* flag: *M* flag is defined in the Reserved field of the original Binding Update message. When the *M* flag is set to 1, it indicates a binding update to MAP.
- The Reserved field length is shortened to 13 bits.
- The other remaining fields are same as those from the original Binding Update message of MIPv6 in Fig. 10-19.

### 11.2.5.2 Neighbor Discovery extension – MAP option

MAP option is added to the Neighbor Discovery which is specified in RFC 2461.[5] Especially, MAP option is contained in the Option field of Router Advertisement messages to allow mobile nodes to learn MAP address from ARs. The format of Router Advertisement message is shown in Fig. 5-6. MAP option format is shown in Fig. 11-9. The detailed explanation of each field of MAP option is described as follows.

- The Type field specifies IPv6 Neighbor Discovery option, as shown in Table 5-4. Type value for MAP option is not defined yet.
- The Length field specifies the length of the option and should be set to 3.
- The Distance field specifies the distance between MAP and a receiving node of this message. Default value is 1. The distance does not mean the number of hops between MAP and a mobile node. MAP distance has same meaning within the same domain. It is set to 1 when MAP and a mobile node are present in the same link. The highest distance number indicates the furthest distance between MAP and a mobile node.
- The Preference field specifies the preference of a MAP. This field contains decimal value, and the value 15 means the highest preference.

- When the *R* bit is set to 1, a mobile node should build RCoA based on the prefix information in the MAP option.
- The Valid Lifetime field specifies both of the preferred and valid prefix lifetimes in a MAP's domain in second.
- The Global IP Address for MAP field contains one of MAP's global addresses. Once a mobile node receives the Router Advertisement message with the MAP option, it extracts 64-bit prefix for RCoA from the address of the MAP.

## 11.3 FAST HANDOVER FOR MOBILE IPv6

### 11.3.1 Concept

To reduce delay and packet loss, a fast handover scheme (FMIPv6)[2] is introduced into MIPv6. In the fast handover, several portions of the layer 3 handover are performed in advance prior to the handover, such as new care-of address configuration and movement detection[4] to reduce the handover latency. A tunnel is established between a currently attached access router and an anticipated access router not to lose packets from correspondent nodes during the handover. The fast handover enables the mobile node to quickly detect that it has moved to a new subnet by providing the new access point and the associated subnet prefix information when the mobile node is still connected to its current subnet. Operations of the fast handover are composed of predictive mode and reactive mode.

### 11.3.2 Terminology

Following terminologies are defined for the Fast Handover.

- Access Point (AP): a layer 2 device connected to an IP subnet that offers wireless connectivity to a mobile node.[6] An access pointer identifier (AP-ID) refers to the access point's layer 2 address. Sometimes, AP-ID is also referred to as a base station subsystem ID (BSSID).
- Previous Access Router (PAR): mobile node's default router prior to its handover.
- New Access Router (NAR): mobile node's anticipated default router subsequent to its handover.
- Previous care-of address (PCoA): mobile node's care-of address which is valid on the previous access router's subnet

- New care-of address (NCoA): mobile node's care-of address which is valid on new access router's subnet
- Router Solicitation for Proxy Advertisement (RtSolPr): a message from the mobile node to the previous access router requesting information for a potential handover.
- Proxy Router Advertisement (PrRtAdv): a message from the previous access router to the mobile node that provides information about neighboring links facilitating expedited movement detection. The message also acts as a trigger for the network-initiated handover.
- Access Point Identifier, Access Router Information tuple, [AP-ID, AR-Info]: It contains an access router's layer 2 and IP addresses, and prefix which is valid on the interface to which the access point (identified by access point identifier) is attached. The triplet, [router's layer 2 address, router's IP address, prefix], is called access router information.
- Assigned Addressing: a particular type of new care-of address configuration in which the new access router assigns an IPv6 address for the mobile node.
- Fast Binding Update (FBU): a message from the mobile node instructing its previous access router to redirect its traffic (towards new access router).
- Fast Binding Acknowledgment (FBAck): a message from the previous access router in response to Fast Binding Update.
- Fast Neighbor Advertisement (FNA): a message from the mobile node to the new access router to announce attachment, and to confirm the use of new care-of address when the mobile node has not received Fast Binding Acknowledgment.
- Handover Initiate (HI): a message from the previous access router to the new access router regarding a mobile node's handover.
- Handover Acknowledge (HAck): a message from the new access router to the previous access router as a response to handover initiate.

### 11.3.3 Operation

The mobile node initiates the fast handover when a layer 2 trigger takes places. Then, the mobile node sends a Router Solicitation for Proxy advertisement message to its access router to resolve one or more access point identifiers to subnet-specific information. In response, the access router (e.g. previous access router) sends a Proxy Router Advertisement message which contains one or more [AP-ID, AR-Info] tuples.

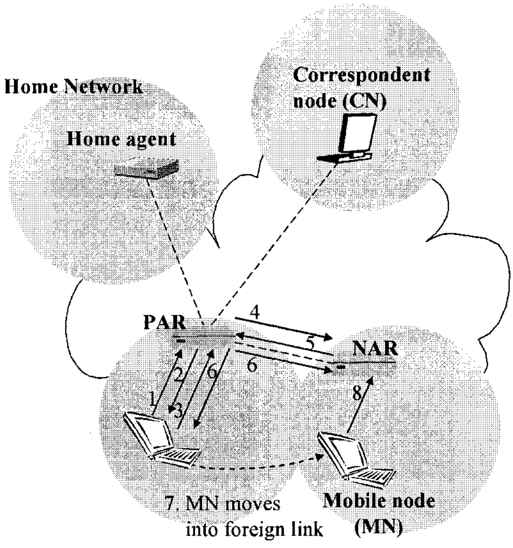
With the information provided in the Proxy Router Advertisement message, the mobile node forms a prospective new care-of address and sends a Fast Binding Update message. The purpose of the Fast Binding update is to make the previous router to bind the previous care-of address to the new care-of address and establish tunnel between the previous access router and the new access router, so that packets arrived from correspondent nodes can be tunneled to the new location of the mobile node.

The Fast Binding Update message should be sent from the mobile node at the previous access router's link if possible. When the mobile node could not send the Fast Binding Update message at the previous access router's link, the Fast Binding Update message is sent from the new link. It is encapsulated within a Fast Neighbor Advertisement message to ensure that the new care-of address does not conflict with an address already in use by some other node on link.

When the previous access router receives the Fast Binding Update message, it sends Handover Initiate message to the new access router to determine whether the new care-of address is acceptable at the new access router. When the new access router verifies the new care-of address, DAD is performed to avoid duplication on links when stateless address autoconfiguration is used. Confirmed new care-of address must be returned in the Handover Acknowledge message from the new access router. Then, the previous access router must in turn provide the new care-of address in a Fast Binding Acknowledgment. Thus, new care of address is determined by the exchange of Handover Initiate and Handover Acknowledge messages.

DAD adds delays to a handover. The probability of interface identifier duplication on the same subnet is very low. However, this probability can not be neglected. In the fast handover, certain precautions are necessary to minimize the effects of duplicate address occurrences. In some cases, the new access router may already have the knowledge required to assess whether the mobile node's address is a duplicate or not before the mobile node moves to the new subnet. The result of this search is sent back to the previous access router in the Handover Acknowledge message. The new access router can also rely on its trust relationship with the previous access router before providing forwarding support for the mobile node. That is, it may create a forwarding entry for the new care-of address subject to approval from the previous access router which it trusts.

For preventing packet loss, this protocol provides an option to indicate request for buffering at the new access router in the Handover Initiate message. When the previous access router requests this feature for the mobile node, it should also provide its own support for buffering. Such buffering can be useful when the mobile node leaves without sending the Fast Binding Update message from the previous access router's link. The previous access router should stop buffering after processing the Fast Binding Update message.



⟶ Signaling message ---- Data path during handover
PAR: previous access router NAR: new access router
1. RtSolPr: Router Solicitation for Proxy Advertisement
2. PrRtAdv: Proxy Router Advertisement 3. FBU: Fast Binding Update
4. HI: Handover Initiate 5. HAck: Handover Acknowledge
6. FBAck: Fast Binding Acknowledgement 8. FNA: Fast Neighbor Advertisement

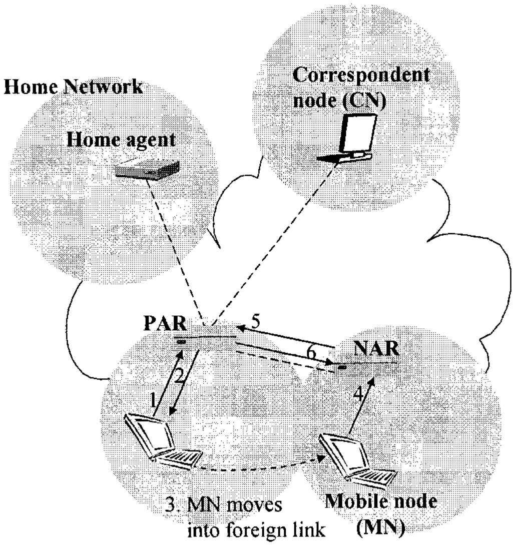
*Figure 11-10.* Fast handover by predictive mode.

Depending on whether a Fast Binding Acknowledgment message is received on the previous link, there exist two kinds of mode of operation. The first one is predictive mode of operation. In this mode of operation, the mobile node receives the Fast Binding Acknowledgment message on the previous link. This means that packet tunneling would already be in progress by the time when the mobile node handovers to the new access router. As soon as the mobile node establishes link connectivity with the new access router, it should send a Fast Neighbor Advertisement message immediately, so that buffered packets can be forwarded to the mobile node right away. Fig. 11-10 shows predictive mode of operation of the fast handover.

The other is reactive mode of operation. In this mode, the mobile node does not receive the Fast Binding Acknowledgment message on the previous link. This occurs if the mobile node has not sent the Fast Binding Update. The other case is that the mobile node has left the link after sending the Fast Binding Update message but before receiving the Fast Binding Acknowledgment message.

Without receiving the Fast Binding Acknowledgment message, the mobile node can not ascertain whether the previous access router has successfully processed the Fast Binding Update message. Hence, as soon as it attaches to the new access router, it sends a Fast Binding Update message. The mobile node should encapsulate the Fast Binding Update message in the Fast Neighbor Advertisement message, which enables the new access router to forward packets immediately when the Fast Binding Update message has been processed and allows the new access router to verify if the new care-of address is acceptable.

The new access router must discard the inner Fast Binding Update message and send a Router Advertisement message with a Neighbor Advertisement Acknowledge option. In Router Advertisement message, the new access router may include an alternate IP address for the mobile node to use if it detects that the new care-of address is already in use when processing the Fast Neighbor Advertisement message. This discarding can avoid rare but undesirable outcomes resulting from address collision. Fig. 11-11 shows reactive mode of operation of the fast handover.



⟶ Signaling message ---- Data path during handover
PAR: previous access router NAR: new access router
1. RtSolPr: Router Solicitation for Proxy Advertisement
2. PrRtAdv: Proxy Router Advertisement 4. FNA: Fast Neighbor Advertisement
5. FBU: Fast Binding Update 6. FBAck: Fast Binding Acknowledge

*Figure 11-11.* Fast handover by reactive mode.

The registrations of the new care-of address to the home agent and correspondent nodes are performed after it is registered at the new access router. These registrations are the same procedure as MIPv6.

## 11.3.4 Message formats

Four new ICMP message types are defined for FMIPv6. General ICMP message format is shown in Fig. 4-1. The first three fields, such as Type, Code, Checksum, should be present in all of ICMP messages.[7] The Type field of all four messages is still under discussion.[8] After three fields, 8-bit Subtype field is defined to distinguish ICMP messages for HMIPv6 from original ICMP messages.

- Router Solicitation for Proxy Advertisement
- Proxy Router Advertisement
- Handover Initiate
- Handover Acknowledgement

A message contained in the Mobility Header following IPv6 Header is called the mobility message. In addition to mobility message defined in MIPv6, three new mobility messages defined for FMIPv6. Mobility Header format is shown in Fig. 10-13.

- Fast Binding Update
- Fast Binding Acknowledgment
- Fast Neighbor Advertisement

These new seven message types are explained in following subsections.

### 11.3.4.1 Router Solicitation for Proxy Advertisement (RtSolPr)

A mobile node sends the Router Solicitation for Proxy Advertisement to request Proxy Router Advertisement from routers. The message format is shown in Fig. 11-12. The message format follows the same ICMP message format in Fig. 4-1. For unexplained fields, refer to Chapter 4.

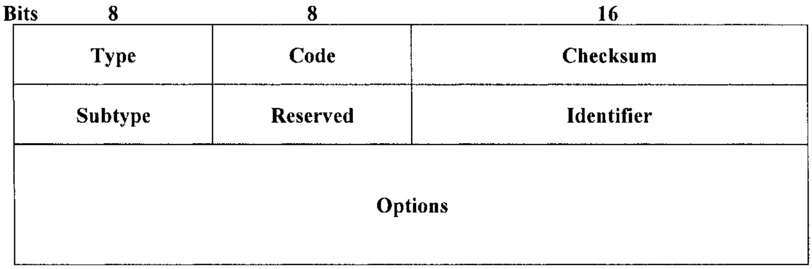
When the Subtype field is set to 2, it indicates Router Solicitation for Proxy Advertisement message as follows:

- In IP Header:

  The source address of this ICMP message is chosen by the source address selection rules, specified in Chapter 4.

  The destination address may be the address of access router or all routers multicast address.

  If any security association for the IP Authentication Header exists between a sender and a destination address, then the sender should include Authentication Header.



| Bits 8 | 8 | 16 |
|---|---|---|
| Type | Code | Checksum |
| Subtype | Reserved | Identifier |
| Options | | |

*Figure 11-12.* Router Solicitation for Proxy Advertisement (RtSolPr) message format.

- In ICMP message:

  The Code field is set to 0.

  The Subtype is set to 2.

  The Reserved field should bet set to 0 by the sender and ignored by the receiver.

  The Identifier field must be set by the sender so that replies can be matched to this Solicitation.

  In the Option field, the source link layer address of the mobile node and link layer address or identification of AP which the mobile node requests advertisement messages should be included.

### 11.3.4.2 Proxy Router Advertisement (PrRtAdv)

Access routers send out the Proxy Router Advertisement message gratuitously as a response to the Router Solicitation for Proxy Advertisement message from a mobile node. This message will provide link layer address, IP address and subnet prefixes of neighboring routers to the soliciting node. The message format of the Proxy Router Advertisement is identical to the Router Solicitation for Proxy Advertisement message.

When the Subtype field is set to 3, it indicates Proxy Router Advertisement message as follows:

- In IP Header:

  The source address should be set to the link local address assigned to router's interface.

  The destination address may be copied form the source address of the received Router Solicitation for Proxy message. If this message is not solicited, the router directs a mobile node to handover.

If any security association for the IP Authentication Header exists between a sender and a destination address, then the sender should include Authentication Header.

- In ICMP message:
  The Code value for Proxy Router Advertisement message is listed in Table 11-1.
  The Subtype is set to 3.
  The Reserved field should bet set to 0 by the sender and ignored by the receiver.
  The Identifier field is copied from the previous Router Solicitation for Proxy Advertisement message. If this message is not solicited, this field should be zero.
  In the Option field, the source link layer address of router and link layer address or identification of AP copied from the Proxy Router Advertisement message should be included.

### 11.3.4.3 Handover Initiate (HI)

The Handover Initiate is an ICMPv6 message sent by an access router to another access router to initiate the process of a mobile node's handover. Fig. 11-13 shows the Handover Initiate message format.

When the Subtype field is set to 4, it indicates Handover Initiate message as follows.

- In IP Header:
  The source address of this ICMP message is the IP address of previous access router.
  The destination address is IP address of the new access router.
  IP Authentication Header should be included.[9]
- In ICMP message:
  When PCoA is used as a source IP address, the previous access router will set the Code field to 0. When any other address type is used as a source address, the Code field is set to 1.
  The Subtype is set to 4.
  The Reserved field should bet set to 0 by the sender and ignored by the receiver.
  The *S* field is an assigned address configuration flag. When it is set, this message requests a new care-of address to be returned by the destination.
  The *U* field is a buffer flag. When it is set, the destination should buffer any packets towards the node indicated in the options of this message.
  The Identifier field must be set by the sender so that replies can be matched to this Solicitation.

Valid options are a link layer address of mobile node, a previous care-of address and a new care-of address.

### 11.3.4.4 Handover Acknowledge (HAck)

The Handover Acknowledge message is a new ICMPv6 message that must be sent as a reply to the Handover Initiate message. A message format of the Handover Acknowledge is identical to the Router Solicitation for Proxy Advertisement message.

When the Subtype field is set to 5, it indicates Handover Acknowledge message as follows:

- In IP Header:

  The source address is copied from the destination address of the received Handover Initiate message.

  The destination address copied from the source address of Handover Initiate message.

  IP Authentication Header should be included.[9]

*Table 11-1.* Code value for Proxy Router Advertisement message.

| Code value | Description |
|---|---|
| 0 | The mobile node should use [AP-ID, AR-INFO] for movement detection and NCoA generation. |
| 1 | This Proxy Router Advertisement message is unsolicited and used as network-initiated handover trigger. |
| 2 | No new router information is present. The mobile node process along the Option-Code field in the New Access Point LLA option. (section 3.5.3) |
| 3 | No new router information is present only for a subnet of APs requested. |
| 4 | This Proxy Router Advertisement message is similar one of Code 1 but is not a handover trigger. |

*Table 11-2.* Code value for Handover Acknowledge message.

| Code value | Description |
|---|---|
| 0 | The NAR accepts Fast Handover, and new care-of address is valid. |
| 1 | The NAR accepts Fast Handover, and new care-of address is not valid. |
| 2 | The NAR accepts Fast Handover, and new care-of address is in use. |
| 3 | The NAR accepts Fast Handover, and new care-of address is assigned. |
| 4 | The NAR accepts Fast Handover, and new care-of address is not assigned |
| 128 | The NAR does not accept Fast Handover. Unspecified reason. |
| 129 | Administratively prohibited. |
| 130 | Insufficient resources. |

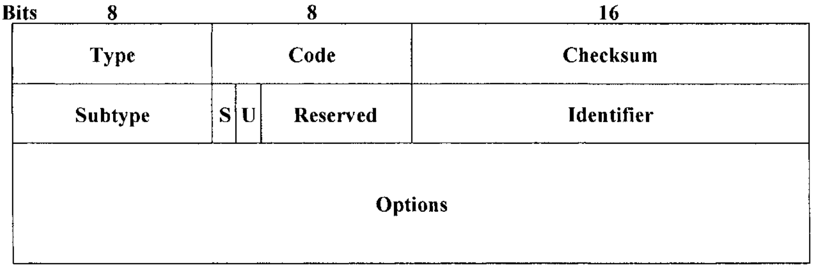


*Figure 11-13.* Handover Initiate (HI) message format.

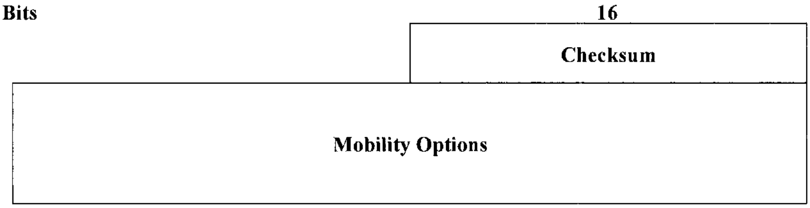


*Figure 11-14.* Fast Neighbor Advertisement (FNA) message format.

- In ICMP message:
  The Code value for Proxy Router Advertisement message is listed in Table 11-2.
  The Subtype is set to 5.
  The Reserved field should bet set to 0 by the sender and ignored by the receiver.
  The Identifier field is copied from the Handover Initiate message.
  Valid option is a new care-of address.

### 11.3.4.5 Fast Binding Update (FBU)

The Fast Binding Update message is identical to the MIPv6 Binding Update message. However, the processing rules are slightly different. The source IP address is a previous care-of address when the Fast Binding Update message is sent from the previous access router's link, and the source IP address is a new care-of address when sent from the new access router's link.

#### 11.3.4.6 Fast Binding Acknowledgment (FBAck)

The Fast Binding Acknowledgment message is sent by the previous access router to acknowledge the receipt of a Fast Binding Update message in which the '*A*' flag is set. The Fast Binding Acknowledgment message should not be sent to the mobile node before the previous access router receives a Handover Acknowledge message from the new access router. The Fast Binding Acknowledgment may also be sent to the mobile node on the old link.

#### 11.3.4.7 Fast Neighbor Advertisement (FNA)

A mobile node sends a Fast Neighbor Advertisement message to announce itself to the new access router. When the mobility header type is Fast Neighbor Advertisement, the Payload Protocol field (=Next Header field) may be set to IPv6 to assist Fast Binding Update encapsulation. Fig. 11-14 shows the Fast Neighbor Advertisement message format.

### 11.3.5 Options

Option-Code field is added to the general option format.

#### 11.3.5.1 IP Address option

This option is sent in the Proxy Router Advertisement, the Handover Initiate, and Handover Acknowledge messages. Fig. 11-15 shows the IPv6 Address Option message format. Following Option-Code field values are defined.

- 1: old care-of address.
- 2: new care-of address.
- 3: new access router's IP address.

#### 11.3.5.2 New Router Prefix Information option

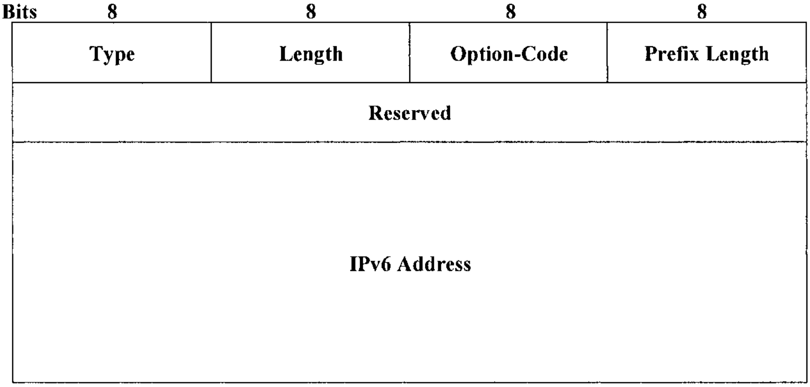
This option is sent in the Proxy Router Advertisement message to provide the prefix information valid on the new access router.

#### 11.3.5.3 Link-Layer Address (LLA) option

Following Option-Code field values are defined.

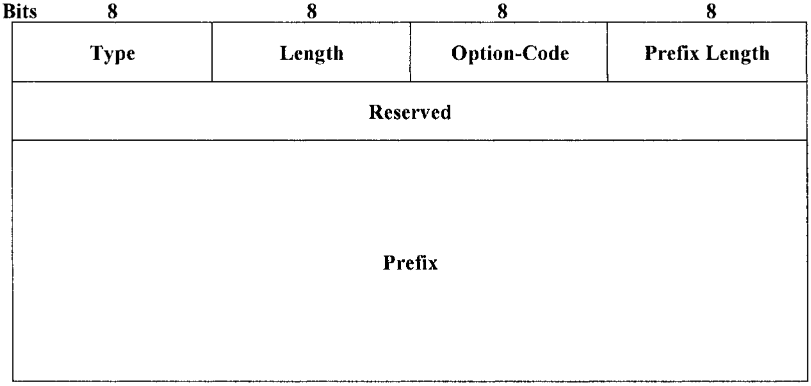
- 0: wildcard requesting resolution for all nearby access points.
- 1: link-layer address of the new access point.

- 2: link-layer address of the mobile node.
- 3: link-layer address of the new access router.
- 4: link-layer address of the source of router solicitation for proxy advertisement or proxy router advertisement message.
- 5: the access point identified by the link layer address belongs to the current interface of the router.
- 6: no prefix information available for the access point identified by the link layer address.
- 7: no fast handovers support available for the access point identified by the link layer address.



| Bits 8 | 8 | 8 | 8 |
|---|---|---|---|
| Type | Length | Option-Code | Prefix Length |
| Reserved | | | |
| IPv6 Address | | | |

*Figure 11-15.* IPv6 Address option format.



| Bits 8 | 8 | 8 | 8 |
|---|---|---|---|
| Type | Length | Option-Code | Prefix Length |
| Reserved | | | |
| Prefix | | | |

*Figure 11-16.* New Router Prefix Information option format.

| Bits | 8 | 8 | 8 | 8 |
|---|---|---|---|---|
| | **Type** | **Length** | **Option-Code** | **LLA** |

*Figure 11-17.* Link-layer Address option format.

| Bits | 8 | 8 | 8 | 8 |
|---|---|---|---|---|
| | **Type** | **Length** | **Option-Code** | **Status** |
| | **Reserved** | | | |

*Figure 11-18.* Neighbor Advertisement Acknowledgment option format.

### 11.3.5.4 Neighbor Advertisement Acknowledgment (NAACK) option

The new access router responds to the Fast Neighbor Advertisement message with the Neighbor Advertisement Acknowledgment option to notify the mobile node to use a different new care-of address if there is address collision.

Following Status field values are defined.

- 1: the new care-of address is invalid.
- 2: the new care-of address is invalid, use the supplied care-of address. The new care-of address must be present following the reserved field.
- 123: link layer address unrecognized.

## 11.4 EARLY BINDING UPDATE

### 11.4.1 Concept

In MIPv6, a mobile node performs correspondent registration after the home registration. The mobile node must process a return routability procedure before binding update procedure. The return routability procedure consists of two address tests; a home address test and a care-of address test, as explained in Chapter 10.

Home Test Init and Home Test messages are used for the home address test, and Care-of Test Init and Care-of Test messages are used for the care-of address test. In MIPv6, the Home Test Init and Care-of Test Init messages are sent simultaneously from the mobile node to the correspondent node. Home test and Care-of Test messages are sent from the correspondent node to the mobile node as the response to former two messages. The home address test provides reasonable assurance to the correspondent node that the mobile node is the legitimate owner of the home address which the mobile node claims to own. The care-of address test provides reasonable assurance to the correspondent node that the mobile node is addressable through the care-of address which the mobile node wishes to register.

The return routability procedure requires minimum processing resources of both the mobile node and the correspondent node. A weak point, however, is that the two address tests, though typically processed in parallel, consist of a considerable fraction of the latency of the binding update procedure. These two address tests are potentially performed along a very long distance. Latencies of two address tests make a harmful effect on the seamless mobility support for the mobile nodes. Thus, Early Binding Update scheme has been proposed to reduce these latencies.

The Early Binding Update procedure is a minor extension to the binding update procedure of MIPv6.[10] This procedure is fully compatible to the binding update procedure defined in MIPv6. In Early Binding Update, the messages of return routability procedure are the same as the ones used for the return routability procedure in MIPv6. All messages related to binding update of MIPv6 remain unchanged from their original meanings.

With an Early Binding Update procedure, the home address test is triggered by the mobile node whenever a handover is expected before the home registration. When the mobile node moves into a new network, it has a fresh Home Keygen Token. The mobile node starts the home registration at the new network. Then, the mobile node initiates the care-of address test with the correspondent node.

The mobile node uses two new binding update message; an Early Binding Update and Early Binding Acknowledgment messages. The mobile node sends the Early Binding Update message to the correspondent node when it wants to register a new care-of address without performing care-of address test. The mobile node requests the correspondent node to return the Early Binding Acknowledgment message. All messages of binding update procedure in the MIPv6 remain unchanged with their original meanings. Therefore, the mobile node initiates an Early Binding Update procedure without knowledge whether or not the correspondent node supports it. In case that the correspondent node does not support the Early Binding Update procedure, the general binding update procedure in the MIPv6 is automatically performed.

### 11.4.2 Terminology

Following terminologies are defined for the Early Binding Update scheme.

- Early Binding Update (EBU): a message from the mobile node to register a new care-of address without performing care-of address test.
- Early Binding Acknowledgment (EBA): a message from the correspondent node in response to Early Binding Update message.

### 11.4.3 Operation

Whenever a handover is expected, a mobile node triggers a home address test, a Home Test Init message (1) and a Home Test message (2) in Fig. 11-19. If the handover is not expected, the mobile node repeats periodically the home address test. Thus, the mobile node has a fresh Home Keygen Token when it moves between networks. The Home Test Init and Home Test messages used for the Early Binding Update are the same as the ones used for the binding update procedure in the Mobile IPv6.

The interval time by which the mobile node repeats the home address test should be conservatively determined as the minimum time, MAX_TOKEN_LIFETIME[23] in the MIPv6. It is time that a token is expected to be valid.

The Home Test Init and Home Test messages are typically protected by an IPsec tunnel between the mobile node and its home agent. The home agent must update the corresponding security association to the mobile node's new care-of address when the mobile node moves between networks. With the Early Binding Update procedure, the changed new care-of address does not affect a home address test because the home address test is performed before the mobile node's movement.

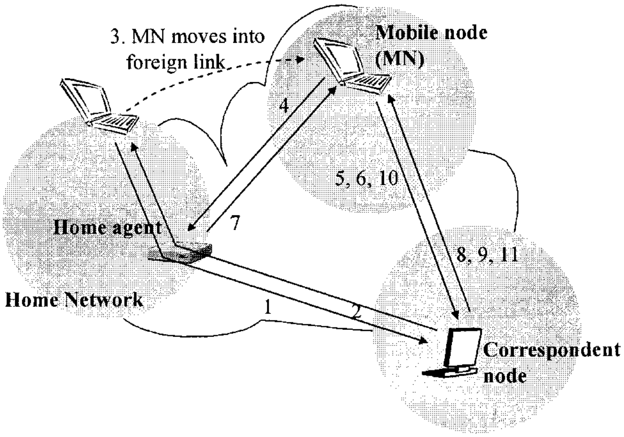
This is a difference to the general binding update procedure in the MIPv6, where the home agent must update the security association upon performing the home registration. When the mobile node detects its movement between networks, it configures a new care-of address. The mobile node sends a Binding Update message (3) to its home agent. The Binding Update message informs the home agent of the mobile node's new location information, new

[23] MAX_TOKEN_LIFETIME value is set to 210 seconds. When this lifetime is expired, this token should not be used for authentication of binding update no longer.

care-of address. It is protected by means of the IPsec tunnel and security association between the mobile node and the home agent.

Having sent the Binding Update message to its home agent, the mobile node creates an Early Binding management key, which is a one-way hash on the Home Keygen Token from the most recently received Home Test message. The mobile node then sent an Early Binding Update message (4) to the correspondent node to be authenticated with the Early Binding management key. The Early Binding management key does not incorporate a Care-of Keygen Token as the binding management key used for a binding update procedure in the MIPv6 does. Since the Care-of Keygen Token proves that the mobile node can be reached with the given care-of address, it provides reason that the Early Binding management key is sufficient for authenticating the Early Binding Update message.

The mobile node will provide proof of its addressability at the new care-of address at a later step when it sends a Binding Update message to the correspondent node. The mobile node will need to create binding management key in the MIPv6 using both the Home Keygen Token and the Care-of Keygen Token.



1. Home Test Init (HoTI)
2. Home Test (HoT)
4. Binding Update (BU)
5. Early Binding Update (EBU)
6. Care-of Test Init (CoTI)
7. Binding Acknowledgment (BA)
8. Early Binding Acknowledgment (EBA)
9. Care-of Test (CoT)
10. Binding Update (BU)
11. Binding Acknowledgment (BA)

*Figure 11-19.* Procedure of Early Binding Update.

The Early Binding Update message includes a message authentication code and the Home Nonce Index copied from the most recently received Home Test message. There are two different points between the Early Binding Update and the Binding Update message used for the binding update procedure in the MIPv6. First, the message authentication code in the Early Binding Update message is generated with the Early Binding management key, which does not incorporate the Care-of Keygen Token. Second, the Early Binding Update message does not include the Care-of Nonce Index. The mobile node requests the correspondent node to return an Early Binding Acknowledgment message by setting the *A* flag in the Early Binding Update message. In the Early Binding update procedure, the Early Binding Update message will arrive at the correspondent node ahead of all data packets which the mobile node has sent from its new care-of address. Thus, the mobile node can use its new care-of address about one round-trip time earlier with Early Binding Updates than with the binding updates procedure in the MIPv6.

The mobile node sends the Care-of Test Init message (5) to the correspondent node in parallel with sending the Early Binding Update message. The Care-of Test Init message used for an Early Binding Update is the same as the one used for the binding update procedure in the MIPv6. When the home agent receives the Binding Update message from the mobile node, it registers the mobile node's new care-of address. If the home agent maintains an IPsec tunnel and security association with the mobile node, it also updates the corresponding security association to the new care-of address.[11]

The home agent sends the Binding Acknowledgment message (6) to the mobile node to inform successful care-of-address registration. The Binding Acknowledgment message is authenticated by means of the IPsec tunnel and security association between the mobile node and the home agent. When the correspondent node receives the Early Binding Update message, it can reproduce the Home Keygen Token with the help of the Home Nonce Index.

The token allows the correspondent node to reproduce the Early Binding management key. The correspondent node can then compute the message authentication code. If the result matches the message authentication code in the Early Binding Update message, the mobile node must have received the Home Keygen Token to construct the Early Binding management key with which the message authentication code was produced. Therefore, the correspondent node can assume that the mobile node is the legitimate owner of the home address. The message authentication code validates the Early Binding Update message's integrity and authenticity.

If the result matches the message authentication code in the Early Binding Update message, the correspondent node creates a tentative Binding

Cache Entry with the mobile node's new care-of address. The correspondent node uses the mobile node's new care-of address upon creating Binding Cache Entry. Thus, with Early Binding Updates, the correspondent node can use the mobile node's new care-of address about one round-trip time sooner than with the binding update procedure in the MIPv6.

Since the care-of-address test still has to be performed for the mobile node's new care-of address, the new Binding Cache Entry's lifetime is limited to TENTATIVE_BINDING_LIFETIME.[24] The lifetime will be extended when the correspondent node completes a correspondent registration with the mobile node. In case that the *A* flag in the Early Binding Update message is set, the correspondent node sends the Early Binding Acknowledgement message (7) to the mobile node to indicate the tentative care-of-address registration.

When the correspondent node receives the Care-of Test Init message from the mobile node, it sends the Care-of Test message (8) to the mobile node. The Care-of Test message used for the Early Binding Update is the same as the one used for the binding update procedure in the MIPv6.

The Care-of Test message includes the Care-of Keygen Token. The mobile node uses this Care-of Keygen Token together with the Home Keygen Token from the most recently received Home Test message to produce a Binding management key. This Binding management key is a one-way hash on the concatenation of the Home Keygen Token and the Care-of Keygen Token. It is equivalent to the Binding management key used for the binding update procedure in the MIPv6.

The mobile node then sends the normal Binding Update message (9) to the correspondent node. This Binding Update message includes a message authentication code which is produced with the Binding management key. It is equivalent to the Binding Update message used for the binding update procedure in the MIPv6. The mobile node requests the correspondent node to return a Binding Acknowledgment message by setting the *A* flag in the Binding Update message. When the correspondent node receives the Binding Update message, it verifies the message's authenticity and integrity. If the result matches with the message authentication code in the Binding Update message, the correspondent node extends the lifetime of the mobile node's tentative Binding Cache Entry to the regular lifetime proposed in MIPv6. In case that the *A* flag in the Binding Update message is set, the correspondent node sends the Binding Acknowledgment message (10) to the mobile node to inform successful care-of-address registration. This Binding

[24] TENTATIVE_BINDING_LIFETIME value is defined as 3seconds.

Acknowledgment message is equivalent to the one used for the binding update procedure in the MIPv6.

## REFERENCES

1. D. Johnson, C. Perkins, and J. Arkko, Mobility Support in IPv6, RFC 3775 (June 2004).
2. R. Koodli, Fast Handovers for Mobile IPv6, work in progress (October, 2004).
3. H. Soliman, C. Catelluccia, K. Malki, and L. Bellier, Hierarchical Mobile IPv6 mobility management (HMIPv6), work in progress (December 2004).
4. S. Thomson and T. Narten, IPv6 Stateless Address Autoconfiguration, RFC 2462 (December 1998).
5. T. Narten, E. Nordmark, and W.Simpson, Neighbor Discovery for IP version 6, RFC 2461 (December 1998).
6. M. Gast, 802.11 Wireless Networks: The Definitive Guide (O'REILLY, April 2002).
7. A. Conta and S. Deering, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC 2463 (December 1998).
8. J. Kempf, Instructions for Seamoby and Experimental Mobility Protocol IANA Allocations, work in progress (June 2004).
9. S. Kent, R. Atkinson, IP Authentication Header, RFC 2402 (November 1998).
10. C. Vogt, R. Bless, M. Doll, and T. K'fner, Early Binding Updates for Mobile IPv6, work in progress (February 2004).
11. P. Nikander, Mobile IP version 6 Route Optimization Security Design Background, work in progress (October 2004).

Chapter 12

# SECURITY IN MOBILE IP

## 12.1 INTRODUCTION

Several important security implications for Mobile IP are explained in this chapter. VPN and firewall are two of the most widely used security technologies nowadays. Since they are not designed for mobile terminals, careful considerations are needed to be effective even for mobile terminals. In the next section, Mobile IP with VPN is considered. Since no concrete work has been done for Mobile IPv6 (MIPv6) with VPN so far, Mobile IPv4 (MIPv4) with VPN is briefly described. Since it is expected that some problems which might occur in MIPv6 to coexist with VPN will be similar to those in MIPv4 environments, considerations of MIPv4 with VPN will help readers to understand situations for MIPv6 with VPN in the future. Section 3 describes cryptographically generated address. As described in Chapter 10, the default security mechanism for mobile nodes in MIPv6 Binding Update is return routability. It tries to prove the ownership of home addresses by mobile nodes. However, it exposes weakness to various attacks such as man-in-the-middle attack. Thus, stronger security mechanisms for the proof of ownership of home addresses are needed. Cryptographically generated address receives widest consensus as an optional security mechanism for MIPv6. Firewall traversal problem is described in Section 4.

## 12.2 VPN PROBLEMS AND SOLUTIONS IN MIPv4

### 12.2.1 Concept

Mobile IP[1] agents are being deployed in enterprise networks to enable mobility across wired and wireless networks while roaming inside the enterprise intranet. With the growing deployment of IEEE 802.11 access points ('hot spots') in public places such as hotels, airports, and convention centers, and wireless WAN data networks such as GPRS, needs for enabling mobile users to maintain their transport connections and constant reachability while connecting back to their home networks protected by Virtual Private Network (VPN) technology are increasing. This implies that Mobile IP and VPN technologies have to coexist and work together in order to provide mobility and security to the enterprise mobile users.

Mobile node is always addressable with its home address which is allocated from its home network. The basic idea of IP mobility support is to maintain the address pair consisting of home and care-of address to update the current location of mobile nodes to minimize the handover latency and reduce the service disruption without needs of additional protocol overhead of adjacent layers. The care-of address changes for every movement. Thus, it is hard to maintain the pre-configured Security Association (SA) between the mobile node, and home agent or security entities such as VPN gateway or firewall.

The VPN gateway controls the flow of incoming and outgoing packets with pre-defined filtering rule. Addresses and ports known to VPN are only allowed to pass the gateway which in turn drops ingress packets from unknown sources. This becomes an obstacle to deploy the Mobile IP with coexisting VPN environment because CoAs of mobile nodes which are obtained outside VPN domain are usually unknown to VPN. The goal of this section is to identify and describe practical deployment scenarios for Mobile IPv4 and VPN in enterprise and operator environments. Deployment scenarios and possible problems are briefly described in the next section.

### 12.2.2 Mobile IP and VPN deployment scenarios

In all scenarios, a mobile node runs both the MIPv4 and IPsec-based VPN client software. The foreign network might or might not employ foreign agents. The term Intranet refers to a private network protected by a VPN gateway and perhaps a layer-3 transparent or non-transparent firewall.

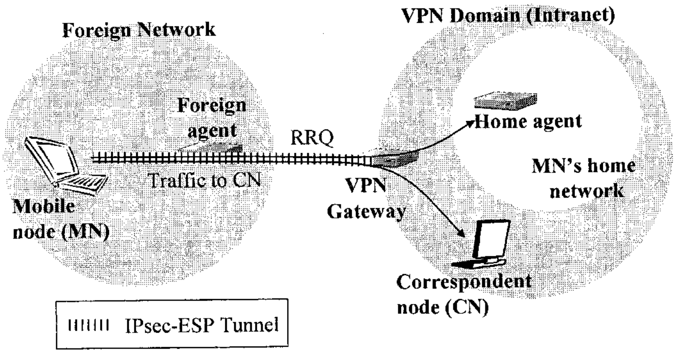


*Figure 12-1.* Home agent is inside the Intranet behind a VPN gateway.

The scenarios assume that encryption is not enforced inside the VPN domain because (1) the VPN domain (Intranet) is viewed as a trusted network and users residing inside the Intranet are also trusted, and (2) the VPN is used to guard the Intranet resources from unauthorized users attached to a untrusted network and to provide a secure communication channel for authorized users to access resources inside the Intranet from outside.

#### 12.2.2.1 HA(s) inside the Intranet protected by a VPN gateway

The MIPv4 home agents are deployed inside the Intranet protected by a VPN gateway, and are not directly reachable by the mobile nodes outside the Intranet as shown in Fig. 12-1.

Direct application of the MIPv4 protocol is successfully used to provide mobility for users inside the Intranet. However, mobile users outside the Intranet can only access the intranet resources (e.g., Mobile IP agents) through the VPN gateway, which will allow only authenticated IPsec traffic inside. This implies that the MIPv4 traffic has to run inside IPsec, which leads to two distinct problems:

1. When the foreign network has a foreign agent deployed (as in e.g. CDMA 2000), MIPv4 registration becomes impossible because the traffics between mobile node and VPN gateway must be encrypted but foreign agents between them are not able to decrypt those traffics.
2. In co-located mode, successful registration is possible but the VPN tunnel has to be re-negotiated every time the mobile node changes its point of network attachment.

This scenario may not be common yet, but it is practical and becoming important as there are increasing needs for providing corporate remote users with continuous access to the Intranet resources.

### 12.2.2.2 VPN gateway and HA(s) in parallel

A MIPv4 home agent is deployed in parallel with the VPN gateway in Fig. 12-2. Then, the home agent is directly reachable by mobile nodes inside or outside the Intranet.

The MIPv4 home agent has a public interface connected to the Internet, and a private interface attached to the Intranet. Mobile users will most likely have a virtual home network associated with the MIPv4 home agent's private interface, so that the mobile users are always away from home and hence registered with the MIPv4 home agent.

If the VPN gateway and the home agent are deployed in a corporate DMZ, MIPv4 traffics will always be routed through the DMZ regardless of whether mobile nodes are located outside or inside the Intranet. This deployment can be used with two different configurations such as MIPv4 inside IPsec-ESP tunnel and IPsec-ESP inside MIPv4 tunnel. Alternatively, running MIPv4 home agent and VPN on the same machine as shown in Fig. 12-3 resolves routing related issues when an ‘IPsec-ESP inside MIPv4 tunnel‘ configuration is used. However, this configuration can not be assumed if equipments running MIPv4 home agent and VPN equipment must be acquired from separate vendors.

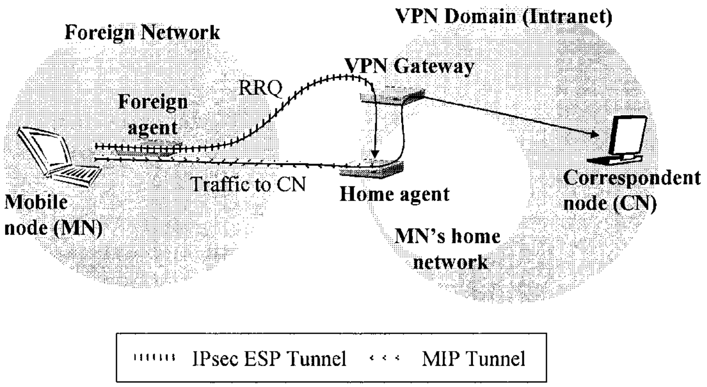


*Figure 12-2.* Home agent and VPN gateway exist in parallel.

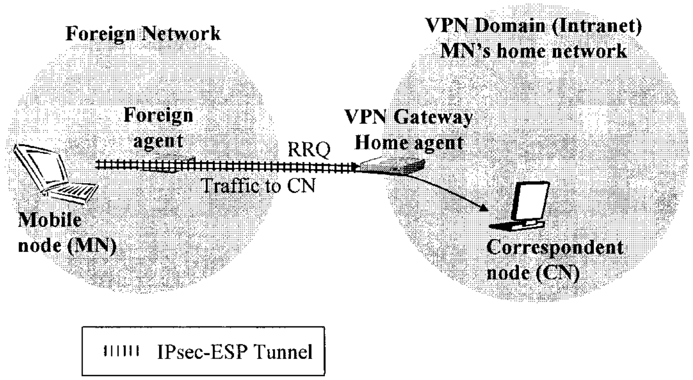


*Figure 12-3.* Home agent coexists with VPN gateway.

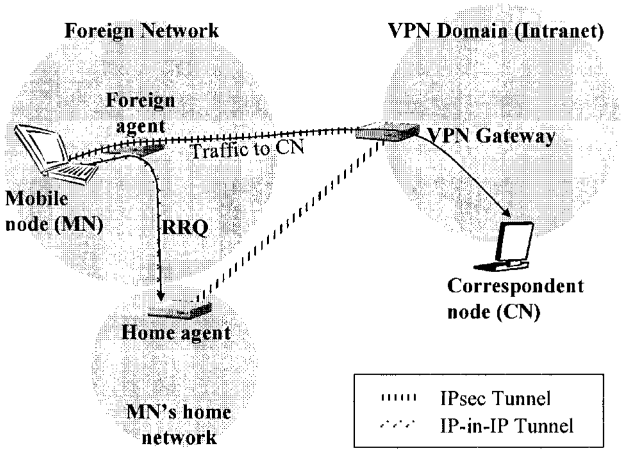


*Figure 12-4.* Home agent is outside the VPN domain.

### 12.2.2.3 HA(s) outside the VPN domain

In this configuration, MIPv4 home agents are deployed outside the Intranet (e.g., in an operator network), as shown in Fig. 12-4. In this deployment scenario, the goal is to provide remote users with seamless access to the Intranet resources while they are roaming outside the Intranet (i.e., mobility is not supported inside the Intranet). In this case it is most practical to run IPsec-ESP inside a MIPv4 tunnel. MIPv4 tunnel end points are the mobile node and the home agent; the IPsec-ESP packet from the mobile node and to the VPN gateway is encapsulated in the MIPv4 tunnel.

Thus, mobile nodes can register with the home agent without establishing an IPsec tunnel to the VPN gateway. This should work without any technical problems. The IPsec tunnel end points will be the mobile node and the VPN gateway. The 'home network' will be a virtual home network, located at the home agent, from which it is possible to reach the corporate intranet through the VPN gateway.

### 12.2.2.4 Combined VPN gateway and HA(s) on the local link

The VPN gateway and home agent would most naturally be co-located in the same box, although this is in no way a requirement. This deployment works today without any technical problems with IPsec-ESP running inside a MIPv4 tunnel. If however MIPv4 is run inside the IPsec-ESP tunnel, it has the same problems explained before. In other words, MIPv4 registration becomes impossible when the registration is to be done via a foreign agent. In co-located mode, the VPN tunnel has to be re-negotiated every time the mobile node changes its point of attachment. This deployment is not common or practical for large size deployments, i.e. on the order of thousands of users, because of the large and distributed security perimeter.

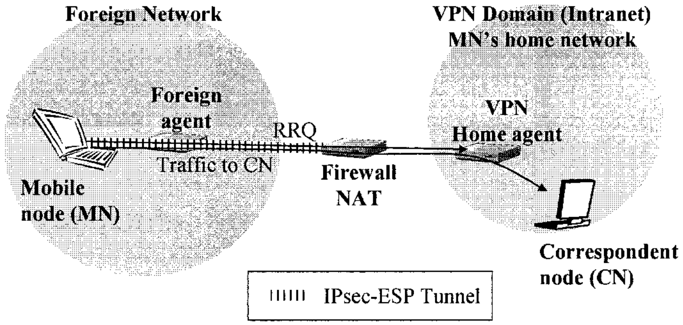


*Figure 12-5.* Home agent is combined with VPN gateway on the local link.

## 12.3 APPLYING CGA TO OPTIMIZE MIPv6

### 12.3.1 Concept

For the route optimization mechanism, the return routability procedure is needed for the proof of ownership of home addresses. The return routability procedure can remove the triangular routing for packets for mobile nodes. There are, however, some overhead and latency problems in current route optimization scheme.

Optimization of MIPv6 based on cryptographically generated address provides a new enhanced route optimization security for MIPv6. The primary motivation for this new mechanism is the reduction of signaling load and handoff delay as well as additional security benefits.

Proof of ownership based on cryptographically generated address in MIPv6 reduces the overhead of signal and handoff delay. The primary reason for using route optimization is to avoid routing all traffic through a home agent. Fig. 10-8 shows return routability procedure for the calculation of the latency.

Basic home registration introduces a latency of zero to one roundtrip before payload traffic can be transferred, depending on the direction of traffic and whether the mobile node chooses to wait for an acknowledgement. With route optimization, the combined latency is one to three roundtrips, depending again on the direction of packets and waiting condition for acknowledgements. Home agents and correspondent nodes can start to send data packets once they have sent the binding acknowledgement. The overall latency until inbound traffic can start flowing to the mobile is therefore at least 1.5 roundtrips.

Current route optimization requires a periodic return routability procedure and the reestablishment of the binding at the correspondent node. This results in average bandwidth of about 7 bits/second which is an insignificant bandwidth for nodes that are actually communicating. However, it can still represent a burden for hosts which just have the bindings ready for a possible packet but are not currently communicating. This can be problematic for hosts in standby mode.

### 12.3.2 Generating CGA

Cryptographically generated address[8] is an IPv6 address, which contains a set of bits generated by hashing the IPv6 address owner's public key. Such feature allows the user to provide a proof of ownership of its IPv6 address. Cryptographically generated address is associated with a public key and auxiliary parameters. Fig. 12-6 shows a flow chart to generate a cryptographically generated address.

The cryptographically generated address verification takes an IPv6 address and auxiliary parameters as input. In order to sign a message, a node needs the cryptographically generated address, the associated cryptographically generated address parameters, the message and the private cryptographic key that corresponds to the public key. If a verification of signature succeeds, it can be assumed that a message is from the right owner of the address.

It is important to note that because cryptographically generated address themselves are not certified, an attacker can create a new cryptographically generated address from any subnet prefix and its own or anyone else's public key. What the attacker can not do is to take a cryptographically generated address created by someone else and send signed messages that appear to come from the owner of that address.

The cryptographically generated address has some advantages. It makes the spoofing attack against the IPv6 address much harder and allows signing messages with the owner's private key. Cryptographically generated address does not require any upgrade or modification in the infrastructure.

### 12.3.3 Protocol performance

This protocol is implemented with similar principles as the original return routability protocol. This protocol, however, adds some mechanisms in order to make it more efficient.

Cryptographically generated address provides stronger proof of ownership of claimed address than the pure use of routing paths. This also decreases the number of signaling messages and provides the public key which is used to secure certain data.

This protocol provides semi-permanent security associations, created with the help of the cryptographically generated address public keys. Because cryptographically generated address is unable to guarantee that a particular address is actually reachable at a given prefix, there is a need for both home and care-of address tests. Due to the higher security of the cryptographically generated address technique, however, we can make these tests much less frequent.

The protocol is divided into two separate cases such as establishing the initial contact and subsequent messaging. Fig. 12-7 shows the signaling diagram for the initial contact.

The Pre-Binding Update, Acknowledgement, Test messages are to ensure that the home and care-of addresses are reachable. They ensure also that at least some communications have taken place before the exchange. They provide Keygen Token which is used to construct a binding management key to the mobile mode. This binding update is the original binding update, but includes the mobile node's public key, signature, and its extended sequence number. The binding acknowledgment carries the semi-permanent security association key in the SKey option, which is encrypted with the mobile node's public key. As a result of the initial process, a standard Binding Cache Entry is created and a semi-permanent security association is established with a key which is called Kbmperm. Fig. 12-8 shows the signaling diagram for subsequent movements.

Care-of Test Init and Care-of Test messages implement the care-of address test operation. In the subsequent messaging, home address tests are now needed. The Binding Update and Acknowledgement messages are authenticated using Kbmperm. It is used in HMAC_SHA1 (Care-of Keygen Token | Kbmperm).

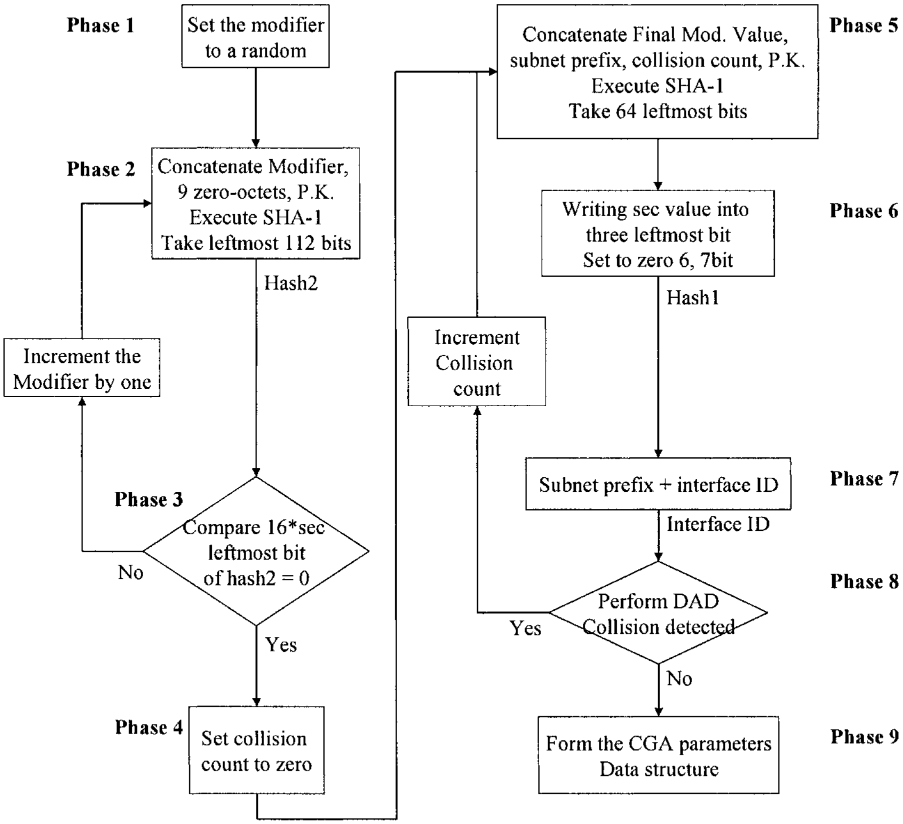
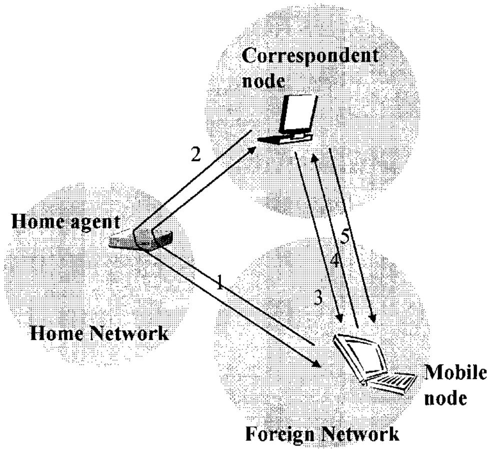

Phase 1
Set the modifier to a random
Phase 2
Concatenate Modifier, 9 zero-octets, P.K. Execute SHA-1 Take leftmost 112 bits
Hash2
Increment the Modifier by one
Phase 3
Compare 16*sec leftmost bit of hash2 = 0
No
Yes
Phase 4
Set collision count to zero
Concatenate Final Mod. Value, subnet prefix, collision count, P.K. Execute SHA-1 Take 64 leftmost bits
Phase 5
Writing sec value into three leftmost bit Set to zero 6, 7bit
Phase 6
Hash1
Increment Collision count
Subnet prefix + interface ID
Phase 7
Interface ID
Perform DAD Collision detected
Phase 8
Yes
No
Form the CGA parameters Data structure
Phase 9

*Figure 12-6.* Algorithm to generate a CGA.



1. Pre Binding Update
2. Pre Binding Acknowledgment
3. Pre Binding Test
4. Binding Update (ESN, CGA key, SIG, BAD)
5. Binding Acknowledgment (ESN, Skey, BAD)

*Figure 12-7.* Signaling diagram to establish the initial contact between mobile node and correspondent node.

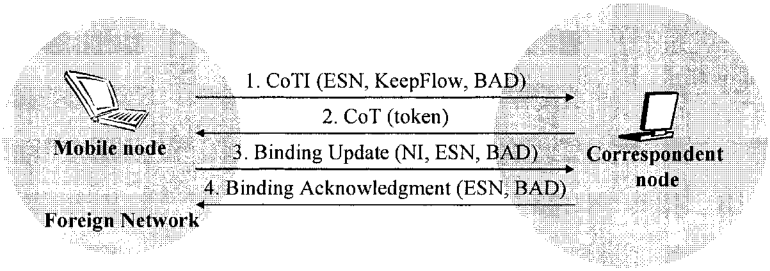

1. CoTI (ESN, KeepFlow, BAD)
2. CoT (token)
3. Binding Update (NI, ESN, BAD)
4. Binding Acknowledgment (ESN, BAD)
Mobile node
Foreign Network
Correspondent node

*Figure 12-8.* Signaling diagram for the subsequent movement.

Performance of protocol depends on whether we look at the initial or subsequent runs. The number of messages to establish the initial contact is one less than that in base MIPv6, but the size of the message is increased. There is, however, a significant signaling reduction, as the lifetime can be set higher than in return routability. The maximum allowed lifetime is 24 hours. In the subsequent movements, there is a significant impact of latency, since the home address test is eliminated. Message formats are explained in the following section.

## 12.3.4 Message formats

### 12.3.4.1 The Pre-Binding Update message

This message is used to initiate two address (home address and care-of address) tests. This does not yet establish any state at the correspondent node.

- The Reserved field is unused and must be initialized to zero by the sender. If this field has some value other than zero, then the receiver must ignore it.
- Care-of Address: The current care-of address of the mobile node.
- Pre-Binding Update Cookie: This field which is 64-bit long, contains a random value. The value ensures that responses match to requests.
- Mobility Options: This field has variable-length. The complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options.

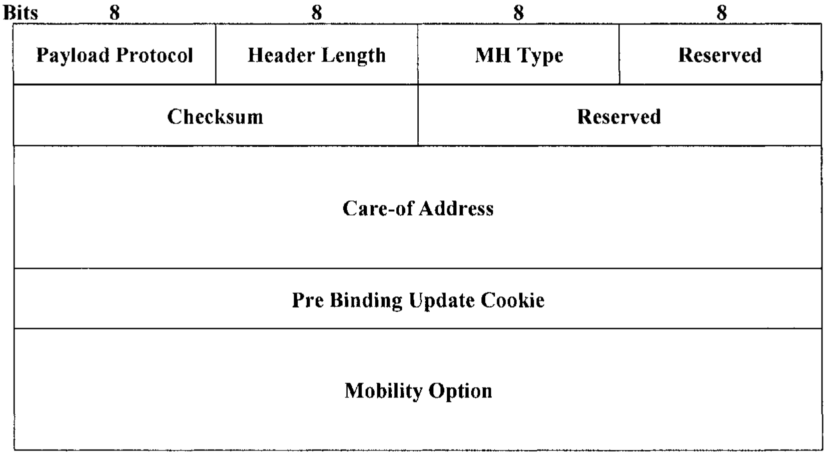


*Figure 12-9.* Pre-Binding Update message format.

### 12.3.4.2 The Pre-Binding Acknowledgement message

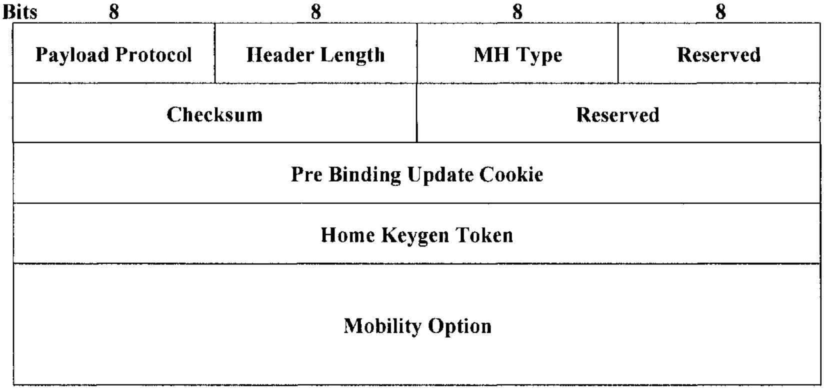
This message is a response to Pre-Binding Update message. This message is transmitted to the mobile node via home agent. This message provides a binding management key material which is Home Keygen Token to the mobile node. The binding management key is required in the initial phase.

- The Reserved field is unused and must be initialized to zero by the sender. If this field has some value other than zero, then the receiver must ignore it.
- Pre-Binding Update Cookie: This field contains the value from the same field in the Pre-Binding Update message.
- Home Keygen Token: This field contains a Home Keygen Token.
- Mobility Options: This field has variable-length. The complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options.

### 12.3.4.3 The Pre-Binding Test message

This message is a response to Pre-Binding Update message. This message is transmitted to the mobile node directly. This message provides a binding management key material which is Care-of Keygen Token to the mobile node. The binding management key is required.

- The Reserved field is unused and must be initialized to zero by the sender. If this field has some value other than zero, then the receiver must ignore it.



| Bits 8 | 8 | 8 | 8 |
|---|---|---|---|
| Payload Protocol | Header Length | MH Type | Reserved |
| Checksum | | Reserved | |
| Pre Binding Update Cookie | | | |
| Home Keygen Token | | | |
| Mobility Option | | | |

*Figure 12-10.* Pre-Binding Acknowledgement message format.

<table>
<tr><th>Bits</th><th>8</th><th>8</th><th>8</th><th>8</th></tr>
<tr><td></td><td>Payload Protocol</td><td>Header Length</td><td>MH Type</td><td>Reserved</td></tr>
<tr><td></td><td colspan="2">Checksum</td><td colspan="2">Reserved</td></tr>
<tr><td></td><td colspan="4">Pre Binding Update Cookie</td></tr>
<tr><td></td><td colspan="4">Care-of Keygen Token</td></tr>
<tr><td></td><td colspan="4">Mobility Option</td></tr>
</table>

*Figure 12-11.* Pre-Binding Test message format.

- Pre-Binding Update Cookie: This field contains the value from the same field in the Pre-Binding Update message.
- Care-of Keygen Token: This field contains a Care-of Keygen Token.
- Mobility Options: This field has variable-length. The complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options.

## 12.4 NSIS FIREWALL TRAVERSAL

### 12.4.1 Concept

Most of the firewalls deployed today are MIPv6 unaware. Route optimization is an integral part of MIPv6 specification. However, this operation does not work with firewalls that employ stateful packet filtering. The other mode in MIPv6, bi-directional tunneling and triangular routing also do not work under firewalls.[9] Therefore, a signaling protocol which can make MIPv6 messages to traverse several firewalls according to certain rules is needed. In Next Steps in Signaling (NSIS) working group in IETF, a NAT/Firewall NSIS Signaling Protocol (NSLP) is proposed to make MIPv6 messages to traverse firewalls.[10]

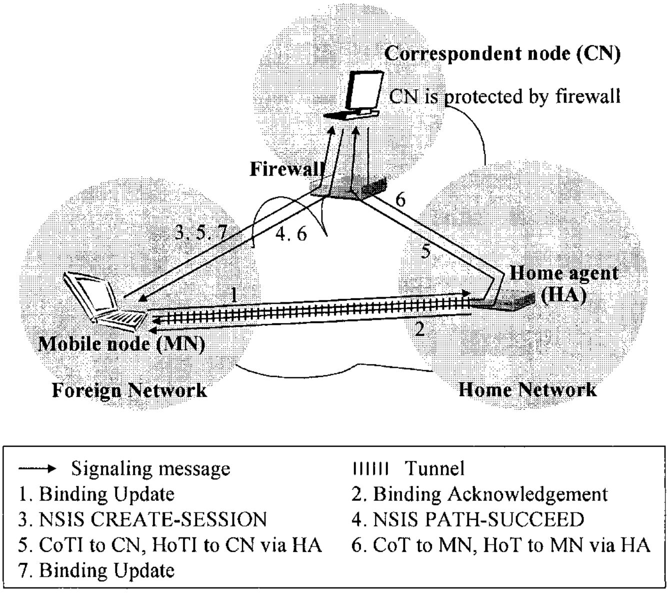


*Figure 12-12.* Signaling diagram for optimized communication between a correspondent node and a mobile node when the correspondent node is behind a firewall and the mobile node is in a foreign network.

## 12.4.2 Route optimization

In Route Optimization mode, correspondent node and mobile node deliver packets directly to each other. However, mobile node has to perform return routability procedure, where it sends a Home Test Init message and a Care-of Test Init message to the correspondent node. Then, the correspondent node returns a Home Test message and a Care-of Test message to the mobile node. As a result of return routability procedure, the mobile node has a binding key which is used in the binding update procedure.

### 12.4.2.1 Correspondent node behind a firewall

In Fig. 12-12, the correspondent node is protected by a firewall. The mobile node is in home network and communicating with the correspondent node. When the mobile node moves out of home network, it has to perform the return routability procedure before sending binding update to the correspondent node. The mobile node sends a Home Test Init message through the home agent to the correspondent node and a Care-of Test Init message directly to the correspondent node. However a firewall will drop these packets. Thus, the return routability procedure can not be completed.

The mobile node initiates the NSIS session by sending a CREATE message to the correspondent node. The firewall may not necessarily know the mobile node and the firewall may not be able to authenticate the mobile node. The correspondent node approves the request and the firewall will install the relevant policy. When the mobile node receives Home Test message and Care-of Test message, the mobile node generates the binding key and performs binding update with the correspondent node.

#### 12.4.2.2 Mobile node behind a firewall

When the mobile node moves to the new network, the mobile node creates a new care-of address and it performs the binding update to a home agent. Signaling messages should be exchanged between the mobile node and the home agent. Thus, the mobile node receives a Home Test message from the home agent.

Once the return routability procedure is successful, the Binding Update message is sent to the correspondent node. If the mobile node want to send data traffic, then no NSIS signaling is needed. However, if the correspondent node want to send data traffic, it has to initiate Signaling-D to mobile node after return routability procedure.

#### 12.4.2.3 Home agent behind a firewall

Binding Update message between a mobile node and a home agent is protected by IPsec. However, primitive firewall does not recognize IPsec traffic and drop packets. Hence, UDP encapsulation of IPsec traffics might be needed. The present firewalls use the security parameter index instead of the port number for IPsec traffic. The mobile node initiates the NSIS Signaling-C to create rules. Then it performs the binding update to the home agent.

The installed firewall rules will not allow the Home Test Init message. Hence, the mobile node has to install different rules to allow these messages. The mobile node initiates the NSIS session by sending a CREATE message and sends Home Test Init message to the home agent. Then the home agent forwards it to the correspondent node. If the home agent receives a Home Test message as a response to the Home Test Init message from the correspondent node, then it sends it to the mobile node. Therefore, the return routability procedure is successfully completed. Fig. 12-13 shows signal message flow for above processes.

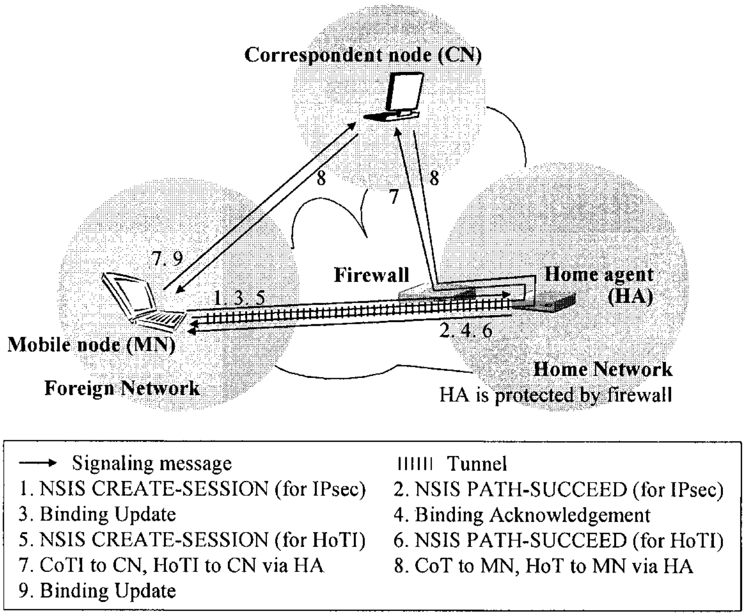

Correspondent node (CN)
8
7
8
7, 9
Firewall
Home agent (HA)
1, 3, 5
2, 4, 6
Mobile node (MN)
Foreign Network
Home Network
HA is protected by firewall
→ Signaling message
IIIII Tunnel
1. NSIS CREATE-SESSION (for IPsec)
2. NSIS PATH-SUCCEED (for IPsec)
3. Binding Update
4. Binding Acknowledgement
5. NSIS CREATE-SESSION (for HoTI)
6. NSIS PATH-SUCCEED (for HoTI)
7. CoTI to CN, HoTI to CN via HA
8. CoT to MN, HoT to MN via HA
9. Binding Update

*Figure 12-13.* Signaling diagram for optimized communication between a correspondent node and a mobile node when a home agent is behind a firewall and the mobile node is in a foreign network.

## 12.4.3 Bi-directional tunneling

When a mobile node moves to a new network, it creates a care-of address on the current link. The mobile node registers its location with a home agent. If the correspondent node sends data to the home address of the mobile node, then the home agent encapsulates this packet and sends it to the mobile node. The mobile node should decapsulate this packet after receiving it from the home agent. In the opposite direction, packets are reverse tunneled to the home agent.

### 12.4.3.1 Correspondent node behind a firewall

If the correspondent node initiates data traffic, then there is no need for any signaling. The correspondent node sends the data traffic and hence a firewall will store relevant connection information.

#### 12.4.3.2 Mobile node behind a firewall

If a mobile node is protected by a firewall, the correspondent node is generally unaware that the mobile node is behind the firewall. The home agent is forced to perform NSIS signaling. The correspondent node does not know the care-of address of the mobile node and hence has no chance of opening the pin-hole. If the correspondent node sends data traffic, then it require an NSIS aware home agent. If the mobile node sends data traffic, no signaling is needed.

#### 12.4.3.3 Home agent behind a firewall

If a home agent is protected by a firewall, the home agent requires also to be NSIS aware. The home agent has the capabilities of NSIS responder. The correspondent node has to open pin-holes in the firewall by initiating Signaling-D. Hence, it is allowed to send data traffics through the firewall. Then the home agent decapsulates packets and sends them to the mobile node.

### 12.4.4 Triangular routing

The triangular routing differs from the bi-directional routing in the reverse direction only. In this routing mode, a correspondent node sends a packet to a home address of the mobile node. Then, a home agent intercepts the packet and performs standard Mobile IP processing. The home agent sends the encapsulated packet to the mobile node. The mobile node decapsulates the packet and eventually knows the address of the correspondent node. Therefore, the mobile node can send the packets directly to the correspondent node.

#### 12.4.4.1 Correspondent node behind a firewall

If a correspondent node is protected by a firewall, data traffics from the correspondent node will be bypassed by the firewall. However, if the mobile node sends data traffics, then the firewall will not allow them. Hence, the mobile node has to initiate Signaling-D by sending the CREATE message to the correspondent node. Firewall will install the policies when it receives the SUCCEED message. As a result, the mobile node is allowed to communicate in the reverse direction.

#### 12.4.4.2 Mobile node behind a firewall

If a mobile node is protected by a firewall, data traffic from a correspondent node to the mobile node will be forwarded to home agent. Then, the home agent recognizes that the mobile node is behind the firewall and initiates signaling to the mobile node to send the tunneled packets. The correspondent node is not aware of the fact that the mobile node is behind the firewall. The mobile node could also install the firewall rules.

#### 12.4.4.3 Home agent behind a firewall

If a home agent is protected by a firewall, a correspondent node initiates NSIS signaling to open pin-holes in the firewall protecting the home agent when the correspondent node sends data traffics to a home address of a mobile node. Therefore, the correspondent node can send data traffics to the home address of the mobile node.

## REFERENCES

1. C. Perkins, IP Mobility Support for IPv4, RFC 3344 (August 2002).
2. F. Adrangi and H. Levkowetz, Problem Statement: Mobile IPv4 Traversal of VPN Gateways, work in progress (June 2003).
3. S. Vaarala, Mobile IPv4 Traversal Across IPsec-based VPN Gateways, work in progress (September 2003).
4. F. Adrangi, M. Kulkarni, G. Dommety, E. Gelasco, Q. Zhang, S. Vaarala, D. Gellert, N. Baider, and H. Levkowetz, Problem Statement and Solution Guidelines for Mobile IPv4 Traversal Across IPsec-based VPN Gateways, work in progress (January 2003).
5. S. Vaarala and O. Levkowetz, Mobile IP NAT/NAPT Traversal using UDP Tunneling, work in progress (November 2002).
6. T. Kivinen, Negotiation of NAT-Traversal in the IKE, work in progress (May 2003).
7. G. Montenegro, Reverse Tunneling for Mobile IP, revised, RFC 3024 (January 2001).
8. T. Aura, Cryptographically Generated Addresses (CGA), work in progress (December 2003).
9. F. Le, Mobile IPv6 and Firewalls Problem statement, work in progress (August 2004).
10. M. Stiemerling, A NAT/Firewall NSIS Signaling Layer Protocol (NSLP), work in progress (July 2004).

# Index