



**National Institute of
Standards and Technology**
U.S. Department of Commerce

Special Publication 800-119

Guidelines for the Secure Deployment of IPv6

**Recommendations of the National Institute
of Standards and Technology**

Sheila Frankel
Richard Graveman
John Pearce
Mark Rooks

NIST Special Publication 800-119

Guidelines for the Secure Deployment of IPv6

*Recommendations of the National
Institute of Standards and Technology*

Sheila Frankel
Richard Graveman
John Pearce
Mark Rooks

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

December 2010



U.S. Department of Commerce

Gary Locke, Secretary

National Institute of Standards and Technology

Dr. Patrick D. Gallagher, Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-119
Natl. Inst. Stand. Technol. Spec. Publ. 800-119, 188 pages (Dec. 2010)

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

Acknowledgments

The authors, Sheila Frankel of the National Institute of Standards and Technology (NIST), Richard Graveman of RFG Security, John Pearce of Booz Allen Hamilton and Mark Rooks of L-1 Identity Solutions (formerly of Booz Allen Hamilton) wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content.

The authors would like to acknowledge Tim Grance of NIST for his keen and insightful assistance and encouragement throughout the development of the document. The authors particularly want to thank Mark Carson, Doug Montgomery and Stephen Nightingale of NIST and Scott Hogg for their careful review and valuable contributions to improving the quality of this publication.

The authors also appreciate the efforts of those individuals, agencies, and other organizations that contributed input during the public comment period, including John Baird, DREN; Alistair de B Clarkson, nCipher; Vint Cerf, Google; John Curran, ARIN; Terry Davis, Boeing; Francois Donze and Michael Scott Pontillo, HP; Jeffrey Dunn, Chern Liou, and Jeffrey Finke, Mitre; Fernando Gont, the UK Centre for the Protection of National Infrastructure (UK CPNI); Bob Grillo, US Army; Cecilia Hall, Don Radeke and Joseph Bertrand, USMC; J. Holland, David Leach, Sam Nguyen, M. Roed, Beth Scruggs, D. Wellington and Joe Williams, Aerospace Corp.; Ed Jankiewicz, SRI International; Ralph Kenyon, Caida; Lovell King II, Dept. of State; Joe Klein, IPv6 Security Researcher; Dan Luu, VA; Trung Nguyen, FAA; Carroll Perkins, Serco-NA; and Martin Radford, University of Bristol.

Table of Contents

Executive Summary	1
1. Introduction	1-1
1.1 Authority	1-1
1.2 Purpose and Scope	1-1
1.3 Audience	1-1
1.4 Document Structure	1-1
2. Introduction to IPv6	2-1
2.1 Early History of IPv6	2-1
2.2 Limitations of IPv4	2-1
2.3 Major Features of the IPv6 Specification	2-2
2.3.1 Extended Address Space	2-3
2.3.2 Autoconfiguration	2-3
2.3.3 Header Structure	2-3
2.3.4 Extension Headers	2-4
2.3.5 Mandatory Internet Protocol Security (IPsec) Support	2-4
2.3.6 Mobility	2-4
2.3.7 Quality of Service (QoS)	2-5
2.3.8 Route Aggregation	2-5
2.3.9 Efficient Transmission	2-5
2.4 IPv4 and IPv6 Threat Comparison	2-5
2.5 Motivations for Deploying IPv6	2-7
3. IPv6 Overview	3-1
3.1 IPv6 Addressing	3-2
3.1.1 Shorthand for Writing IPv6 Addresses	3-5
3.1.2 IPv6 Address Space Usage	3-6
3.1.3 IPv6 Address Types	3-7
3.1.4 IPv6 Address Scope	3-7
3.1.5 IPv4 Addressing	3-9
3.1.6 IPv4 Classless Inter-Domain Routing (CIDR) Addressing	3-10
3.1.7 Comparing IPv6 and IPv4 Addressing	3-11
3.2 IPv6 Address Allocations	3-12
3.2.1 IPv6 Address Assignments	3-12
3.2.2 Obtaining Globally Routable IPv6 Address Space	3-14
3.3 IPv6 Header Types, Formats, and Fields	3-16
3.4 IPv6 Extension Headers	3-18
3.5 Internet Control Message Protocol for IPv6 (ICMPv6)	3-22
3.5.1 ICMPv6 Specification Overview	3-22
3.5.2 Differences between IPv6 and IPv4 ICMP	3-25
3.5.3 Neighbor Discovery	3-26
3.5.4 Autoconfiguration	3-28
3.5.5 Path Maximum Transmission Unit (PMTU) Discovery	3-29
3.5.6 Security Ramifications	3-30
3.6 IPv6 and Routing	3-34
3.6.1 Specification Overview	3-34
3.6.2 Security for Routing Protocols	3-35

3.6.3	Unknown Aspects	3-36
3.7	IPv6 and the Domain Name System (DNS)	3-36
3.7.1	DNS Transport Protocol	3-37
3.7.2	DNS Specification Overview	3-37
3.7.3	Security Impact and Recommendations	3-39
4.	IPv6 Advanced Topics	4-1
4.1	Multihoming	4-1
4.1.1	Differences between IPv4 and IPv6 Multihoming	4-1
4.1.2	Site Multihoming by IPv6 Intermediation (SHIM6) Specification Overview ..	4-2
4.1.3	Security Ramifications for Multihoming	4-4
4.2	IPv6 Multicast	4-5
4.2.1	IPv6 Multicast Specifications	4-6
4.2.2	Differences between IPv4 and IPv6 Multicast	4-8
4.2.3	Multicast Security Ramifications	4-9
4.2.4	Unresolved Aspects of IPv6 Multicast	4-9
4.3	IPv6 Quality of Service (QoS)	4-10
4.3.1	IPv6 QoS Specifications	4-10
4.3.2	Differences between IPv4 and IPv6 QoS	4-11
4.3.3	Security Ramifications	4-11
4.3.4	Unresolved Aspects of IPv6 QoS	4-12
4.4	Mobile IPv6 (MIPv6)	4-12
4.4.1	MIPv6 Specification Overview	4-12
4.4.2	Differences from IPv4 Standards	4-16
4.4.3	Security Ramifications	4-16
4.4.4	Unknown Aspects	4-26
4.5	Jumbograms	4-27
4.5.1	Specification Overview	4-27
4.5.2	Security Ramifications	4-27
4.6	Address Selection	4-28
4.6.1	Specification Overview	4-28
4.6.2	Differences from IPv4 Standards	4-30
4.6.3	Security Ramifications	4-30
4.6.4	Unknown Aspects	4-31
4.7	Dynamic Host Configuration Protocol (DHCP) for IPv6	4-31
4.7.1	Specification Overview	4-32
4.7.2	Differences from IPv4 Standards	4-34
4.7.3	Security Ramifications	4-34
4.7.4	Unknown Aspects	4-35
4.8	IPv6 Prefix Renumbering	4-35
4.8.1	Specification Overview	4-36
4.8.2	Differences from IPv4 Standards	4-38
4.8.3	Security Ramifications	4-38
4.8.4	Unknown Aspects	4-39
5.	IPv6 Security Advanced Topics	5-1
5.1	Privacy Addresses	5-1
5.2	Cryptographically Generated Addresses	5-3
5.3	IPsec in IPv6	5-4
5.3.1	Specification Overview	5-5
5.3.2	Differences from IPv4 Standards	5-8

5.3.3	Support for Multicast	5-8
5.3.4	Status of IPsec and On-Going Work.....	5-9
5.3.5	Security Ramifications.....	5-15
5.3.6	Unknown Aspects	5-16
5.4	Secure Stateless Address Autoconfiguration and Neighbor Discovery	5-17
5.4.1	Using IPsec to Secure Autoconfiguration and ND	5-18
5.4.2	Using SEND to Secure Autoconfiguration and ND	5-19
5.4.3	Current Status and Unknown Aspects.....	5-19
6.	IPv6 Deployment	6-1
6.1	Security Risks	6-1
6.1.1	Attacker Community.....	6-1
6.1.2	Unauthorized IPv6 Clients.....	6-2
6.1.3	Vulnerabilities in IPv6.....	6-2
6.1.4	Dual Operations	6-4
6.1.5	Perceived Risk	6-4
6.1.6	Vendor Support.....	6-4
6.2	Addressing Security	6-5
6.2.1	Numbering Plan	6-5
6.2.2	Hierarchical Addressing to Support Security Segmentation.....	6-6
6.2.3	Problems with EUI-64 Addresses.....	6-7
6.2.4	Address Management.....	6-7
6.2.5	Privacy Extensions.....	6-8
6.3	Transition Mechanisms.....	6-8
6.4	Dual Stack IPv4/IPv6 Environments	6-9
6.4.1	Deployment of a Dual Stack Environment	6-9
6.4.2	Addressing in a Dual Stack Environment	6-10
6.4.3	Security Implications of a Dual Stack Environment.....	6-11
6.5	Tunneling	6-11
6.5.1	General Security Considerations for Tunneling	6-13
6.5.2	Configured Tunneling.....	6-15
6.5.3	Automatic Tunneling	6-16
6.5.4	6over4 Protocol.....	6-16
6.5.5	6to4 and 6rd Protocols	6-17
6.5.6	Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)	6-19
6.5.7	Teredo Protocol.....	6-22
6.5.8	Tunnel Brokers.....	6-27
6.5.9	Automatic Tunneling of IPv4 over IPv6 (Dual Stack Transition Mechanism [DSTM])	6-28
6.5.10	Carrier-Grade NAT and Dual-Stack Lite	6-30
6.6	Translation	6-32
6.6.1	SIIT	6-33
6.6.2	NAT-PT.....	6-33
6.6.3	Replacing NAT-PT	6-34
6.6.4	TRT.....	6-35
6.6.5	Application Layer Translation	6-36
6.7	Other Transition Mechanisms.....	6-37
6.8	The IPv6 Deployment Planning Process for Security.....	6-37
6.9	IPv6 Deployment	6-38
6.9.1	Initiation Phase	6-39
6.9.2	Acquisition / Development Phase.....	6-41

6.9.3	Implementation Phase.....	6-44
6.9.4	Operations / Maintenance Phase	6-46
6.9.5	Disposition Phase	6-46
6.10	Summary.....	6-47

List of Appendices

Appendix A— Acronyms and Abbreviations	A-1
---	------------

Appendix B— References and Other IPv6 Resources	B-1
--	------------

List of Figures

Figure 2-1. The IPv6 Packet Header Format (Field Sizes in Bits).....	2-4
Figure 3-1. IPv6 Address Format	3-3
Figure 3-2. 32-Bit Network Prefix	3-4
Figure 3-3. 48-Bit Network Prefix	3-4
Figure 3-4. 64-Bit Network Prefix	3-5
Figure 3-5. A Comparison of IPv4 and IPv6 Addressing.....	3-11
Figure 3-6. The IPv6 Packet Header Format (Field Sizes in Bits) (RFC 2460)	3-16
Figure 3-7. Example IPv6 Packet Header	3-18
Figure 3-8. Next Header Fields in IPv6 and Extension Headers.....	3-18
Figure 3-9. IPv6 Extension Header Chaining	3-19
Figure 3-10. ICMPv6 Message Format.....	3-23
Figure 3-11. Example of Neighbor Discovery	3-27
Figure 3-12. Example of Stateless Address Autoconfiguration (SLAAC)	3-29
Figure 3-13. Significance of MTU under IPv6.....	3-30
Figure 4-1. SHIM6 Protocol Stack.....	4-4
Figure 4-2. The Main MIPv6 Components.....	4-14
Figure 4-3. IKEv1 Identifiers used between a MN and its HA	4-20
Figure 4-4. IKEv2 identifiers used between a MN and its HA	4-20
Figure 4-5. Return Routability—Init Messages	4-22
Figure 4-6. Return Routability—Keygen Replies	4-23
Figure 4-7. Reverse Routability—BU and BUA Protected with Kbm.....	4-24
Figure 5-1. Example of IPv6 Privacy Addressing.....	5-2
Figure 5-2. Generating Cryptographic Addresses from Public-Private Key Pairs.....	5-3

Figure 5-3. IPsec in the TCP/IP Protocol Stack	5-5
Figure 5-4. Encryption and Authentication Algorithms for the IPsec Protocol	5-9
Figure 5-5. Cryptographic Algorithms for Use in IKEv2	5-10
Figure 6-1. Example of Tunneling IPv6 over IPv4 Networks.....	6-12
Figure 6-2. IPv6 over IPv4 Tunnels Transparent to the IPv4 Infrastructure	6-14
Figure 6-3. Example - Tunneling IPv6 over IPv4 Networks with ISATAP	6-21
Figure 6-4. Example - Tunneling IPv6 over IPv4 Networks with Teredo	6-23
Figure 6-5. Teredo Address	6-24

List of Tables

Table 3-1. Differences between IPv4 and IPv6.....	3-1
Table 3-2. IPv6 Address Types	3-6
Table 3-3. Assignment of Leftmost, Centermost, and Rightmost Bits	3-13
Table 3-4. IPv6 Extension Headers and Upper Layer Protocols.....	3-21
Table 3-5. ICMPv6 Error Messages and Code Type	3-24
Table 3-6. ICMPv6 Informational Messages.....	3-24
Table 3-7. ICMPv6 Recommended Filtering Actions – Must Not Drop & Should Not Drop ...	3-33
Table 4-1. IPv6 Scoped Multicast Values (from RFC 4291).....	4-7

Executive Summary

Due to the exhaustion of IPv4 (Internet Protocol version 4) address space, and the Office of Management and Budget (OMB)¹ mandate that U.S. federal agencies begin to use the IPv6 (Internet Protocol version 6) protocol, NIST undertook the development of a guide to help educate federal agencies about the possible security risks during their initial IPv6 deployment. This document provides guidelines for organizations to aid in securely deploying IPv6. Since the majority of organizations will most likely run both IPv6 and IPv4 on their networks for the foreseeable future, this document speaks about the *deployment of IPv6* rather than the *transition to IPv6*.²

The deployment of IPv6 can lead to new challenges and types of threats facing an organization. The goals of this document are:

- To educate the reader about IPv6 features and the security impacts of those features
- To provide a comprehensive survey of mechanisms that can be used for the deployment of IPv6
- To provide a suggested deployment strategy for moving to an IPv6 environment

The migration to IPv6 services is inevitable as the IPv4 address space is almost exhausted. IPv6 is not backwards compatible with IPv4, which means organizations will have to change their network infrastructure and systems to deploy IPv6. Organizations should begin now to understand the risks of deploying IPv6, as well as strategies to mitigate such risks. Detailed planning will enable an organization to navigate the process smoothly and securely.

Federal agencies will most likely face security challenges throughout the deployment process, including:

- An attacker community that most likely has more experience and comfort with IPv6 than an organization in the early stages of deployment
- Difficulty in detecting unknown or unauthorized IPv6 assets on existing IPv4 production networks
- Added complexity while operating IPv4 and IPv6 in parallel
- Lack of IPv6 maturity in security products when compared to IPv4 capabilities
- Proliferation of transition-driven IPv6 (or IPv4) tunnels, which complicate defenses at network boundaries even if properly authorized, and can completely circumvent those defenses if unauthorized (e.g. host-based tunnels initiated by end users)

Organizations planning the deployment of IPv6 should consider the following during the planning process:

- IPv6 is a new protocol that is not backward compatible with IPv4
- In most cases IPv4 will still be a component of IT (Information Technology) infrastructure. As such, even after the deployment of IPv6, organizations will require mechanisms for IPv6 and IPv4 co-existence.

¹ OMB [Memo M-05-22](#), *Transition Planning for Internet Protocol Version 6 (IPv6)*, August 2005; OMB [Memo](#), *Transition to IPv6*, September 2010

² Since many of the IPv6-related protocols, tools and mechanisms are typically referred to as *transition mechanisms*, this document does use the word transition in that context.

- IPv6 can be deployed just as securely as IPv4, although it should be expected that vulnerabilities within the protocol, as well as with implementation errors, will lead to an initial increase in IPv6-based vulnerabilities. As a successor to IPv4, IPv6 does incorporate many of the lessons learned by the Internet Engineering Task Force (IETF) for IPv4.
- IPv6 has already been deployed and is currently in operation in large networks globally.

To overcome possible obstacles associated with deploying IPv6, organizations should consider the following recommendations:

- Encourage staff to increase their knowledge of IPv6 to a level comparable with their current understanding of IPv4
- Plan a phased IPv6 deployment utilizing appropriate transition mechanisms to support business needs; don't deploy more transition mechanisms than necessary
- Plan for a long transition period with dual IPv4/IPv6 co-existence

Organizations that are not yet deploying IPv6 globally should implement the following recommendations:

- Block all IPv6 traffic, native and tunneled, at the organization's firewall. Both incoming and outgoing traffic should be blocked.
- Disable all IPv6-compatible ports, protocols and services on all software and hardware.
- Begin to acquire familiarity and expertise with IPv6, through laboratory experimentation and/or limited pilot deployments.
- Make organization web servers, located outside of the organizational firewall, accessible via IPv6 connections. This will enable IPv6-only users to access the servers and aid the organization in acquiring familiarity with some aspects of IPv6 deployment.

Organizations that are deploying IPv6 should implement the following recommendations to mitigate IPv6 threats:

- Apply an appropriate mix of different types of IPv6 addressing (privacy addressing, unique local addressing, sparse allocation, etc) to limit access and knowledge of IPv6-addressed environments.
- Use automated address management tools to avoid manual entry of IPv6 addresses, which is prone to error because of their length.
- Develop a granular ICMPv6 (Internet Control Protocol for IPv6) filtering policy for the enterprise. Ensure that ICMPv6 messages that are essential to IPv6 operation are allowed, but others are blocked.³
- Use IPsec (Internet Protocol Security) to authenticate and provide confidentiality to assets that can be tied to a scalable trust model (an example is access to Human Resources assets by

³ [NIST SP 500-267](#), *A Profile for IPv6 in the US Government*, specifies the capability to perform selective ICMPv6 filtering as a mandatory function. However, currently, that capability is not available in all products.

internal employees that make use of an organization's Public Key Infrastructure (PKI) to establish trust).

- Identify capabilities and weaknesses of network protection devices in an IPv6 environment.
- Enable controls that might not have been used in IPv4 due to a lower threat level during initial deployment (implementing default deny access control policies, implementing routing protocol security, etc).
- Pay close attention to the security aspects of transition mechanisms such as tunneling protocols.
- Ensure that IPv6 routers, packet filters, firewalls, and tunnel endpoints enforce multicast scope boundaries and make sure that Multicast Listener Discovery (MLD) packets are not inappropriately routable.
- Be aware that switching from an environment in which NAT (Network Address Translation) provides IP (Internet Protocol) addresses to unique global IPv6 addresses could trigger a change in the FISMA (Federal Information Security Management Act) system boundaries.

After reviewing this document, the reader should have a reasonable understanding of IPv6 and how it compares to IPv4, as well as security impacts of IPv6 features and capabilities, and increased knowledge and awareness about the range of IPv4 to IPv6 transition mechanisms.

1. Introduction

1.1 Authority

The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets; but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), “Securing Agency Information Systems,” as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This guideline has been prepared for use by Federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright, though attribution is desired.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority, nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official.

1.2 Purpose and Scope

The purpose of *Guidelines for the Secure Deployment of IPv6* is to provide information security guidance to organizations that are planning to deploy IPv6 technologies or are simply seeking a better understanding of IPv6. The scope of this document encompasses the IPv6 protocol and related protocol specifications. IPv6-related security considerations are discussed with emphasis on deployment-related security concerns. The document also includes general guidance on secure IPv6 deployment and integration planning.

1.3 Audience

This document is intended primarily for network engineers and administrators who are responsible for planning, building, and operating IP networks, as well as security engineers and administrators who are responsible for providing Information Assurance support. Anyone interested in deploying IPv6 technologies and related security implications may also find the document useful. It includes a discussion of the major features and protocols that constitute IPv6. For each of these, the description is comprised of an introductory section, a more in-depth description, and three analytical sections: differences between the IPv4 and the IPv6 versions, security ramifications and unknown aspects. Managers or users who are trying to understand IPv6 might want to skip the in-depth descriptions but read the other sections (Introduction, Differences, Security Ramifications and Unknown Aspects). They should also read [Section 1](#) (Introduction) and [Sections 6.8-6.9](#) (IPv6 Deployment). It is assumed that readers are already familiar with basic IPv4, data networking, and network security concepts.

1.4 Document Structure

The remainder of this document is composed of the following sections and appendices:

- [Section 2](#) provides an introduction to IPv6, including its history, features, and comparisons with IPv4.
- [Section 3](#) discusses in more detail IPv6 addressing, allocation, packet organization, and ICMPv6.
- [Section 4](#) examines some of the more advanced features of IPv6 and their security implications, including multihoming, multicast, QoS (Quality of Service), Mobile IPv6, Jumbograms and address selection.
- [Section 5](#) provides an introduction to some of the advanced security features included in IPv6, including privacy addresses; IPsec; and secure stateless address autoconfiguration and neighbor discovery.
- [Section 6](#) covers the process of securely deploying IPv6 and discusses the risks, addressing security, various transition mechanisms and the deployment process.

[Appendix A](#) provides a list of acronyms and abbreviations used in this document.

[Appendix B](#) lists references and other resources related to IPv6.

2. Introduction to IPv6

Internet Protocol version 6 (IPv6) is a new network layer protocol. It is an enhancement to Internet Protocol version 4 (IPv4), the protocol in use since the 1980s. There are numerous upgrades in IPv6. Most significantly, in comparison with IPv4, IPv6 has increased its network address size from 32 bits to 128. This provides more than enough addresses to satisfy the global demand for unique IP addresses.

This chapter provides an overview of IPv6 as a foundation for later sections. The section starts with the early history of IPv6 and the limitations of IPv4, followed by descriptions of the major features of the IPv6 specifications. This is followed by a threat comparison between IPv4 and IPv6 and concludes with motivations for deploying to IPv6.

2.1 Early History of IPv6

IPv4 was developed in the 1970s and early 1980s for use in government and academic communities in the United States to facilitate communication and information sharing. Today's networking demand, in particular web pages, email, peer-to-peer services, and the use of mobile devices, has grown well beyond its originators' expectations. Widespread deployment and growth of networking technologies and mobile communications have surpassed IPv4's ability to provide adequate globally unique address space⁴.

Efforts to develop a successor to IPv4 started in the early 1990s within the Internet Engineering Task Force (IETF)⁵. The objective was to solve the address space limitations as well as provide additional functionality. The IETF started the Internet Protocol Next Generation (IPng) work in 1993 to investigate different proposals and to make recommendations for further actions. The IETF recommended IPv6 in 1994. (The name IPv5 had previously been allocated to an experimental stream protocol.) Their recommendation is specified in [RFC 1752](#), *The Recommendation for IP Next Generation Protocol*. Several proposals followed; the Internet Engineering Steering Group approved the IPv6 recommendation and drafted a Proposed Standard on November 17, 1994. [RFC 1883](#), *Internet Protocol, Version 6 (IPv6) Specification*, was published in 1995. The core set of IPv6⁶ protocols became an IETF Draft Standard on August 10, 1998. This included [RFC 2460](#), which replaced [RFC 1883](#).

IPv6 is a protocol designed to handle the growth rate of the Internet and to cope with the demanding requirements of services, mobility, and end-to-end security. The following sections describe the limitations of IPv4, the major features of IPv6, and motivations for deploying IPv6.

2.2 Limitations of IPv4

IPv4 ([RFC 791](#)) was designed over 30 years ago for a relatively small number of users. At that time, it seemed unlikely that personal computing technology would become as widespread as it is today in the United States and worldwide. The rapid, universal adoption and growth of personal computing technologies, including IP networking, were unforeseen in 1981. At that time, the Internet was used almost exclusively by scholars and researchers, and IPv4's 4.3 billion theoretically available addresses were considered to be more than sufficient.

⁴ Hagen, *IPv6 Essentials 2nd Edition*.

⁵ The [IETF](#) is an open international community charged with the evolution of the Internet architectures and standards. An Internet standard begins as an Internet Draft, which generally evolves during the publication of successive versions. It may then be published as a Request for Comments (RFC) document. Some RFCs define IETF standards; others are informational documents or describe experimental protocols.

⁶ Two current IETF working groups that concentrate on IPv6 operations and protocols are the IPv6 Operations ([v6ops](#)) Working Group and the IPv6 Maintenance ([6man](#)) Working group.

As a result of growing Internet use, IPv4's address capacity could not meet the demand. In practice, the supply of available IPv4 addresses has been limited since the early 1990s. Previously, an organization could apply for and receive an order of magnitude more IPv4 addresses than it could actually justify. However, as a result of regulatory advances, IP address allocations are now bound by strict policies that include formal justification to a Regional Internet Registry (RIR). During the 1990s, address allocation policies, along with address reuse and restriction technologies, were put into place to conserve IPv4 addresses.

Technologies widely adopted in response to the constrained supply of IPv4 addresses are network address translation (NAT [[RFC 3022](#)]) and classless inter-domain routing (CIDR [[RFC 4632](#)]); both are discussed in detail in Chapter 3. NAT essentially makes private IPv4 addresses (also known as non-routable addresses) at least partially functional on the global Internet. Despite their adaptation to other uses, private IPv4 addresses were designed for testing and other non-production purposes and never intended to be usable on the Internet. Nevertheless, a NAT-capable router positioned at an organization's boundary has the ability to connect an entire network of privately addressed nodes within the organization to the Internet via a single routable IP address.

This technology saves IPv4 address space because nodes bearing private addresses are essentially "on" the Internet but do not have globally unique IP addresses. Nevertheless, this address conservation technology can actually defeat certain aspects of the design intent of IPv4: network layer end-to-end security, peer-to-peer (host-to-host connectivity), and interoperability. A host using private addressing behind a NAT device cannot have a full peer-to-peer relationship with another host via the Internet or backbone enterprise network using globally unique addressing. This is because NAT does not allow communication sessions to be initiated from globally addressed nodes to the privately addressed nodes.

NAT traversal technologies are available to work around some of these barriers. They typically work in one of two ways: (1) by maintaining stateful address lookup tables and redirecting inbound traffic to appropriate private addresses; (2) by employing application layer gateways that listen for specific port numbers and redirect traffic according to pre-configured parameters. Neither of these approaches to NAT traversal lends itself to scalability or guarantees compatibility with all forms of NAT, not to mention the efforts put into each of these work-arounds. In addition, neither approach lends itself to dynamic configuration when, for example, hosts move or networks are renumbered.

Another limitation of IPv4 is that its design favored interoperability over security and did not contain features that protected the confidentiality, integrity, or availability of communications. For example, IPv4 could not cryptographically protect data from eavesdropping or manipulation, and IPv4 did not provide a method for endpoints to authenticate each other. Over time, the open nature of IPv4 was increasingly a target of exploitation. The multi-path nature of the Internet, which was designed for high availability, also allows multiple attack vectors for a variety of threats. As a response, new technologies were added to IPv4 to provide needed security functionality. With IPv6, these features were designed into the new protocol as mandatory components.

2.3 Major Features of the IPv6 Specification

IPv6 has many new or improved features that make it significantly different from its predecessor. These features include extended address space, autoconfiguration, header structure, extension headers, IPsec, mobility, quality of service, route aggregation, and efficient transmission. This section discusses these features and compares specific aspects of IPv4 and IPv6 to help establish an understanding of the protocols' similarities and differences.

2.3.1 Extended Address Space

Each IPv4 address is typically 32 bits long and is written as four decimal numbers representing 8-bit octets and separated by decimal points or periods. An example address is 172.30.128.97. Each IPv6 address is 128 bits long (as defined in [RFC 4291](#)) and is written as eight 16-bit fields in colon-delimited hexadecimal notation (an example is fe80:43e3:9095:02e5:0216:cbff:feb2:7474). This new 128-bit address space provides an enormous number of unique addresses, 2^{128} (or 3.4×10^{38}) addresses, compared with IPv4's 2^{32} (or 4.3×10^9) addresses. That is enough for many trillions of addresses to be assigned to every human being on the planet. Moreover, these address bits are divided between the network prefix and the host identifier portions of the address. The *network prefix* designates the network upon which the host bearing the address resides. The *host identifier* identifies the node or interface within the network upon which it resides. The network prefix may change while the host identifier can remain static. The static host identifier allows a device to maintain a consistent identity despite its location in a network. This enormous number of addresses allows for end-to-end communication between devices with globally unique IP addresses and can better support the delivery of peer-to-peer services with data-rich content such as voice and video. Chapter 3 describes IPv6 addressing in detail.

2.3.2 Autoconfiguration

Essentially plug-and-play networking, autoconfiguration, defined in [RFC 4862](#), *IPv6 Stateless Address Autoconfiguration*, is one of the most interesting and potentially valuable addressing features in IPv6. This feature allows devices on an IPv6 network to configure themselves independently using a stateless protocol. In IPv4, hosts are configured manually or with host configuration protocols like Dynamic Host Configuration Protocol (DHCP); with IPv6, autoconfiguration takes this a step further by defining a method for some devices to configure their IP addresses and other parameters without the need for a server. Moreover, it also defines a method, *renumbering*, whereby the time and effort required to renumber a network by replacing an old prefix with a new prefix are vastly reduced. [Section 3.5.4](#) describes autoconfiguration in detail.

2.3.3 Header Structure

The IPv6 header is much simpler than the IPv4 header and has a fixed length of 40 bytes (as defined in [RFC 2460](#)).

Even though this header is almost twice as long as the minimum IPv4 header, much of the header is taken up by two 16-byte IPv6 addresses, leaving only 8 bytes for other header information. This allows for improved fast processing of packets and protocol flexibility. IPv6 datagrams use a structure that always includes a 40-byte *base header* and, optionally, one or more *extension headers*. This base header is like the header of IPv4 datagrams, though it has a different format. Five IPv4 header fields have been removed: IP header length, identification, flags, fragment offset, and header checksum. The IPv6 header fields are as follows: version (IP version 6), traffic class (replacing IPv4's type of service field), flow label (a new field for Quality of Service (QoS) management), payload length (length of data following the fixed part of the IPv6 header), next header (replacing IPv4's protocol field), hop limit (number of hops, replacing IPv4's time to live field), and source and destination addresses. The IPv6 header format is illustrated in Figure 2-1. The payload can be up to 64KB in size in standard mode, or larger with a *jumbo payload* option. [Section 3.3](#) describes these headers in detail.

Version (4)	Traffic Class (8)	Flow Label (20 bits)	
Payload length (16)		Next Header (8)	Hop Limit (8)
Source Address (128 bits)			
Destination Address (128 bits)			

Figure 2-1. The IPv6 Packet Header Format (Field Sizes in Bits)⁷

2.3.4 Extension Headers

An IPv4 header can be extended from 20 bytes to a maximum of 60 bytes, but this option is rarely used because it impedes performance and is often administratively prohibited for security reasons. IPv6 has a new method to handle options, which allows substantially improved processing and avoids some of the security problems that IPv4 options generated. IPv6 [RFC 2460](#) defines six *extension headers*: hop-by-hop option header, routing header, fragment header, destination options header, authentication header (AH), and encapsulating security payload (ESP) header. Each extension header is identified by the *Next Header* field in the preceding header. [Section 3.4](#) describes extension headers in detail.

2.3.5 Mandatory Internet Protocol Security (IPsec) Support

IP security (IPsec) is a suite of protocols for securing Internet Protocol (IP) communications by authenticating the sender and providing integrity protection plus optionally confidentiality for the transmitted data. This is accomplished through the use of two extension headers: the Encapsulating Security Payload (ESP) and the Authentication Header (AH). The negotiation and management of IPsec security protections and the associated secret keys is handled by the Internet Key Exchange (IKE) protocol. IPsec is a mandatory part of an IPv6 implementation; however, its use is not required. IPsec is also specified for securing particular IPv6 protocols (e.g., Mobile IPv6 and OSPFv3 [Open Shortest Path First version 3]). [Section 5.3](#) describes IPsec in detail.

2.3.6 Mobility

Mobile IPv6 (MIPv6) is an enhanced protocol supporting roaming for a mobile node, so that it can move from one network to another without losing IP-layer connectivity (as defined in [RFC 3775](#)). [RFC 3344](#), *IP Mobility Support for IPv4*, describes Mobile IP concepts and specifications for IPv4. Nevertheless, using Mobile IP with IPv4 has various limitations, such as limited address space, dependence on address resolution protocol (ARP), and challenges with handover when a device moves from one access point to another. Mobile IPv6 uses IPv6's vast address space and *Neighbor Discovery* ([RFC 4861](#)) to solve the handover problem at the network layer and maintain connections to applications and services if a device changes its temporary IP address. Mobile IPv6 also introduces new security concerns such as *route optimization* ([RFC 4449](#)) where data flow between the home agent and mobile node will need to be appropriately secured.

⁷ Additional illustration and explanation of the major differences between the IPv6 and IPv4 headers can be found in the GAO report, [Internet Protocol Version 6: Federal Agencies Need to Plan for Transition and Manage Security Risks](#).

[Section 4.4](#) describes Mobile IPv6 in detail.

2.3.7 Quality of Service (QoS)

IP (for the most part) treats all packets alike, as they are forwarded with best effort treatment and no guarantee for delivery through the network. TCP (Transmission Control Protocol) adds delivery confirmations but has no options to control parameters such as delay or bandwidth allocation. QoS offers enhanced policy-based networking options to prioritize the delivery of information. Existing IPv4 and IPv6 implementations use similar QoS capabilities, such as Differentiated Services and Integrated Services, to identify and prioritize IP-based communications during periods of network congestion. Within the IPv6 header two fields can be used for QoS, the *Traffic Class* and *Flow Label* fields. The new Flow Label field and enlarged Traffic Class field in the main IPv6 header allow more efficient and finer grained differentiation of various types of traffic. The new Flow Label field can contain a label identifying or prioritizing a certain packet flow such as voice over IP (VoIP) or videoconferencing, both of which are sensitive to timely delivery. IPv6 QoS is still a work in progress and security should be given increased consideration in this stage of development. [Section 4.3](#) describes QoS in detail.

2.3.8 Route Aggregation

IPv6 incorporates a hierarchical addressing structure and has a simplified header allowing for improved routing of information from a source to a destination. The large amount of address space allows organizations with large numbers of connections to obtain blocks of contiguous address space. Contiguous address space allows organizations to aggregate addresses under one prefix for identification on the Internet. This structured approach to addressing reduces the amount of information Internet routers must maintain and store and promotes faster routing of data⁷⁷. Additionally, it is envisioned that IPv6 addresses will primarily be allocated only from Internet Service Providers (ISPs) to customers. This will allow for ISPs to summarize route advertisements to minimize the size of the IPv6 Internet routing tables. This is covered in more detail in [Section 3.2](#).

2.3.9 Efficient Transmission

IPv6 packet fragmentation control occurs at the IPv6 source host, not at an intermediate IPv6 router. With IPv4, a router can fragment a packet when the Maximum Transmission Unit (MTU) of the next link is smaller than the packet it has to send. The router does this by slicing a packet to fit into the smaller MTU and sends it out as a set of fragments. The destination host collects the fragments and reassembles them. All fragments must arrive for the higher level protocol to get the packet. Therefore, when one fragment is missing or an error occurs, the entire transmission has to be redone. In IPv6, a host uses a procedure called *Path Maximum Transmission Unit (PMTU) Discovery* to learn the path MTU size and eliminate the need for routers to perform fragmentation. The IPv6 Fragment Extension Header is used when an IPv6 host wants to fragment a packet, so fragmentation occurs at the source host, not the router, which allows efficient transmission. PMTU is discussed in [Section 3.5.5](#), and Section 4.5 describes efficient transmission in detail.

2.4 IPv4 and IPv6 Threat Comparison

The deployment of IPv6 can lead to new challenges with respect to the types of threats facing an organization. This section provides a high-level overview as to how threats differ from an IPv4 environment to an IPv6 environment and combined IPv4-IPv6 environment. The following chapters provide additional details to these threats as required. It should be noted that many IPv6 threat discussions rely on IPsec to provide protection against attack. Due to issues with key management and overall configuration complexity (including applications), it is possible that IPsec will not be deployed

much more than it is with IPv4 today for initial IPv6 use. IPsec is covered in detail in [Section 5.3](#).

Network reconnaissance is typically the first step taken by an attacker to identify assets for exploitation ([RFC 5157](#)).⁸ Reconnaissance attacks in an IPv6 environment differ dramatically from current IPv4 environments. Due to the size of IPv6 subnets (2^{64} in a typical IPv6 environment compared to 2^8 in a typical IPv4 environment), traditional IPv4 scanning techniques that would normally take seconds could take years on a properly designed IPv6 network. This does not mean that reconnaissance attacks will go away in an IPv6 environment; it is more likely that the tactics used for network reconnaissance will be modified. Attackers will still be able to use passive techniques, such as Domain Name System (DNS) name server resolution, to identify victim networks for more targeted exploitation. Additionally, if an attacker is able to obtain access to one system on an IPv6 subnet, the attacker will be able to leverage IPv6 neighbor discovery to identify hosts on the local subnet for exploitation. Neighbor discovery-based attacks will also replace counterparts on IPv4 such as ARP spoofing.

Prevention of unauthorized access to IPv6 networks will likely be more difficult in the early years of IPv6 deployments. IPv6 adds more components to be filtered than IPv4, such as extension headers, multicast addressing, and increased use of ICMP. These extended capabilities of IPv6, as well as the possibility of an IPv6 host having a number of global IPv6 addresses, potentially provides an environment that will make network-level access easier for attackers due to improper deployment of IPv6 access controls. Moreover, security related tools and accepted best practices have been slow to accommodate IPv6. Either these items do not exist or have not been stress tested in an IPv6 environment. Nevertheless, global aggregation of IPv6 addresses by ISPs should allow enhanced anti-spoofing filtering across the Internet where implemented.

Attacks that focus on exploitation above the IP layer, such as application-based attacks and viruses, will not see a difference in the types of threats faced in an IPv6 environment. Most likely, some worms will use modified IPv6 reconnaissance techniques for exploitation. Additionally, because many IPv4 broadcast capabilities have been replaced with IPv6 multicast functionality, broadcast amplification attacks will no longer exist in an IPv6 environment.

From this comparison of IPv4 and IPv6 threats, one can surmise that IPv6 will not inherently be either more or less secure than IPv4. While organizations are in the process of deploying IPv6, the lack of robust IPv6 security controls (described in [Section 6](#)) and a lack of overall understanding of IPv6 by security staff may allow attackers to exploit IPv6 assets or leverage IPv6 access to further exploit IPv4 assets. There is a very likely possibility that many IPv6 services will rely on tunneling IPv6 traffic in IPv4 for infrastructures that do support the protocol, which will also increase the complexity for security staff. Additionally, since IPv6 systems and capabilities are not yet widely used in production environments, there is a distinct possibility that the number of vulnerabilities in software from implementing IPv6 capabilities could rise, as IPv6 networks are increasingly deployed.

Based on of the threat comparison between IPv4 and IPv6, the following actions are recommended to mitigate IPv6 threats during the deployment process:

- Apply different types of IPv6 addressing (privacy addressing, unique local addressing, sparse allocation, etc) to limit access and knowledge of IPv6-addressed environments.
- Assign subnet and interface identifiers randomly to increase the difficulty of network scanning.

⁸ Bellovin, Cheswick and Keromytis, [Worm propagation strategies in an IPv6 Internet](#).

- Develop a granular ICMPv6 filtering policy for the enterprise. Ensure that ICMPv6 messages that are essential to IPv6 operation are allowed, but others are blocked.
- Use IPsec to authenticate and provide confidentiality to assets that can be tied to a scalable trust model (an example is access to Human Resources assets by internal employees that make use of an organization's Public Key Infrastructure (PKI) to establish trust).
- Identify capabilities and weaknesses of network protection devices in an IPv6 environment.
- Enable controls that might not have been used in IPv4 due to a lower threat level during initial deployment (implementing *default deny* access control policies, implementing routing protocol security, etc).
- Pay close attention to the security aspects of transition mechanisms such as tunneling protocols.
- On networks that are IPv4-only, block all IPv6 traffic.

2.5 Motivations for Deploying IPv6

IP technologies were invented in the United States, and the early adoption of those technologies occurred predominantly in the United States. As mentioned in [Section 2.2](#), early address allocation policies were relatively relaxed and large quantities of IPv4 addresses were assigned upon request, even when those allocations were not thoroughly justified. This resulted in a high concentration of IPv4 address allocations in the United States, with more than half of all routable IPv4 addresses assigned to U.S.-based organizations. Some large U.S.-based Internet backbone service providers have more IPv4 addresses than all of the nations that comprise the Asian region of the world.

These circumstances have left most of the world, especially Asia, with little choice other than to adopt the IPv6 specification if they are to become pervasive participants in IP technologies or the global Internet at large. Nations such as Japan have built IPv6-capable Internet infrastructures to support their growing demand for Internet connectivity. Further, the advanced state of wireless telecommunications in Asia produced an environment where globally unique IP addresses are required to enable the features of Third Generation (3G) wireless technologies. In essence, every mobile 3G device becomes a mobile personal computing platform, and each of those devices requires true end-to-end connectivity to realize its full potential.

All organizations making use of IP networking should study and consider IPv6's feature set when designing and managing their networks. Even with no intent to replace IPv4, the IPv6 security controls discussed later in this document should be planned and deployed to detect unauthorized use of IPv6. Fundamental knowledge of IPv6—what it is, what its attributes are, and how it operates—is critical to any organization.

As the IPv6 protocol becomes increasingly ubiquitous, all enterprise and Internet-connected networks need to be prepared for specific threats and vulnerabilities that the new protocol will bring. For example, an IPv4-only network segment may contain several newly installed hosts that are both IPv4 and IPv6-capable, as well as hosts that have IPv6 enabled by default. This circumstance can come about simply as a result of the normal systems life cycles. Additionally, IPv6 could be enabled on a host by an attacker to circumvent security controls that may not be IPv6-aware; these hosts can then be leveraged to create covert or backdoor channels. Taken further, IPv6 traffic could be encapsulated within IPv4 packets using readily available tools and services and exchanged with malicious hosts via the Internet.

Interoperability of geographically dispersed Internet-connected nodes may become a profit motivation for some organizations to deploy IPv6. For instance, content providers are making more multimedia features

available via a diverse set of customer platforms. Mobile phones, handheld personal computers, notebook computers, desktop PCs, and home multimedia and gaming centers are all IPv4-capable today. Delivering multimedia content to those platforms is increasingly viable given the broadband network bandwidths available. Nevertheless, IPv4 clearly cannot address all of these devices without using an address conservation technology like NAT, and NAT by its nature denies true end-to-end IP connectivity. Multimedia service offerings and ultimately the market for those offerings are likely always to be constrained by IPv4, while IPv6 may prove to be an enabling technology.

If an organization is not constrained by IPv4 address availability or the disruption that NAT causes to true end-to-end connectivity between nodes, it should still plan for a world in which IPv6 will eventually be ubiquitous. All major vendors of IT products are shipping IPv6-capable products. Wholesale replacement of computing platforms and network infrastructure as a deployment requirement is less likely now than only five years ago, since many operating systems and networking products contain a native IPv6 protocol stack. Also, tunneling IPv6 over the existing IPv4 Internet is possible today by using free, readily available tunnel clients. An end user may download client software, obtain a routable IPv6 address, and begin tunneling IPv6 over IPv4 networks with few technical or administrative barriers. Many open source IP networking tools are IPv6-capable, as are many consumer-oriented wireless access points. Many consumers of personal computing and home networking equipment are IPv6-capable, even if they do not use the features.

Because of the increasing availability and use of IPv6, as well as many years of coexistence between IPv6 and IPv4, management and technical experts within any organization should understand IPv6 technology—its background, basis, and capabilities, and how they can mitigate risks associated with running *dual stack* IPv4 and IPv6 networks. In the context of this document, dual stack means that nodes are running both IPv4 and IPv6 protocols concurrently. The remainder of this document examines certain aspects of the IPv6 specification in detail, and discusses threats, vulnerabilities, and the mitigation of risks, in detail.

3. IPv6 Overview

From the standpoint of header design, IPv6 is both more powerful and more flexible than its IPv4 predecessor. [Section 2.3](#) introduced a number of enhancements and features in IPv6. Most significant is the vast amount of address space, along with support for orderly address assignment and efficient network address aggregation on the Internet. Illustrated in Table 3-1 are some of the major differences between IPv4 and IPv6 followed by basic IPv6 terminology used later in this guide. These differences can have implications for IPv6 security and are discussed throughout this and subsequent sections.

Table 3-1. Differences between IPv4 and IPv6⁹

Property	IPv4	IPv6
Address size and network size	32 bits, network size 8-30 bits	128 bits, network size 64 bits
Packet header size	20-60 bytes	40 bytes
Header-level extension	limited number of small IP options	unlimited number of IPv6 extension headers
Fragmentation	sender or any intermediate router allowed to fragment	only sender may fragment
Control protocols	mixture of non-IP (ARP), ICMP, and other protocols	all control protocols based on ICMPv6
Minimum allowed MTU	576 bytes	1280 bytes
Path MTU discovery	optional, not widely used	strongly recommended
Address assignment	usually one address per host	usually multiple addresses per interface
Address types	use of unicast, multicast, and broadcast address types	broadcast addressing no longer used, use of unicast, multicast and anycast address types
Address configuration	devices configured manually or with host configuration protocols like DHCP	devices configure themselves independently using stateless address autoconfiguration (SLAAC) or use DHCP

Basic Terms ([RFC 2460](#), [RFC 4862](#))

The following basic IPv6 definitions are important for any IPv6 discussion.

- **Address.** An IPv6-layer identifier for an interface or a set of interfaces.
- **Node.** A device on the network that sends and receives IPv6 packets
- **Deprecated address.** An address, assigned to an interface, whose use is discouraged, but not forbidden (e.g., site-local addresses such as FEC0::/10). A deprecated address should no

⁹ NSA Report, [Router Security Configuration Guide Supplement – Security for IPv6 Routers](#).

longer be used as a source address in new communications, but packets sent from or to deprecated addresses are delivered as expected.

- **Router.** A node that sends and receives packets, and also accepts packets and forwards them on behalf of other nodes.
- **Host.** A node that may send and receive packets but does not forward packets for other nodes.
- **Link.** A communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IPv6. Examples are Ethernets (simple or bridged); Point-to-Point Protocol (PPP); X.25, Frame Relay, or Asynchronous Transfer Mode (ATM) networks; and layer three (or higher) tunnels, such as tunnels over IPv4 or IPv6 itself.
- **Link MTU.** The maximum transmission unit (MTU), i.e., maximum packet size in octets, which can be conveyed over a link.
- **Path MTU.** The minimum link MTU of all the links in a path between a source node and a destination node.
- **Upper Layer.** A protocol layer immediately above IPv6. Examples are transport protocols such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), control protocols such as Internet Message Control Protocol (ICMP), routing protocols such as Open Shortest Path First (OSPF), and internet or lower-layer protocols being *tunneled* over (i.e., encapsulated in) IPv6 such as Internetwork Packet Exchange (IPX), AppleTalk, or IPv6 itself.
- **Interface.** The point at which a node connects to a link. Unicast IPv6 addresses are always associated with interfaces.
- **Packet.** An IPv6 header plus payload.
- **Neighbors.** Nodes attached to the same link.

This section provides general information about IPv6 as a foundation for later sections. The rest of this section is a resource for understanding the similarities and differences between IPv4 and IPv6, with a focus on addressing ([RFC 4291](#)). [Section 3.1](#) discusses IPv6 addresses, how the IPv6 address space is used, and IPv6 address types and scope. This is followed by a review of IPv4 addressing and IPv4 Classless Inter-Domain Routing (CIDR) addressing. Then IPv4 and IPv6 addressing are summarized and compared. [Section 3.2](#) covers IPv6 address allocation. IPv6 headers, their formats, and fields are discussed in [Section 3.3](#). [Sections 3.4](#) through [3.7](#) cover extension headers, ICMPv6, IPv6 routing, and IPv6 Domain Name System (DNS) respectively.

3.1 IPv6 Addressing

Described in [RFC 4291](#), IPv6 addresses are 128 bits long and are written in what is called colon-delimited hexadecimal notation. An IPv6 address is comprised of eight distinct numbers representing 16 bits each and written in base-16 (hexadecimal or *hex*) notation. The valid hex digits are 0 through 9 and A through F and together with the colon separator are the only characters that can be used for writing an IPv6 address. A comparison of IPv4 and IPv6 addressing conventions is illustrated in Figure 3-5 and discussed in more detail in [Section 3.1.7](#).

An example of an IPv6 address is:

2001:0db8:9095:02e5:0216:cbff:feb2:7474

Note that the address contains eight distinct four-place hex values, separated by colons. Each of these values represents 16 bits, for a total of 128 bits in the entire address.

IPv6 addresses are divided among the network prefix, the subnet identifier and the host identifier portions of the address. The *network prefix* is the high-order bits of an IP address, used to identify a specific network and, in some cases, a specific type of address (see Table 3-2). The *subnet identifier* (ID) identifies a link within a site. The subnet ID is assigned by the local administrator of the site; a single site can have multiple subnet IDs. This is used as a designator for the network upon which the host bearing the address is resident. The *host identifier* (host ID) of the address is a unique identifier for the node within the network upon which it resides. It is identified with a specific interface of the host. Figure 3-1 depicts the IPv6 address format with the network prefix, subnet identifier and host identifier.

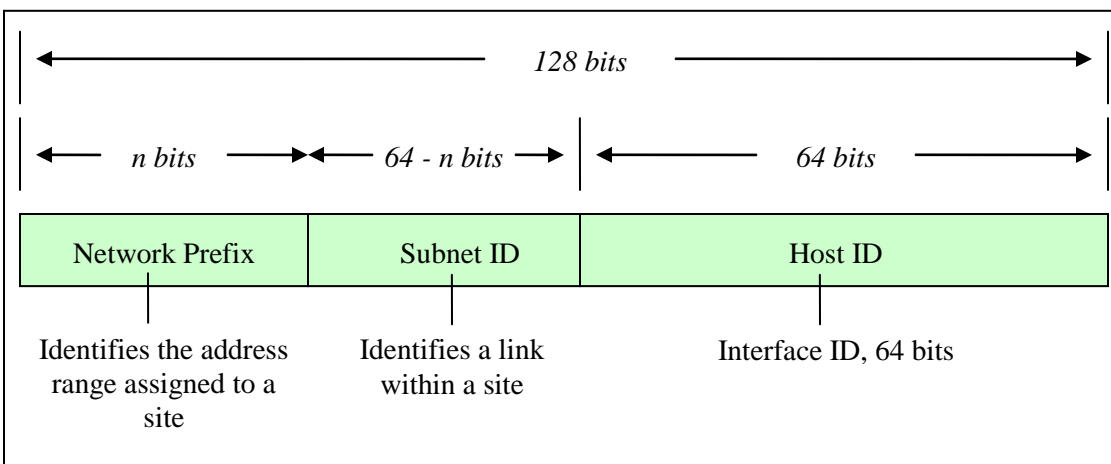


Figure 3-1. IPv6 Address Format

[RFC 4291](#) also describes the notation for prefixes. The network prefix is analogous, but not equivalent, to the subnet mask in IPv4. IPv4 addresses are written in Classless Inter-domain Routing (CIDR) notation, with a subnet mask that contains “1”s in the bit positions that identify the network ID (see [Section 3.1.6](#)). There is no subnet mask in IPv6, although the slash notation used to identify the network address bits is similar to IPv4’s subnet mask notation. The IPv6 notation appends the prefix length and is written as a number of bits with a slash, which leads to the following format:

IPv6 address/prefix length

The prefix length specifies how many of the address’s left-most bits comprise the network prefix. An example address with a 32-bit network prefix is:

2001:0db8:9095:02e5:0216:cbff:feb2:7474/32

Quantities of IPv6 addresses are assigned by the international registry services and Internet service providers (ISP) (see [Section 3.2.2](#)) based in part upon the size of the entity receiving the addresses. Large, top-tier networks may receive address allocations with a network prefix of 32 bits as long as the need is justified. In this case, the first two groupings of hex values, separated by colons, comprise the network prefix for the assignee of the addresses. The remaining 96 bits are available to the local

administrator primarily for reallocation of the subnet ID and the host ID. The subnet ID identifies a link within a site, which can have multiple subnet IDs. The host ID within a network must be unique and identifies an interface on a subnet for the organization, similar to an assigned IPv4 address. Figure 3-2 depicts an IPv6 address with 32 bits allocated to the network prefix.

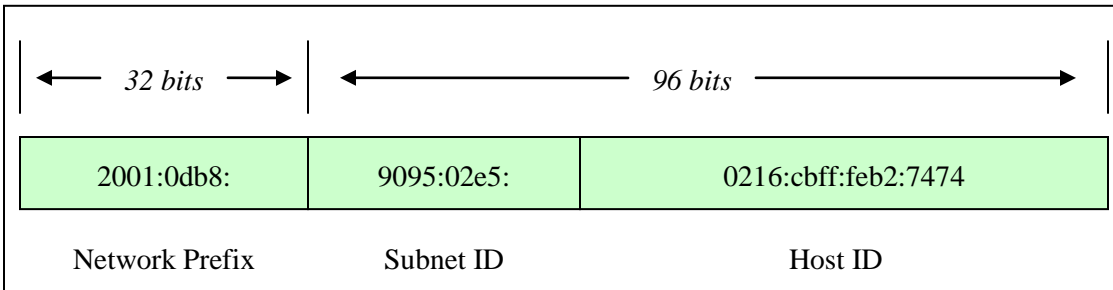


Figure 3-2. 32-Bit Network Prefix

Government, educational, commercial, and other networks typically receive address allocations from top-tier providers (ISPs) with a network prefix of 48 bits (/48), leaving 80 bits for the subnet identifier and host identifier. Figure 3-3 depicts an IPv6 address with 48 bits allocated to the network prefix.

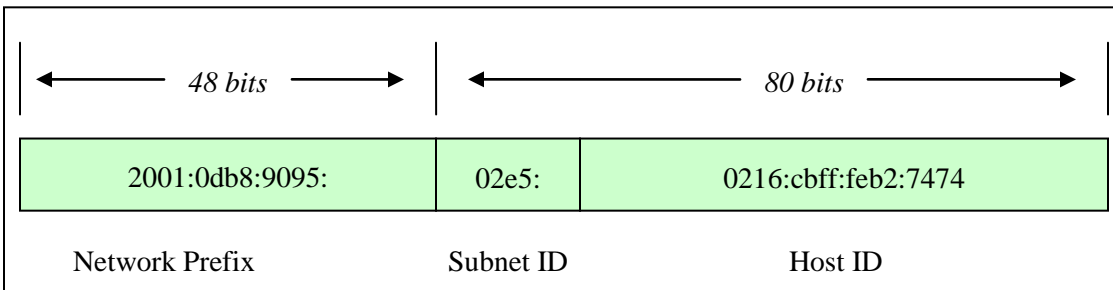


Figure 3-3. 48-Bit Network Prefix

Subnets within an organization often have network prefixes of 64 bits (/64), leaving 64 bits for allocation to hosts' interfaces. The host ID should use a 64-bit interface identifier that follows EUI-64 (Extended Unique Identifier) format when a global network prefix is used (001 to 111), except in the case when multicast addresses (1111 1111) are used¹⁰. Figure 3-4 depicts an IPv6 address with 64 bits allocated to the network prefix.

¹⁰ IEEE EUI-64, [Guidelines for 64-Bit Global Identifier \(EUI-64\) Registration Authority](#).

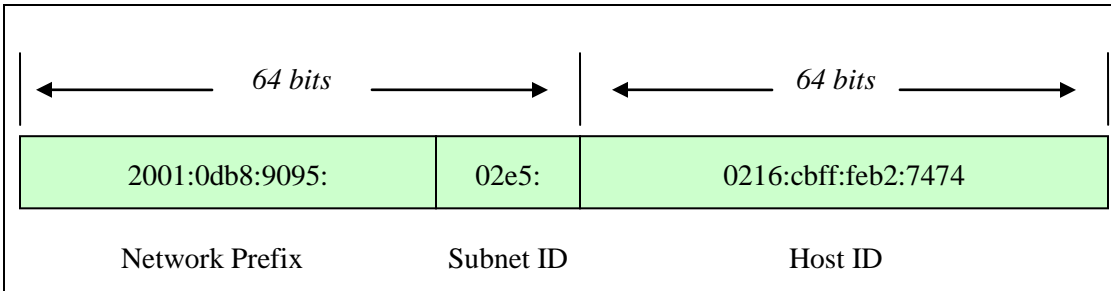


Figure 3-4. 64-Bit Network Prefix

3.1.1 Shorthand for Writing IPv6 Addresses

Due to their length, IPv6 addresses do not lend themselves to human memorization. Administrators of IPv4 networks typically can recall multiple IPv4 network and host addresses; remembering multiple IPv6 network and host addresses is more challenging. The notation for IPv6 addresses may be compressed and simplified under specific circumstances.

One to three zeroes that appear as the leading digits in any colon-delimited hexadecimal grouping may be dropped. This simplifies the address and makes it easier to read and to write. For example:

2001:0db8:0aba:02e5:0000:0ee9:0000:0444/48 becomes
2001:db8:aba:2e5:0:ee9:0:444/48

It is important to note that trailing zeroes may *not* be dropped, because they have intrinsic place value in the address format.

Further efficiency is gained by combining all-zero portions of the address. Any colon-delimited portion of an address containing all zeros may be compressed so that nothing appears between the leading and trailing colons. For example:

2001:0db8:0055:0000:cd23:0000:0000:0205/48 becomes
2001:db8:55:0:cd23::205/48

In this example, the sixth and seventh 16-bit groupings contain all zeroes; they were compressed by eliminating the zeroes completely, as well as the colon that divided the two groupings. Nevertheless, compressing an address by removing one or more consecutive colons between groups of zeroes may only be done once per address. The fourth 16 bit-grouping in the example also contains all zeroes, but in the condensed form of the address, it is represented with a single zero. A choice had to be made as to which group of zeroes was to be compressed. The example address could be written:

2001:db8:55::cd23:0:0:205/48, but this is not as efficient as 2001:db8:55:0:cd23::205/48.

It is important to note that both of the addresses in the preceding paragraph are properly formatted, but the latter address is shorter. Compression is just a convention for writing addresses, it does not affect how an address is used, and it makes no difference whether compression falls within the network prefix, host identifier, or across both portions of the address.

3.1.2 IPv6 Address Space Usage

This section introduces the different types of IPv6 addresses, their scope, and use. It introduces IPv6 addressing as basic information needed for secure adoption and deployment of the protocol. [RFC 4291](#), *IP Version 6 Addressing Architecture*, is the authoritative source for information on IPv6 addressing, and it should be referenced for comprehensive details. Mechanisms for generating and assigning IPv6 addresses are discussed in detail in subsequent sections of this document.

Table 3-2. IPv6 Address Types

Address Type	Binary Prefix	IPv6 notation	Uses
Embedded IPv4 address	00...1111 1111 1111 1111 (96 bits)	::FFFF/96	Prefix for embedding IPv4 address in an IPv6 address
Loopback	00...1 (128 bits)	::1/128	Loopback address on every interface [RFC 2460]
Global unicast	001	2000::/3	Global unicast and anycast (allocated) [RFC 4291]
Global unicast	01 – 1111 1100 0	4000::/2 – FC00::/9	Global unicast and anycast (unallocated)
Teredo	0010 0000 0000 0001 0000 0000 0000 0000	2001:0000::/32	Teredo [RFC 4380]
Nonroutable	0010 0000 0000 0001 0000 1101 1011 1000	2001:DB8::/32	Nonroutable. Documentation purposes only [RFC 3849]
6to4	0010 0000 0000 0010	2002::/16	6to4 [RFC 3056]
6Bone	0011 1111 1111 1110	3FFE::/16	Deprecated. 6Bone testing assignment, 1996 through mid-2006 [RFC 3701]
Link-local unicast	1111 1110 10	FE80::/10	Link local unicast
Reserved	1111 1110 11	FEC0::/10	Deprecated. Formerly Site-local address space, unicast and anycast [RFC 3879]
Local IPv6 address	1111 110	FC00::/7	Unicast Unique local address space, unicast and anycast [RFC 4193]
Multicast	1111 1111	FF00::/8	Multicast address space [RFC 4291]

IPv6 addressing differs from IPv4 in several ways aside from the address size. In both IPv4 and IPv6, addresses specifically belong to interfaces, not to nodes. However, because IPv6 addresses are not in short supply, interfaces often have multiple addresses. As discussed in 3.1, IPv6 addresses consist of a network prefix in the higher order bits and an interface identifier in the lower order bits. Moreover, the prefix indicates a subnet or link within a site, and a link can be assigned multiple subnet IDs.

Many IPv6 address ranges are reserved or defined for special purposes by the IETF's IPv6 standards and by the Internet Assigned Number Authority (IANA). Table 3-2 lists the major assignments and how to identify the different types of IPv6 address from the high-order bits.

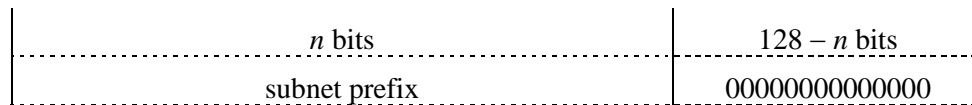
All address ranges not listed in Table 3-2 are reserved or unassigned. IANA currently assigns only out of the binary range starting with 001.¹¹

3.1.3 IPv6 Address Types

IPv6 uses the notion of address types for different situations. These different address types are defined below:

- **Unicast Addresses.** Addresses that identify one interface on a single node; a packet with a unicast destination address is delivered to that interface.
- **Multicast Addresses.** [RFC 4291](#) defines a multicast address as, “An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.” Although multicast addresses are common in both IPv4 and IPv6, in IPv6 multicasting has new applications. The single most important aspect of multicast addressing under IPv6 is that it enables fundamental IPv6 functionality, including neighbor discovery (ND) and router discovery. Multicast addresses begin with **FF00::/8**. They are intended for efficient one-to-many and many-to-many communication. The IPv6 standards prohibit sending packets from a multicast address; multicast addresses are valid only as destinations. Multicast Addressing is discussed in [Section 4.2](#).
- **Anycast Addresses.** Addresses that can identify several interfaces on one or more nodes; a packet with an anycast destination address is delivered to one of the interfaces bearing the address, usually the closest one as determined by routing protocols. Anycast addressing was introduced as an add-on for IPv4, but it was designed as a basic component of IPv6.

The format of anycast addresses is indistinguishable from unicast addresses.



The *subnet prefix* in an anycast address is the prefix that identifies a specific link. Anycast addresses are intended for efficiently providing services that any one of a number of nodes can perform (e.g., a Home Agent for a Mobile IP node). Anycast addresses may not be used as source addresses and, as of the writing of this guide, may only be assigned to routers. It should be noted that there are no defined mechanisms for security or registration for anycast, nor is there a way to verify that a response to a packet sent to an anycast address was sent by an interface authorized to do so. This leaves open the possibility of impersonating anycast servers.

- **Broadcast Addresses.** Broadcast addressing is a common attribute of IPv4, but is not defined or implemented in IPv6. Multicast addressing in IPv6 meets the requirements that broadcast addressing formerly fulfilled.

3.1.4 IPv6 Address Scope

The shortage of IPv4 addresses led to the designation of non-routable addresses in [RFC 1918](#) and the

¹¹ IANA, [Internet Protocol Version 6 Address Space](#).

widespread use of Network Address Translation (NAT) to share globally routable addresses (with certain limits placed on the hosts using so-called RFC 1918 addresses). IPv6 has no such shortage, so the use of NAT is unnecessary; nevertheless, the usefulness of addresses with limited scope was identified and maintained in IPv6. IPv6 addresses with different scopes were defined. In the original design for IPv6, link local, site local, and global addresses were defined; later, it was realized that site local addresses were not well enough defined to be useful. Site local addresses were abandoned and replaced with unique local addresses. Older implementations of IPv6 may still use site local addresses, so IPv6 firewalls need to recognize and handle site local addresses correctly.

The IPv6 standards define several scopes for meaningful IPv6 addresses:

- **Interface-local.** This applies only to a single interface; the loopback address has this scope.
- **Link-local.** This applies to a particular LAN (Local Area Network) or network link; every IPv6 interface on a LAN must have an address with this scope. Link-local addresses start with **FE80::/10**. Packets with link-local destination addresses are not routable and must not be forwarded off the local link.

Link-local address:

10 bits	54 bits	64 bits
1111 1110 10	0000.....0000	Interface ID
FE80/10	0000.....0000	Interface ID

Link-local addresses are used for administrative purposes such as neighbor and router discovery.

- **Site-local.** This scope was intended to apply to all IPv6 networks or a single logical entity such as the network within an organization. Addresses with this scope start with **FEC0::/10**. They were intended not to be globally routable but potentially routed between subnets within an organization. Site local addresses have been deprecated and replaced with unique local addresses.
- **Unique local unicast.** This scope is meant for a site, campus, or enterprise's internal addressing. It replaces the deprecated site-local concept. Unique local addresses (ULAs) may be routable within an enterprise. Use of unique local addresses is not yet widespread; see [RFC 4193](#), *Unique Local IPv6 Unicast Addresses*, for more information.
- **Global.** The global scope applies to the entire Internet. These are globally unique addresses that are routable across all publicly connected networks.
- **Embedded IPv4 Unicast.** The IPv6 specification has the ability to leverage existing IPv4 addressing schemes. The transition to IPv6 will be gradual, so two special types of addresses have been defined for backward compatibility with IPv4: IPv4-compatible IPv6 addresses (rarely used and deprecated in [RFC 4291](#)) and IPv4-mapped IPv6 addresses. Both allow the protocol to derive addresses by embedding IPv4 addresses in the body of an IPv6 address. An IPv4-mapped IPv6 address is used to represent the addresses of IPv4-only nodes as an IPv6 address, which allows an IPv6 node to use this address to send a packet to an IPv4-only node.

IPv4-compatible IPv6 address:

80 bits	16 bits	32 bits
0000.....0000	0000	IPv4 address

IPv4-mapped IPv6 address:

80 bits	16 bits	32 bits
0000.....0000	FFFF	IPv4 address

The two IPv4 embedded address types are similar. The only difference is the sixth group of 16 bits. IPv4-compatible addresses set these to 0; IPv4-mapped addresses set these to 1.

A more generalized form of IPv4-embedded IPv6 addresses has been defined ([RFC 6052](#)), to aid the process of automated translation from one type of address to the other. Two new variants of IPv4-embedded IPv6 addresses are:

- IPv4-converted IPv6 addresses: "IPv6 addresses used to represent IPv4 nodes in an IPv6 network"
- IPv4-translatable IPv6 addresses: "IPv6 addresses assigned to IPv6 nodes for use with stateless transition"

It is quite likely that additional special-use variants will be defined in the future.

- **Other address or Special Address types.** IPv6 makes use of addresses other than those shown above. The *unspecified address* consists of all zeros (0:0:0:0:0:0:0:0 or simply ::) and may be the source address of a node soliciting its own IP address from an address assignment authority (such as a DHCPv6 [DHCP for IPv6] server). IPv6-compliant routers never forward a packet with an unspecified address. The *loopback address* is used by a node to send a packet to itself. The loopback address, 0:0:0:0:0:0:0:1 (or simply ::1), is defined as being interface-local. IPv6-compliant hosts and routers never forward packets with a loopback destination.

An essential design consideration for IPv6 is to simplify routing in enterprise and global networks. One of the intents of the IPv6 address schema is to facilitate hierarchical routing. Hierarchical routing in turn accelerates the end-to-end routing function, and routing table convergence and maintenance are vastly simplified.

A typical IPv6 interface is configured to receive packets sent to several addresses. In addition to its link local and global unicast addresses, it may have a unique local address. It can also receive multicast messages sent to the *all hosts* and *solicited node* multicast addresses, as well as possibly to other multicast addresses. Finally, because of renumbering, multiple instances of some of these addresses may be active at once. How these addresses are selected is covered in the [Sections 4.6](#), Address Selection, and [4.2](#), Multicast.

3.1.5 IPv4 Addressing

Each IPv4 address is 32 bits long and is typically written as four decimal numbers (0-255) representing eight bits each and separated by decimal points or periods. This is called dotted decimal. An example of an IPv4 address is 172.30.128.97. Each IPv4 address is associated with an additional component called a

subnet mask, which denotes how many high-order bits of the address are assigned to the network address ([RFC 950](#)). The remaining lower-order bits are used to identify the node.

Three primary subnet types or network classifications were designed for IPv4: Class A, Class B, and Class C ([RFC 791](#)). Typically, Class A networks were assigned to the early pioneers of the Internet. Class B networks typically were assigned to larger enterprises and service providers, and Class C network addresses usually were allocated to smaller organizations and treated as subnets of larger networks. The following are examples of IPv4 network addresses and their related subnet masks:

- **Class A: 10.0.0.0 netmask 255.0.0.0** The first octet denotes the network and the remaining three octets (24 bits) are available to identify a node on that network. This means that over 16 million host addresses are available on this single Class A network. Class A allocations were often made to organizations that could never put 16 million distinct host addresses to use.
- **Class B: 172.30.0.0 netmask 255.255.0.0** The first two octets denote the network and the remaining two octets (16 bits) are available to identify a node on that network. More than 65,000 distinct addresses are available to network nodes in each Class B network. As with Class A allocations, this also produced a wasteful situation, because many recipients of Class B address allocations did not need to employ more than a small fraction of the addresses.
- **Class C: 192.168.1.0 netmask 255.255.255.0** The first three octets denote the network and the final octet (8 bits) is available to identify a node on that network. This provides 254 addresses for allocation to network nodes (the all ones and all zeros addresses are reserved for other uses). More than two million Class C networks were available. Class C was the smallest, most granular network and host address allocation possible until the introduction of CIDR in 1993.

3.1.6 IPv4 Classless Inter-Domain Routing (CIDR) Addressing

CIDR addresses do not follow the Class A/B/C model. Netmasks in CIDR addresses are not confined to the octet boundaries of an IPv4 address. For example, the CIDR address 192.168.1.1/27 indicates that the IP address is 192.168.1.1 and the netmask splits the address after the 27th bit.¹² The first 27 bits are designated for the network address, and the final five bits are available to provide 30 node or host addresses within that network. This allows for a much more granular approach to address allocation because ranges of addresses can be sized appropriately to the organization receiving them. Of equal importance to address conservation is the related mechanism for routing efficiency that CIDR brings. CIDR addressing allows multiple subnets, defined by common netmasks and having adjacent addresses, to be *supernetted* together. This means that multiple networks are aggregated and reachable under one routing table entry.

The Internet and many large enterprise networks are comprised of core routers (also known as backbone routers) that move vast amounts of data between networks. These routers connect disparate networks and thus make the Internet what it truly is: a network of networks. This same concept applies to large, geographically dispersed enterprise networks. Core routers maintain large, complex routing tables that contain accurate and timely information about how to reach nearly every network that is a part of the global Internet.

The number of entries in these backbone routing tables has increased dramatically since CIDR addressing was introduced in 1993 ([RFC 4632](#)), despite the best intentions of supernetting CIDR address space

¹² If written in the classful notation described previously, it would be represented as 192.168.1.1 netmask 255.255.255.224.

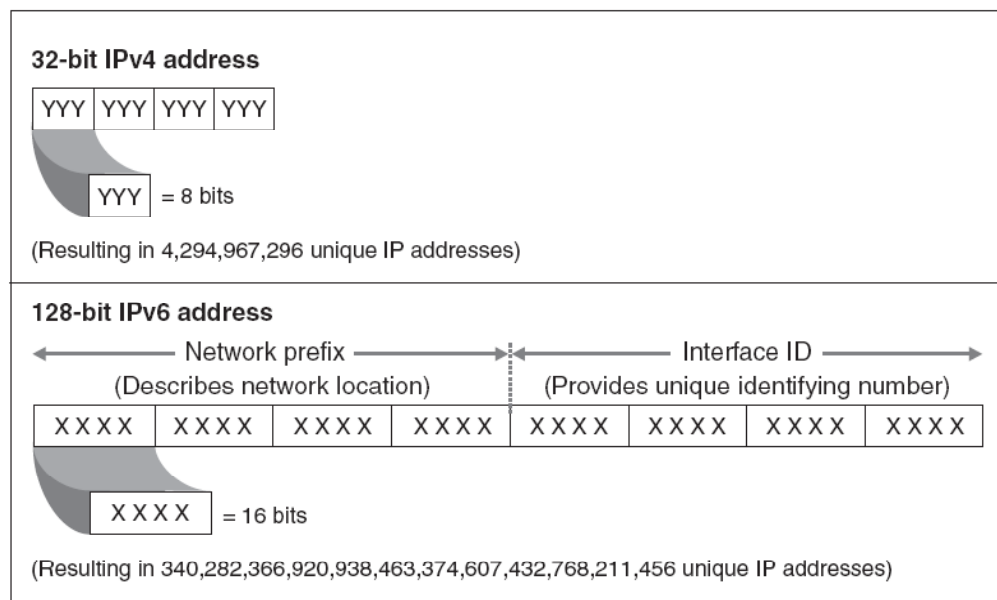
together. As a result, core routers are burdened with ever increasing demands on their memory and processing capacities. In short, IPv4 does not lend itself to a highly scalable and efficient Internet backbone infrastructure.

Routing prefix aggregation allows contiguous groupings of CIDR addresses to be advertised to the global Internet as a single network rather than as multiple, distinct networks. Separate routing table entries no longer need to be made for each allocation of address space. Much like the concept of supernetting, this means that two distinct organizations sharing only one common attribute, their Internet Service Provider (ISP), can be attached to the Internet with unique IP addresses from an appropriately sized allocation. Yet those two distinct entities are reachable through the global Internet using only one globally unique network route. The two concepts discussed here, scalability of address allocations and routing efficiency through prefix aggregation, are integral aspects of the design of IPv6.

3.1.7 Comparing IPv6 and IPv4 Addressing

IPv6 was designed to provide sufficient numbers of globally unique IP addresses to enable true peer-to-peer communication between nodes on interconnected networks. It was also designed to provide a simplified hierarchical routing architecture across the Internet backbone—one that does not suffer from inefficiencies and increasing demands for memory and processing capacities on backbone Internet routers. Several accommodations have been made to retrofit these concepts onto IPv4, while these same concepts are native to the IPv6 specification.

IPv6 provides an enormous number of unique addresses, about 3.4×10^{38} compared with IPv4's roughly 4.3×10^9 addresses. The number of possible IPv6 addresses is so large that many analogies and metaphors have been created that attempt to convey its magnitude. For example, if each IPv6 address weighed one gram, the sum total weight of all IPv6 addresses would be greater than the weight of 56 Earths. The available address space under IPv6 is generally considered to be sufficient for the foreseeable future, even considering the historical growth of the Internet and the devices expected to connect to it in the future. See Figure 3-5 for a comparison of IPv4 and IPv6 addressing conventions.



Source: GAO.

Figure 3-5. A Comparison of IPv4 and IPv6 Addressing⁷

The constraints of IPv4 addressing were major considerations when IPv6 addressing was designed. The IPv6 addressing architecture is different not only in terms of address length, but also in terms of address types, address notation, and address aggregation. As discussed in [Section 2](#), as well as later in [Sections 3](#) and [4](#), each of these differences enables new features in IPv6.

In both IPv4 and IPv6, Dynamic Host Configuration Protocol (DHCP) and the Domain Name System (DNS) can be used to assign, monitor, administer, and change IP addresses. IPv6 also includes an autoconfiguration capability for assigning IP addresses to hosts. Due to the smaller amount of address space available with IPv4, address management was often not complex, with some organizations manually tracking address assignments. The longer, more complex IPv6 addresses, as well as the much larger amount of address space, will most likely require the use of address management tools to avoid errors. In IPv4, it is customary to allocate addresses sequentially, whether they are allocated manually or using DHCP. In IPv6, with an address space large enough to defeat attackers' scanning attempts, addresses should be allocated non-sequentially (e.g., randomly), to preserve that advantage.

3.2 IPv6 Address Allocations

IPv6 addresses have a flexible structure for address assignments. This enables registries, ISPs, network designers, and others to assign address ranges to organizations and networks based on different criteria, such as size of networks and estimated growth rate. Often, an initial assignment does not scale well if a small network becomes larger than expected and hence needs more addresses. The assignment authority may not be able to allocate contiguous addresses if they were already assigned to another network.

[Section 3.2.1](#) describes address assignments using leftmost, rightmost, and centermost strategies. With these methods, organizations have the flexibility to aggregate their IPv6 address allocations efficiently. [Section 3.2.2](#) explains how organizations can obtain IPv6 addresses allocations globally through several regional registry services.

3.2.1 IPv6 Address Assignments

IPv6 network prefix assignment is the first step in network deployment. Understanding several methods such as leftmost, rightmost, and centermost helps provide for flexibility and efficient aggregation of an assigned IPv6 block, as described in [RFC 3531](#), *A Flexible Method for Managing the Assignment of Bits of an IPv6 Address Block*. If done without foresight, boundaries between sub-allocations become difficult to move, and future increases in the use of address space cannot be kept contiguous.

The easiest but least flexible solution is to make block address assignment in order from the beginning of the organization's allocated IPv6 block. For example, if an organization is assigned the prefix 2001:0db8:9095::/48, prefixes can be distributed in simple sequential order:

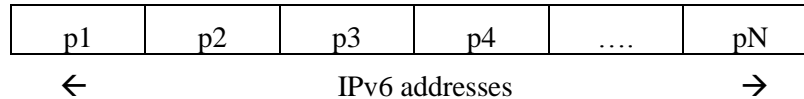
2001:0db8:9095:0001::/64

2001:0db8:9095:0002::/64

2001:0db8:9095:0003::/64

This is the simplest way to distribute address assignments, but it lacks consideration for future needs and does not take into account grouping networks by site for clean routing aggregation. Additionally, this method makes it impossible to make an existing network assignment larger and keep its address space contiguous.

[RFC 3531](#) proposes a method to manage the assignment of bits of an IPv6 address block or range. First, the scheme defines parts of the IP address as p1, p2, p3,pN in order, so that an IP address is composed of these parts contiguously. Boundaries between each part are based on the prefix assigned by the next level authority. Part (p1) is the leftmost part probably assigned to a registry, Part (p2) can be allocated to a large ISP or national registry. Part (p3) can be allocated to a large customer or a smaller provider, etc. Each part can be of different length.



The algorithm for allocating addresses is as follows: (p1) for the left-most part, assign addresses using the leftmost bits first; (pN) for the rightmost part, assign addresses using the rightmost bits first; and for all other parts (center parts), predefine an arbitrary boundary (prefix) and then assign addresses using center bits of the part being assigned first.

This algorithm increases the assigned bits in such way that it keeps unassigned bits near the boundaries between the parts. This means that the boundary between any two parts can be changed forward or backward, later on, up to the assigned bits. See Table 3-3 for the assignment of leftmost, centermost, and rightmost bits.

Table 3-3. Assignment of Leftmost, Centermost, and Rightmost Bits

— Leftmost		— Centermost		— Rightmost	
Binary	Hex	Binary	Hex	Binary	Hex
0000 0000	00	0000 0000	00	0000 0000	00
1000 0000	80	0000 1000	08	0000 0001	01
0100 0000	40	0001 0000	10	0000 0010	02
1100 0000	C0	0001 1000	18	0000 0011	03
0010 0000	20	0000 0100	04	0000 0100	04
1010 0000	A0	0000 1100	0C	0000 0101	05
0110 0000	60	0001 0100	14	0000 0110	06
1110 0000	E0	0001 1100	1C	0000 0111	07
0001 0000	10	0010 0000	20	0000 1000	08

A brief example based on [RFC 3531](#) uses a provider called P1. This provider has been assigned the 3ffe:0b00/24 prefix and wants to assign prefixes to its connected networks. It expects in the foreseeable future a maximum of 256 customers consuming 8 bits. One of these customers, named C2, expects a maximum of 1024 customers' assignments under it, consuming 10 other bits (see [RFC 3531](#) for greater detail). The assignment will be as follows, not showing the first 24 leftmost bits (3ffe:0b00/24 or 0011 1111 1111 1110 0000 1011):

P1 assigns address space to its customers using leftmost bits:

1000 0000 : assigned to customer 1 (C1)
 0100 0000 : assigned to customer 2 (C2)
 1100 0000 : assigned to customer 3 (C3)
 0010 0000 : assigned to customer 4 (C4)

C2 assigns address space to its customers (C2C1, C2C2,) using centermost bits:

0000 10000 : assigned to C2C1
 0001 00000 : assigned to C2C2
 0001 10000 : assigned to C2C3

Customer of C2 uses centermost bits for maximum flexibility and then the last aggregators (which should be networks within a site) will be assigned using rightmost bits.

Putting all bits together for C2C3:

	P1	C2	C2C3
Hex	3ffe:0b00	40	0C
Binary	0011 1111 1111 1110 0000 1011	0100 0000	0000 1100 00
		←	→ ← →
			growing bits

By using this method, P1 will be able to expand the number of customers, and the customers will be able to modify their first assumptions about the size of their own customers, until the reserved bits are assigned.

Predicting future network requirements will always be a challenge with ever changing business needs and unforeseen technological advances. Nonetheless, a strategy to account for organizational needs, possible growth areas, and consideration to address assignment will provide as much downstream flexibility as possible.

3.2.2 Obtaining Globally Routable IPv6 Address Space

The Internet Corporation for Assigned Names and Numbers (ICANN)¹³ and IANA¹⁴ have delegated most IPv6 address allocation to five Regional Internet Registries (RIR):

- Africa and the Indian Ocean (AfriNIC), <http://www.afrinic.net/> or <http://www.afrinic.net/registrationServices.htm>
- Australia, Oceania, and most of Asia (APNIC), <http://www.apnic.net/> or <http://www.apnic.net/policy/ipv6-address-policy>
- Europe, parts of Asia, and the Middle East (RIPE NCC), <http://www.ripe.net/> or <http://www.ripe.net/rs/index.html>

¹³ ICANN, [Address Support](#).

¹⁴ IANA, [Number Resources](#).

- Latin America and the Caribbean (LACNIC), <http://www.lacnic.net/> or <http://lacnic.net/en/bt-IPv6.html>
- North America (ARIN)¹⁵, <http://www.arin.net/> or https://www.arin.net/resources/request/ipv6_initial_assign.html.

ISPs find information about their regional registries at these Web sites. Organizations and end users get their address allocations from their ISPs. Normally, a RIR allocates a /32 address to qualified ISPs, which are called Local Internet Registries (LIR), and the ISP allocates /48 addresses to its customers¹⁶.

American Registry for Internet Numbers (ARIN) is allowing some large government agencies to get provider independent (PI) IPv6 address assignments, defined in ARIN's Number Resource Policy Manual (NRPMP) Section 6.5.8, *Direct Assignments from ARIN to end-user organizations*¹⁷. To qualify for a direct assignment, an organization must not be an IPv6 local Internet registry and must qualify for an IPv4 assignment or allocation from ARIN under the IPv4 policy currently in effect. If these criteria are met, an organization is eligible to receive a minimum assignment of /48, and with justification can request additional subnets. These assignments will be made from a distinctly identified prefix with a reservation of growth of at least /44.

From one point of view, the case for PI assignments can allow for a small number of large organizations to avoid a significant expense due to address renumbering. In addition, organizations may not want to be locked in to a specific Internet provider. On the other hand, the main concerns regarding PI assignment include two major issues. The first is the possibility of a large increase in the size of the IPv6 default-free routing table; these tables generally point only to top-level domains of aggregated routes. PI assignments do not fit into the normal aggregation and will increase the size of these tables. Secondly, the fear is that early adopters, similarly to IPv4, would have an unfair advantage vis à vis those who adopted later.

IPv6 address allocation is designed to allow routing prefix aggregation. IPv6 network addresses may be aggregated in the same sense that IPv4 CIDR addresses are. IPv6 address allocation is based on the hierarchy mentioned previously, and allocated blocks of addresses are widely dispersed with top-tier allocations having network prefixes of 32 bits. This leaves 96 bits' worth of addresses that can all be aggregated through a single route advertisement on the Internet backbone.

Consider routing prefix aggregation for a large backbone service provider. The service provider, hypothetically, receives a block of address space with a 32-bit network prefix. In turn, the provider allocates this address space to customers. Those customers could be multiple regional network service providers or large enterprises that receive blocks of addresses with 48-bit network prefixes from that single large backbone service provider. Subnets within those enterprises and smaller regional service providers may have address space with 64-bit network prefixes.

This arrangement may easily result in tens of millions of nodes attached to millions of subnets, all of which are aggregated and reachable via the global Internet through one route on the Internet's backbone routers.

IPv6 address allocation is a work in progress. Organizations should familiarize themselves with

¹⁵ ARIN's [wiki](#) contains general information about IPv6 as well as pointers to relevant ARIN policies.

¹⁶ ICANN, [Global Policy for Allocation of IPv6 Space](#).

¹⁷ ARIN, [IPv6 Policies section 6.5.8 Direct Assignments from ARIN to end-user organizations](#) and Policy Proposal 2005-1, [Provider-independent IPv6 Assignments for End Sites](#).

assignment and reporting requirements that differ from those for IPv4. [RFC 3177](#), *IAB/IESG Recommendations on IPv6 Address Allocations to Sites*, documents an ongoing effort to provide the latest information for the Internet community regarding current practices, status, and clarifications for IPv6 address allocations. ARIN¹⁸ will document its policy as well.

3.3 IPv6 Header Types, Formats, and Fields

The design of the IPv6 header is the culmination of lessons learned from more than 20 years of experience with IPv4. Two primary design goals for the new header were efficiency and extensibility. The IPv6 header is always 40 bytes long and contains only eight fields, whereas IPv4 headers may be as short as 20 bytes or as long as 60 bytes and contain at least 12 different fields (some of which may be unused). When comparing these attributes, it becomes apparent that the IPv6 header is simpler and more efficient to process. Three examples are:

- The checksum has been removed, because error checking is usually performed in link layer and transport layer protocols.
- Fragmentation has been relegated to an extension header, the minimum MTU has been increased to 1280 bytes, and fragmentation and reassembly are only performed by endpoints.
- Routers have to examine more than the 40-byte header only when the Next Header (NH) field is zero.

The design also pays careful attention to alignment for 64-bit processors; for example, the addresses are aligned on 64-bit boundaries.

The constant size of IPv6 headers makes the header length field found in IPv4 unnecessary. Routers and intermediate nodes handling the packets are not required to accommodate variability in the length of the headers, which expedites packet handling. The IPv6 header format is illustrated in Figure 3-6.

Version (4)	Traffic Class (8)	Flow Label (20 bits)	
Payload length (16)		Next Header (8)	Hop Limit (8)
Source Address (128 bits)			
Destination Address (128 bits)			

Figure 3-6. The IPv6 Packet Header Format (Field Sizes in Bits) ([RFC 2460](#))

The fixed length of the IPv6 header does not preclude flexibility in favor of function. Options are handled with extension headers, which are described in detail in the next section. The following are the eight fields in the fixed IPv6 header:

- **Version.** This is the version of the protocol. This is a 4-bit value and must equal 6 (in binary, 0110).

¹⁸ ARIN, [IPv6 Policies](#).

- **Traffic Class.** Traffic class indicates the type of traffic or service. This eight-bit value is the same as type of service (TOS) in IPv4 and tags packets for special treatment during transmission. [RFC 2474](#), *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, describes how the Traffic Class field in IPv6 can be used. The Traffic Class field in the IPv6 header is referred to as the Differentiated Services (DS) field in [RFC 2474](#), as well as the TOS field in the IPv4 header.
- **Flow Label.** This is a 20-bit value used to identify packets belonging to the same flow or stream of data. It plays an important role in Quality of Service (QoS) differentiation under IPv6 (discussed in [Section 4.3](#)).
- **Payload Length.** Payload length is the length of data carried after the fixed IPv6 header. This 16-bit field identifies the length of the payload to which the 40-byte header is attached. The Payload Length field represents the payload length as an unsigned integer with a maximum value of 65535, so the maximum length is $40 + 65535 = 65575$. (This limit can be extended with the Jumbogram Hop-by-Hop Option discussed in [Section 4.5](#).)
- **Next Header.** Next Header (NH) contains a protocol number for an extension header or upper layer protocol. Called the Protocol Type field in IPv4, the NH field is part of a chain of headers. See [Section 3.4](#) for a complete description.
- **Hop Limit.** Hop Limit defines the maximum number of hops a packet can transit, the same as the Time to Live (TTL) field in IPv4. (The IPv4 TTL was originally defined as a number of seconds, but it almost always means hop count.) Intermediate nodes decrement this unsigned eight-bit value by one for each node the packet traverses. For example, if a packet source sets the Hop Limit to four and there are four routers and five hops in the path between source and destination, the packet is discarded by the fourth router. That packet never reaches the destination, because the packet has a hop limit value of zero after being processed by the fourth router. Generalized TTL Security Mechanism (GTSM) is designed to protect a router or host's TCP/IP (Transmission Control Protocol/Internet Protocol) based control plane from various attacks originating off of the local link. It is a simple security mechanism that avoids these remote attacks by requiring a maximum TTL or Hop Limit for incoming packets. Any packets from a remote attacker would have to travel through at least one intervening router, would have a lower-than-maximum TTL or Hop Limit (255 hops), and would be dropped on receipt. This mechanism based on an expected TTL value can provide a simple and reasonable defense against infrastructure attacks based on forged protocol packets from external sources. This is further described in [RFC 5082](#).
- **Source Address.** This is a 128-bit value representing the unicast IPv6 address of the packet's source. IPv6 addressing is discussed in [Section 3.1](#).
- **Destination Address.** This is a 128-bit value representing the IPv6 address of the packet's destination. It may be a unicast, multicast, or anycast address.

A packet capture of an IPv6 header is illustrated in Figure 3-7.

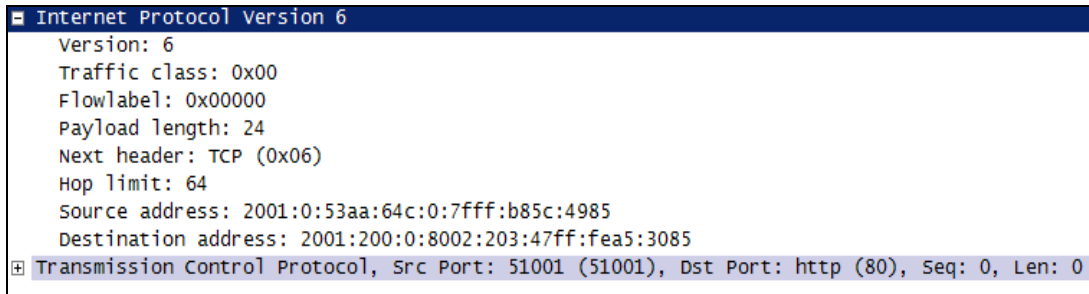


Figure 3-7. Example IPv6 Packet Header

3.4 IPv6 Extension Headers

Extension headers provide major services and functions for the IPv6 protocol. As discussed, the IPv6 header is much simpler with its eight fields and 40-byte header allowing faster processing. Moreover, IPv6 has a new way to deal with options that has substantially improved processing: it handles options in additional headers called extension headers.⁴ Extension headers are inserted into a packet only if the options are needed.

Within the IPv6 packet header and its eight fields, the first extension header is identified in the NH field. In IPv6, optional Internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the upper-layer header in a packet ([RFC 2460](#)). There are a small number of such extension headers, each identified by a distinct NH value (see Table 3-4, IPv6 Extension Headers and Upper Layer Protocols). An IPv6 packet may carry zero, one, or more extension headers, each identified by the NH field of the preceding header and thus forming a chain illustrated below in Figure 3-8. The NH fields indicate the Routing extension header; next, the Fragment extension header; the ESP extension header; and, finally, the TCP header.

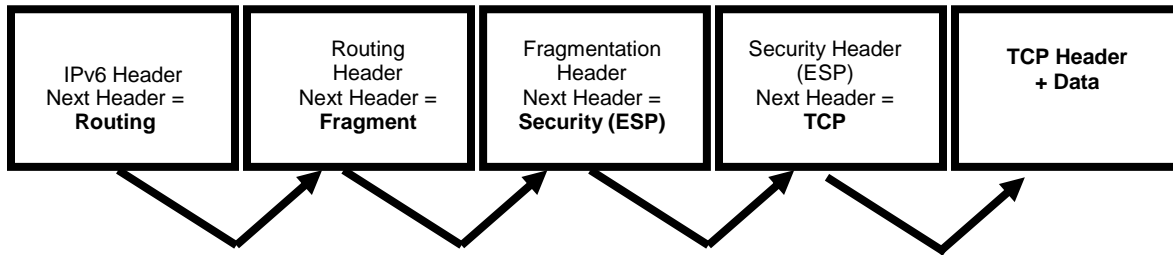


Figure 3-8. Next Header Fields in IPv6 and Extension Headers

Except for Hop-by-Hop Options, extension headers are examined or processed only by the node identified in the Destination address field of the IPv6 header (or a set of nodes, in the case of multicast)⁴ and must be processed strictly in the order in which they appear in the packet. The Hop-by-Hop Option Header is indicated by the value zero (0) in the NH field and requires that information within the packet be examined and processed by every node along the path of the packet.

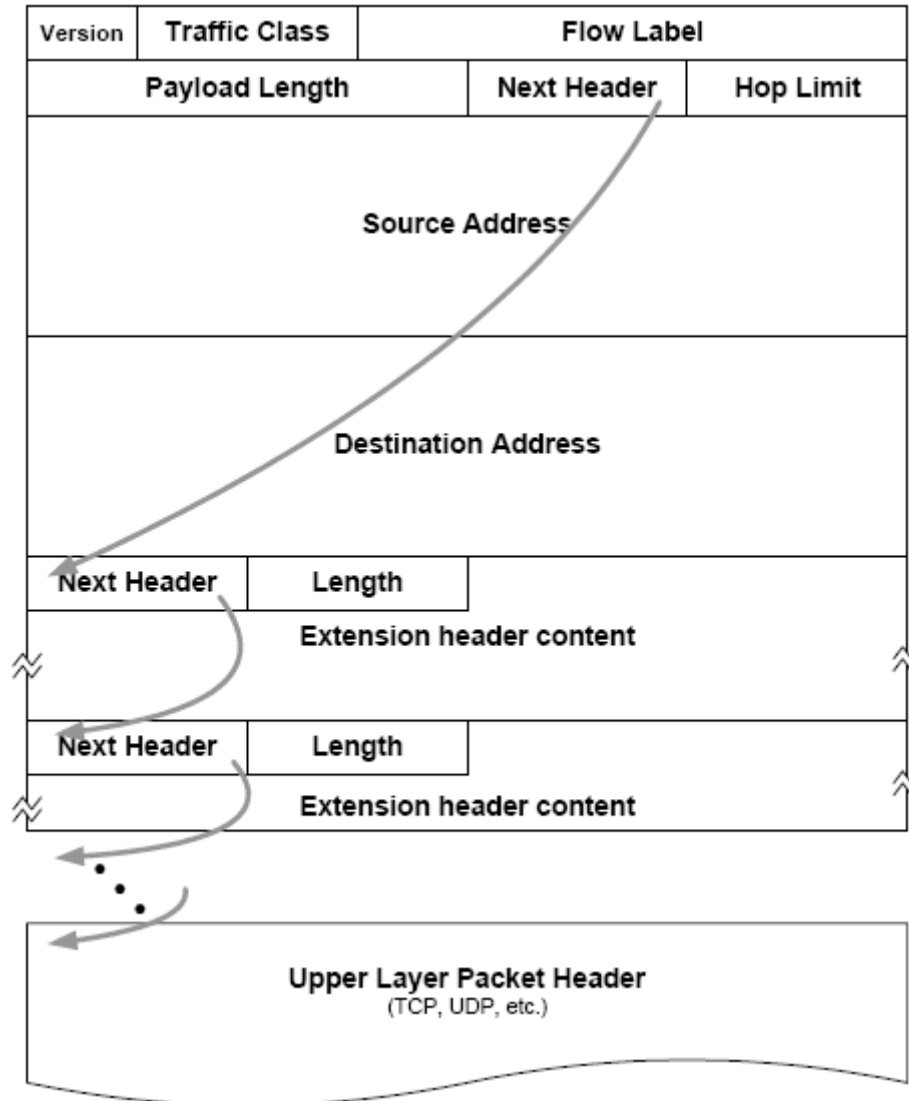
Figure 3-9. IPv6 Extension Header Chaining⁹

Figure 3-9 illustrates further how the NH field in the IPv6 packet header points to the following header in a chain of NHs, which defines the different parts of the payload in the packet. In this example, the NH field of the last extension header in the packet contains the type of the upper-layer protocol, e.g., TCP (value 6) or UDP (value 17). See Table 3-4 for the most common NH values. The IPv6 specification ([RFC 2460](#)) defines six NH values or extension headers. Other extension headers such as the Mobility Header are defined elsewhere.

- **Hop-by-Hop Option header.** NH value of 0 in the IPv6 base header. This header must occur first and is used to carry optional information that must be examined by every node along a packet's delivery path. When the NH value is zero, then the node knows to examine the contents of one or more options contained in the extension header. An example is the Jumbogram IPv6 option, which allows IPv6 to transport packets larger than normal. Jumbograms are discussed in [Section 4.5](#).

- **Routing Header.** NH value of 43 in the immediately preceding header. The Routing Header is used by an IPv6 source (sending host) to list one or more intermediate nodes to be traversed on the way to a packet's destination. The sending host sets the NH extension to direct a packet through a certain path. In IPv4, this is called the *Loose Source and Record Route* option. The Type 0 Routing Header was identified as a security risk, because it may allow attackers to bypass firewalls and carry out denial of service or other attacks. For these reasons the IETF deprecated the use of Type 0 Routing Headers ([RFC 5095](#)). The Mobile IPv6 specification defines a Type 2 Routing Header which allows the data exchange between the care-of address of a mobile node and a correspondent node without being routed through the home agent. Mobile IPv6 is discussed in [Section 4.4](#).
- **Fragment header.** NH value of 44 in the immediately preceding header. This header is used by an IPv6 source to send a packet larger than the Path Maximum Transmission Unit (PMTU) to its destination. Each fragment of a packet must have the same identification value as well as identical source and destination addresses. Fragmentation in IPv6 is performed only by source nodes, not by routers along a packet's delivery path as allowed in IPv4.
- **Authentication header.** NH value of 51 in the immediately preceding header. The Authentication header is used to provide connectionless integrity and data origin authentication for IP datagrams and to provide protection against replays. Also known as IPsec Authentication Header (AH), this capability is used in IPv4 as well as IPv6. IPsec is covered in [Section 5.3](#).
- **Encapsulating Security Payload header.** NH value of 50 in the immediately preceding header. This header is used to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and limited traffic flow confidentiality. Also known as the IPsec Encapsulating Security Payload header (ESP), this capability is used in IPv4 as well as IPv6 to provide the same functions as AH and also, optionally, confidentiality. IPsec as well as AH and ESP is covered in [Section 5.3](#).
- **Destination Options header.** NH value of 60 in the immediately preceding header. This header is used to carry optional information that needs to be examined only by a packet's destination node(s). Intermediate nodes (for which the IPv6 base header destination address is not the address of the current node or more Routing Header processing is required) do not examine the Destination Options header. The Destination Options header works similarly to the Hop-by-Hop Option header, as it may carry one or more options, where each option is encoded in type-length-value format. A node receiving a packet that matches the IPv6 destination address examines the NH field, notes the presence of a Destination Options, and processes the Destination Options before forwarding the payload to the upper layer protocol. A Destination Options header could appear before or after an ESP header; however, it should be placed after an ESP header for increased protection.
- **Mobility header.** NH value of 135 in the immediately preceding header. This header is used by a mobile node, correspondent node, and home agent to manage mobile IPv6 bindings. Mobile IPv6 (MIPv6) provides enhanced security, streamlined administrative protocols, and greater efficiency compared with Mobile IPv4. MIPv6 is covered in [Section 4.4](#).

Extension headers are established at the source before transmission of the packet. Their order and contents are not altered by intermediary nodes.

Extension headers provide important services and functions for the IPv6 protocol. They may direct intermediary nodes about how packet payloads are to be handled prior to reaching their ultimate destination. Defined extension headers for the IPv6 protocol should occur in a recommended order. [RFC 2460, Internet Protocol, version 6 \(IPv6\) Specification](#), indicates that future extension header specifications may be more precise when it comes to ordering. To accommodate the definition and deployment of additional extension headers, each extension header includes information that instructs the receiver how to behave if it does not recognize a new extension header type. The two basic types of behavior are: skip the unrecognized extension header but continue processing the packet or discard the packet.

The following table illustrates the recommended order for extension headers. Only the Hop-by-Hop Options are *required* by the specification to be placed immediately after the IPv6 header (literally the IPv6 “next header”). Note that more than one extension header may be used in a packet. In those cases, [RFC 2460](#) recommends the first four extension headers be prioritized according to the first four entries in the table below.

Table 3-4. IPv6 Extension Headers and Upper Layer Protocols¹⁹

Extension Header	Type	Remarks
Hop-by-hop Options	0	used for options that apply to intermediate routers
Routing	43	used for source routing
Fragment	44	processed only by the final recipient
Destination Options	60	used for options that apply only for the final recipient
Authentication header (AH)	51	used for IPsec integrity protection
Encapsulating Security Payload (ESP)	50	used for IPsec integrity and confidentiality protection
Mobility	135	used for managing mobile IPv6 bindings
Protocol	Type	Remarks
TCP	6	protocol type for Transmission Control Protocol
UDP	17	protocol type for User Datagram Protocol
IPv6-in-IPv6	41	protocol type for IPv6 in IPv6 tunnels
GRE	47	protocol type for Generic Routing Encapsulation tunnels
ICMPv6	58	protocol type, Internet Control Message Protocol for IPv6
No next header	59	dummy packet, often used with ESP
OSPF	89	protocol type, Open Shortest Path First version 3 routing protocol
PIM	103	protocol type, Protocol Independent Multicast routing
SCTP	132	protocol type, Stream Control Transmission Protocol

A NH value of 43 causes the node specified in the Destination Address to examine the Routing Header for further routing instructions. A Routing Header initially contains (1) an ordered list of intermediate Destination Addresses that the packet traverses on its way to its final destination at the end of the list and (2) a pointer to the first one of these. Each time the Destination Address in the IPv6 header is reached, the next Destination Address from the Routing Header is swapped with the Destination Address in the IPv6 header, and the pointer in the Routing Header is advanced until the final destination is reached.

¹⁹ A full list of Extension Headers and Upper Layer Protocols can be found at IANA, [Protocol Numbers](#).

Note that, from a security point of view, simply examining the Destination Address in the IPv6 header may not provide all of the information needed to make packet filtering decisions. An amplification attack, resulting in severe congestion and DoS, can be caused by a packet with a Routing Header containing multiple instances of the same address. Thus, it is crucial to perform ingress filtering that prohibits the forwarding of packets with a Type 0 Routing Header ([RFC 5095](#)).

The NH field and the related extension headers provide IPv6 with flexibility and extensibility, while taking advantage of fixed-length headers with a reduced number of fields as compared with IPv4. NH functionality in IPv6 provides the foundation for enhanced services such as IPv6 security and mobility. They should be kept in mind when securing IPv6 networks, to be discussed later in this document.

Using extension headers can have a number of security implications. Packets containing hop-by-hop extension headers must be analyzed at every node along the forwarding path, and can potentially cause a resource consumption attack. Extension headers also incorporate additional complexity for the purpose of traffic filtering. An example is enforcing a policy that blocks IPv6 traffic with mobility headers if IPv6 mobility is not being used by an organization. Incorporating extension header filtering policies may also impact a device's overall performance. Filtering is discussed in greater detail in [Section 6](#), and ICMPv6 recommended filtering recommendations are provided in Table 3-7. Extension headers can also be used as a "covert channel" to hide communications between two systems, e.g., in Destination Options.

3.5 Internet Control Message Protocol for IPv6 (ICMPv6)

The IPv6 specifications redefine the Internet Control Message Protocol (ICMP) of IPv4 with a number of additions and changes. The resulting protocol is documented in [RFC 4443](#) and called ICMPv6. Specific details regarding ICMPv6 are provided here, along with examples of how the protocol differs from its counterpart under IPv4.

ICMPv6 is an integral aspect of the IPv6 specification. It reports errors if packets cannot be processed properly and sends informational messages about the status of the network. An operational IPv6 network depends upon proper implementation and functionality of ICMPv6. To achieve secure IPv6 operations, it is crucial that network administrators and managers understand the design of ICMPv6 and how it functions. Managers of IPv4-only networks should consider adding the capability of detecting ICMPv6 traffic to enhance security on their networks.

ICMPv6 provides IPv6 with administrative and network diagnostic functions. ICMPv6 provides familiar capabilities like *ping* and *destination unreachable*. In IPv6, as in IPv4, ping can be used by network administrators as a diagnostic tool to confirm that a node's address is properly configured and responsive to specific ICMPv6 requests, called echo requests. ICMPv6 also makes new features like *Neighbor Discovery* (ND) and path MTU discovery possible within IPv6. ND, described in [RFC 4861](#), is the process by which an IPv6 node may learn important information such as link layer addresses of interfaces on its own link.

ND effectively replaces the Address Resolution Protocol (ARP) used with IPv4. ND's multicast messages eliminate the need for the link-level broadcast messages associated with ARP.

3.5.1 ICMPv6 Specification Overview

This section provides an overview of the ICMPv6 specification. The intent of this section is to introduce the reader to fundamentals of the ICMPv6 specification: message format, error handling, and diagnostics. As mentioned previously, specific applications of ICMPv6 (MTU discovery, ND, etc.) are defined in their respective RFCs.

A standard IPv6 header precedes every ICMPv6 message. The IPv6 header may or may not contain extension headers. The IPv6 header identifies the ICMPv6 header with a NH value of 58.

ICMPv6 messages have the following general format, illustrated in Figure 3-10:

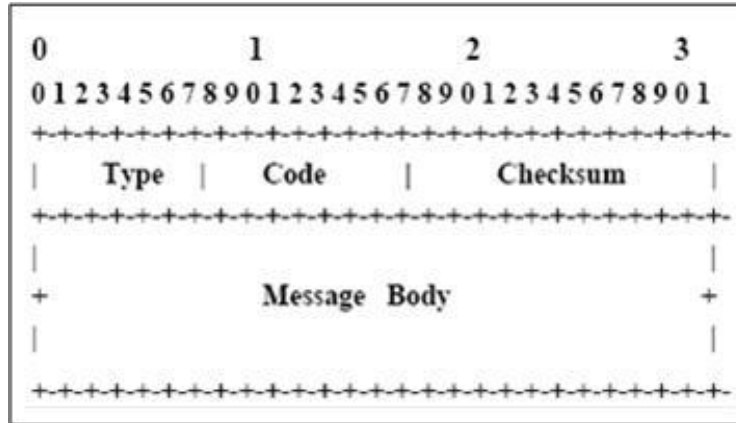


Figure 3-10. ICMPv6 Message Format

- **Type.** Indicates the message type. This eight-bit value determines the format of the remaining data; see Tables 3-5 and 3-6, below.
- **Code.** Depends on the message type (see below). This eight-bit value is used to create an additional level of message granularity.
- **Checksum.** This 16-bit field is used to detect data corruption in the ICMPv6 message and parts of the IPv6 header. In ICMPv6, the checksum calculation includes portions of the IPv6 header, called a pseudo-header, since these fields are not contained within the ICMPv6 data. The reason for the change is to protect ICMP from misdelivery or corruption of those fields of the IPv6 header on which it depends, which, unlike IPv4, are not covered by an Internet-layer checksum. The Next Header field is included in the pseudo-header for ICMP and contains the value 58, which identifies the IPv6 version of ICMP.
- **Message Body.** Field length varies depending on the type and code of the message. ICMPv6 messages are grouped into two classes: *error messages* and *informational messages*.

The ICMPv6 specification defines two classes of ICMP messages: error and informational. Other protocols such as Mobile IPv6 define additional messages within each class. Tables 3-5 and 3-6 provide an overview of some of the most common message types²⁰, along with the additional code information, which depends on the message type.

²⁰ The full list of ICMPv6 Message Numbers and Codes can be found at the IANA [ICMPv6 Registry](#).

Table 3-5. ICMPv6 Error Messages and Code Type

Message Number	Message Type	Code Field
1	Destination Unreachable	0 = No route to destination 1 = Communication with destination administratively prohibited 2 = Beyond scope of source address 3 = Address unreachable 4 = Port unreachable 5 = Source address failed ingress/egress policy 6 = Reject route to destination
2	Packet Too Big	Set to 0 (zero) by the originator and ignored by the receiver
3	Time Exceeded	0 = Hop limit exceeded in transit 1 = Fragment reassembly time exceeded
4	Parameter Problem	0 = Erroneous header field encountered 1 = Unrecognized Next Header type encountered
100 and 101	Private Experimentation	RFC 4443
127	Reserved for expansion of ICMPv6 error messages	RFC 4443

Table 3-6. ICMPv6 Informational Messages

Message Number	Message Type	Code Field
128	Echo Request	RFC 4443 . Used for the ping command
129	Echo Reply	
130	Multicast Listener Query	RFC 2710 . Used for multicast group management
131	Multicast Listener Report	
132	Multicast Listener Done	
133	Router Solicitation	RFC 4861 . Used for neighbor discovery and autoconfiguration
134	Router Advertisement	
135	Neighbor Solicitation	
136	Neighbor Advertisement	
137	Redirect Message	
200 and 201	Private Experimentation	RFC 4443
255	Reserved for expansion of ICMPv6 informational messages	RFC 4443

Error messages are used, for example, if a transmitting node sends a packet of 1500 bytes, but an intermediary hop has an MTU of only 1300. The intermediary node then sends an ICMPv6 Packet Too Big (Type = 2) error message to the sender with information about the problem. For ICMPv6 error messages, the sender includes, at least, the start of the packet causing the error to allow the originator of

the packet to identify the upper-layer protocol and perhaps the process that sent the packet. Informational messages may contain configuration information such as a local router's address in a Router Advertisement (RA) message.

3.5.2 Differences between IPv6 and IPv4 ICMP

Several differences exist between the ICMP specifications for IPv4 and IPv6. These include using ND to replace ARP, dynamic PMTU discovery, and several automated administrative functions unique to IPv6. In particular, some of these differences are:

- **Next Header Value.** IPv6 identifies ICMPv6 messages with a NH value of 58. In IPv4, the corresponding next protocol value is 1.
- **Neighbor Discovery (ND) replaces ARP.** The ICMPv6 ND function serves to locate link-local neighbors and is similar to the function of ARP with IPv4. However, IPv4 has no means to detect whether a neighbor is reachable. With IPv6, ND locates link-local routers, identifies duplicate IPv6 addresses, and eliminates the link-local broadcast traffic generated by ARP. This substantially improves packet delivery in case of failed routers or link interfaces that changed their link-layer address, which solves the problem of outdated ARP caches.
- **Increased PMTU.** The minimum MTU that nodes are required to handle under IPv4 is 576 bytes. In IPv6, all links must handle a datagram size of at least 1280 bytes, and the minimum recommended MTU is 1500 bytes. This is a dramatic increase in the minimum payload each packet must be able to carry, and it results in higher efficiency because fewer headers may need to be processed for a given amount of data.
- **Elimination of in-transit packet fragmentation through the use of PMTU discovery.** In IPv4, packets may be fragmented at any point: at the source or in transit by routers forwarding those packets. In IPv6, only the source may fragment packets. The result of this requirement is that the packet source must use ICMPv6 to determine the PMTU prior to sending traffic and perform fragmentation where needed. The destination node performs reassembly of fragmented packets under both IPv4 and IPv6.
- **Multicast Listener Discovery (MLD).** This is a set of three ICMPv6 messages equivalent to version 2 of the Internet Group Management Protocol (IGMP) for IPv4 used to manage subnet multicast membership. Instead of using IGMP, IPv6 uses ICMPv6 messages for the same functionality, now called MLD. MLD is the protocol that allows multicast listeners to register for multicast addresses they want to receive. Unlike IPv4, IPv6 does not have broadcast addresses. In IPv6, multicast is used with ICMPv6 for infrastructure applications like neighbor discovery and autoconfiguration on local links. IPv6 multicast addresses have new capabilities such as scope, which limits the network realm in which a multicast address is applicable; and embedded unicast prefixes, which limit the scope of the address to the portion of the network that is addressed by that prefix.

ICMPv6 specifies a framework for control messages to provide IPv6 with error handling and parameter establishment functions. Several of ICMPv6's functions are new or different from ICMP under IPv4. Furthermore, ICMPv6 is a fundamental and essential component of any IPv6 implementation. For example, no IPv6 or dual stack IPv4/IPv6 network can function properly without ICMPv6. The next sections contain more detail about ND, Autoconfiguration, and the security ramifications of ICMPv6.

3.5.3 Neighbor Discovery

ND, described in [RFC 4861](#), is the process by which an IPv6 node may learn important information such as link layer addresses of interfaces on the same local segment. ND effectively replaces the ARP found in IPv4. Additionally, it combines this with ICMP Router Discovery and Redirect capabilities. This subsection describes ND in brief, and improvements over the IPv4 set of protocols are noted.

IPv4 is limited in determining whether a neighbor is reachable. The Neighbor Discovery Protocol (NDP) for IPv6 specification is used by all nodes, hosts and routers. IPv6 nodes use ND for the following purposes:⁴

- For autoconfiguration of IPv6 addresses
- To determine network prefixes and other configuration information
- For Duplicate Address Detection (DAD)
- To determine layer two addresses of nodes on the same link (address resolution)
- To find neighboring routers that can forward their packets
- To keep track of which neighbors are reachable and which are not (*Neighbor Unreachability Detection*, or NUD)
- To detect changed link-layer addresses.

As described in [RFC 4861](#), nodes use ND to determine the link-layer addresses for neighbors known to reside on attached links and to purge cached values that become invalid. Hosts also use ND to find neighboring routers that are willing to forward packets on their behalf. Finally, nodes use the protocol to keep track of which neighbors are reachable and which are not, and to detect changed link-layer addresses. When a router or the path to a router fails, a host actively searches for functioning alternates.

ND plays an important role in addressing because it provides address resolution and address autoconfiguration. These are accomplished through the different processes in the ND protocol, which consists of five different ICMP packet types: a pair of Router Solicitation and Router Advertisement messages, a pair of Neighbor Solicitation and Neighbor Advertisement messages, and a Redirect message (listed above under ICMPv6 informational messages). [RFC 4861](#) defines the purpose of these messages, which are all sent via multicast ICMPv6:

- **Router Solicitation (RS).** When an interface becomes enabled, hosts may send RSs that request routers to generate RAs immediately rather than at their next scheduled time.
- **Router Advertisement (RA).** Routers advertise their presence together with various link and Internet parameters either periodically or in response to a RS message. RAs contain prefixes used for on-link determination and address configuration, a suggested hop limit value, the Maximum Transmission Unit (MTU) for the link, etc.
- **Neighbor Solicitation (NS).** Nodes send NSs to determine the link-layer address of a neighbor or to verify that a neighbor is still reachable via a cached link-layer address. NSs are also used for Duplicate Address Detection (DAD).

- **Neighbor Advertisement (NA).** A response to a NS message. A node may also send unsolicited NAs to announce a link-layer address change.
- **Redirect Message.** Used by routers to inform hosts of a better first hop for a destination.

A common use of the NS and NA messages is to resolve IP addresses by discovering the MAC (Media Access Control) addresses of nodes on the same link. A sender must discover the recipient's MAC address to send a data packet. Figure 3-11 illustrates this example of the ND process.

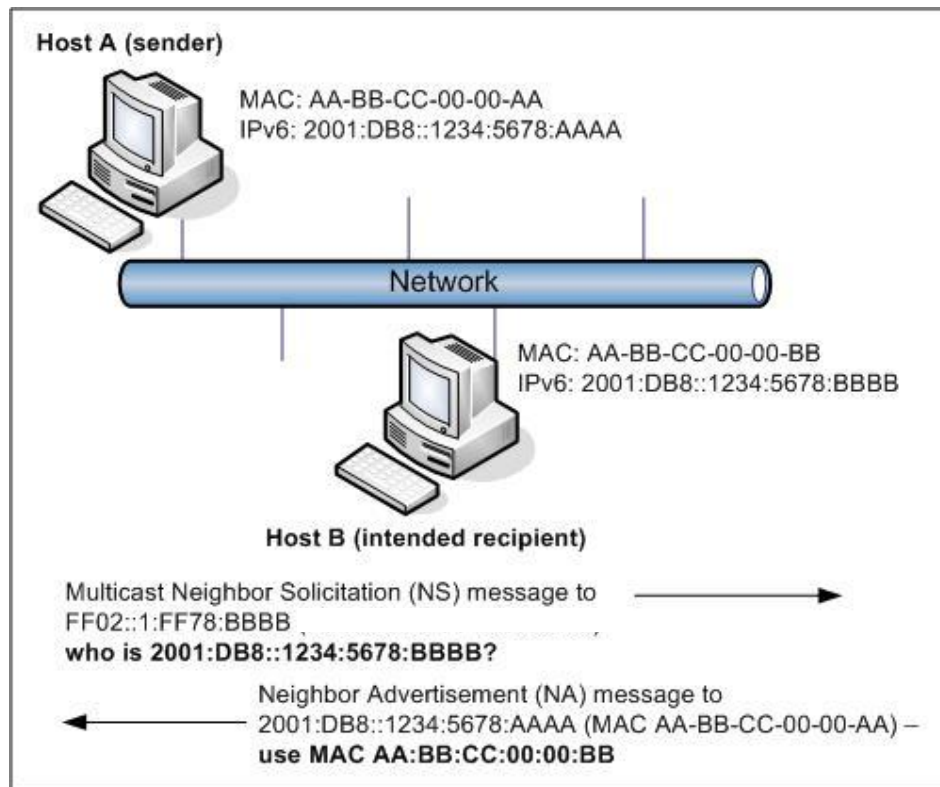


Figure 3-11. Example of Neighbor Discovery

For NS, the sending node (Host A) knows the unicast address of the destination node (Host B) is local because of the network prefix. Knowing this, it uses a link-local solicited-node multicast address to send its NS message. The process that allows the sending node to resolve this link-local MAC address is as follows:

Destination Node Unicast Address:
2001:DB8::1234:5678:BBBB

NS solicited node multicast destination address (last 24-bits of the unicast address):
FF02::1:FFxx:xxxx final form → FF02::1:FF78:BBBB

The sending node takes the low-order 24-bits of the unicast address of the destination node and uses them to fill in the link-local solicited-node multicast address with format “FF02::1:FFxx:xxxx”. The NS message includes both the IPv6 and MAC address of the sender, so the recipient can answer directly and supply its own MAC address. The destination node, knowing its unicast address, will listen for the corresponding solicited node multicast address. When the multicast message arrives at the destination,

the destination node analyses the packet. Seeing the FF02::1 multicast with the ICMPv6 MAC request, the destination checks that it is the intended recipient, updates its neighbor cache, and replies with a NA sending its true MAC address.

ND is essential to the correct operation of an IPv6 network, as well as associated procedures like NUD and DAD. Routers must provide multicast services correctly and obey scope rules, and interfaces must be able to receive solicited node multicast messages. ND security ramifications are discussed in [Section 3.5.6](#); Secure Neighbor Discovery is discussed in [Section 5.4](#).

3.5.4 Autoconfiguration

Autoconfiguration, described in [RFC 4862](#), is essentially plug-and-play networking.

One of the most interesting and potentially valuable addressing features implemented in IPv6, this new feature allows devices on an IPv6 network to configure addresses independently using stateless address autoconfiguration (SLAAC). Whereas in IPv4, hosts were originally configured manually or with host configuration protocols like DHCP, IPv6 autoconfiguration goes a step further by defining a method for devices to configure their IP address and other parameters automatically without the need for a server. Moreover, it also defines a method, *renumbering*, whereby large numbers of IP addresses on a network can be renumbered. This subsection describes autoconfiguration in brief as well as several improvements over the IPv4 set of protocols.

IPv6 defines both Stateful and Stateless address autoconfiguration. SLAAC, illustrated in Figure 3-12, requires no manual configuration of hosts, minimal (if any) configuration of routers, and no additional servers. This allows a host to generate its own addresses using a combination of locally available information and information advertised by routers. Locally available information is delivered to a host when routers advertise prefixes that identify the subnets associated with a link. In turn, a host generates an *interface identifier* that uniquely identifies an interface on a subnet. As previously discussed, an address is formed by combining the two. If a router is not available to advertise subnet prefixes, a host can only generate link-local addresses, which are sufficient for allowing communication among nodes attached to the same link; in the presence of a router, a host will generate its link-local address in addition to other addresses.

Stateful autoconfiguration for IPv4 is known as DHCP. In this case, hosts obtain interface addresses or configuration information and parameters from a server. DHCP servers maintain a database to keep track of which addresses have been assigned to each host. The IPv6 version, DHCPv6, is described in [Section 4.7](#).

As defined in [RFC 4862](#), an IPv6 address can have different states:

- **Tentative Address.** An address whose uniqueness on a link is being verified, prior to its assignment to an interface. A tentative address is not considered assigned to an interface in the usual sense. An interface discards packets addressed to a tentative address except for ND packets related to DAD.
- **Preferred Address.** An address assigned to an interface whose use by upper layer protocols is unrestricted. Preferred addresses may be used as the source (or destination) address of packets sent from (or to) the interface.

- **Valid Address.** A preferred or deprecated address. A valid address may appear as the source or destination address of a packet, and the internet routing system is expected to deliver packets sent to a valid address to their intended recipients.
- **Invalid Address.** An address that is not assigned to any interface. A valid address becomes invalid when its valid lifetime expires. Invalid addresses should not appear as the destination or source address of a packet. In the former case, the Internet routing system will be unable to deliver the packet; in the latter case the recipient of the packet will be unable to respond to it.

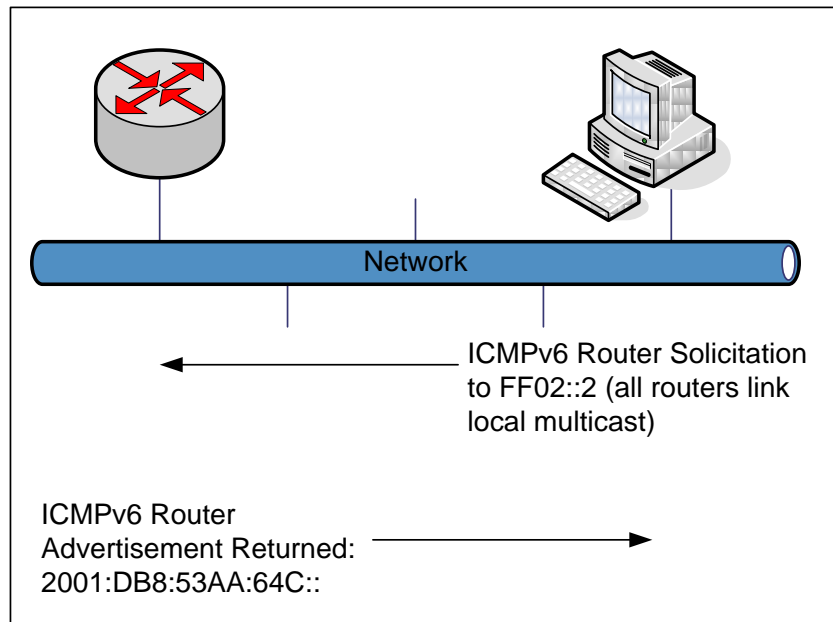


Figure 3-12. Example of Stateless Address Autoconfiguration (SLAAC)

3.5.5 Path Maximum Transmission Unit (PMTU) Discovery

With IPv4, a router can fragment a packet when the Maximum Transmission Unit (MTU) of the next link is smaller than the packet it has to send. The router does this by slicing the packet to fit into the smaller MTU and sending it out as a series of fragments. The packet is then reassembled at the final destination, which can be very inefficient. It can also introduce additional traffic into the network, in the form of an increased number of smaller-than-necessary packets, as well as necessitating re-transmission of packets if all of the fragments do not arrive within a specific time interval. With IPv6, routers do not fragment packets; instead, the sender discovers the maximum packet size by using *Path Maximum Transmission Unit* (or PMTU) for the entire path. PMTU discovery is the process by which each node on the network establishes an important IPv6 parameter for a given communication session. The PMTU establishes the maximum packet size, measured in bytes, which may be carried across a sequence of network nodes. PMTU for a local network segment can be determined directly from the MTU in RA messages.

The IPv6 specification calls for a MTU of at least 1280 bytes. In other words, all links must accept packets of any size up through 1280 bytes. The minimum recommended MTU for IPv6 is 1500 bytes. The IPv6 specification does not allow intermediate nodes (such as routers) to fragment packets in transit. Instead, PMTU is determined dynamically and maintained throughout each communication session. By disallowing fragmentation by intermediary devices, IPv6 achieves a level of efficiency that is generally not available under IPv4. Intermediate nodes operate faster and with less processing overhead by not

having to fragment packets, and destinations receive fewer packets requiring reassembly. Since originating nodes know the PMTU, they can in most cases ensure that traffic is sent with no fragmentation at all.

The PMTU of any given network path is equal to the smallest MTU along that path. For example, Figure 3-13 depicts two nodes establishing a communications session across three intermediate nodes. The links to the two end nodes (A and B) have MTUs of 1500, whereas the links connecting the three intermediary nodes (1 and 2; 2 and 3) have MTUs of 1300 and 1800, respectively. The PMTU of the network path between nodes A and B is 1300 bytes. This is because the MTU from node 1 to node 2 is only 1300 bytes, the smallest of the four hops.

In this example, ICMPv6 is employed by all nodes to recognize and configure the PMTU automatically. The two endpoints of the path set the PMTU to 1300 without administrative intervention. The size of an IPv6 packet traversing this path for this session needs to conform to the PMTU for the path; otherwise it is dropped and the sender receives an ICMPv6 error message requesting it to retransmit using a smaller packet size.

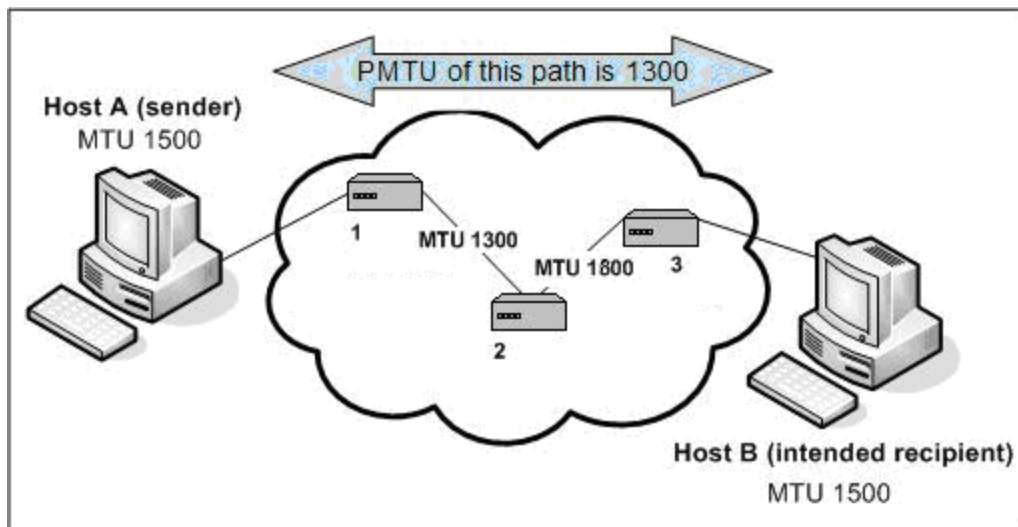


Figure 3-13. Significance of MTU under IPv6

The PMTU protocol is detailed in [RFC 1981](#).

3.5.6 Security Ramifications

This section describes security considerations for ICMPv6. Examples include denial of ICMPv6 traffic that can effectively result in a denial of service condition for a given network. The functions of Neighbor Discovery and PMTU discovery are entirely dependent upon the proper functioning of ICMPv6. Not only is proper Neighbor Discovery configuration essential to a functional IPv6 network, but misconfiguration of RA and RS parameters can compromise the security of the network.

Rogue routers or misconfigured nodes may be inserted on a local network segment and configured to propagate false or inaccurate RA messages. Network nodes on this segment might then learn false routing information that would result in network traffic being sent through the rogue router. Further, forged RA messages may be sent from legitimate hosts on an IPv6 network segment, possibly causing

other nodes to forward packets to non-existent routers and resulting in a denial of service or man-in-the-middle attack. Additionally, malicious responses to DAD messages can cause denial of service conditions on a local network segment; DAD can be disabled on specific network or security device interfaces to prevent Denial of Service (DOS) attacks. Spoofed NS/NA packets can cause redirect or DOS attacks. Secure Neighbor Discovery (SEND) was developed to help mitigate some of the security issues caused by rogue devices. SEND is discussed in more detail in [Section 5.4](#).

Router access control lists (ACLs), firewalls, and other security components must be carefully managed to retain ICMPv6 functionality. Any security measures on a network segment must allow IPv6 nodes to use ICMPv6 to accomplish Neighbor Discovery, PMTU discovery, and other essential tasks. If an IPv6 default router on a network segment is unable to receive and reply to legitimate RS messages, nodes sending those messages may experience a denial of service condition. PMTU messages should also be allowed to traverse network perimeters, to enable proper functioning of IPv6 end-to-end communications.

Some ICMPv6 traffic (e.g., RS and RA) is only useful on the local segment, uses link-local addresses, and should never be routed. Port-based ACLs can be used to prevent any Ethernet port from receiving RA messages coming inbound to the network from an end-user computer. ACLs implemented on common routers can be configured to allow other appropriate ICMPv6 traffic to pass despite the presence of the default *deny all* parameter. Network administrators should confirm the inherent ICMPv6 capabilities of their router's operating system and confirm the configuration parameters that govern the handling of ICMPv6. There may be places in the network (e.g., a Demilitarized Zone [DMZ]) where it makes sense to disable Neighbor Discovery and use static neighbor cache entries instead.

Any IP network, whether it is IPv4-only or a dual stack IPv4/IPv6 network, must have the capability to detect and examine ICMPv6 and IPv6 packets. Without this capability, rogue IPv6 nodes may be operating on a network that is intended to handle only IPv4. ICMPv6 is fundamental to the operation of IPv6 networks, and even malicious IPv6 nodes will depend upon ICMPv6 to operate. Network administrators and managers should evaluate the capability of their existing tools used to monitor ICMP traffic for similar support for ICMPv6.

Use of IPsec to authenticate the sender and validate the contents of ICMPv6 messages is often not possible. Additionally, establishing security associations with all possible sources of ICMPv6 messages is generally not possible. Furthermore, some ICMP messages (e.g., PMTU) may be returned from intermediate routers, not from the message's ultimate destination. A further discussion about the use of IPsec with ICMPv6 is included in [Section 5.4.1](#). Due to this inability to establish security associations, alternatives should be used to reduce the vulnerability to ICMPv6-based attacks.

It is essential to establish strict filtering policies in site firewalls to limit ICMPv6 messages that can pass between the site and the Internet. Firewall treatment of ICMPv6 traffic requires finer-grained handling than may have sufficed for IPv4. Certain IPv6 features, such as Path MTU Discovery, require the transmission of the associated ICMPv6 messages (e.g., Packet Too Big) for proper functionality. On the other hand, messages associated with other features, like Network Discovery, should **not** be transmitted from outside the local link.

Hence, providing only a blanket "allow all"/"block all" choice for ICMP traffic handling, which may have been adequate for IPv4, is not appropriate for IPv6. The Federal IPv6 Profile²¹ requires firewalls to at least be able to block/allow ICMPv6 traffic selectively based on type and code.

More sophisticated firewalls can do better: by associating connection-specific ICMPv6 types (such as

²¹ [NIST SP 500-267](#), *A Profile for IPv6 in the U.S. Government – Version 1.0*.

Packet Too Big or, in some situations, Destination Unreachable) with the corresponding state records for existing connections (TCP, UDP, or SCTP streams), such stateful firewalls can limit even allowed types to those that could legitimately be part of permitted traffic flows.

Table 3-7 gives recommended ICMPv6 firewall settings for various possible traffic types. The “Maintenance of Communications” and “Error Messages” sections of the table assume the more sophisticated connection-association capabilities mentioned above; less-sophisticated firewalls would have to allow these messages whenever a corresponding connection could occur, based on other rules (e.g., allowed source and destination addresses).

The recommendations for ICMPv6 firewall filtering in Table 3-7 are based on [RFC 4890](#), *Recommendation for Filtering ICMPv6 Messages in Firewalls*. These recommendations allow propagation of ICMPv6 messages needed to maintain functionality of the network but drop messages posing potential security risks. Many ICMPv6 messages should only be used in a link-local context, rather than end-to-end, and filters need to be concerned with the types of addresses in ICMPv6 packets as well as the specific source address, destination addresses, and ICMPv6 Type. [RFC 4890](#) classifies ICMPv6 messages according to whether they are designed for end-to-end communications (traffic to transit a firewall) or local communications within a link (local traffic addressed to an interface on a firewall). All experimental and undefined ICMPv6 messages should be dropped. ACLs should permit only those ICMPv6 messages that are required, based on specific local needs and policies; all others should be dropped.

Table 3-7. ICMPv6 Recommended Filtering Actions – Must Not Drop & Should Not Drop

Message (Type)	Must Not Drop		Should Not Drop	
	Transit	Local	Transit	Local
Maintenance of Communication:	Allow non-local when associated with allowed connections			
Destination Unreachable (1) – All codes	•	•		
Packet Too Big (2)	•	•		
Time Exceeded (3) – Code 0 only	•	•		
Parameter Problem (4) – Codes 1 and 2 only	•	•		
Connectivity Checking:	Allow/disallow non-local based on topology/information concealment policy			
Echo Request (128)	•	•		
Echo Response (129)	•	•		
Address Configuration and Router Selection:	Allow in link-local traffic only			
Router Solicitation (133)		•		
Router Advertisement (134)		•		
Neighbor Solicitation (135)		•		
Neighbor Advertisement (136)		•		
Inverse Neighbor Discovery Solicitation (141)		•		
Inverse Neighbor Discovery Advertisement (142)		•		
Link-Local Multicast Receiver Notification:	Allow in link-local traffic only			
Listener Query (130)		•		
Listener Report (131)		•		
Listener Done (132)		•		
Listener Report v2 (143)		•		
SEND Certification Path Notification:	Allow in link-local traffic only			
Certification Path Solicitation (148)		•		
Certification Path Advertisement (149)		•		
Multicast Router Discovery:	Allow in link-local traffic only			
Multicast Router Advertisement (151)		•		
Multicast Router Solicitation (152)		•		
Multicast Router Termination (153)		•		
Error Messages:	Allow non-local when associated with allowed connections			
Time Exceeded (3) – Code 1			•	•
Parameter Problem (4) – Code 0			•	•
Mobile IPv6:	Allow non-local for predefined endpoints			
Home Agent Address Discovery Request (144)			•	
Home Agent Address Discovery Reply (145)			•	
Mobile Prefix Solicitation (146)			•	
Mobile Prefix Advertisement (147)			•	

3.6 IPv6 and Routing

Routing protocols fall into two general types. Interior Gateway Protocols (IGPs) are designed for use within an autonomous system (AS), that is, among routers that are all controlled by the same enterprise or organization. Exterior Gateway Protocols (EGPs) are designed for exchanging routes between autonomous systems, such as between network carriers or between a large enterprise and its network service providers. To support IPv6, routing protocols, such as Routing Information Protocol (RIP), OSPF, Intermediate System to Intermediate System (IS-IS), Enhanced Interior Gateway Routing Protocol (EIGRP), and Border Gateway Protocol (BGP), had to be updated. This section covers changes to routing protocols, as well as security support for such protocols.

3.6.1 Specification Overview

Open Shortest Path First (OSPF) is a link-state hierarchical Interior Gateway Protocol (IGP). Dijkstra's algorithm is used to calculate the shortest path tree. It uses path cost as its routing metric. Path cost is determined generally by the speed (i.e., bandwidth) of a given route. IPv4 networks run OSPF version 2 (OSPFv2) as specified in [RFC 2328](#).

OSPF version 3 (OSPFv3) is designed specifically for IPv6 and is specified in [RFC 5340](#). OSPFv3 for IPv6 is a completely independent routing protocol from OSPFv2 for IPv4. OSPFv3 expands on OSPFv2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses. OSPF is commonly used in large enterprise networks.

The Routing Information Protocol (RIP) is a distance-vector routing protocol that employs hop count as a routing metric. IPv4 uses RIP version 2 (RIPv2) as specified in [RFC 2453](#). The IPv6 version of RIP is standardized in [RFC 2080](#). It is easy to configure, but offers limited flexibility and scalability. RIP enhancements for IPv6, detailed in [RFC 2080](#), also known as RIPng, include support for IPv6 addresses and prefixes and the use of the all-RIP-routers multicast group address FF02::9 as the destination address for RIP update messages. Each RIPng update contains a copy of the entire routing table. RIPng is suited for networks of modest size only.

IS-IS is an IGP that advertises link-state information throughout the network to create a picture of the network topology. IS-IS is an Open Systems Interconnection (OSI) hierarchical routing protocol that designates an intermediate system as a Level 1 or Level 2 device. Level 2 devices route between Level 1 areas to create an intradomain routing backbone. Integrated IS-IS uses a single routing algorithm to support several network address families, such as IPv6, IPv4, and OSI. It runs point to point over the link layer protocol; it does not *use* IPv4 or IPv6. Each IS-IS update contains only changes to the network topology. IS-IS is flexible, efficient, and suitable for large IPv4/IPv6 networks.

Enhanced Interior Gateway Routing Protocol (EIGRP) is Cisco's proprietary routing protocol. EIGRP is an advanced distance-vector routing protocol, with optimizations to minimize both the routing instability incurred after topology changes, as well as the use of bandwidth and processing power in the router. EIGRP and IGRP are compatible with each other. EIGRP for IPv4 and EIGRP for IPv6 are configured and managed separately.

BGP4 is the Exterior Gateway Protocol (EGP) used mainly to connect separate routing domains that contain independent routing policies (autonomous systems). Connecting to a service provider for access

to the Internet is a common use for Border Gateway Protocol (BGP). BGP version 4 with multi-protocol extensions supports both IPv4 and IPv6. Each BGP update contains only changes to the network topology. BGP is efficient and flexible. Multi-protocol BGP is standardized in [RFC 4760](#).

3.6.2 Security for Routing Protocols

Routing protocols can be subject to threats such as unauthorized updates for either IPv4 or IPv6 routes. Security capabilities have been designed for routing protocols to mitigate unauthorized update threats. Some IPv6 routing protocols rely on similar mechanisms to those in IPv4 for protection, while others have incorporated IPsec for protection. These security mechanisms do not provide end-to-end security for routing protocols across multiple hops, because while they provide integrity assurance for routing protocol messages between nodes, they do not verify the integrity of messages received from other nodes that are not part of a security association. This is a major security concern for EGPs and a somewhat lesser one for IGPs.

RIPng

RIP for IPv4 uses an MD5-based integrity mechanism; this was removed from RIPng. RIPng offers no integrity assurance features. Per [RFC 2080](#), RIPng leverages IPsec for security. It should be noted that hardware vendors have not incorporated IPsec features as a configuration option, instead relying on native IPv6 IPsec support from the operating platform for protection. RIPng is suitable only for small, private networks where the threat of routing attacks is substantially reduced.

OSPFv3

Securing OSPFv2 in a dual stack environment will protect neither the OSPFv3 protocol nor the OSPFv3 routing table. OSPFv2 allows null, password-based, or cryptographic authentication using MD5-based integrity for routing updates. The authentication fields found in OSPFv2 have been removed from the OSPFv3 packet for IPv6, so MD5 is not an authentication option. OSPFv3 offers no integrity assurance features itself and relies on IPsec AH or ESP for authentication, integrity, and confidentiality. Note that OSPFv3 uses unicast and multicast, and IKE does not work with multicast, so the default method is to use manual keying. Since replay protection cannot be provided using manual keying, OSPFv3 messages are vulnerable to replay attacks, which can lead to DoS attacks, Central Processing Unit (CPU) overload, and localized routing loops. IPsec for OSPFv3 is detailed in [RFC 4552](#).

With routing protocols, routing integrity is usually a greater concern than confidentiality. The ESP parameter NULL indicating no encryption is generally regarded to be an acceptable choice for OSPF security.

IS-IS and EIGRP

Both IS-IS and EIGRP support simple MD5-based integrity for protecting IPv6 routing updates, similar to protecting routing updates for IS-IS and EIGRP for IPv4.

BGP

The use of BGP as an inter-AS routing protocol means that it can be subject to serious threats. Three mechanisms exist to mitigate threats to BGP. The first is the use of MD5-based integrity to protect routing updates. The second mechanism to mitigate threats to BGP is GTSM ([RFC 5082](#)). GTSM is a simple security mechanism for rejecting spoofed BGP messages based on their IP TTL or Hop Limit.

The sending BGP router always uses a TTL=255, and the receiving BGP router checks that the TTL has the expected value of 255. Any packets from a remote attacker would have to travel via intervening routers, would have a smaller-than-maximum TTL, and would be dropped on receipt. Note however that a router operating as the endpoint of a tunneling protocol may not decrement the hop count upon receiving packets through the tunnel, so these could conceivably come from anywhere with TTL=255. The third mechanism to mitigate threats to BGP is IPsec. IPsec key management can use shared secrets or public key certificates, which allow IPsec to offer scalability. GTSM has the lowest overhead of the three mechanisms, and is the easiest to configure. It also offers the least effective protection. The MD5 signature mechanism offers low overhead and effective protection, but it forces administrators to disrupt their BGP sessions at each key update, and it does not scale well. IPsec offers the most effective protection, least disruption, and best scalability. It also imposes the highest overhead (although the overhead is still small), and it is the most complex mechanism to configure. In summary, using an MD5 checksum is certainly better than nothing, but MD5 itself can be attacked successfully, and most of these methods have no easy ways to change hash functions or even change keys. IPsec is preferable for routing protocols that support its use. All of the above security mechanisms protect against unauthorized insertion or manipulation of routing protocol messages.²² They do not protect against a corrupted or malfunctioning router that may construct and pass along incorrect routing information. Many approaches to providing better end-to-end security for BGP have been proposed, but consensus on a single solution has not yet been achieved.

3.6.3 Unknown Aspects

As IPv6 deployment grows globally so will the Internet's IPv6 global routing tables. It is very well possible that this expanded growth can impact service providers that already are faced with large IPv4-only routing tables. By assigning IPv6 addresses through service providers, in a hierarchical fashion, there is a greater possibility that efficient aggregation for IPv6 addressing will occur. If more organizations succeed in obtaining PI addressing, then there is a higher risk that global routing tables could explode in size, and hence cause possibly costly hardware upgrades to deal with extremely large routing tables. To assist with developing a scalable Internet architecture, use of techniques that separate end-systems' addressing space and routing locators' space (locator-ID split) are also being investigated.

3.7 IPv6 and the Domain Name System (DNS)

The Domain Name System (DNS) is essential for almost all use of the Internet. It must be available, and it must provide accurate information. Threats such as denial of service against the top-level servers are taken extremely seriously. Perhaps the biggest network security story of 2008 was the demonstration of a new attack that can insert false information into a DNS server's cache. Upgrading DNS security is one of the major current challenges for ISPs²³.

The *Secure Domain Name System (DNS) Deployment Guide*²⁴ contains background information on DNS as it is used with IPv4, the possible attacks against DNS, and appropriate security measures. The *Domain Name System – Security Technical Implementation Guide version 4r1*²⁵ covers similar topics for Department of Defense (DoD) networks and also has advice on using DNS on particular computing platforms and operating systems. Both of these documents emphasize extensive experience with IPv4. Architecting and structuring DNS services is a complex topic that is outside the scope of this publication; extensive guidance about secure DNS services can be found in NIST SP (Special Publication) 800-81.

²² [NIST SP 800-54](#), *Border Gateway Protocol Security*.

²³ ISOC Conference, [Securing the DNS](#).

²⁴ [NIST SP 800-81](#), *Secure Domain Name System (DNS) Deployment Guide*.

²⁵ DISA Report, [Domain Name System – Security Technical Implementation Guide version 4r1](#).

This section reviews the main aspects of DNS briefly to help understand the changes needed to make DNS work with IPv6 and how these affect the secure operation of DNS.

DNS is a hierarchical, distributed database that translates logical, human readable names such as `www.example.com` into binary IP addresses used by applications like email and web browsers. The most common format of a DNS name is referred to as the Uniform Resource Locator (URL), or simply as the web address. DNS database entries are called *resource records*²⁶. IPv4 and IPv6 addresses are different types of resource records, but the DNS handles both essentially the same. What is important to remember is that a single URL may have IPv4 addresses, IPv6 addresses, both, or neither.

DNS services are fully defined for IPv6 in [RFC 3596](#). Additional information can be found in [RFC 4472](#), *Operational Considerations and Issues with IPv6 DNS*. The root and common top-level domain (TLD) servers are all IPv6 capable today. Nevertheless, IPv6 addresses for production services usually cannot be found in the DNS, particularly for services based in the United States or North America. Most of these services do not have IPv6 access enabled, and most users would not have easy IPv6 connectivity to them if they were. Getting IPv6 DNS entries close to parity with IPv4 DNS on the Internet is unlikely to happen without demand for IPv6 access to popular, high-volume web sites and other services.

Most DNS implementations have been upgraded to support IPv6, but older software applications may assume that DNS address queries return only 32-bit IPv4 addresses. Fully IPv6-capable DNS implementations not only need to handle 128-bit IPv6 addresses but also need to run over IPv6 with the same UDP and TCP port number, 53, as used by IPv4 ([RFC 3901](#), *DNS IPv6 Transport Operational Guidelines*)²⁷.

The primary components of DNS are the DNS root and TLD servers, authoritative DNS servers, local caching servers, and clients called *resolvers*. A resolver requests resource records from a local caching server. If the local caching server does not have a requested record, it uses the information it does have to start querying authoritative servers.

3.7.1 DNS Transport Protocol

It is important to note that while IPv6 address queries may be made over an IPv4 or IPv6 network, IPv6 transport of DNS messages is not required for looking up IPv6 addresses in DNS. The query, not the transport protocol, should always determine what information is returned in the Answer, Authority, and Additional sections of a response. A host can request an IPv6 address even if the network on which that host resides is IPv4-only. To obtain both the IPv4 and IPv6 addresses, a single DNS query could be used, but it is recommended that two separate requests be used. The choice as to which protocol to use first is configurable. Hosts and DNS servers running both IPv4 and IPv6 should have no problem with this aspect of DNS, but all zones should be set up so that they have at least one IPv4-enabled authoritative server, and IPv6-only systems should follow or exceed the minimum configuration guidelines in [RFC 3901](#) to ensure that they do not get cut off from the rest of the DNS tree because they cannot communicate with any IPv4-only DNS servers.

3.7.2 DNS Specification Overview

[RFC 3596](#), *DNS Extensions to Support IP Version 6*, defines the changes needed to DNS to support IPv6.

²⁶ Technically, we are discussing the IN (Internet) CLASS of the DNS. A complete list of resource record types can be found at the IANA [DNS Registry](#).

²⁷ It is incumbent upon authoritative name servers accessible with IPv6 transport to maintain connectivity with today's predominant part of the DNS accessible with IPv4 transport to avoid splitting the namespace.

A new resource record type, AAAA (pronounced quad-A), is defined to store a host's IPv6 address. A 128-bit IPv6 address is encoded in the data portion of a AAAA resource record in network byte order (high-order byte first). A AAAA resource record stores a single IPv6 address, so a host with more than one IPv6 address may have more than one such record. AAAA queries for a specified domain name return all associated AAAA resource records in the answer section of a response.

A special domain is defined to look up names corresponding to an IPv6 address. The intent of this domain is to provide a reverse mapping of an IPv6 address to a host name (stored as a DNS PTR resource record). The domain is rooted at IP6.ARPA. An IPv6 address is represented as a name in the IP6.ARPA domain with a sequence of four-bit nibbles written as hexadecimal digits and separated by dots with the suffix IP6.ARPA. The sequence of 32 nibbles is encoded in reverse order, i.e., the low-order nibble is encoded first, followed by the next low-order nibble and so on. For example, the reverse lookup domain name corresponding to the address 2001:db8:2:3:4:5:678:90ab is:

b.a.0.9.8.7.6.0.5.0.0.4.0.0.3.0.0.2.0.0.0.8.b.d.0.1.0.0.2.IP6.ARPA

Note that zero suppression and double-colon compression cannot be used in reverse DNS names.

All existing query types that perform type A (IPv4 address) additional section processing, i.e., name server (NS), service location (SRV), and mail exchange (MX) query types, must be modified or redefined to perform both type A and type AAAA additional section processing. This means that a name server must add any relevant IPv4 addresses *and* any relevant IPv6 addresses available locally to the Additional Section of a response when processing any one of the above queries.

Several DNS implementations have been observed handling queries for AAAA records incorrectly, and these have been documented in [RFC 4074](#). They may return the wrong data, wrong error codes, or nothing at all. Most of these errors result in unreachable services, delays, timeouts, or faulty assumptions by caching servers, but the RFC indicates places where these errors can also be exploited in denial-of-service attacks. In addition, some firewalls and Intrusion Detection Systems (IDSs) have experienced problems handling DNS responses that are larger than 512 bytes.

When performing a DNS address query, DNS is responsible only for resolving a domain name to a set of IP addresses. Applications and operating systems are responsible for choosing how to use the IPv6 AAAA or IPv4 A records that may be returned. This topic is called address selection, and it is an important part of using IPv6 and dual IPv4-IPv6 networks correctly. As a result, receiving unexpected AAAA records may cause an application that is not IPv6 aware to fail. Combining both IPv6 and IPv4 records into the same domain can lead to application problems that are beyond the scope of the DNS administrator. The *Domain Name System – Security Technical Implementation Guide version 4r1* mentioned above recommends using different DNS names for IPv6-enabled hosts until all such problems (or at least the critical ones) are fixed. It also may be useful during pilot tests or early deployment to use names in separate domains for IPv6 servers, e.g., imap.IPv6.example.com, although one would like such names to handle IPv4 and IPv6 completely transparently in the long term.

A dual stack host needs a mechanism for choosing between IPv4 and IPv6. When DNS returns a set of different addresses, resolvers need to be configured either to choose which addresses to pass to an application or to forward all of the addresses and leave the choice to the application.

IPv6 link-local addresses should never be put into the DNS (and site-local addresses should not be used at all). Temporary ([RFC 3041](#)) addresses are usually meant to be anonymous, so putting them into DNS would be an unusual choice and would require frequent updates. Putting 6to4 addresses into DNS may be sensible, but one may need cooperation from a local or regional registry to set up the IP6.ARPA PTR

records for 6to4 addresses and for other address formats with specific prefixes and embedded IPv4 addresses.

For additional details about adding DNS records for new services, handling time-to-live values in caches, obtaining a list of DNS servers when DHCPv6 is not used, updating forward (AAAA) and reverse (PTR) entries, handling dynamic DNS, and renumbering, see [RFC 4472](#).

3.7.3 Security Impact and Recommendations

3.7.3.1 General DNS Security Recommendations

All of the general accounts of threats against DNS ([RFC 3833](#)) and advice for securing DNS, independent of IPv6, apply. This begins with host security and software security for DNS servers. It includes keeping up with new software releases, vulnerability alerts, and patches. The security of DNS servers can be improved with isolation, redundancy, geographic diversity, network path diversity, and potentially platform diversity. Administrative access should be controlled and secured. Tools for checking that a zone file is well formed or that other configuration variables are correctly set should be used. DNS servers should be included in any penetration testing exercises.

It is good practice to implement certain well-known security best practices, which rely on not divulging more information than necessary. The IP address of a hidden master server should not be advertised, nor should the DNS software version number, lest an attacker easily exploit bugs known to be in a certain release. Common implementations of DNS support access control lists based on IP addresses, and at least some of these support access control lists based on both IPv4 and IPv6 addresses. Address-based security is regarded as a rather weak form of authentication, particularly for important actions like DNS dynamic update, but nevertheless it is an efficient way to provide some protection if it is used together with ingress and egress address filtering. The widely-used technique of “split DNS,” whereby an enterprise’s DNS servers provide different answers to internal and external queries, can be used the same way with either A records or AAAA records.

Based on recent experiences, two attack scenarios are particularly likely. One is denial of service. Excess capacity and diversity help. Firewalls and intrusion detection systems can help protect a server located in a “demilitarized zone.” Administrators should be prepared to contact the appropriate emergency response teams and law enforcement agencies. [RFC 5358](#) contains advice for configuring DNS servers to make using them as amplifiers in a denial-of-service attack against a third party more difficult.

The second scenario is so-called cache poisoning—inserting false information into a server’s cache, so that, for example, users may be misdirected to a bogus web site. This is an old idea, but a much more effective method for accomplishing it became well known in 2008. With or without cryptographic protection, the recommended remedy is to force the attacker to guess two randomly chosen 16-bit values simultaneously instead of just one ([RFC 5452](#)).

3.7.3.2 Cryptographic Protection of DNS

Two standard cryptographic protocols are available for securing DNS. They can be used equally with IPv4, IPv6, and combined IPv4-IPv6 implementations. DNSSEC, the DNS security extensions, is defined in three RFCs ([RFC 4033](#), [4034](#), [4035](#)), and TSIG, the Secret Key Transaction Authentication protocol is described in a fourth RFC ([RFC 2845](#)).

The TSIG protocol provides data origin authentication and message integrity for DNS transactions by adding message authentication codes based on shared secrets. Originally, only the HMAC-MD5

algorithm was specified, but TSIG now requires HMAC-SHA-1 and HMAC-SHA-256 as well ([RFC 4635](#))²⁸. It can be used to authenticate dynamic updates as coming from an approved client, responses as coming from an approved recursive name server, or zone transfers as coming from an authoritative server. Its most common use is to protect zone transfers, but protecting dynamic updates is an important application as well.

TSIG is widely deployed and strongly recommended. Normally, network administrators use an out-of-band mechanism to configure name servers and resolvers with shared secrets. However, a secure, automated mechanism for key distribution and key update is a much more desirable solution. It simplifies operations and enhances security. The TKEY (Transaction Key) protocol ([RFC 2930](#)) offers several options—the Diffie-Hellman method is a practical choice. One limitation with TSIG is that there are no levels of authority, so any host with the secret key may update any record.

An alternative to TSIG called SIG(0) and described in [RFC 2931](#) uses public keys and digital signatures instead of message authentication codes based on shared secrets. It is far less widely used than TSIG, but it may be a practical alternative especially for securing dynamic updates.

DNSSEC provides an entirely different set of cryptographic security mechanisms. Its objective is to secure the DNS database itself by deploying a hierarchical infrastructure of signed resource records and its own built-in public key infrastructure. It accomplishes this by defining four new types of resource record. The RRSIG resource record (RR) contains a digital signature of another resource record. The DNSKEY (Domain Name System Key) resource record contains a signature verification key; it, in turn, is signed with an RRSIG resource record. The DS (delegation signer) RR names the signer of a delegation. If DNSSEC is fully deployed, the DS records can form a chain from any zone to the root. The NSEC (next secure) resource record specifies the name of the next secured entry in a zone (in lexicographic order), so that the non-existence of a resource record can be verified cryptographically.

Besides its added complexity, especially for signing large zones, DNSSEC also increases the size of files and messages substantially. Also, administrators need to protect the secrecy of their private signing keys carefully. DNSSEC has not been deployed widely yet, but this is gradually changing. Newer releases of DNS software support DNSSEC, and plans are underway to sign the root and many major top-level domains. The US Government's .gov root was signed in Feb. 2009, and OMB mandated²⁹ that the Government's second-level domains should be signed by Dec. 2009. OMB Memorandum M-08-23 states: "The Government's reliance on the Internet to disseminate and provide access to information has increased significantly over the years, as have the risks associated with potential unauthorized use, compromise, and loss of the .gov domain space. Almost every instance of network communication begins with a request to the Domain Name System (DNS) to resolve a human readable name for a network resource (e.g., www.usa.gov) into the technical information (e.g., Internet Protocol address) necessary to actually access the remote resource. DNSSEC provides cryptographic protections to DNS communication exchanges, thereby removing threats of DNS-based attacks and improving the overall integrity and authenticity of information processed over the Internet." It is hoped that this will provide the impetus and experience necessary to get DNSSEC deployed for commonly used services and large enterprises throughout the Internet.

3.7.3.3 IPv6-Specific DNS Security Recommendations

During any IPv6 deployment, DNS services may have to support both IPv4 and IPv6. In fact, DNS

²⁸ [RFC 4635](#), *HMAC SHA TSIG Algorithm Identifiers*, specifies additional mandatory and optional TSIG algorithms and how to handle truncation of the message digest.

²⁹ [OMB Memorandum M-08-23](#), *Securing the Federal Government's Domain Name System Infrastructure*.

services should be among the first to be fully dual stack capable in any deployment effort. Software applications need to be modified to query for both forms of addresses and to choose between them. IPv4 may need to remain the network protocol between a caching server and authoritative DNS servers to ensure continuity of service. Many factors can affect network performance and availability during this conversion process, and these can impact both IPv4 and IPv6 access. DNS may supply AAAA records before services are fully turned on and reachable, and timeouts may occur. Applications may be unprepared to handle them. Caching resolvers may have to deal with A and AAAA records having different time-to-live values.

It is important to verify that client resolvers are receiving the correct responses and resource records. An ongoing testing phase for IPv6-capable DNS services is a necessary step during any organization's IPv6 deployment process. When this process includes testing IPv6 application software, separate DNS servers and domain names should be used to support simultaneous IPv6 testing and operational IPv4 name resolution.

DNS responses with AAAA records are longer than similar IPv4 responses, because the records are simply larger than comparable A records and interfaces may have more than one IPv6 address in the DNS. Zone file transfer sizes will increase and, potentially, responses returning multiple AAAA resource records (e.g. MX requests) may require fragmentation. Since IPv6 fragments are discouraged, and sometimes are blocked, this could impact availability.

Some authoritative servers ignore queries for an AAAA record and cause a resolver first to wait and timeout and then to fall back to a query for an A record, which may cause a fatal timeout at the application that called the resolver. Even if the resolver and application eventually succeed, the result can be an unacceptable delay for the application's user, especially with interactive applications like web browsing.

Advertising IPv6 addresses in DNS during the initial deployment requires additional care. Availability problems can easily arise if AAAA records are inserted into the DNS zone before IPv6 services are working. The recommendation is that AAAA records for a service should not be added to a DNS zone until the address is assigned to an interface on a host, the address is configured and enabled on the host's interface, and finally the interface is on a link connected to the IPv6 infrastructure.

4. IPv6 Advanced Topics

This chapter provides specific details about the status, requirements, capabilities, and security impacts of more advanced IPv6 topics such as multihoming, multicast, quality of service, mobile IPv6, jumbograms, address selection, DHCPv6, and IPv6 renumbering. As of the writing of this guide, some of these topics have not yet been fully specified or implemented and are not ready to deploy. In these cases, this is noted and interim methods are recommended where appropriate.

4.1 Multihoming

Multihoming means having the ability to utilize more than one connection to the Internet. A host, for example, may have more than one network connection (e.g., 100baseT and WiFi); a connection to a LAN with more than one router to the Internet on it; or a connection to a single router that has more than one Internet connection. Frequently, one refers to multihoming for an entire site, which may be a home, small office, or campus location within a large enterprise.

Multihoming is extremely useful but has a potentially large impact on the global Internet architecture. This section describes the motivations for multihoming, the problems it creates, the requirements for a good solution, potential solutions, and their security implications.

Users have several strong motivations for multihoming. First and foremost, having more than one Internet connection provides greater reliability and resiliency should one link fail or one ISP have a prolonged outage. Other reasons for multihoming include better performance (load balancing or QoS differentiation) and policy enforcement (including security policy).

While multihoming offers obvious advantages, it is complicated by another issue: the use of Internet addresses for two purposes. On the one hand, addresses are used for forwarding packets to the right location. On the other hand, they are used for identifying an endpoint, e.g., a transport protocol (TCP or UDP addresses plus port numbers) or IPsec security association (destination address plus security parameters index). For multihoming to work, either the entire Internet has to know multiple paths to a multihomed site, or the nodes at a site have to be able to use multiple addresses seamlessly and transparently. The former implies enormous growth in the Internet's core forwarding tables. The latter implies that nodes have to cope with TCP connections, UDP responses, IPsec security associations, and other upper layer protocols (ULPs) tied to addresses.

Multihoming solutions also need to satisfy other requirements. They need to avoid causing problems with fragmentation, renumbering, and domain names. They need to scale to the size of the global Internet and not affect performance too greatly. They also need to work with firewalls and ingress filtering.

4.1.1 Differences between IPv4 and IPv6 Multihoming

Any multihoming solution must satisfy two main goals. The first is to make multihoming transparent to upper layer protocols. Otherwise, multihoming is no better than changing service providers and renumbering one's network manually. The second is to avoid causing explosive growth in the global routing and forwarding tables. Dealing with this growth is the number one challenge Internet engineers face today, with IPv4 or IPv6, with or without multihoming.

Compared with IPv6, IPv4 interfaces are normally limited to a single address, the entire supply of IPv4 network prefixes is much more limited, and the addresses themselves are four times shorter, all of which constrains the problem somewhat. IPv4 users with provider independent (PI) addresses can achieve fully transparent multihoming with resiliency and load balancing at the expense of global router table growth.

If NAT is used, resiliency is not transparent, and load balancing within a connection is impossible, so IPv6 multihoming without NAT potentially offers more powerful capabilities.

On the one hand, obtaining PI addresses is the most practical way to achieve IPv6 multihoming, whereas, on the other hand, the router table growth caused by IPv4 multihoming (frequently called CIDR address prefix de-aggregation) is problematical for the Internet, and it is a potentially overwhelming problem with IPv6. Therefore, provisioning IPv6 PI addresses has been a vigorously debated topic. Policies differ among the Regional Internet Registries (RIRs). The ARIN Number Resource Policy Manual³⁰ requires that an organization must be an end site and not a local registry and “qualify for an IPv4 assignment or allocation from ARIN under the IPv4 policy currently in effect, or demonstrate efficient utilization of all direct IPv4 assignments and allocations, each of which must be covered by any current ARIN RSA”.

Qualifying organizations may obtain a /48 PI IPv6 assignment. Until a solution both satisfying the global forwarding problem and providing host transparency is available, IPv6 sites needing multihoming and not qualifying for a PI assignment should attempt to get their primary ISP to accept /48 prefixes from secondary ISPs and thus achieve partial multihoming, although such requests may or may not be honored.

4.1.2 Site Multihoming by IPv6 Intermediation (SHIM6) Specification Overview

Several architectural approaches to IPv6 multihoming have been considered. These are described in detail in [RFC 4177](#):

- + Use the global routing infrastructure, as is done with IPv4.
- + Base the solution on Mobile IPv6.
- + Modify protocols in hosts to accommodate dynamic changes of locators.
- + Design the intelligence, including rewriting addresses, into site exit-routers.
- + Add a network layer protocol element to split addresses into locators and identifiers.

The most forward looking solutions to the multihoming problem introduce the notion of splitting an IP address into an identifier and locator. The idea is that upper layer protocols use the identifier, and core network routing and forwarding use the locator. Multihoming is accomplished by dynamically managing the bindings between the two. To make this work, protocols for establishing and maintaining these relationships must be provided. This requires protocol elements that update locator lists, switch locators in use, and so forth.

More than one proposal has been made along these lines, but the active standards track work on specifying split IPv6 identifiers and locators is called SHIM6 and is being done in the IETF’s SHIM6 Working Group. The name is derived from the way the additional address is specified in a shim header inserted into the packet.

SHIM6 ([RFC 5533](#)) is a network layer, host-based protocol to establish identifier-locator bindings. Its goals are³¹:

- + To preserve established communications in the presence of certain classes of failures, for example, TCP connections and UDP streams

³⁰ ARIN, [Number Resource Policy Manual](#).

³¹ IETF Site Multihoming by IPv6 Intermediation (shim6), [Working Group Charter](#).

- + To have minimal impact on upper layer protocols in general and on transport protocols and applications in particular
- + To address security threats identified in [RFC 4218](#) through the combination of hash-based or cryptographically-generated addresses and additional measures.
- + Not to require extra roundtrip up front to set up shim-specific state
- + To take advantage of multiple locators for load spreading so that different sets of communication to a host (e.g., different connections) may use different locators of the host.

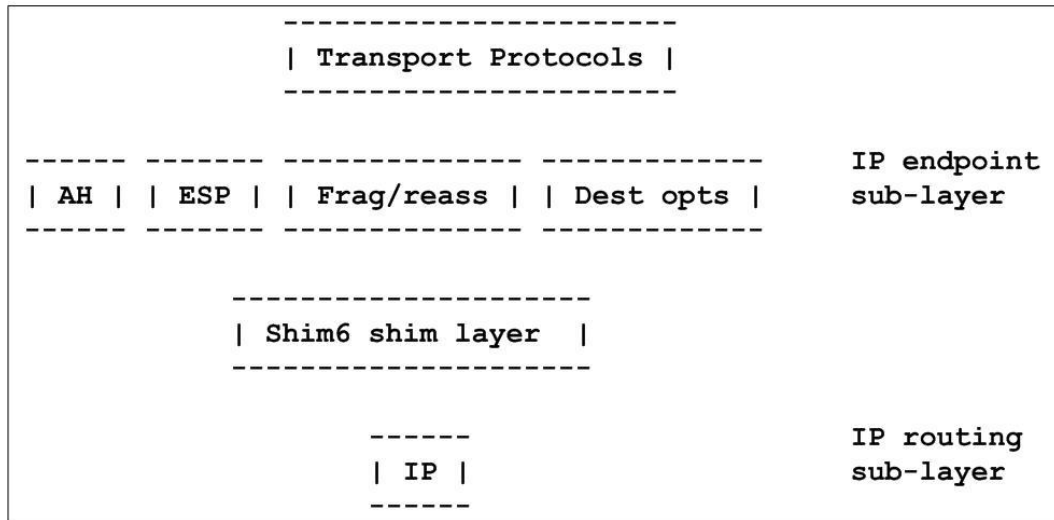
The general idea is to use IPv6 normally but, through SHIM6 signaling, set up alternative locators that can be used when needed. The steps, roughly, are:

- + An AAAA DNS query provides a (possibly incomplete) set of locator addresses.
- + The source chooses a locator address to establish a conversation (choose a different Locator ID if the first attempt is not successful).
- + Once upper layer communication begins, end hosts can signal SHIM6 capabilities and exchange a complete set of locators.
- + The current source and destination locators are used as source and destination Upper Layer IDs.
- + In case of an outage, the source or destination can detect path failure in the forwarding plane and change source or destination locator to any in the locator set.
- + Existing sessions continue uninterrupted using the unchanged Upper Layer ID.

An important design paradigm is that no new name space is needed. The SHIM6 protocol uses four IPv6 extension header messages called I1, I2, R1, and R2 to accomplish all of this. Figure 4-1 shows where the shim header fits into the IPv6 protocol stack.

Although the most promising work on IPv6 multihoming is SHIM6, this is likely to get much more discussion and revision. Some concerns about this approach have been expressed. In particular, it may be difficult to provide adequate traffic engineering with a host-based solution like SHIM6. For large enterprises with complex routed networks, site-based multihoming may be more useful. One example of site-based multihoming that is currently being considered within the IETF is the Locator/ID Separation Protocol (LISP)³². On large servers with many simultaneous connections, the overhead of maintaining SHIM6 state information may impact performance significantly.

³² IETF LISP [Working Group Charter](#).

Figure 4-1. SHIM6 Protocol Stack³¹

4.1.3 Security Ramifications for Multihoming

[RFC 4218](#) presents an overview of security vulnerabilities inherent in multihoming. In summary, one must be concerned about denial of service, re-routing packets to unintended destinations or black holes, and multicast issues. It advises, in general, that connectionless transport protocols like UDP present more security problems than connection-oriented transport protocols. Also, if identifiers and locators are split, security should be tied to the identifiers to lessen the impact of attacks on the identifier-locator binding. Securing other parts of the infrastructure such as the DNS and routing protocols helps minimize the potential attacks on multihoming. Multihoming solutions also need to account for and work with ingress filtering so that spoofed addresses can not be used to attack systems. ([RFC 2827](#) and [RFC 3704](#)).

Networks need to ensure that they do not announce their prefixes in a way that generates asymmetric traffic flows. Traffic following asymmetrical paths might get blocked by strict Reverse Path Forwarding (RPF) checks or stateful packet filters. It may also make it impossible to implement IPsec at site border routers.

The SHIM6 protocol contains several security measures:

- + Hash-based addresses ([RFC 5535](#)) allow one to prove address ownership and to prevent redirection attacks.
- + Reachability probes ([RFC 5534](#)) allow one to identify third party flooding attacks.
- + Two-way communications are required before the responder creates any state. This means that a state-based DOS attack (trying to use up all available memory on the responder) at least reveals an IPv6 address that the attacker was using.
- + Context establishment messages use nonces to prevent replay attacks and to prevent off-path attackers from interfering with the establishment.
- + After context establishment, every SHIM6 control message contains the context tag assigned to the particular context. This implies that an attacker needs to discover a valid context tag before being able to spoof any SHIM6 control message. This also helps protect the SHIM6 protocol from off-path attackers.

Packet filters may need to be aware of SHIM6 and modify their actions accordingly. First, along with upper layer protocol and port numbers, they may need to match on the upper layer identifiers as well as or instead of the IPv6 locator addresses. Second, they may need to understand when established sessions begin using shim headers or change locators.

4.2 IPv6 Multicast

Multicast refers to sending a packet to an IP address designated as a multicast address; one or more hosts specifically interested in the communication then receive a copy of that single packet. This differs from *broadcast*, which delivers packets to all hosts on a subnet, because multicast traffic is only sent to hosts subscribed to the multicast group. Multicasting is often used, for example, to stream audio and video more efficiently. Senders achieve two primary advantages by using multicast. First, the sender only needs to create and send one packet, instead of creating and sending a separate packet to each recipient. Second, the sender does not need to keep track of who the actual recipients are. Multicasting can also be advantageous from a network perspective, because it reduces network bandwidth consumption.

This section describes how multicast works in an IPv6 environment. In IPv6, broadcast has been eliminated and multicast takes on a much larger role. In addition to replacing broadcast, it also works with ICMPv6 neighbor discovery and router discovery on the local link to perform stateless address autoconfiguration and address resolution.

First, consider an example of how IPv6 multicast makes Neighbor Discovery in IPv6 more efficient than using ARP with IPv4. Running IPv4, when a host has an IP address on its own subnet and needs to know the corresponding link layer address, it broadcasts an ARP request containing the IPv4 address. Every host on the subnet gets a copy.

Suppose, running IPv6, an interface wants to find the link layer address for the link local IP address FE80::4DF2:54C8:B8C7:113A. It takes the low-order 24 bits of this address (C7:113A) and appends them to the well-known solicited node multicast prefix, FF02:0:0:0:1:FF00::/104 to form the solicited node multicast address FF02::1:FFC7:113A. Then it sends an ICMPv6 Neighbor Solicitation message to this multicast address. The message gets delivered to the interface at FE80::4DF2:54C8:B8C7:113A because it belongs to the multicast group FF02::1:FFC7:113A. Interfaces using any unicast or anycast address must join the solicited node multicast group corresponding to the above prefix and low-order 24 bits of their address. They use ICMPv6 Multicast Listener Discovery (MLD) to join a multicast group. The result is a more efficient address resolution procedure, in which a smaller number of hosts are queried to determine the address.

The solicited node multicast addresses range from FF02::1:FF00:0 to FF02::1:FFFF:FFFF, so the Neighbor Solicitation traffic is partitioned into 2^{24} solicited node multicast groups. This makes receiving a Neighbor Solicitation intended for a different address highly unlikely.

IPv6 multicast addresses are easy to recognize. They always begin with eight 1 bits: FF. The next eight bits, 02 in this case, specify that this is a well-known multicast address with link local scope. Other examples of well-known multicast addresses with link-local scope are:

FF02::1	All Nodes
FF02::2	All Routers
FF02::1:2	All DHCP Agents

In addition to scoped multicast addresses, other features such as source-specific multicast have been added to IPv6. These different features and their applications are described below, and areas still needing

work are noted (e.g., multicast with SHIM6 and IPsec).

4.2.1 IPv6 Multicast Specifications

In addition to providing an essential part of the IPv6 infrastructure, multicast applications include groupware, multimedia distribution, searching, routing, database replication, grid computing, and real-time information delivery.

With IPv6, multicast addresses have scope ranging from a single interface or link to the global Internet. They can be permanently assigned and well-known, as in the examples above, or they can be used transiently for specific purposes.

[RFC 4291](#), *IP Version 6 Addressing Architecture*, defines a multicast address as: “An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.” Normally, these are the addresses that have previously joined a given multicast group. The Multicast Listener Discovery (MLD) Protocol is the method interfaces use to join and leave multicast groups, and routers keep track of these groups for each interface on which they forward packets. Version 2 (MLDv2) ([RFC 3810](#)) manages multicast group membership with two ICMPv6 message types:

- + Multicast Listener Query (Type = 130)
- + Version 2 Multicast Listener Report (Type = 143).

MLDv2 is backward compatible with MLDv1 ([RFC 2710](#)), so MLDv2 also supports:

- + Version 1 Multicast Listener Report (Type = 131)
- + Version 1 Multicast Listener Done (Type = 132).

All of these messages are sent with a link-local IPv6 source address (or the unspecified source address if necessary), an IPv6 Hop Limit of 1, and an IPv6 Router Alert option ([RFC 2711](#)) in a Hop-by-Hop Options header. (The Router Alert option forces routers to examine MLD messages sent to IPv6 multicast addresses in which the routers themselves previously had no interest.)

How routers actually implement multicast depends on the Layer 2 networking technology. It is trivial on point-to-point links; it may be implemented by using *promiscuous mode* on link layers with a natural broadcast capability; or it may require Layer 2 protocol logic on non-broadcast multi-access link layers.

The biggest change introduced in MLDv2 is Source Specific Multicast. MLDv2 allows an interface to specify, for each multicast address, from which source addresses it does or does not want to receive packets.

[RFC 4604](#) updates MLDv2 and describes source-specific multicast for IPv4 (IGMPv3) and IPv6 (MLDv2) in a single document.

As stated above, IPv6 multicast addresses are easy to recognize. They are exactly the addresses beginning with eight 1 bits or hexadecimal FF:

```

      8      4      4      112 bits
+-----+-----+-----+-----+-----+
|11111111|flgs|scop|                      group ID                      |
+-----+-----+-----+-----+-----+

```

The next eight bits in a multicast address specify the *flags* and *scope*. The final 112 bits in the multicast address are called the *Group ID* and are used to specify the set of nodes that are members of a multicast group. In a previous version, the Group ID was only 32 bits, and some implementations still stick to 32 bits.

In the IPv6 specification, multicast addresses always have a *scope* that limits the set of receiving nodes. The scope is the group of nodes for which the packet is intended relative to its source. Its values are shown in Table 4-1.

Table 4-1. IPv6 Scoped Multicast Values (from [RFC 4291](#))

Value	Scope
1	Interface Local
2	Link Local
4	Admin. Local
5	Site Local
8	Organization Local
E	Global

Some well-known multicast Group IDs are defined for variable scopes. A good example is the “All NTP [Network Time Protocol] Servers” address:

FF02::101	All NTP Servers	Link Local
FF04::101	All NTP Servers	Admin Local
FF05::101	All NTP Servers	Site Local
FF08::101	All NTP Servers	Organization Local
FF0E::101	All NTP Servers	Global

[RFC 2375](#) contains a list of well-known IPv6 multicast addresses categorized by scope, and that list has been extended by newer specifications. The complete and current version is at <http://www.iana.org/assignments/ipv6-multicast-addresses>.

The flags specify, first, whether a multicast address is a well-known, pre-defined address, or whether it is a transient address not permanently defined. Second, the flags specify whether a transient multicast address has an authorized and properly scoped unicast prefix embedded in it, and, if so, whether it also specifies a rendezvous point ([RFC 3956](#)). The unicast prefix eliminates the need for an additional protocol to allocate unique multicast addresses.

A rendezvous point for a multicast group is the root of a tree used with Protocol Independent Multicast—Sparse Mode (PIM-SM) ([RFC 4601](#)), a sophisticated multicast routing protocol. PIM-SM builds a tree of senders and group members. Then, traffic from senders is first routed upstream towards the rendezvous point and then downstream to group members (receivers).

For well-known multicast addresses, the four flag bits are always all zeros. The first flag bit is reserved and must be 0. The remaining three flag bits are called R, P, and T (for rendezvous, prefix, and transient) and are non-zero for transient multicast addresses:

```

      flgs
+--+--+--+
|0|R|P|T|
+--+--+--+

```

The flags are set as follows (see [RFC 3306](#) and [RFC 3956](#)):

0 0 0 0	Well-known, pre-defined multicast address (as in all of the examples above)
0 0 0 1	Transient multicast address without an embedded unicast prefix
0 0 1 1	Transient multicast address with an embedded unicast prefix and no rendezvous point
0 1 1 1	Transient multicast address with an embedded unicast prefix and rendezvous point

This provides a simple way to allocate multicast addresses belonging to or allocated by a given network prefix in the following format, without adding any new multicast address allocation protocol:

```

| 8      | 4  | 4  | 8      | 8      | 64      | 32      |
+-----+---+---+-----+-----+-----+-----+
|11111111|flgs|scop|reserved| p-len  | network prefix | group ID |
+-----+---+---+-----+-----+-----+-----+

```

The eight bits after the flags and scope must be zero. The next eight bits specify a prefix length, up to 64, followed by the prefix, left justified and zero filled.

This example comes from [RFC 3306](#). The address FF38:0030:3FFE:FFFF:0001:0:1234:5678 is:

- Multicast (FF)
- Transient with embedded prefix (3)
- Organization Local scope (8)
- Using a prefix of length 48 (30)
- Specifying the prefix as 3FFE:FFFF:0001::/48
- Using a 32-bit Group ID of 1234:5678

Note that the scope of such multicast addresses must not be greater than the scope of their embedded prefixes.

4.2.2 Differences between IPv4 and IPv6 Multicast

Although multicast addresses are common in both IPv4 and IPv6, several important differences exist:

- + Unlike IPv4, IPv6 does not have broadcast addresses. Instead, IPv6 uses optimizations like the Solicited Node multicast groups and the *all routers* multicast addresses, which make better use of network resources than broadcast.
- + In IPv6, multicast is used with ICMPv6 for infrastructure applications like neighbor discovery and autoconfiguration on local links.
- + IPv6 multicast addresses have new capabilities such as scope and embedded unicast prefixes. In general, IPv6 extensions to multicast have been added to make multicast more useful over internets.
- + Multicasting is managed with ICMPv6 message types collectively called MLD instead of IGMP.

4.2.3 Multicast Security Ramifications

IPv6 routers, packet filters, firewalls, and tunnel endpoints need to enforce multicast scope boundaries. Firewalls should inspect all source IPv6 addresses and filter any packets with a multicast source address. Furthermore, firewalls in Layer 3 mode should never forward link-layer multicast packets.

Attackers may take advantage of well-known multicast addresses to find hidden resources such as routers or particular servers. These addresses need to be blocked at the appropriate places according to local security policy.

Denial of service attacks may use multicast to amplify bandwidth consumption or attempt to exhaust other resources. So-called reflector attacks may send packets with a source address of the target of attack and a multicast destination address, to try to get all multicast receivers to respond to the target. These attacks need to be intercepted and dropped.

IPsec coverage for multicast is incomplete. If a multicast group has more than one sender, the replay protection mechanism does not work. More importantly, IKE is a unicast UDP protocol that only works between two parties, so automated key management for multicast IPsec is lacking. For additional details about using IPsec with IPv6 multicast, see [Section 5.3.3](#).

Attacks on the MLD protocol include denial of service, causing unwanted traffic to be delivered, and downgrading capabilities from MLDv2 to MLDv1. By properly enforcing the unicast scope and hop count rules for MLD, these attacks can be confined to a local link.

Forging Protocol Independent Multicast (PIM) messages can cause unwanted traffic to be sent to replace the role of designated routers ([RFC 5294](#)). IPsec can be used for cryptographic protection of these messages. Cryptographically protecting PIM messages ([RFC 5796](#)) also stops many denial-of-service attacks. Other precautions include limiting the set of neighbors from which Join, Prune, Assert, and Hello messages are accepted. Routers should check that a valid Hello message was received first and that source addresses are legal for the interfaces on which they are received. See also [RFC 4609](#) for a list of suggestions for rate limiting PIM messages, in particular, the inter-domain multi-source discovery protocol (MSDP).

Issues concerning how link-local ICMPv6 multicast traffic used for neighbor discovery and autoconfiguration can be secured are covered in [Section 5.4](#).

4.2.4 Unresolved Aspects of IPv6 Multicast

IPsec and IKE were not designed with multicast security in mind, and three important unresolved aspects of multicast security are related to IPsec and key management for IPsec.

The fundamental IPsec security association architecture and protection protocols, ESP and AH, were designed mainly for unicast, and, although they work with multicast in principle, they are incompletely specified for multicast, and many open issues exist. (For example, what happens to a multicast security association when members join or leave the group?)

Key management for IPsec, provided by IKE, is inherently a two-party protocol. Different protocols for group key management have been proposed, but an agreed-upon and widely implemented standard for IPsec multicast key management has not emerged. Protocol specifications like PIM suggest using IPsec with manual keying, but this solution that does not scale well over time or space and has other limitations. For example, the IPsec replay detection feature is not supposed to be used with manual keying,

The security recommendations for PIM call for using IPsec AH, even for unicast messages. Where IPsec is used with IPv4, AH has generally fallen out of use in favor of ESP, with NULL encryption when authentication-only is desired. Thus the IPsec standard ([RFC 4301](#)) no longer requires AH, and many implementations omit it. See [Section 5.3.6](#) for additional discussion of this topic.

4.3 IPv6 Quality of Service (QoS)

The TCP/IP Network Layer—IPv4 and IPv6—was intentionally designed without any of the features normally associated with QoS such as admission controls; resource guarantees; and in-order, lossless delivery. QoS on TCP/IP networks or the Internet is a somewhat imprecise concept, which may have different meanings varying from “anything except undifferentiated best effort” to specific service level contracts between a provider and user. It may mean:

- + Providing a given user with certain levels of overall availability, throughput, low latency, maximum packet loss, or even security
- + Treating different types of traffic differently, according to content: real-time audio or video requires high throughput and consistently low latency but can tolerate small losses, whereas file transfer can tolerate delay but no losses whatsoever.

Many aspects of engineering QoS depend on technologies running at multiple layers, such as MPLS (MultiProtocol Label Switching) and ATM (Asynchronous Transfer Mode). IETF work on QoS for TCP/IP began with Integrated Services (intserv), which was designed to provide QoS guarantees. Intserv has been replaced with Differentiated Services (diffserv), which simply recognizes that different types of traffic have different QoS requirements and need to be marked accordingly. The signaling protocol (i.e., the protocol used for QoS setup and specification) to establish QoS requests is RSVP—Resource Reservation Protocol. IPv6 end-to-end addressing allows services that are difficult to deploy with NAT as well as end-to-end use of diffserv and RSVP. Many of these services may have real-time and multimedia content, so QoS is likely to become a more important topic with the widespread use of IPv6.

The notion of improved QoS has always been linked with IPv6. In fact, IPv6 was designed to support certain QoS improvements, but not all of these have been completely specified or implemented.

4.3.1 IPv6 QoS Specifications

Several aspects of IPv6 implicitly or explicitly support QoS. These include:

- + A streamlined header with fewer fields, no checksum processing, sufficient address space to make address translation unnecessary, and a simple test for whether routers need to examine anything past the fixed length header promote efficient packet forwarding.

- + Requiring a larger minimum MTU and PMTU discovery also increase efficiency.
- + Eliminating in-route fragmentation removes one of the greatest sources of performance degradation in IPv4.
- + Eliminating broadcast and building in better support for multicast and anycast make better use of network resources.
- + A new Flow Label field and larger Traffic Class field in the main IPv6 header allow more efficient and finer grained differentiation of various types of traffic.

The IPv6 Traffic Class field replaces the IPv4 Type of Service field. The original intent of the IPv4 Type of Service field has been replaced by diffserv ([RFC 2474](#)), which provides a use for the field. Although the RFCs are vague on this point, this is the way the IPv6 Traffic Class is usually used, and the functionality in IPv6 is equivalent to that in IPv4. Because of the way diffserv works, this field may be rewritten in transit. For example, [RFC 2474](#) describes how packet marking is performed by traffic conditioners at network boundaries, including the edges of the network (first-hop router or source host) and administrative boundaries, and [RFC 3168](#) sets aside two bits in this field called Explicit Congestion Notification for routers to indicate network congestion to end hosts.

When IPv6 QoS is mentioned, most frequently the last of these aspects, especially the Flow Label, is cited. Rudimentary use of the Flow Label is defined in [RFC 1809](#) and [RFC 3697](#). A Flow label value is always associated with a source and destination address pair with the same Hop-by-Hop options and Routing Header. (A zero value means that the field is not being used.) This has the advantage of specifying flows completely in the main header. It is not necessary to examine extension headers, upper layer protocols, and port numbers to identify the packet. [RFC 3697](#) recommends that each new transport connection and application data stream be given a new value, and it requires that applications be able to specify this value. It stipulates that Flow Labels be delivered intact, and it gives rules for timeouts and reuse. Finally, it prohibits using specific bits or mathematical interpretations of the value—it is just a 20-bit label.

Many IPv6 implementations do choose different Flow Label values for each TCP connection, for example, but few if any make additional use of the field. Thus, realizing the potential for providing better QoS offered by the IPv6 Flow Label lies in finding improved ways to use this feature in the future.

4.3.2 Differences between IPv4 and IPv6 QoS

The specific differences in QoS capabilities between IPv4 and IPv6 are covered point-by-point in the preceding section. The overall design of IPv6 is better thought out with respect to QoS; several specific improvements in IPv6 allow for more efficient network usage, and room has been left for additional QoS capabilities when these are defined.

4.3.3 Security Ramifications

One aspect is securing the QoS mechanisms themselves, to prevent theft of service, traffic analysis, or other attacks. For example, the Type of Service and Flow Label in the IPv6 header are not protected, even by AH. This is because the Type of Service can be altered while a packet is in transit. Although [RFC 3697](#) specifies a nonalterable Flow Label field, when AH was originally designed, the Flow Label was also thought to be capable of alteration; the updated version of AH maintained that view for backwards compatibility. An application able to forge these fields may be able obtain preferred service fraudulently. If this is a major concern, IPsec can be run in tunnel mode, the QoS parameters can be copied from the inner header to the outer header, and the protected inner header can be compared with the

outer header upon delivery.

Because the QoS fields in the outer header are not protected, firewalls cannot blindly trust the Type of Service and Flow Label alone for access control decisions. Also, traffic analysis may become simpler, because someone monitoring traffic flows can take advantage of the same efficiencies as legitimate routers forwarding traffic.

A different aspect is making sure that security does not impede the required QoS. QoS may need to be applied to packets secured with IPsec, in which case information about the upper layer protocols may not be accessible, but in this case, the methods in [RFC 2207](#) can be used to differentiate IPsec-protected traffic. When planning to provide QoS, one must take into account that cryptographic protection, packet filtering, and examination by intrusion prevention systems all add some delay.

A third consideration is that securing QoS signaling protocols such as RSVP presents some difficulties, because these protocols often do not run strictly end to end but presume that intermediate points examine (or, worse, modify) the contents. For an overview of RSVP security, see [RFC 4230](#).

Since the flow label field is currently for the most part unused, it could be used as a covert channel. Therefore, unless it is in use, its contents should always be zero.

4.3.4 Unresolved Aspects of IPv6 QoS

General aspects of using Flow Labels have been specified, but details needed to take advantage of them are still missing. Because of the ways QoS depends on lower-layer protocols, it is unclear where and how progress on this front will be made.

4.4 Mobile IPv6 (MIPv6)

IP-layer mobility has long been considered a useful and important feature, but today, with the exploding growth of laptop computers, PDAs, and mobile phones connecting to the Internet from more than one location, and, tomorrow, with the Internet in every motor vehicle and yet unimagined portable devices, it is becoming an essential ingredient in advanced services.

The work on Mobile IPv4 (MIPv4) envisioned many of the features and functions of Mobile IPv6 (MIPv6), but MIPv4 has never been practical on a large scale. The capabilities built into IPv6 make widespread use of MIPv6 practical: plentiful end-to-end addresses, security, optimized routing, increased reliability, and more. Thus, it is expected that use of mobility will increase with IPv6.

The central issue with IP mobility is the same as with multihoming: disconnecting from one network and reconnecting to another is easy; changing IP addresses while keeping one's TCP connections, IPsec security associations, and streaming protocols running takes more work. Not surprisingly, MIPv4 and MIPv6 have common elements, but mobility is inherently a difficult problem, and MIPv6 involves quite a bit of sophisticated protocol design.

4.4.1 MIPv6 Specification Overview

MIPv6 allows an IPv6 interface to disconnect and reconnect physically in an internet topology while logically retaining its "home" IPv6 address, so MIPv6 nodes can change their point of attachment to the network while maintaining seamless connectivity. The primary document describing MIPv6 is [RFC](#)

[3775](#), *Mobility Support in IPv6*,³³ but many other published RFCs and work still in progress support MIPv6.

As one might expect, MIPv6 has its own terminology, a certain amount of which is essential for understanding how it works. The most important terms are:

- + **Mobile Node (MN).** A node using MIPv6 to change its point of network attachment
- + **Home Address (HoA).** The permanent, routable unicast address of the MN
- + **Home Link.** The link on which the MN's HoA is defined
- + **Foreign Link.** Any link except the home link
- + **Care-Of Address (CoA).** A routable unicast address used by the MN on a foreign link
- + **Correspondent Node (CN).** A peer with which the MN is communicating
- + **Home Agent (HA).** A router on the MN's Home Link with which the MN registers its CoA and which forwards traffic to and from the MN at its CoA
- + **Binding.** The association of a HoA and CoA for a given amount of time
- + **Binding Cache (on HA or CN).** A table of other nodes' bindings and their lifetimes
- + **Binding Update List (on MN).** A MN's table of HA and CN bindings
- + **Route optimization.** Direct communications between a MN and CN without involving a HA

As with SHIM6, MIPv6 solves the problems created by using IP addresses for both identity and location. A MN's identity is its HoA, and its location is its CoA. The goal is to establish and use bindings between these securely and efficiently, and the first step is for a MN to set up IPsec-secured communications with a HA. This may be done before leaving the home link. Figure 4-2 shows the main components of MIPv6.

³³ [RFC 3775](#) is currently being revised to include items like changes to IKE and IPsec, new work on bootstrapping, and an update to the IPv6 addressing architecture, in the current version of the IETF Internet Draft, [Mobility Support in IPv6](#), Work in Progress.

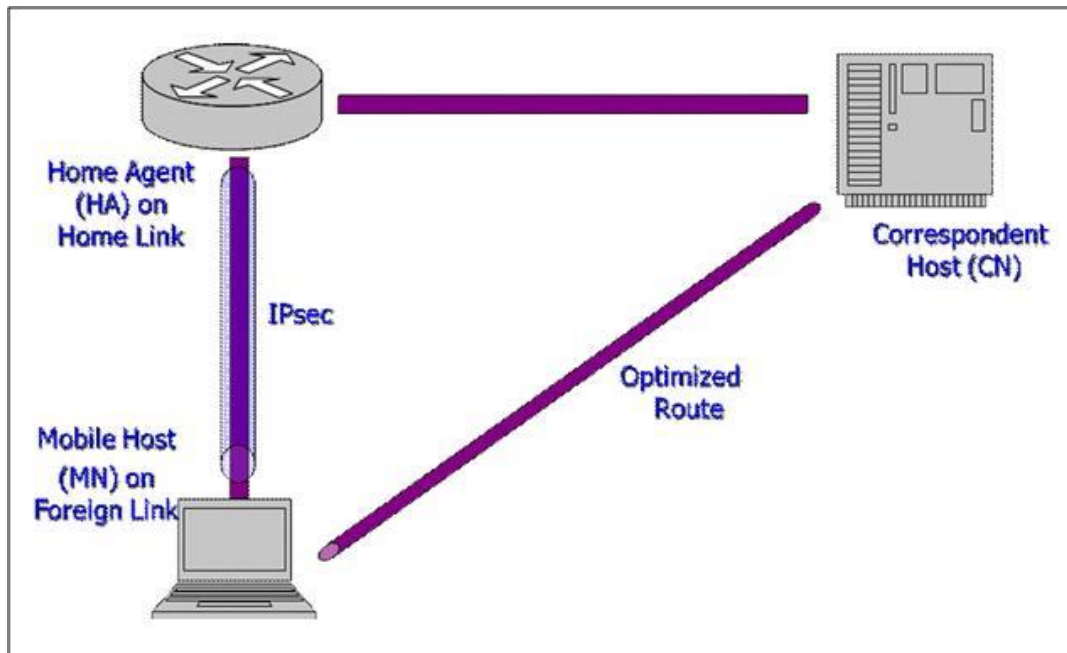


Figure 4-2. The Main MIPv6 Components

To make this work, several functions must be provided, and these require underlying protocol elements. The four new protocol elements added for MIPv6 are:

- + A new IPv6 Extension Header, the Mobility Header (MH), with eight different message types. The first four are used to set up bindings and the last four to run a security protocol called return routability, which is described in [Section 4.4.3.3](#), below:
 - Binding Update
 - Binding Acknowledgement
 - Binding Refresh Request
 - Binding Error
 - Home Test Init
 - Home Test
 - Care-of Test Init
 - Care-of Test
- + Four new ICMPv6 message types:
 - Home Agent Address Discovery Request
 - Home Agent Address Discovery Reply
 - Mobile Prefix Solicitation

- Mobile Prefix Advertisement
- + Two new types of Destination Options Extension Header called the Home Address Option and Alternate Care-of-Address Option
- + A new Routing Header, Type 2.

To get an idea of how these are used, consider the following examples:

A. A MN announces a new CoA:

- A Binding Update (BU) and Binding Update Acknowledgement (BUA) are exchanged between the MN and its HA and between the MN and each of its CNs.
- The BU uses a Binding Update (Type 5) IPv6 Mobility Header and the Home Address Option. The BUA uses a Binding Update Acknowledgment (Type 6) IPv6 Mobility Header and a Type 2 Routing Header.

B. A MN finds a HA:

- The MN initiates the Dynamic Home Agent Address Discovery (DHAAD) protocol.
- To do this, the MN sends an ICMPv6 Home Agent Address Discovery Request to the Mobile IPv6 Home-Agent's anycast address (with which the MN was previously configured) for its home subnet prefix. A HA returns an ICMPv6 Home Agent Address Discovery Reply.

C. A MN learns about home link renumbering:

- A MN receives an unsolicited ICMPv6 Mobile Prefix Advertisement (MPA) indicating that renumbering is occurring.

D. A MN uses route-optimized communications between it and a CN:

- The MN sends normal traffic with a Home Address Option.
- The CN sends normal traffic with an IPv6 Type 2 Routing Header.

E. A MN receives a Binding Refresh request from a CN:

- The CN sends a Binding Refresh Request (Type 0) Mobility Header to the MN's HoA.
- The MN checks that the CN is in its Binding Update List and starts Return Routability and a new BU.

F. A CN sends a Binding Error to a MN:

- A CN receives an unrecognized HoA and sends the MN a Binding Error with a Binding Error (Type 7) MH, error status 1 for "unknown binding," and the HoA it received in error.
- The MN checks that the CN is in its Binding Update List; if it has indication that communications with the CN are working, it ignores the message; otherwise it deletes the binding and sends subsequent communications with the CN through the HA or alternatively starts Return Routability and a new BU.

G. A MN includes an Alternate CoA in a BU:

- The BU contains a Home Test (Type 3) MH option to indicate a CoA different from the source address (because of network topology or security, for example).
- The CN uses the Alternate CoA instead of the original CoA.

4.4.2 Differences from IPv4 Standards

MIPv4 is standardized in [RFC 3344](#). MIPv4 and MIPv6 share much of the same motivation and have somewhat similar designs, but MIPv6 provides enhanced security, streamlined administrative protocols, and greater efficiency. This is not an accident: the reason MIPv6 has so many advantages over MIPv4 is that it uses the new features and capabilities found in IPv6 but not in IPv4. IPv6 has autoconfiguration, globally unique addressing (without NAT), flexible extension headers, and mandatory IPsec. The much larger IPv6 address space makes MIPv6 easier to deploy. Some of the biggest differences are:

- + Route optimization is standard with MIPv6, and it uses a new approach to security called return routability.
- + MIPv6 does not use the MIPv4 last-hop foreign agent. The tunnel endpoint is built directly into the MN, which also allows end-to-end security.
- + MIPv6 does not have any new, special-purpose authentication, authorization and accounting (AAA) support. MIPv6 uses standard link layer and IP network access methods for AAA along with IPsec. Using standard, already understood methods is always preferable.
- + MIPv6 uses two-way tunneling, which works better with ingress filtering.
- + MIPv6 has many security improvements.

MIPv4 and MIPv6 use different protocols. MIPv4 uses ICMP(v4) Router Discovery, Port 434 (UDP or TCP), and “home-grown” security, whereas Mobile IPv6 uses ICMPv6, IPv6 Routing and Mobility Extension Headers, Destination Options, and IPsec.

4.4.3 Security Ramifications

For MIPv6, security has always been a primary design concern. A thorough approach to security requires looking at all of the potential vulnerabilities and choosing appropriate measures to deal with them. The designers of MIPv6 actually used a security threat analysis as the basis for the design. MIPv6 security starts with the base specification, *Using IPsec to Protect Mobile IPv6 Signaling between Mobile Nodes and Home Agents* ([RFC 3776](#)), and extends into several other RFCs, primarily:

- + [RFC 4225](#), *Mobile IP version 6 Route Optimization Security Design Background*
- + [RFC 4285](#), *Authentication Protocol for Mobile IPv6*
- + [RFC 4487](#), *Mobile IPv6 and Firewalls: Problem Statement*
- + [RFC 4449](#), *Securing Mobile IPv6 Route Optimization Using a Static Shared Key*
- + [RFC 4877](#), *Mobile IPv6 Operation with IKEv2 and the revised IPsec Architecture*
- + [RFC 4882](#), *IP Address Location Privacy and Mobile IPv6: Problem Statement.*

The goals set by those specifying MIPv6 security have been to address the most serious security vulnerabilities first, to use existing security methods where they fit and can be deployed easily, to avoid introducing new vulnerabilities, and, as a last resort, to design new security methods where necessary.

Attacks clearly exist if someone can forge or modify any of the main MIPv6 messages:

- + BU between MN and HA
- + BU between MN and CN
- + IPv6 Routing and Mobility Headers
- + IPv6 Home Address and Alternate CoA options

The most serious new vulnerabilities introduced with MIPv6 involve BUs. Many of these can lead to Denial of Service (DOS) of one type or another. It is possible to starve the MN or to flood another host. Beyond DOS, attacks on these protocol messages may attempt connection hijacking, eavesdropping, or other variations of man-in-the-middle or impersonation.

A forged BU between a MN and its HA may be sent by another legitimate MN or any other party. Such an attack can be prevented if the MN is forced to show “ownership” of its HoA, and this can usually be arranged through the on-going relationship a MN has with its HA.

On the other hand, a bogus BU between a MN and CN presents a more challenging problem, because a prior relationship between a MN and CN is less likely to exist, and in any case, protocol designers cannot count on their having one. This attack can be used to redirect traffic between *any* pair of hosts. For example, suppose Alice is communicating with Bob. Eve sends Bob a BU that Alice’s new CoA is Eve’s address. Alice does not have to be mobile to make this work—Bob does not know. It may be possible to play this attack ahead of time, so that it takes effect when Alice and Bob communicate later. Conversely, this attack can be turned around into a DOS attack against Eve.

BUs, however, are not the only vulnerability in MIPv6. Attacks on prefix propagation and HA discovery are possible. Spoofing an ICMPv6 Mobile Prefix Solicitation (MPS) or Mobile Prefix Advertisement (MPA) can break HA-MN connectivity, and merely eavesdropping on these messages can reveal addressing and topology information about the Home Link.

The ICMPv6 Home Agent Discovery is sent to the HA anycast address on the Home Link. Both this and the ICMPv6 Home Agent Reply are unprotected.

Other attacks include many denial-of-service opportunities, for example:

- + Inducing extra BUs with bogus CNs. Although no satisfactory defense exists, route optimization is optional, and the tradeoff is to risk suboptimal routing. A MN can be selective about route optimization.
- + Preventing a legitimate BU from completing while sending a bogus BU to a CN (where the attacker is on the same link as the victim)
- + Reflection attacks, whereby the victim’s address is forged as the source, so that the victim is flooded with replies
- + Replaying old route optimization BUs, especially if sequence numbers are unreliable because of crashes or rollover

- + Bypassing firewall egress filtering with a forged Home Address Option

The next three sections describe how security is provided between a MN and, first, its HA, and, second, a CN. In summary:

- + Mobile IPv6 security is based on a security goal and a threat analysis.
- + Mobile IPv6 uses IPsec where clearly practical and has updated its specifications to use IKEv2 and [RFC 4301](#).
- + Return routability was added as a practical method for securing route optimizations between a MN and CN.

Unfortunately, many implementations of MIPv6 don't support using IPsec between the MN and the HA, so they are vulnerable to man-in-the-middle attacks. IKEv2 is also not widely supported by these implementations.

4.4.3.1 Securing MN to HA Binding Updates

The top priority for MIPv6 security is stopping a forged BU. Given the goal of using existing security systems wherever practical, IPsec was the logical choice. A working relationship between a MN and its HA naturally exists, and IPsec has always been a mandatory part of IPv6. The latest version of IPsec ([RFC 4301](#)) contains several improvements for securing communications between a MN and its HA. In addition to more efficient cryptographic transformations and the simplifications in IKEv2, selectors for ICMPv6 message types and the Mobility Header are now included, and the Peer Authorization Database (PAD) can also be used on the HA. The HA uses the PAD to specify how to authenticate the MN and tie the MN's identity to its HoA to prevent attacks that impersonate a MN. IPsec ESP in transport mode is used for:

- + BU: MN → HA
- + BUA: HA → MN

The MN's security associations must use its HoA in either the source address, the Home Address Destination Option, or a Type 2 Routing Header. The integrity transform (ESP-NULL) is required, confidentiality is optional, and replay detection is recommended if dynamic keying (IKEv1 or IKEv2) is used. The following protocol elements are used:

BU Message:

- + IPv6 Header: Source = CoA; Destination = HA
- + Home Address Option: Address = HoA
- + ESP header: transport mode, authentication
- + Mobility header: Alternate CoA Option = CoA

BU Acknowledgement:

- + IPv6 Header: Source = HA; Destination = CoA
- + Type 2 Routing Header: Address = HA

- + ESP header: transport mode, authentication
- + Mobility header: BU Acknowledgement Option

To use IPsec for BUs and BUAs between a MN and its HA, both the MN and HA must have appropriate entries in their IPsec Security Policy Database (SPD) and Security Association Database (SAD). The following examples show how to do this with Transport Mode. [RFC 4877](#) contains examples using Tunnel Mode.

MN SPD:

- + SPD in: Use SA1 for: Source = HA; Destination = HoA; Protocol = Mobility Header
- + SPD out: Use SA2 for: Source = HoA; Destination = HA; Protocol = Mobility Header

MN SAD:

- + SA1 (IN, SPI, ESP, TRANSPORT): Source = HA; Destination = HoA; Protocol = Mobility Header
- + SA2 (OUT, SPI, ESP, TRANSPORT): Source = HA; Destination = HoA; Protocol = Mobility Header

HA SPD:

- + SPD in: Use SA1 for: Source = HoA; Destination = HA; Protocol = Mobility Header
- + SPD out: Use SA2 for: Source = HA; Destination = HoA; Protocol = Mobility Header

HA SAD:

- + SA1 (IN, SPI, ESP, TRANSPORT): Source = HoA; Destination = HA; Protocol = Mobility Header
- + SA2 (OUT, SPI, ESP, TRANSPORT): Source = HoA; Destination = HA; Protocol = Mobility Header

Setting up security associations (SAs) with IKEv1 or IKEv2 is optional, but necessary for enabling replay detection. It may be done with public keys or pre-shared secrets. With IKEv1 and pre-shared secrets, aggressive mode must be used in Phase 1. One cannot use the source address of the MN, which is the CoA, to select the pre-shared secret (there is no identity hiding in aggressive mode), and, similarly, one cannot use ID_IPV6_ADDR in Phase 1, so the recommendation is to use a Fully Qualified Domain Name (FQDN) in Phase 1 and the HoA in Phase 2. The HA must verify the relationship between these. Thus, the HoA cannot be dynamically assigned. How the HA does this is not specified, but it could be done with DNSSEC, X.509, or Cryptographically Generated Addresses. The general problem of configuring MIPv6 is defined in [RFC 4640](#), *Problem Statement for Bootstrapping Mobile IPv6 (MIPv6)*. Figure 4-3 illustrates the IKEv1 identifiers used between a MN and its HA.

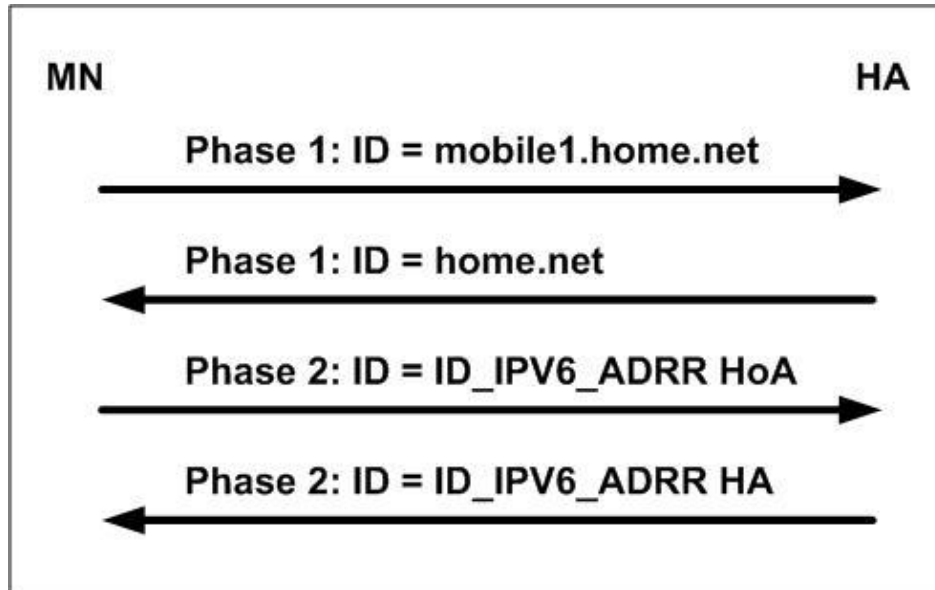


Figure 4-3. IKEv1 Identifiers used between a MN and its HA

[RFC 4877](#) specifies the entire IKEv2 exchange. Figure 4-4 shows the IKEv2 Identifiers used between a MN and its HA.

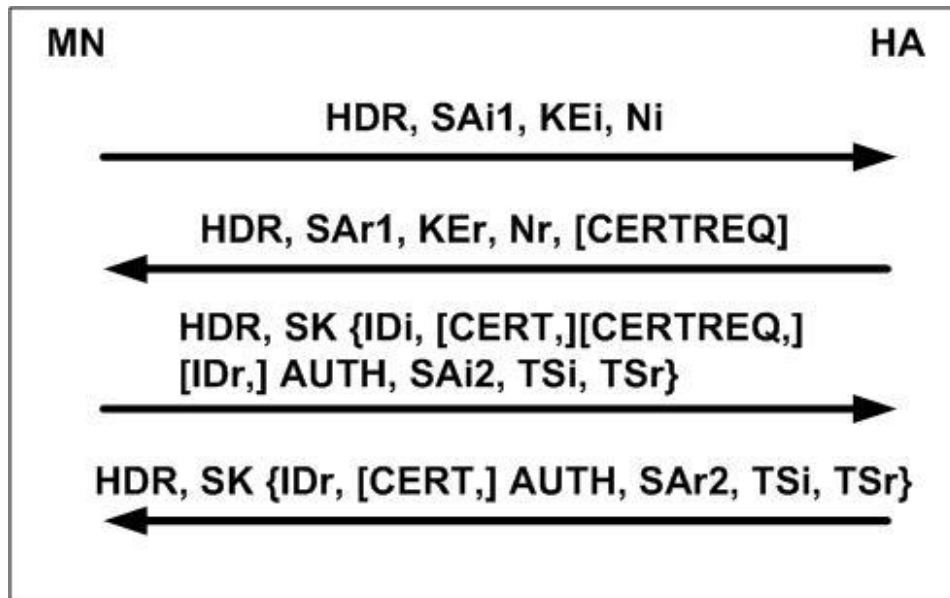


Figure 4-4. IKEv2 identifiers used between a MN and its HA

The MN inserts its identity (e.g., its HoA or FQDN) in the *IDi* payload in the third message, which is encrypted and authenticated. The HoA and traffic selectors for protecting BUs and BUAs are included in the *TSi* (traffic selector—initiator). The MN or HA can then send CREATE_CHILD_SA exchanges to protect other traffic. Again, the MN uses its HoA in the *TSi*.

4.4.3.2 Securing Other MN to HA Traffic

Once an IPsec SA is established between a MN and its HA for BUs and BUAs, it can be used for other protocol elements:

- + ICMPv6 between the HA and MN for MPS and MPA prefix discovery and for DHAAD
- + The return routability messages Home Test Init and Home Test (see [Section 4.4.3.3](#))
- + User traffic (everything else)

At this point, the security associations at the MN include four sets of ESP SPD and SAD entries to and from the HA. All must use the ESP data authentication (integrity) service. Confidentiality may be used if needed. The four sets of entries are:

- + Transport mode for Mobility Headers for BU and BUA with the HA
- + Transport mode for ICMPv6 for home network prefix discovery
- + Tunnel mode for Mobility Headers for return routability messages to and from a CN
- + Optionally, tunnel mode for all other traffic

4.4.3.3 Securing MN to CN Communications

A MN needs to exchange a BU and BUA with a CN to establish route optimization. The major threats are that a CN gets a forged BU, or a CN processes a forged Home Address Option. The original drafts recommended using IPsec, but it was decided that sufficient infrastructure for IPsec authentication did not exist to make this widely deployable, and a new security protocol called *return routability* was invented to authenticate and share a key (called *Kbm*) between a MN and CN. The idea was to make sure the MN can receive messages *both* directly from the CN over its optimized route *and* indirectly over its IPsec connection with its HA. Return routability works as follows: the MN sends the CN Care-of Test Init and Home Test Init over these two paths, respectively. These paths are illustrated in Figure 4-5.

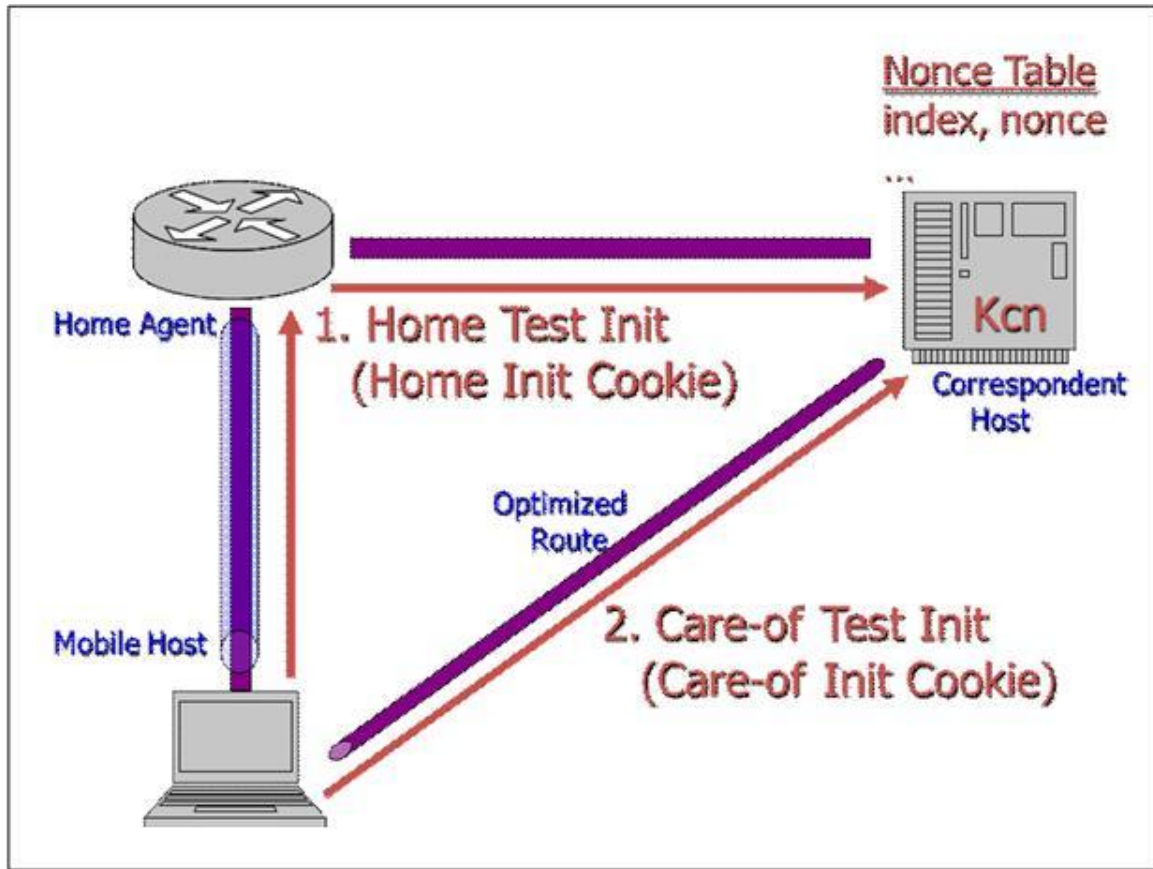


Figure 4-5. Return Routability—Init Messages

Note that route optimization is optional for both parties. A MN always has a tradeoff between optimal packet forwarding and location privacy, and a CN may ignore BUs (when it suspects an attack, for example).

The CN can now compute and return Keygen Tokens for each path:

- + Home Keygen Token = First(64, HMAC_SHA1(*Kcn*, (Home Address, HoA Nonce, 0)))
- + Care-of Keygen Token = First(64, HMAC_SHA1(*Kcn*, (Care-of Address, CoA Nonce, 1)))

The CN sends these back and forgets them so as to avoid a denial-of-service attack. Later, when addresses and nonce indices are returned, it can re-compute these and:

- + $K_{bm} = \text{SHA1}(\text{Home Keygen Token}, \text{Care-of Keygen Token})$

The CN does not allocate Binding Cache storage until authentication completes. Figure 4-6 illustrates the Keygen replies.

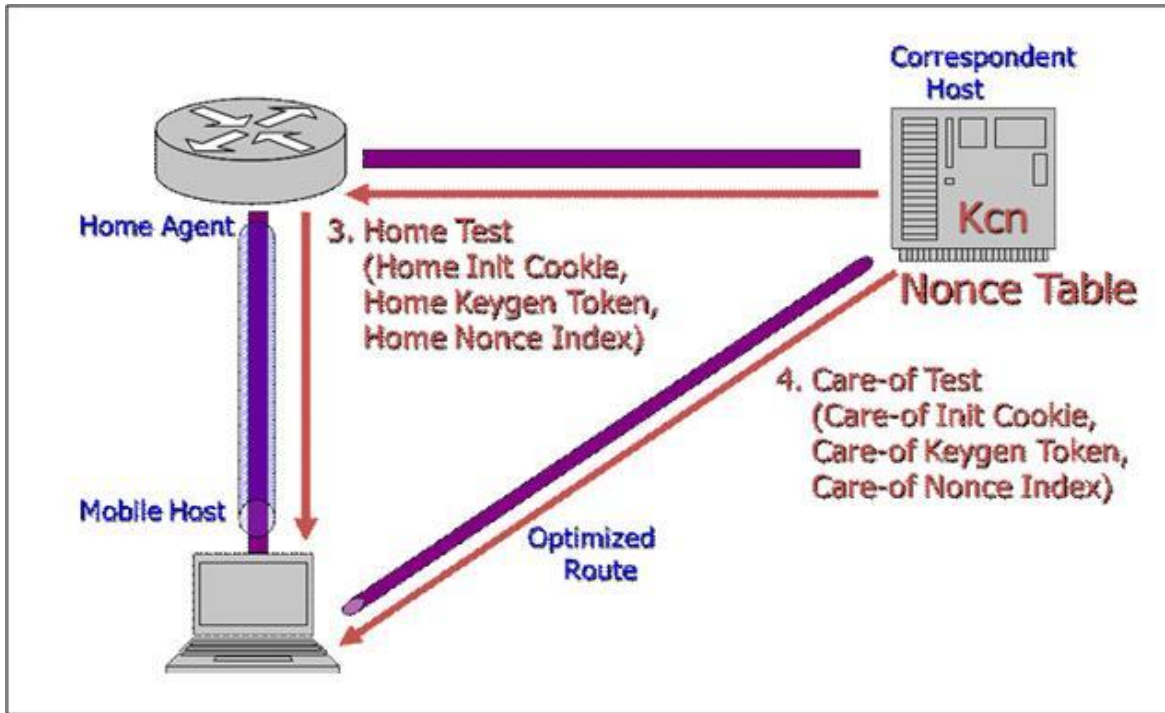


Figure 4-6. Return Routability—Keygen Replies

The MN now computes the following and sends the BU:

- + $K_{bm} = \text{SHA1}(\text{Home Keygen Token}, \text{Care-of Keygen Token})$
- + Binding Update Message Authentication Code (BU MAC) = $\text{First}(96, \text{HMAC-SHA1}(K_{bm}, (\text{Care-of Address} \mid \text{CN address} \mid \text{BU}^*)))$
- + BU = (Home Address Option, BU MAC, sequence number, Home Address Nonce Index, Care-of Address Nonce Index).

Figure 4-7 illustrates the BU and BUA path.

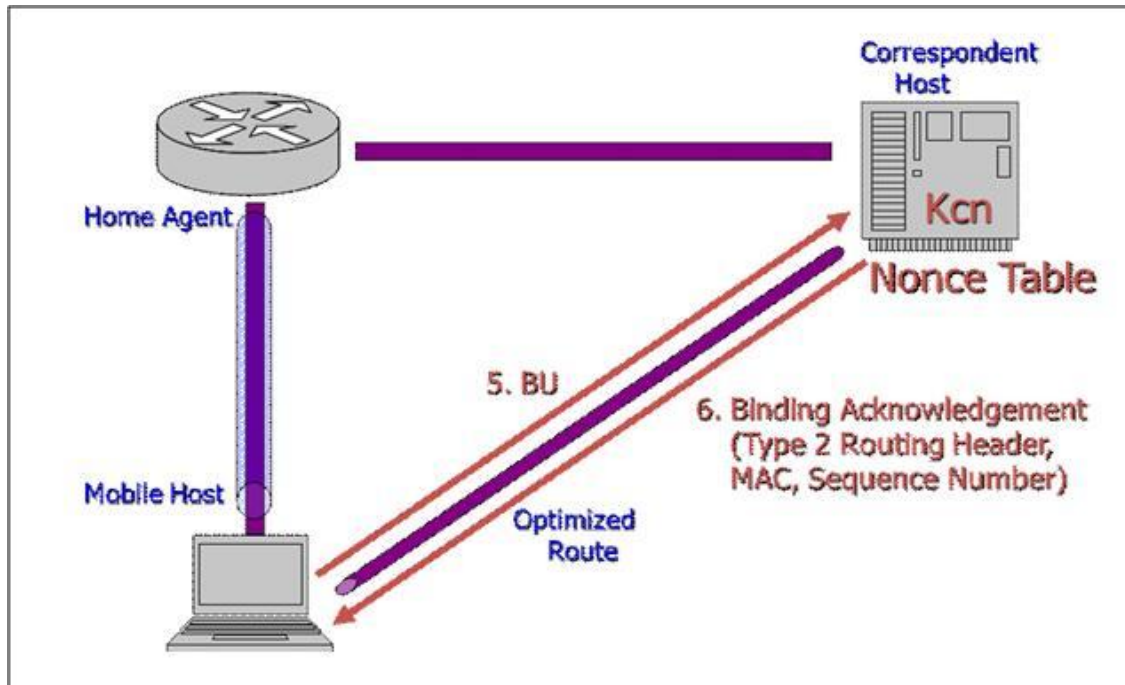


Figure 4-7. Reverse Routability—BU and BUA Protected with K_{bm}

Now, over the optimized route, the MN sends normal traffic with the Home Address Option, and the CN sends normal traffic with a Type 2 Routing Header.

To prevent various abuses, restrictions exist on using the Home Address Option and Type 2 Routing Header. For the former, the rules are:

- + Only one Home Address Option is allowed per packet, it must not be altered en route, it must contain a routable, unicast address, and it must not cause changes in the routing or binding cache.
- + A BU at a CN must be authenticated with a K_{bn} established with return routability.
- + A BU at a HA must be authenticated with transport mode ESP.
- + All other cases must correspond to an entry in the binding cache.

The rules for Type 2 Routing Headers are:

- + Only one type 2 routing header is allowed per packet, and it may only have one segment remaining.
- + The HoA in the Type 2 Routing Header cannot have smaller scope than the CoA in the destination, it must be routable and unicast, and it must be the correct one for the MN.

In summary, an attacker has to intercept two messages sent along different paths to get K_{bm} . Perhaps the greatest danger of this is on the MN's local link, but here IPsec with encryption protects the Home Keygen Token in the Home Test (message 3).

[RFC 4449](#) describes an efficient alternative to return routability, whereby the MN and CN have pre-shared a K_{cn} and nonces, and both can compute K_{bm} directly. This may be useful, for example, when the MN accesses servers at its home site. The CN needs to trust the MN not to use its pre-shared data to

launch DOS attacks. Also, the IETF is working on a draft specification, *Using IPsec between Mobile and Correspondent IPv6 Nodes*, for using IPsec where it is suitable or desirable to secure BUs and other communications between a MN and CN.

4.4.3.4 Other Security Considerations for MIPv6

This section briefly covers several topics that do not fit into the above sections: an alternative to IPsec for MN-HA security proposed by the Third Generation Partnership Project (3GPP2), issues with making mobility and firewalls work together, security protection for the HA, and a discussion of location privacy.³⁴

- + Some designers of 3GPP2 networks consider including IPsec in handsets too difficult. [RFC 4285](#), *Protocol for Mobile IPv6 Authentication*, describes a shared-key, lightweight alternative to IPsec for securing communications between a MN and its HA designed specifically for 3GPP2 networks. The IETF has considered this document informational, not standards track, and it has not been recommended for use in other environments.

However, additional questions have been raised about the suitability of IPsec for securing MIPv6, and a standards-track alternative has been proposed³⁵. The reasons given for supporting the standardization of alternatives to IPsec are:

1. *Software complexity*: Because IPsec is implemented in the operating system and no convenient application programming interface exists, it is difficult for a third party to implement MIPv6 on a system supporting IPsec and IKEv2.
2. *NAT traversal*: The need to run MIPv6 on dual stack IPv4-IPv6 systems ([RFC 5555](#)) across an IPv4 network address translation (NAT) component requires UDP encapsulation and makes running IKEv2 and IPsec more complicated.
3. *Dynamic HoA assignment*: Unless this is built into the IKEv2 processing, securely configuring a dynamic address leads to a chicken-and-egg problem. Using this capability within IKEv2 requires communicating the home prefix to IKEv2.
4. *Scalability*: The number of security associations a HA must support may exceed its capacity.
5. *Availability*: Some IPv6 platforms may not have IPsec and IKEv2.
6. *Communications overhead*: The total overhead for IKEv2 and ESP exceeds that of other security solutions.

While these opinions, taken in total, may have some merit, they are controversial. Nevertheless, it is likely that at least one alternative to IPsec will proceed along the IETF standards track. It is unclear what all of the details will look like and what the time frame for completing this work will be.

- + Many IPv6 firewalls are not compatible with MIPv6. Since many MIPv6 messages are encapsulated, most firewalls and Intrusion Prevention Systems (IPSs) cannot parse these messages accurately. Current firewall policy configuration rules don't allow a security administrator to specify both the external and internal (tunneled) packet contents before allowing or blocking the traffic. More

³⁴ For a discussion of other security issues including attacks by a legitimate but misbehaving MN that creates routing loops or tries to bypass ingress filtering, see the expired IETF Internet Draft [Mobile IPv6 Residual Threats](#).

³⁵ See the current version of the IETF Internet Draft, [Transport Layer Security-based Mobile IPv6 Security Framework for Mobile Node to Home Agent Communication](#), Work in Progress.

sophisticated firewall policies (AND/OR/NOT logic rules in place of the current first-match approach) are needed to accomplish these goals.

[RFC 4487](#), *Mobile IPv6 and Firewalls: Problem Statement*, considers four cases:

- *The MN is in a network protected by firewalls:* The MN needs to be able to get IPsec ESP packets through the firewall for BUs and Home Test Init messages. The firewall also has to understand these protocols and allow appropriate responses (BUA and Home Test). The Care-of Test Init and reply also must be allowed through. If the MN moves from a network protected by one firewall to a network protected by another, the firewalls need to retain state jointly.
- *The CN is in a network protected by firewalls:* The firewall needs to understand inbound Home Test Init and Care-of Test Init messages, so that these are not dropped. The firewall has no way to examine a BU, distinguish legitimate instances from an attack, and update its state accordingly.
- *The HA is in a network protected by firewalls:* The firewalls need to handle ESP and unsolicited incoming connections. Movement by the MN may result in traffic arriving at the HA through a different firewall from before, so stateful firewalls need to maintain this state jointly.
- *The MN moves into a network protected by firewalls:* First, the BU with the HA has to get through. Then, existing connections, which have no prior state, need to continue. Finally, return routability with CNs needs to work.

It is unclear how the combined goals of accommodating these cases and still repelling attacks will be satisfied, so more work on this topic is needed.

- + The HA is likely to reside on a DMZ or a perimeter network, where it is exposed to the Internet. If this is the case, the HA itself needs the appropriate level of security controls.
- + Many privacy issues exist at all protocol layers, from MAC addresses to application-layers. [RFC 4882](#), *IP Address Location Privacy and Mobile IPv6: Problem Statement*, considers only *location* privacy with MIPv6, and only the IP layer. Two issues are identified: disclosure of a MN's HoA to eavesdroppers and disclosure of its CoA to CNs. A solution to the former is to use confidentiality with ESP and not to use route optimization. A solution to the latter is not to use route optimization.

The major threat to mobile communications is the ability of a nearby attacker to eavesdrop on these messages. Since mobility implies wireless communications, it is likely that the MN will be on an insecure wireless access medium and, thus, subject to eavesdropping by an attacker. This is what makes the use of IPsec, or an equally secure alternative (e.g. encryption at the application layer such as SSL/TLS [Secure Sockets Layer/Transport Layer Security], SSH [Secure Shell], etc.), such a critical requirement.

4.4.4 Unknown Aspects

MIPv6 is a flexible yet complex capability. Some aspects of the specification are not yet complete and not all implementations support mobility yet. This section highlights some considerations for minimizing unknown or unforeseen behaviors on networks using MIPv6. Emerging topics such as bootstrapping, dual stack operation, and reliability are covered.

A mobile node needs a HoA, a HA address, and a security association with its HA. Statically provisioning this information can be administratively expensive, so work has started on obtaining it automatically through a process called bootstrapping. Variations exist in types of service providers and

methods of address assignment and authentication. These issues are discussed in [RFC 4640](#).³⁶ A distinction is made between the case in which the same service provider offers mobility and provides authorization and the case in which these functions are split between service providers. Proposals for providing bootstrapping information with DHCPv6 and making bootstrapping work with RADIUS (Remote Authentication Dial-In User Service) authentication are under discussion. A solution has been developed for the split case (different home network and mobile providers) in [RFC 5026](#), *Mobile IPv6 Bootstrapping in Split Scenario*.

A MN with dual IPv4 and IPv6 protocol stacks could run both MIPv4 and MIPv6, but this would be expensive and also awkward if the mobility protocols are not integrated. It also raises security issues, not the least of which is the need to include the completely different approaches to security in MIPv4 and MIPv6. One immediate question is which versions of IP the foreign link supports. A second is whether the same HA supports both MIPv4 and MIPv6. A third is which protocols the CNs are capable of using. None of the tunneling mechanisms designed to help IPv4-to-IPv6 transition includes specific support for mobility. Because mobility is essentially a tunneling protocol, ways exist, in principle, to use a single HA and a single CoA (IPv4 or IPv6) to deliver either IPv4 or IPv6 packets. One such solution, documented in [RFC 5555](#), allows dual stack systems to run both IPv6 and IPv4 over MIPv6. It also handles Network Address Translation (NAT) in the IPv4 networks.

The MIPv6 HA is a single point of failure. Switching HAs requires detecting failure, finding an alternative HA, transferring state, and reestablishing security. Different solutions are possible: a MN could maintain connectivity with a “hot spare” HA, the HA could share state with backup systems, or the MN could, in effect, start over when a failure is detected. These involve different tradeoffs, and solutions are under discussion³⁷. Another case being considered is when, for load balancing, planned outages, or other reasons, a HA can announce that it will no longer be available.

4.5 Jumbograms

[RFC 2675](#) defines an IPv6 Hop-by-Hop Option called jumbograms, which allows an IPv6 packet to contain a payload longer than 65,535 octets. Jumbograms are only defined for IPv6 interfaces that may be attached to links with a MTU greater than 65,575 octets. Other interfaces can ignore jumbograms. Many IPv6 implementations allow jumbograms but do not support any link layers that can handle them (except the loopback interface).

4.5.1 Specification Overview

The 16-bit Payload Length in the IPv6 Header is set to zero and the Hop-by-Hop Jumbo Payload Option, Type C2 base 16 (1100 0010), must be included right after the header. It specifies a 32-bit Payload Length. Jumbograms must be at least 65,536 bytes and must not be fragmented.

Common transport payloads such as TCP and UDP have packet size limits (in the UDP length field and the TCP MSS [Maximum Segment Size] option and Urgent field), so modifications are needed for them to use jumbograms.

4.5.2 Security Ramifications

[RFC 2675](#) says, “The Jumbo Payload option and TCP/UDP jumbograms do not introduce any known new

³⁶ Also see Kempf, Arkko and Nikander, “Mobile IPv6 Security,” *Wireless Personal Communications*, Vol. 29, (2004): 389-414; and Soliman, *Mobile IPv6*, Addison-Wesley, 2004.

³⁷ See the current version of the IETF Internet Draft, [Home Agent Reliability Protocol \(HARP\)](#), Work in Progress.

security concerns,” but the BSD and KAMI implementations claim that IPsec AH and jumbograms are incompatible due to an unexplained “quirk in the specs” regarding the AH header size. Given the limited use of jumbograms, it is not surprising that some (perhaps most) IPsec implementations do not support them.

Jumbograms may be used as a vehicle for DOS. They can cause substantial latency, increase the cost of retransmission, and interfere with QoS. A maximum length jumbogram will tie up a one gigabit per second link for over a half a minute, and if the UDP checksum fails, the receiving interface may request retransmission. A jumbogram, used together with a hop-by-hop header, could possibly be used to perform a resource-consumption attack.

Intermediate systems such as application layer firewalls and intrusion detection systems may be unprepared to handle huge jumbograms (which can be up to four gigabytes long).

Finally, many IPv6 implementations may contain places in which the Payload Length is stored as an unsigned 16-bit integer. Finding and exploiting these may be a method of causing buffer overflows, kernel crashes, or other unintended effects.

4.6 Address Selection

An IPv4 interface typically has one IPv4 unicast address, which may or may not be globally routable, plus the loopback address (127.0.0.1). An IPv6 interface, in contrast, typically starts with an interface local loopback address, a link local address, a unique local address, and a globally routable address. Renumbering and multihoming can result in having more than one address of certain types. There is nothing to stop one from generating and using additional addresses, and renumbering may cause more than one address of a given type to be active. Some addresses should still be accepted or handled by an implementation, but their use is no longer recommended. A choice of source addresses exists for every packet. Often, a choice of destination addresses exists as well.

In a system running both IPv4 and IPv6, address selection is more complicated, and rules need to be given for which of these to prefer.

[RFC 3484](#) specifies rules for choosing source and destination addresses, but more recent experience has shown that additional rules are needed for networks with multiple prefixes. This section examines IPv6 address selection rules and their consequences.

4.6.1 Specification Overview

This section summarizes the rules for source and destination address selection specified in [RFC 3484](#). Criteria for choices and preferences in each case are covered.

Address selection may have to choose among IPv4 versus IPv6, addresses with different scopes, public or private address, and so forth. Some of the rules seem obvious, for example, choosing an address that has not been deprecated over one that has, but nevertheless, software has to be written to make the right choice. In general, source and destination pairs are chosen to match scopes and types (e.g., native IPv6, 6to4, or IPv4-mapped), to prefer smaller scopes, and to prefer native addresses over transitional addresses. If the destination is a multicast address, the multicast scope is used in choosing the scope of the unicast source address. One of the principles introduced in address selection is *longest prefix match*, which means, in the absence of other criteria, one chooses source and destination addresses to try to take advantage of route aggregation.

Another new concept is the *address selection policy table*, which allows administrators to add or override address selection rules. IPv4 addresses are represented in the table as IPv4-mapped IPv6 addresses, and these are assigned scopes corresponding to IPv6 link-local or global addresses. [RFC 3484](#) gives an example of a policy table:

Prefix	Precedence	Label	Use
::1/128	50	0	Loopback
::/0	40	1	Default (including native IPv6)
2002::/16	30	2	6to4
::/96	20	3	IPv4 Compatible
::ffff:0:0/96	10	4	IPv4 Mapped

Given an address, the entire table is scanned to find the entry with the longest common prefix matching the address. Then the corresponding Precedence and Label are returned. This has the effect of, first, matching labels of the source and destinations, and, second, preferring native IPv6 addresses over IPv4 or the different tunneling addresses (6to4 or v4-compatible).

The first step in source address selection is to make a list of candidate choices. In general, the choice should match the interface and scope. The next step is to sort the candidates by going through an ordered list of rules, starting with rule 1, and proceeding to the next rule only to break ties:

1. Prefer a source address that equals the destination.
2. Prefer the smallest scope that is at least as large as the destination's scope. (This rule is mandatory.)
3. Prefer an address that is not deprecated.
4. Prefer a home address over a care-of address, unless the application chooses to reverse this.
5. Prefer an address on the correct outgoing interface for the given destination.
6. Prefer an address that matches the label of the destination in the policy table.
7. Prefer a public address over a temporary address, unless the application chooses to reverse this.
8. Use the address with the longest matching prefix in common with the destination.

Selecting a destination address follows a similar set of rules. One main difference is that choosing a destination involves asking what source would be used *for each choice*. Candidates are listed and compared, starting at rule 1, and continuing down the list of rules to break ties:

1. Avoid unusable destinations (ones that are unreachable or have no usable source address).
2. Prefer a destination with a source of matching scope.
3. Prefer a destination with a source that is not deprecated.
4. Prefer a destination with a source that is just a home address to one with a source that is just a care-of address.

5. Prefer a destination with a source that has a matching label (in the policy table).
6. Prefer a destination with higher precedence (in the policy table).
7. Prefer a destination reached with native transport over one using encapsulation.
8. Prefer a destination with smaller scope.
9. Prefer a destination with a source having the longest matching prefix.
10. Prefer a destination that occurred first in the original list. That is, leave the order unchanged.

4.6.2 Differences from IPv4 Standards

The address selection problem itself is new with IPv6 for several reasons:

- + Freedom to use more addresses because of the larger address space
- + Scoped addresses (link local, unique local, etc.)
- + Explicit rules that allow use of multiple addresses of the same type
- + Prefix renumbering
- + Address deprecation.

The address selection rules recognize MIPv6 and distinguish home addresses from care-of addresses. The simultaneous presence of IPv4, IPv6, and transition tunneling protocols requires address selection rules that integrate the use of both IPv4 and IPv6.

4.6.3 Security Ramifications

The proper use of the address selection rules given here can have an impact on security, affecting data availability and integrity. Improper address selection can result in dropping packets inadvertently or inefficient packet forwarding, which are not security problems *per se*. Many attacks, however, use spoofed addresses, which can be characterized as systems not following the rules.

Proper source address selection is important for ingress filtering. Note that it may be difficult for ingress filtering to distinguish temporary addresses from spoofed “in-prefix” addresses used in DOS attacks. Denying newly established incoming TCP and UDP connections can prevent some spoofed source address attacks by preventing datagrams with spoofed source addresses from transiting network protection devices.

Address selection works across IPv4 and IPv6 addresses and can determine which is used when both are available or which type of tunneling is used. Ensuring that address selection policy is correctly specified can affect other assumptions about security.

The address selection rules can help enable some attacks on privacy. By probing a host with requests that come from different source addresses and observing what source addresses the target uses for replies, one can extract information about the set of addresses used by the target.

If an attacker were to compromise and modify the address selection policy table, this could cause a man-in-the-middle or DOS attack. This can be mitigated by applying best practices of desktop and server security (e.g., malware filtering in web and e-mail content, patching processes, removal of administrative

rights from end-users, etc.), methods that apply equally to IPv6 and IPv4.

4.6.4 Unknown Aspects

Experience using the rules described above for address selection has unveiled several problematic configurations, which require further attention. When a link has more than one router, or a site border router is connected to more than one ISP, multiple prefixes can be used. If the prefix of a source address does not correspond to the connection used, the packet may be dropped because of ingress filtering, or an asymmetrical route may be established, and returned packets may be dropped due to packet filtering. This problem is even worse in the case in which one of the routers is not globally connected but only connected to a closed network segment. Prefix renumbering increases the complexity of this, because choices of appropriate prefixes are not static.

The longest prefix match rule can always distinguish currently allocated global addresses from unique local addresses, but this will become an issue when global addresses with a leading one bit begin to be allocated.

Applications may need to have better control over address selection in certain cases. One case is choosing between temporary and permanent addresses; another is between home and care-of addresses.

The above selection rules prefer IPv6 over IPv4 addresses, but cases exist in which the IPv4 address is the better choice. For example, choosing the IPv6 address may result in use of a less efficient tunnel through the same IPv4 network or choosing an IPv6 ULA will give only local connectivity as opposed to an IPv4 address which may give global connectivity.

Work is underway to find ways to remedy all of these situations, but in the meantime, installations need to be aware of these potential problems and configure ways to work around them, case by case.

4.7 Dynamic Host Configuration Protocol (DHCP) for IPv6

Dynamic Host Configuration Protocol for IPv6 (DHCPv6) is a client-server protocol that provides IPv6 interfaces with address assignments and other configuration information. The alternative automated protocol, stateless address autoconfiguration (SLAAC), is the “native IPv6” way to obtain a dynamic IPv6 address, but it does not supply information like DNS and NTP servers’ addresses or perform dynamic DNS updates. Also, stateless address autoconfiguration does not offer centralized control of address assignments, which some network operators may require. DHCPv6 keeps track of address assignments and is called *stateful*, in contrast to stateless. DHCPv6 is not described in the IPv6 standards as an essential component, but as more enterprises start to use IPv6, demand for DHCPv6 is growing.

The original DHCP for IPv4 (DHCPv4) and DHCPv6 are two separate protocols. One of the reasons for using DHCPv4 with IPv4 is address conservation. Addresses may be in short supply and need to be recycled, for example, on a dial-in network. This is not normally a concern with IPv6. Nevertheless, IPv6 addresses are more difficult to memorize and type in, so having automated tools to manage them is an aid to users and administrators.

DHCPv6 servers assign IPv6 addresses to network interfaces on a *lease* basis. The client may use the assigned IP address for an administratively pre-determined amount of time before the lease expires. This means that IPv6 address assignments made by DHCPv6 servers are not permanent, and over time, more than one node may use a given IP address, but no more than one node can use an address at one time.

4.7.1 Specification Overview

DHCPv6 is defined in [RFC 3315](#), *Dynamic Host Configuration Protocol for IPv6*. It is a quite sophisticated protocol with many message types, options, status codes, and timers. It performs three services:

- + Address configuration: allocating addresses and providing a network prefix and address of the default router
- + Providing other configuration information: primarily, a DNS server's address, but potentially many other types of information
- + Allocating prefixes to routers.

In addition to [RFC 3315](#), many other standards supplement or modify DHCPv6:

- + [RFC 3319](#), *Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers*, defines an option for DHCPv6 clients to obtain a list of domain names or addresses for SIP servers.
- + [RFC 3633](#), *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6*, describes using DHCPv6 between routers for delegation of address prefixes, which is not discussed further this document.
- + [RFC 3646](#), *DNS Configuration Options*, defines options for configuring a list of DNS name servers and domain search lists.
- + [RFC 3736](#), *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*, defines the subset of [RFC 3315](#) that needs to be implemented for stateless DHCPv6.
- + [RFC 3898](#), *Network Information Service (NIS) Configuration Options*, defines four options for configuring NIS services.
- + [RFC 4014](#), *Remote Authentication Dial-In User Service (RADIUS) Attributes Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Information Option*, specifies an option for a DHCPv6 relay agent to pass information obtained during authentication of the client to the DHCP server.
- + [RFC 4075](#), *Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6*, defines an option for obtaining a list of SNTP servers' addresses.
- + [RFC 4361](#), *Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)*, updates DHCPv4 to be compatible with naming and DNS updates used in DHCPv6, so that systems running both IPv4 and IPv6 can work with both versions of DHCP separately but compatibly.

Three primary configuration options are available for IPv6 interfaces:

1. Use autoconfiguration and not DHCPv6.
2. Use DHCPv6 and not autoconfiguration.
3. Get an address with autoconfiguration and then use DHCPv6 to retrieve additional information.

This choice is driven by two flags (called M for *managed* and O for *other*) in Router Advertisements (RA). When the M flag is set to 1, DHCPv6 should be used for address assignment. When the O flag is set to 1, DHCPv6 should be used to obtain other configuration information.

Clients and servers use UDP to exchange DHCPv6 messages; they listen on ports 546 and 547, respectively. Both clients and servers include a DHCP Unique Identifier (DUID) to identify themselves. A client's request for an address is called a Solicit message. It is normally sent from a link local source address to the link-scoped All_DHCP_Relay_Agents_and_Servers multicast address (FF02::1:2). A typical response to a Solicit message is an Advertise message containing a list of allocated addresses, the address of the default router, and the network prefix for the interface. DHCPv6 is capable of assigning multiple addresses to a single IPv6 interface. Being able to choose a default router allows DHCPv6 to perform load balancing. (For load balancing requirements without using DHCPv6, see [RFC 4311](#).) Once the client has determined the address of a DHCPv6 server, it may, in some cases, send unicast messages directly to the server.

If a client reaches a relay agent, the agent normally queries the All_DHCP_Servers site-scoped multicast address (FF05::1:3). This allows a DHCPv6 client to reach a DHCP server not attached to the same link. The relay operation is transparent to the client, and the discussion in the remainder of this section omits message relaying.

4.7.1.1 Client-server Exchanges Involving Four Messages

To request the assignment of one or more IPv6 addresses, a client first locates a DHCP server and then requests the assignment of addresses and other configuration information from the server. The client sends a Solicit message to the All_DHCP_Relay_Agents_and_Servers address to find available DHCP servers. Any server that can meet the client's requirements responds with an Advertise message. The client then chooses one of the servers and sends a Request message to the server asking for confirmed assignment of addresses and other configuration information. The server responds with a Reply message that contains the confirmed addresses and configuration.

4.7.1.2 Client-server Exchanges Involving Two Messages

When a DHCP client does not need to have a DHCP server assign it IP addresses, the client can obtain configuration information such as a list of available DNS servers or NTP servers by exchanging a single message pair with a DHCP server. The client first sends an Information-Request message to the All_DHCP_Relay_Agents_and_Servers multicast address. A server responds with a Reply message containing the configuration information for the client.

Even if a client needs address allocations, if a server has IPv6 addresses and other configuration information committed to a client, the client and server may be able to complete the exchange using only two messages. In this case, the client sends a Solicit message to the multicast All_DHCP_Relay_Agents_and_Servers address requesting the assignment of addresses and other configuration information. The client indicates it is willing to accept an immediate Reply message from the server. A server that is willing to commit the assignment of addresses to the client immediately responds with a Reply message. The configuration information and the addresses in the Reply message are then immediately available for use by the client.

Each address assigned to the client has preferred and valid lifetimes specified by the server. The valid lifetime is always longer. When the preferred lifetime expires, the address is deprecated for new uses but may be used for existing connections (e.g., TCP or IPsec) until the valid lifetime expires. To request an extension of the lifetimes assigned to an address, the client sends a Renew message. The server returns a

Reply message with the new lifetimes, and the client may continue to use the address without interruption.

4.7.2 Differences from IPv4 Standards

DHCPv4 and DHCPv6 perform similar functions, but distinct differences exist. Many of these are due to either the underlying differences between IPv4 and IPv6 or the chance to make improvements in DHCPv6 based on what has been learned running DHCPv4.

One primary difference is that DHCPv4 and DHCPv6, unlike DNS, are two separate protocols and services. To date, no single standard for DHCP on dual stack IPv4-IPv6 systems exists. [RFC 4477](#), *Dynamic Host Configuration Protocol (DHCP) IPv4 and IPv6 Dual Stack Issues*, covers potential problems for clients receiving both DHCPv4 and DHCPv6 information and considers potential solutions, both with a single server and separate servers. In either case, many questions arise about how best to merge the information about IPv4 and IPv6 configurations.

A second major difference is due to the presence of SLAAC in IPv6. Dynamic IPv6 addresses can be configured without DHCPv6, and whether DHCPv6 is used is determined by RAs, not by hosts.

DHCPv6 generates far less traffic than DHCPv4. The differences include use of broadcast with IPv4 versus multicast with IPv6. DHCPv6 also contains many optimizations, for example, the ability to assign multiple network addresses to a single interface.

4.7.3 Security Ramifications

To enhance accountability, many organizations may prefer to use DHCPv6 and disable autoconfiguration to force authentication prior to gaining an IPv6 address. This also aids with forensics if there are verifiable logs that show which user had been leased which IPv6 address at a given time. Those logs will provide corroborating evidence in a forensic case. Another security enhancement related to DHCPv6 use is that DHCPv6 is more resilient to pool consumption attack. Because the DHCPv6 pool is typically comprised of an entire /64 subnet there are many addresses in the pool. Therefore, it is unlikely that an attacker will be able to claim all the IPv6 addresses in the pool to cause the next legitimate user's request to get rejected because all the addresses have been leased out.

On the other hand, a network that relies upon DHCPv6 for address assignment is subject to DOS if DHCPv6 services are interrupted. Also, if the information obtained from DHCP is unreliable or inaccurate, the network may degrade or fail. Even the inability to reach a DNS server makes the Internet essentially unusable. Finally, an attack against DHCPv6 may cause packets to be misrouted, which then allows the attacker to compromise the confidentiality or integrity of their contents.

Another DHCPv4 attack is related to the SOLICIT messages. An attacker may launch a DoS attack with many SOLICIT messages. However, these SOLICIT messages can be rate-limited to a low bandwidth amount. Similarly, if an attacker were to send numerous RENEW messages, most DHCPv6 servers will simply answer these requests out of cache to minimize impact on the DHCPv6 server. Additional examples of attacks include:

- + Rogue responses from unauthorized servers distributing false information, addresses, or otherwise
- + Maliciously high volumes of client requests causing denial of service
- + The potential exhaustion of internal server memory by huge numbers of requests (although address exhaustion is not a possibility on, for example, a /64)

- + Access to a private network or theft of service on a public network by unauthorized devices.

One problem that can occur is having improperly configured servers distribute inaccurate information. Many of these problems can be mitigated by:

- + Blocking DHCPv6 at firewalls (the port numbers are easily recognized)
- + Using the DHCPv6 authentication option between clients and servers. This is defined in [RFC 3118](#) and implemented as a shared-secret-based HMAC-MD5 calculation with replay detection. The drawback to this, of course, is that it requires pre-configured security parameters and authentication credentials.
- + Securing connections between relay agents and servers, for example, with ESP in transport mode
- + Reviewing logs of DHCPv6 operations for unexpected events.

4.7.4 Unknown Aspects

DHCPv6 is not universally supported in all IPv6 implementations, although demand for it is increasing as users gain more operational experience with IPv6. One still has to check whether it is included.

It is expected that dual IPv4-IPv6 protocol stacks may be a standard method of operation for many networks for a significant time. Running separate servers for DHCPv4 and DHCPv6 is somewhat of an inconvenience, but limiting conflicting responses and resolving such conflicts if they arise are more difficult problems. More work on this topic is needed.

Work is also in progress on several extensions to DHCPv6:

- + An option for servers to specify a domain name suffix if a client requests one. This may be useful for IPv6 hosts or residential gateways, because IPv6 provides globally reachable addresses.
- + An option for relay agents to request servers to echo relay agent options in their entirety. This is now specified in [RFC 4994](#). The point is that a relay agent may add options to a request, the server may ignore them, and if they are not echoed, they may not get back to the client. With this echo option, DHCPv6 works more like DHCPv4 and avoids the problem just described.
- + An option for relay agents to use message sequence numbers with servers for replay protection
- + A method to query servers about active leases similar to the method specified in RFC [4388](#) for DHCPv4. This may be useful for administrators who want to poll their networks for active addresses.

4.8 IPv6 Prefix Renumbering

This section covers renumbering of IPv6 sites, that is, changing network prefixes. Renumbering hosts (interfaces) individually is covered in [Section 4.1](#) on Multihoming, [Section 4.4](#) on Mobile IPv6, and [Section 4.6](#) on Address Selection. Site renumbering with IPv4 is regarded as a major chore, although NAT, where it is used, simplifies the problem somewhat. IPv6 networks likely will not use NAT, but the IPv6 specifications contain provisions for making site renumbering without a network outage work smoothly. Nevertheless, one must pay attention to a significant number of details to avoid errors, oversights, and unexpected results.

The most commonly cited reason for renumbering network prefixes is to change ISPs. Renumbering sometimes becomes necessary when companies with large intranets merge or reorganize or when an ISP

itself forces renumbering. Renumbering affects a large number of components: routers, firewalls, filters, DNS, DHCPv6, routing protocols, system configuration tables, applications, network management tools, log files, etc.—anything that specifies addresses. Renumbering an ISP, although it presents many of the same problems discussed here, is outside the scope of this document.

4.8.1 Specification Overview

The underlying basis for network prefix renumbering is specified in [RFC 4861](#) and [RFC 4862](#).

Renumbering is facilitated because RAs *lease multiple* addresses to interfaces. This has two consequences: old addresses time out, are deprecated, and eventually disappear; new addresses can be immediately available. This “make-before-break” switchover is accomplished by having two lifetimes for prefixes in RAs, a preferred lifetime and a valid lifetime. This allows individual addresses to be categorized as preferred or deprecated. The intention is to allow upper layer protocols ample time to complete using old, deprecated addresses and begin using new, preferred ones. At the same time, link-local addresses of routers do not change, so host-to-router ICMPv6, DHCPv6, and such infrastructure communications are not interrupted.

[RFC 4861](#) presents an example in which a prefix is initially advertised as having a long preferred lifetime that is later shortened. The point is that nodes unplugged from the network may miss the shortened announcement, so even after the prefix has expired, the prefix must continue to be advertised with a zero lifetime until the original, long period has expired. This example is accompanied with a warning about advertising infinite lifetimes.

Another complication is that RAs themselves have lifetimes, in addition to the lifetimes of the prefixes they are announcing. Lacking a valid prefix, no global IPv6 address exists; lacking a valid RA, no default route exists.

Just because interfaces can pick addresses with a new prefix does not make renumbering work. [RFC 4192](#) contains a description of the steps one needs to take to renumber a network with a fairly normal complement of services. The process involves allocating sub-prefixes of the new prefix to links and updating all occurrences of addresses from the old prefix to the new one. These include:

- + Manually assigned addresses for interfaces on routers
- + Routing information and link prefixes advertised by routers
- + Addresses on routers, firewalls, and packet filters used for access control or ingress filtering
- + Addresses assigned to interfaces with stateless address autoconfiguration
- + Addresses and other information provided by DHCPv6
- + DNS records (primarily AAAA and PTR records, as well as DNSSEC)
- + All other instances of addresses in applications, command sequences, configuration files, and elsewhere.

Some parts of a network can be renumbered independently of others, which may allow administrators to break the task up and make it more manageable. In other situations, more than one old prefix may be combined under a new prefix.

The steps outlined in [RFC 4192](#) can be summarized as follows:

1. Obtain the new prefix and DNS reverse zone. Plan how sub-prefixes will be allocated to links. Shorten DNS TTLs and DHCP licenses. List critical services and applications and then test each one to discover all of the places where addresses will need to be updated.
2. Update DNS records. This can be done centrally or delegated to individual systems that use dynamic DNS. In any event, it needs to be coordinated with updates to addresses on interfaces.
3. Configure routers and network infrastructure services with the new prefix in parallel with continued use of the old prefix. DHCPv6 prefix delegation can be used to distribute sub-prefixes to links. Check that routing protocols, DNS, DHCPv6, firewalls and packet filters, intrusion detection systems, and any other infrastructure components are working properly with the new prefix before advertising it outside its network.
4. Allocate addresses with the new prefix to interfaces on hosts with stateless address autoconfiguration, DHCPv6, or manual procedures. Check that the new prefix is working correctly.

After this step stabilizes, the network should be working correctly with both prefixes. The rest of the process consists of removing the old prefix.

5. Begin using the new prefix. Phase the old prefix out of internal and external DNS servers. Allow individual systems to switch to the new prefix asynchronously. This spreads the load on systems like dynamic DNS. Individual systems may need to run certain scripts or tools to complete their reconfiguration. Applications may have to be forced to reset their address caches. Utilities can check where the old prefix is still being used.
6. RAs then deprecate the old prefix by setting the preferred lifetime of link prefixes and addresses with the old prefix to zero.
7. Once sessions using the old prefix complete, it is removed from the routing infrastructure, DNS, anything that depends on DNS, and any other configuration files. The old prefix may be reclaimed by whoever allocated it. The DNS reverse zone may be deleted and its delegation removed. All timers, in particular DNS TTLs, can be reset to their normal values.

Two of the more difficult cases involve systems controlled by another administrative entity (such as an external DNS server) and manually configured devices.

[RFC 4192](#) and drafts proposing updates to it continue with advice on various topics from software development to routing protocols on how to make renumbering go easier:

- + Applications should not ignore the DNS's TTL values.
- + Routers should not rely on manual configuration but should either use DNS (correctly!) or provide an integrated capability for renumbering.
- + Care must be taken in updating BGP not to cause routing outages or route flaps that propagate throughout the network. See [RFC 4192](#), Section 3.4.
- + Using the Stream Control Transmission Protocol (SCTP) may be a good choice for critical applications that need to keep running during renumbering, because it allows dynamic changes in an endpoint's IP addresses.
- + For MIPv6, renumbering at the visited site can be handled as if the mobile moved to a new location, and it can learn of renumbering at the home site with the MPS and MPA ICMPv6 messages.

- + Multicast addresses with embedded unicast prefixes should be updated after renumbering to avoid future address clashes.
- + Using ULAs for services within a site during renumbering may be a good idea. Renumbering does not have to change ULAs.
- + During renumbering, some traffic will be sent from the old to the new prefix or vice versa. Routers and packet filters need to be aware of this.

4.8.2 Differences from IPv4 Standards

[RFC 2072](#) contains a description of IPv4 renumbering at a comparable level of detail with [RFC 4192](#). On the one hand, NAT simplifies renumbering, but renumbering plans that involve different size CIDR networks tend to be more complicated than, for example, mapping one /48 to another in IPv6.

Because IPv4 does not provide the basic building blocks for automated renumbering—starting with RAs containing preferred and valid timers and continuing with DHCPv6 prefix delegation and other features—IPv4 networks must be renumbered manually, which usually means continuous operation throughout the renumbering process is not possible. The processes described in this section do not correspond to any processes typically followed in IPv4 networks.

4.8.3 Security Ramifications

Three main points to keep in mind are:

- + Errors and omissions during renumbering can result in timeouts or lost connectivity during the renumbering. This applies especially to infrastructure components like DNS (including the reverse tree) and routing protocols.
- + Renumbering, if not carried out with attention to security, can leave the network in intermediate states vulnerable to attack. If one wants to extend the castle metaphor for firewalls, then renumbering is the changing of the guard.
- + Incomplete renumbering of security systems can leave permanent vulnerabilities.

If renumbering is due to changing service providers, and both prefixes are available during an overlap or transition period, all of the security considerations for multihoming apply here:

- + Multiple addresses and prefixes lead to complications, especially for enforcing access control lists.
- + Ingress filtering, if implemented at both service providers, will cause some loss of connectivity, unless some arrangement can be made to relax the filtering rules temporarily.

Other security considerations include the following:

- + Renumbering will break long-lived IPsec SAs and SSL/TLS connections.
- + Log files need to be interpreted correctly with respect to address changes.
- + Dynamic DNS updates should be secured with the TSIG and SIG(0) mechanisms defined in DNSSEC.

4.8.4 Unknown Aspects

In spite of all that has been written about IPv6 renumbering, little practical experience exists, although there are reports covering some controlled experiments. Guidelines, rough lists of steps, and some warnings exist, but an agreed upon set of best practices has not yet emerged.

Three topics identified as needing more attention are:

- + Detailed prefix renumbering procedures for specific products
- + A stable method to apply renumbering to BGP without causing route flaps or outages
- + Renumbering of DNS across administrative domains, and, in particular, management of the DNS reverse zone.

Two potential methods of simplifying renumbering are:

- + Assigning well-known anycast addresses for routers and servers that provide important parts of the network infrastructure
- + Deploying a split identifier-locator technology like SHIM6, in which case renumbering only involves the locator part of the address.

5. IPv6 Security Advanced Topics

This chapter covers four additional security topics that are new or different with IPv6. The first two topics are privacy addresses and cryptographically generated addresses, which become feasible because of the enormous size of the IPv6 address space. IPsec is the third security topic; new developments and certain aspects specific to IPv6 are covered here. The fourth topic is security for neighbor discovery (ND), which uses ICMPv6 on the local link. The two distinct approaches to security for ND are described and compared.

5.1 Privacy Addresses

IPv6 offers a number of different addressing options that can assist security architectures. Privacy addresses can be used by client applications to inhibit user tracking, which may be useful for protecting external communications. This section defines and discusses privacy addressing in IPv6, differences between the IPv6 and IPv4 standards, and their security ramifications.

According to [RFC 4291](#), interfaces using stateless address autoconfiguration generate interface identifiers based on their IEEE EUI-64 identifiers. This provides strong support for uniqueness, but it allows an interface to be tracked, even if it moves from one network to another, or the network prefix is changed.

Consider, for example, a mobile device that connects to different wireless networks from different locations. Using IPv4, the device is likely to use DHCP in the different locations and receive completely uncorrelated addresses. If it used IPv6 autoconfiguration, its wireless interface's address would have the same interface identifier in each instance. Moreover, the IEEE EUI-64 interface ID, because it is based on the hardware MAC address, might reveal what type of device it is.

An interface that accepts inbound connections and has a DNS name clearly cannot have a private address, but it is still possible to use different addresses for outbound connections. [RFC 4941](#), *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, defines a way to generate and change such temporary addresses. The important requirements are that the sequence of temporary addresses an interface chooses must be totally unpredictable and have low probability of colliding with choices made by other interfaces.

The method recommended in [RFC 4941](#) works approximately as follows:

1. Obtain the interface identifier that would be used without this scheme.
2. Apply a cryptographic hash function to this value and either a saved history value or a randomly chosen 64-bit number.
3. Use the output of the hash function to select the interface identifier and to update the history value.
4. Run Duplicate Address Detection (DAD).
5. Set the appropriate lifetimes and join the solicited node multicast group corresponding to the interface identifier.
6. Continue to use prior interface identifiers for ongoing connections but not for new ones.
7. Repeat this process whenever one connects to a new network or the timers set during the previous iteration expire.

Link encap:Ethernet HWaddr 00:0C:29:6F:8F:98					
inet addr:192.168.1.3 Bcast:192.168.1.255 Mask:255.255.255.0					
inet6 addr: 2002:2::9048:b971:277c:e16c/64 Scope:Global					
inet6 addr: 2002:2::20c:29ff:fe6f:8f98/64 Scope:Global					
inet6 addr: fe80::20c:29ff:fe6f:8f98/64 Scope:Link					
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1					
No. -	Time	Source	Destination	Protocol	Info
1	0.000000	2002:2::9048:b971	2002:2::1	ICMPv6	Echo request
<div> <div>Frame 1 (118 bytes on wire, 96 bytes captured)</div> <div> <div>Ethernet II, Src: Vmware_6f:8f:98 (00:0c:29:6f:8f:98), Dst: Vmware_f2:d2:a1 (00:0c:29:f2:d2:a1)</div> <div>Destination: Vmware_f2:d2:a1 (00:0c:29:f2:d2:a1)</div> <div>Source: Vmware_6f:8f:98 (00:0c:29:6f:8f:98)</div> <div>Type: IPv6 (0x86dd)</div> </div> <div>Internet Protocol Version 6</div> <div> <div>Version: 6</div> <div>Traffic class: 0x00</div> <div>Flowlabel: 0x000000</div> <div>Payload length: 64</div> <div>Next header: ICMPv6 (0x3a)</div> <div>Hop limit: 64</div> <div>Source address: 2002:2::9048:b971:277c:e16c</div> <div>Destination address: 2002:2::1</div> </div> </div>					
Internet Control Message Protocol v6					

Figure 5-1. Example of IPv6 Privacy Addressing

As an example of privacy addressing, Figure 5-1 shows an Ethernet interface with a /24 non-routable IPv4 address, a /64 globally routable IPv6 address generated with privacy addressing, a second /64 globally routable IPv6 address based on its IEEE address, and link local IPv6 address based on its IEEE address.

For several reasons, this privacy extension mechanism should be used with care, if it is used at all:

- How much privacy is actually provided is questionable. Particularly on small networks that do not change much, anyone who can observe network traffic can also correlate activity fairly accurately regardless of whether or not addresses change periodically. An observer may even be able to determine how often each interface is generating such new addresses.
- On some networks, administrators may want to have better control of what is connected and correspondingly, which addresses are used. Local security policy may dictate that for audit or forensic purposes, all addresses must be centrally assigned and logged. In such cases, it is better not to allow either privacy addresses or stateless address autoconfiguration but to require using DHCPv6 for address assignments.
- In general, enterprise security policy does not extend the right of private communications to a user who is on an enterprise computer or accessing an enterprise network. In these cases, the goals of forensics and security may be viewed as more crucial than the goal of protecting a user's privacy while using the Internet. This is a policy decision that should be left up to the individual department or agency.
- Good networking practice is to apply ingress filtering, that is, not allow packets without valid source addresses into the core of the network. Some distributed denial-of-service (DDOS) attacks have used forged source addresses with valid prefixes. Privacy addresses may be difficult to distinguish from addresses used in these attacks without additional measures such as rate limiting or complete reverse path checking.

For these reasons, privacy addresses should not be turned on by default, and careful consideration should be given case-by-case as to whether it is worth using. In many cases where enterprises are operating their own networks, DHCPv6 address assignments may be preferable.

5.2 Cryptographically Generated Addresses

Cryptographically Generated Addresses (CGAs), also called Hash Based Addresses, provide a method to prove ownership of the source address in a packet. The idea is to choose a public and private key pair suitable for creating digital signatures with the private key and then to verify them with the public key (see Figure 5-2). Then, the public key (along with other parameters) is used to generate an interface identifier, the public key is inserted into the packet, and the packet is signed with the private key. Upon receipt, the public key can be used to check both the address and signature. An attacker without the private key could not sign a forged packet.

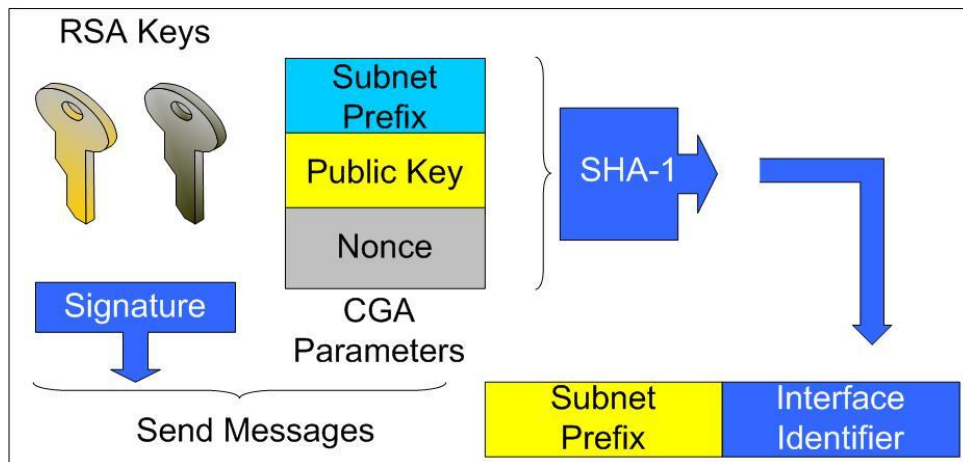


Figure 5-2. Generating Cryptographic Addresses from Public-Private Key Pairs

Four processes are needed to make this work:

The sender must:

1. Generate a key pair and corresponding address.
2. Insert the public key in a packet and sign it with the private key.

The recipient must:

3. Check that the source address corresponds to the public key.
4. Verify the signature with the public key.

Note that using a CGA does not prove one's actual identity, but it does show that the same entity (the one with the private signing key) generated each of the packets, and that the packets were not subsequently modified by another entity. The point is that no one without the private key can legitimately use the CGA.

CGAs may be used to enhance security in any IPv6 network, but in many cases they have no discernible advantage and more overhead than IPsec or higher-layer security protocols. They may be particularly helpful, however, in cases involving ownership of a newly generated address. CGAs have been

standardized as the main security building block for the IPv6 Secure Neighbor Discovery (SEND) protocol [RFC 3971](#), and they have been proposed for use with the Site Multihoming for IPv6 (SHIM6) protocol. A different construction based on the same ideas is used in the experimental Host Identity Protocol (HIP).

In all cases, the hash algorithm currently specified for CGA is SHA-1, the signature algorithm is RSA, and the signature format follows RSA's PKCS (Public Key Cryptography Standard) #1, version 1.5, described in [RFC 3447](#).

CGAs are specified in [RFC 3972](#) and [RFC 4581](#). Implementations need to generate and store cryptographic values securely to use these protocols safely. See in particular [RFC 4086](#) for a discussion of securely generating pseudo-random values.

The strength of the cryptographic security provided by CGAs is limited. The RSA keys are specified to be from 384 to 2096 bits long. The smaller end of this range is certainly vulnerable to factoring attacks, whereas the larger end is in line with current security recommendations. Also, the CGAs themselves have only 59 bits of cryptographic payload, so conceivably one could mount a search for a key that matches a known CGA. Mass searches for such matches are made more difficult by adding a randomly chosen "salt" value to each CGA.

5.3 IPsec in IPv6

In the early 1990s, the IETF began to view the lack of IP-level security as a serious drawback, and it started developing a collection of network-layer security protocols known as IPsec to be used specifically to secure IP communications. The IETF has published three versions of IPsec, which now provides strong, up-to-date confidentiality and integrity protection, access control, replay detection, key management, and strong peer-entity authentication for IPv4. IPsec has proved difficult to deploy with IPv4, and its widespread use has been limited to protecting certain virtual private networks (VPNs) and for secure remote access to enterprise networks when strong security is a requirement. Some explanations include the perceived complexity of IPsec, the lack of sufficient infrastructure for authentication, and inability of application programs to interface with an IPsec subsystem located in an operating system, networking card, or external device. All of these have been or are being addressed to a certain extent. The use of Network Address Translation (NAT) at many edge or customer premise routers complicates the use of IPsec to provide end-to-end security by using non-routable addresses at end systems. Although various workarounds such as UDP encapsulation have been deployed in the above scenarios, full use of IPsec depends upon true peer-to-peer, end-to-end interaction at the network layer.

IPsec was designed long after IPv4. Today, most operating systems, routers, and security appliances bundle or integrate IPsec with their IPv4 protocol stacks, but historically, IPsec was implemented separately from IPv4. IPv6 does not have the addressing limitations that inhibit end-to-end use of IPsec with IPv4. Also, IPsec has been recommended as the way to secure important features of IPv6 such as OSPFv3 routing, mobility, and even neighbor discovery (see [Section 5.4](#)).

When IPv6 was developed ([RFC 1883](#) in 1995, which was replaced by [RFC 2460](#) in 1998), one view (or hope) was that it could provide end-to-end security for all network communications and eliminate the need for intermediate security layers and devices such as firewalls. This meant that hosts would simply be able to establish secure tunnels to other hosts as a complete security solution. Security has turned out to be more complicated than that. Many organizations provide internal applications with weak security and rely on a perimeter defense to exclude outsiders. Others wish to exercise tight controls on information entering or leaving their internal networks. Additionally, an authenticated and encrypted virus or worm is still a virus or worm. Therefore, most organizations are unlikely to rely solely upon

IPsec services implemented on individual hosts to provide sufficient protection for all IPv6 network communications. The security requirements and models that evolve as accepted practices for IPv6 networks are likely to be at least somewhat similar to those currently found in IPv4 networks.

On the other hand, because the transition to IPv6 will last a long time, implementations of IPsec on IPv4 networks are likely to continue to be used indefinitely. It is expected that IPsec will be widely deployed in mixed IPv4 and IPv6 environments for many years to come.

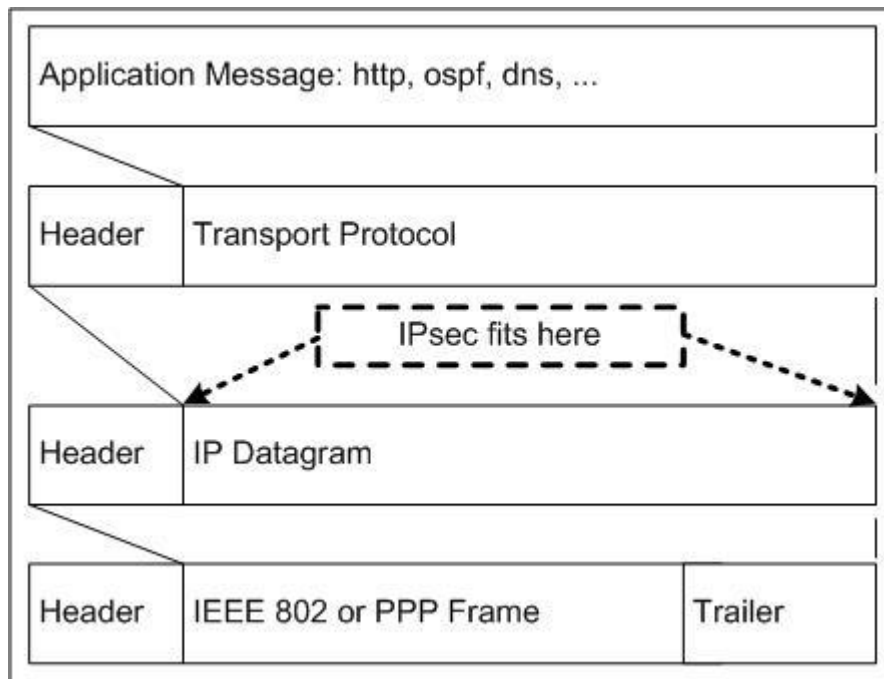


Figure 5-3. IPsec in the TCP/IP Protocol Stack

TCP/IP communications are composed of four layers that work together: application (and session), transport, network, and data link (see Figure 5-3). Security protocols exist for network communications at each of the four layers. Data prepared for transport are passed from the highest to the lowest layer, with each layer adding more information. Because of this, a security control at a higher layer cannot provide full protection for lower layers, because the lower layers add information to the communications after the higher layer security controls have been applied. The primary disadvantages of lower layer security controls is that they are less flexible and granular than higher layer controls and they may not be able to provide end-to-end protection. Accordingly, network layer security, i.e., IPsec, has become a popular choice for securing communications. It provides a balanced approach between the highest layer and lowest layer security protocols and is capable of providing end-to-end protection for all communications between two points.

5.3.1 Specification Overview

IPsec is a framework of open standards for ensuring authenticated and private communications over public networks. It has become the most common network layer security protocol, typically used in IPv4

to create a virtual private network (VPN)³⁸ or to provide secure remote access to a private network. IPsec is a complete set of protocol security services at the IP layer. It can provide entity authentication, confidentiality, data origin authentication, replay detection, and data integrity. It also provides limited traffic flow confidentiality, which masks packet sizes and rates and can make inferential attacks based on traffic analysis substantially more difficult.

IPsec offers three primary models for protection, as follows:

- **Gateway-to-gateway.** This model protects communications between two specific networks, such as an organization's main office network and a branch office network, or two business partners' networks.
- **Host-to-gateway.** This model protects communications between one or more individual hosts and a specific network belonging to an organization. The host-to-gateway model is most often used to allow hosts on unsecured networks, such as traveling employees and telecommuters, to gain secure remote access to internal organizational services, such as the organization's email, Web servers, and custom applications.
- **Host-to-host.** A host-to-host architecture protects communication between two specific computers. It is most often used when a small number of users need to use or administer a remote system that requires network layer security for some or all of its higher layer protocols. OSPFv3 is an example. IPv6 could offer increased opportunities to use this mode, because end-to-end connectivity without Network Address Translation (NAT) makes using IPsec easier.

IPsec consists of the following components:

- **Security Associations (SA).** Each instance of IPsec has a security association database (SAD) of IPsec connections, called security associations. SAs describe the endpoints of the secure connection, exact type of protection provided, cryptographic parameters and keys, expiration time, and label (Security Parameters Index or SPI) to identify packets protected with each SA.
- **Two security protocols: Authentication Header (AH) and Encapsulating Security Payload (ESP).** AH can provide integrity and replay protection for packet headers and data, but it cannot encrypt them. ESP can provide encryption, integrity, and replay protection for packets, but it cannot protect the outermost IP header, as AH can. This protection is not needed in most cases. Accordingly, ESP is used much more frequently than AH because of its encryption capabilities, as well as other operational advantages. The latest IPsec specification states that implementations MAY implement AH, and many choose not to. For a VPN, which requires confidential communications, ESP is the natural choice.
- **Internet Key Exchange (IKEv1 or IKEv2) protocol.** IPsec uses IKEv1 or IKEv2 to authenticate endpoints to each other; define the security parameters of IPsec-protected connections; negotiate and establish IPsec security associations; negotiate secret keys; and manage, update, and delete IPsec's security associations.

³⁸ [NIST SP 800-77](#), *Guide to IPsec VPNs*.

- **Access Control.** A necessary component of IPsec is to specify and enforce proper protection. Therefore, IPsec implementations contain a security policy database (SPD) and packet filter that protects, passes, or drops traffic as specified in a security policy.
- **IP Payload Compression Protocol (IPComp).** Optionally, IPsec can compress packet payloads before encrypting them.

The following points summarize the main ideas behind how IPsec works:

- Both AH and ESP can be used in *transport mode*, which means that the AH or ESP protocol following the IP header encapsulates and protects the upper layer protocol that follows, or they can be run in *tunnel mode*, which means that the AH or ESP protocol encapsulates and protects an entire IP datagram. A given SA uses one mode or the other. Typically, transport mode runs end-to-end, and tunnel mode runs between intermediate security gateways, but exceptions to this exist.
- AH provides integrity protection for all packet headers and data, with the exception of a few IP header fields that change unpredictably in transit. Because AH includes source and destination IP addresses in its calculations, AH is incompatible with NAT. The use of AH has decreased because more recent versions of ESP ([RFC 2406](#) in 1998 and [RFC 4303](#) in 2005) can now provide integrity protection services with or without confidentiality. ESP does not provide the integrity checks for the outermost IP header that AH can.
- In tunnel mode, ESP can provide encryption and integrity protection for an entire encapsulated IP packet including its “inner” header. ESP tunnel mode is the most commonly used IPsec protocol and mode. Because it can encrypt the original IP header, it can conceal the true source and destination of the packet. Also, ESP can add padding to packets and send dummy packets (IPv4 Protocol or IPv6 Next Header 59) to further complicate traffic analysis. Another advantage of ESP tunnel mode is that it can be compatible with NAT.
- In transport mode, ESP can provide encryption and integrity protection for the payload of an IP packet and for the ESP header and trailer. Transport mode is not compatible with NAT.
- IPsec can use IKE to create security associations. IKEv1 phase 1 creates an IKE SA; IKEv1 phase 2 creates an IPsec SA through a channel protected by the IKE SA. IKEv1 phase 1 has two modes: main mode and aggressive mode. Main mode negotiates the establishment of the bidirectional IKE SA through three pairs of messages, whereas aggressive mode uses only three messages. Although aggressive mode is faster, it is also less flexible and secure. IKEv1 phase 2 has one mode: quick mode. Quick mode uses three messages to establish a pair of unidirectional IPsec SAs. Quick mode communications are encrypted by the method specified in the IKE SA created by phase 1.
- IKEv2 contains many simplifications and improvements listed in [Section 5.3.4](#), and whenever possible, it should be used instead of IKEv1.
- Although ESP does not provide integrity-protection for the outer IP addresses, the ability to use the SA correctly implies possession of the message integrity check key for the SA. If the parties at these addresses used IKE to negotiate the SA, they should be the only ones with this key, so the recipient can infer the source of the packet, regardless of whether the source address is intact. This argument does not hold if this key was manually distributed to more than two parties, but IPsec was not intended to be used in this way.

- IPComp can provide lossless compression for IPsec payloads. Because applying compression algorithms to certain types of payloads may actually make them larger, IPComp, when used, is not applied to such data.

5.3.2 Differences from IPv4 Standards

All major aspects of IPsec under IPv6 are the same as they are under IPv4. IPsec is defined for both protocols in the same specifications. AH, ESP, tunnel mode, transport mode, SA, and other aspects of the protocol are identical to those found within the IPv4 version. Coverage of the IP header in transport mode, ICMP selectors, and fragmentation handling are different.

Once IPv6 is widely deployed, it is more likely to be deployed with end-to-end addressing rather than with NAT. Thus, end-to-end IPsec is more feasible with IPv6. Work in the IETF on using IPsec to secure applications and on IPsec application program interfaces (APIs) may prove to be more useful in end-to-end IPv6 environments and is discussed in [Section 5.3.4](#).

5.3.3 Support for Multicast

The current version of IPsec does not fully support protection for multicast traffic, because IPsec was designed specifically for protecting communications between two points, not among many points at once. Each multicast packet may have many recipients (see [Section 4.2](#)), which raises many IPsec-related issues. For example, many recipients may need to decrypt the same packet, but sharing a secret key among them is not a sound security practice.³⁹ Another issue is that many different hosts may be sending packets to the multicast address. Because each of these hosts needs to share the same authentication mechanism, one source host can spoof the identity of another source host, and the recipients may not be able to detect it, eliminating IPsec's data origin authentication capability. Also, the anti-replay protection provided by the sequence number is not available, because multiple senders could simultaneously generate legitimate packets that happen to use the same sequence number. These are but a few examples of the problems caused by attempting to have IPsec provide support for multicast traffic. In addition, IKE is inherently a two-party protocol running over UDP. Entirely different key management methods are needed for multicast.

As of the writing of this guide, researchers have been attempting to find viable ways to extend IPsec so it can support multicast traffic without losing its methods of protection, particularly source authentication. [RFC 5374](#), *Multicast Extensions to the Security Architecture for the Internet Protocol*, discusses the issues and some approaches toward solutions. One of the biggest challenges is to find a solution that is not too resource intensive. Because multicast is typically used for applications such as streaming video that are constantly generating packets, IPsec cannot add too much overhead to the processing of each packet or the applications' functionality may be seriously impaired. Researchers expect that multiple multicast solutions may be created, each addressing a particular multicast need (e.g., single-sender multicast or multicast groups with a small number of members). It is outside the scope of this document to examine the proposed methods. Detailed information is available from research efforts that have been seeking solutions for multicast security issues, including the concluded Group Security Research Group (GSEC) within the Internet Research Task Force (IRTF)⁴⁰ and the Multicast Security (MSEC) Working

³⁹ If the hosts share the same secret key, and one host should no longer have access to the multicast traffic, then the secret key needs to be updated on all the remaining hosts in a timely manner. Distributing the new key in a secure manner to all these hosts may be extremely challenging.

⁴⁰ IRTF [home page](#); GSEC Research Group [Web page](#).

Group within the IETF.⁴¹

5.3.4 Status of IPsec and On-Going Work

The third version of IPsec was published in December 2005. This section summarizes the capabilities in these specifications; so that users can identify features they need and check that vendors supply them. Not all implementations are complete. ICSA Labs (www.icsalabs.com) holds periodic IPsec interoperability tests broken down by features, and their Web site can be consulted along with vendors' documentation.

- [RFC 4301](#), *Security Architecture for the Internet Protocol*. This specifies general requirements for IPsec. It contains a new processing model that eliminates SA bundles and instead processes SAs iteratively. One important IPv6 feature is support for selectors for the different types of ICMPv6 messages. It also adds support for combined integrity and confidentiality algorithms and a Peer Authorization Database (PAD) to specify acceptable authentication credentials. Finally, it downgrades the requirement for implementing AH from MUST to MAY.
- [RFC 4302](#), *IP Authentication Header*. This defines AH. The main new feature is extended, 64-bit replay detection counters.
- [RFC 4303](#), *IP Encapsulating Security Payload (ESP)*. This defines ESP. It allows longer padding to inhibit traffic analysis, combined mode algorithms, and also the extended sequence numbers used in AH.
- [RFC 4835](#), *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*. This specifies the cryptographic algorithms to be used with ESP and AH, shown in Figure 5-4. The encryption algorithms are only applicable to ESP. MUST- means an algorithm may be downgraded and SHOULD+ means potentially upgraded in the future.

Requirement	Encryption Algorithm	
-----	-----	
MUST	NULL	
MUST-	TripleDES-CBC	[RFC2451]
MUST	AES-CBC with 128-bit keys	[RFC3602]
SHOULD	AES-CTR	[RFC3686]
SHOULD NOT	DES-CBC	[RFC2405]
Requirement	Authentication Algorithm	
-----	-----	
MUST	HMAC-SHA1-96	[RFC2404]
SHOULD+	AES-XCBC-MAC-96	[RFC3566]

Figure 5-4. Encryption and Authentication Algorithms for the IPsec Protocol

- [RFC 4306](#), *Internet Key Exchange (IKEv2) Protocol*. This is the replacement for IKEv1. It removes many of the less used features and contains many simplifications and improvements. It does away with phases and modes and takes four messages to set up SAs initially. It has

⁴¹ IETF [home page](#); MSEC Working Group [Web page](#).

improved security, diagnostics, documentation, reliability, and performance. It also is integrated with items that were added only later to IKEv1: DHCPv6, NAT traversal, explicit congestion notification (ECN), extended sequence numbers (ESN) and extensible authentication protocol (EAP). Clarifications to IKEv2 were later published in [RFC 4718](#); [RFC 5996](#) consolidates and updates the two IKEv2 RFCs.

- [RFC 4307](#), *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*. This RFC specifies the cryptographic algorithms for IKEv2, simplifying document maintenance. Figure 5-5 shows the REQUIRED and RECOMMENDED methods. These are the algorithms that IKEv2 uses to protect its own traffic. In addition, IKEv2 has the capability to negotiate the algorithms listed in Figure 5-4 for use in ESP and AH.

Diffie-Hellman Groups			
Group Number	Bit Length	Defined In	Status
2	1024 MODP Group	[RFC2409]	MUST-
14	2048 MODP Group	[RFC3526]	SHOULD+
Encryption			
Name	Number	Defined In	Status
ENCR_3DES	3	[RFC2451]	MUST-
ENCR_AES_CBC	12	[RFC3602]	SHOULD+
Pseudo-Random Function			
Name	Number	Defined In	Status
PRF_HMAC_SHA1	2	[RFC2104]	MUST
PRF_AES128_CBC	4	[RFC4434]	SHOULD+
Message Integrity			
Name	Number	Defined In	Status
AUTH_HMAC_SHA1_96	2	[RFC2404]	MUST

Figure 5-5. Cryptographic Algorithms for Use in IKEv2

- [RFC 4308](#), *Cryptographic Suites for IPsec*. After a long debate about whether individual cryptographic algorithms or entire suites should be defined with IKEv2 and IPsec, the decision was to use the former but provide specific guidance for the latter. This RFC provides such guidance by defining two suites, one oriented towards triply iterated Data Encryption Standard (3DES) and SHA-1, a second oriented towards the Advanced Encryption Standard (AES) in Cipher Block Chaining (CBC) Mode. Additional suites ([RFC 4869](#)) oriented towards elliptic curves and AES in Galois/Counter Mode have emerged more recently.

Work on IPsec standards has continued in several areas, and the following RFCs, categorized by subject, have been published:

Cryptographic methods

- [RFC 2404](#) *The Use of HMAC_SHA-1-96 within ESP and AH*

- [RFC 2410](#) *The NULL Encryption Algorithm and its use with IPsec*
- [RFC 2451](#) *The ESP CBC-Mode Cipher Algorithms*
- [RFC 3526](#) *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)*
- [RFC 3566](#) *The AES-XCBC-MAC-96 Algorithm and its use with IPsec*
- [RFC 3602](#) *The AES-CBC Cipher Algorithm and its use with IPsec*
- [RFC 3686](#) *Using Advanced Encryption Standard (AES) Counter Mode with IPsec Encapsulating Security Payload (ESP)*
- [RFC 4106](#) *The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)*
- [RFC 4309](#) *Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)*
- [RFC 4359](#) *The Use of RSA/SHA-1 Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH)*
- [RFC 4434](#) *The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange (IKE) Protocol*
- [RFC 4543](#) *The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH*
- [RFC 5903](#) *Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2*
- [RFC 4754](#) *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*
- [RFC 4868](#) *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*
- [RFC 4869](#) *Suite B Cryptographic Suites for IPsec*
- [RFC 4894](#) *Use of Hash Algorithms in Internet Key Exchange (IKE) and IPsec*
- [RFC 5114](#) *Additional Diffie-Hellman Groups for Use with IETF Standards*
- [RFC 5282](#) *Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol*
- [RFC 5930](#) *Using Advanced Encryption Standard Counter Mode (AES-CTR) with the Internet Key Exchange version 2 (IKEv2) Protocol*

Three main ideas dominate the work on new cryptographic methods for IPsec:

- Strengthen or find alternatives to hash-based Message Authentication Codes (MACs)

In 2005, vulnerabilities were found to exist in SHA-1. NIST's resulting policy⁴² on the use of hash functions recommends the use of alternative hash functions for digital signatures and other applications that require collision resistance; it requires the use of the SHA-2 family of hash

⁴² NIST, [Policy on Hash Functions](#). NIST is also conducting a competition to define SHA-3, the successor to SHA-2.

functions (SHA-224, SHA-256, SHA-384 and SHA-512) instead of SHA-1 after 2010 for those applications. The use of SHA-1 for key derivation and random number generation is permitted beyond 2010.

Since the SHA-1 vulnerabilities do not affect HMAC-SHA-1, its continued use is permissible. IPsec and IKE use HMAC-SHA-1, and not SHA-1, for integrity protection.

- Use combined encryption-integrity modes for high-speed implementations
- Provide elliptic curve alternatives to public key cryptography based on modular arithmetic in IKEv1 and IKEv2.

Of these, the following are likely to be the most important:

- [RFC 4309](#) specifies the first combined mode encryption and integrity algorithm based on AES in Counter Mode. More recently, a more efficient construction called Galois Counter Mode has become an alternative choice. [RFC 4543](#) describes its use for integrity protection in ESP and AH; [RFC 4106](#) defines its use within ESP as a combined mode algorithm.
- [RFC 4868](#) (HMAC-SHA-256), which is likely to be included in mainstream implementations
- [RFCs 5903](#), [4754](#), and [4869](#), which are of interest to those who want to use IKEv2 with elliptic curve public key cryptography.

Government agencies are required to use only those cryptographic algorithms that are also defined in a FIPS (Federal Information Processing Standard). Most of the mainstream algorithms defined for use within IPsec and IKE satisfy this requirement. The exception is AES-XCBC (both the MAC [Message Authentication Code] and the PRF [Pseudo-Random Function]), which is therefore not approved for government use.

IKE and key management

- [RFC 4430](#) *Kerberized Internet Negotiation of Keys (KINK)*
- [RFC 4718](#) *IKEv2 Clarifications and Implementation Guidelines*
- [RFC 3706](#) *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*
- [RFC 3947](#) *Negotiation of NAT-Traversal in the IKE*
- [RFC 4304](#) *Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)*

[RFCs 3706](#), [3947](#) and [4304](#) define additional features for IKEv1; for IKEv2, these features are included in [RFC 4306](#) and its replacement, [RFC 5996](#).

Applications (routing, mobility, IPv6 over IPv4 tunneling)

- [RFC 4552](#) *Authentication/Confidentiality for OSPFv3*
- [RFC 4555](#) *IKEv2 Mobility and Multihoming Protocol (MOBIKE)*

- [RFC 4621](#) *Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol*
- [RFC 4877](#) *Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture*
- [RFC 4891](#) *Using IPsec to Secure IPv6-in-IPv4 Tunnels*

The three applications all bear directly on IPsec with IPv6. For OSPF on IPv6 networks (which is called v3), integrity is specified, ESP is REQUIRED, and AH is OPTIONAL. IKEv2 Mobility and Multihoming Protocol (MOBIKE) defines an efficient way to move IPsec SAs when an address changes, which happens much more often in IPv6 because of mobility, multihoming, or renumbering. [RFC 4877](#) brings IPv6 mobility up to date with the [RFC 4301](#)-series version of IPsec. Finally, [RFC 4891](#) answers the question of what protocol stack and type of tunneling works best when IPv6 is tunneled through an IPv4 network, and IPsec is applied at the tunnel endpoints. Using Protocol 41 and ESP in transport mode is the easiest configuration to achieve this goal.

Management of security

- [RFC 4807](#) *IPsec Security Policy Database Configuration MIB*

PKI (Public Key Infrastructure) for IPsec and IKE

- [RFC 4945](#) *The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX*
- [RFC 4809](#) *Requirements for an IPsec Certificate Management Profile*

The use of Public Key Certificates, and the interpretation of their contents, has been a source of interoperability problems within IPsec and IKE. These RFCs attempt to pin down some of the details and enhance interoperability. However, they are not specific enough to eradicate these problems.

Weaker authentication and trust models

- [RFC 5387](#) *Problem and Applicability Statement for Better Than Nothing Security (BTNS)*
- [RFC 5386](#) *Better-Than-Nothing-Security: An Unauthenticated Mode of IPsec*

Recognizing that the lack of a global PKI for peer authentication is a major hurdle to the universal deployment of IPsec, the BTNS (Better Than Nothing Security) Working Group defined a method of using IPsec to protect communications, but allowing lesser methods of authentication (e.g., self-signed certificates or raw public keys). This provides protection to the communications themselves, but does not guarantee the identities of the peers. This approach could be used to protect communications for applications that provide their own peer authentication.

The Applicability of IPsec

- [RFC 5406](#) *Guidelines for Specifying the Use of IPsec Version 2*

This contains useful advice for thinking through how IPsec can be applied, but the current version is not up to date with the most recent IPsec RFCs.

IP Compression (IPComp)

- [RFC 3173](#) *IP Payload Compression (IPComp)*
- [RFC 2394](#) *IP Payload Compression Using DEFLATE*
- [RFC 2395](#) *IP Payload Compression Using LZS*

IP payload compression is especially useful when IPsec based encryption is applied to IP datagrams. Encrypting the IP datagram causes the data to be random in nature, rendering compression at lower protocol layers ineffective. If IKE is used to negotiate compression in conjunction with IPsec, compression can be performed prior to encryption.

APIs (Application Program Interfaces) for IPsec

- [RFC 5660](#) *IPsec Channels: Connection Latching*

A number of different API Internet Drafts have been proposed, but have not progressed to standards. One issue that needs to be resolved is how application program security policy and system security policy interact and can be aligned with each other.

IPsecME (IPsec Maintenance and Extensions)

As IPsec and IKE have been deployed, operational issues, feature gaps and additional requirements have surfaced. The IPsecME working group was formed to address several of these issues, including:

- Combining the two IKEv2 RFCs ([RFCs 4306](#) and [4718](#)) into a single RFC, which includes other clarifications and corrections, without introducing changes to the underlying protocol. [RFC 5996](#) replaces the two older IKEv2 RFCs.
- Extending IKEv2 to allow IPv6 remote clients to negotiate all required parameters needed to conduct IPv6-based IPsec communications.
- Adding two new capabilities to IKEv2 that will improve performance for remote clients: allow a client to re-authenticate to a gateway in a streamlined manner, and permit a gateway to re-direct a remote client to another gateway.
- Enabling IKEv2 peers to quickly detect the loss of IKEv2/IPsec SAs.
- Adding two new IKEv2 authentication methods: mutual EAP-based authentication and mutual authentication based on “weak” shared secrets.
- Describing potential issues and requirements to enable gateway clusters to establish and use IKEv2 and IPsec SAs

- Facilitating the ability of middleboxes (firewalls, IDS/IPS systems, etc.) to distinguish unencrypted ESP packets (referred to as ESP-NULL) from encrypted ones. For networks whose policy dictates that end-to-end encrypted packets must be dropped, this will allow high-speed processing and inspection of the ESP-NULL packets. Two approaches have been defined: a heuristics-based approach, requiring no alteration to the communicating peers; and the addition of a new protocol number for ESP-NULL, a more straightforward approach, but one that does require changes to the base ESP protocol. The latter approach, called Wrapped ESP (WESP), can be used with or without encryption
- Writing an IPsec roadmap document to help novices navigate the interconnected and ever-increasing maze of IPsec- and IKE-related documents.

These capabilities have been addressed IPsecME Working Group documents. Most have already progressed to RFCs; the others are also expected to progress to RFCs:

- [RFC 5996](#) *Internet Key Exchange Protocol: IKEv2*
- [RFC 6071](#) *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*
- [RFC 5739](#) *IPv6 Configuration in IKEv2*
- [RFC 5685](#) *Re-direct mechanism for IKEv2*
- [RFC 5723](#) *IKEv2 Session Resumption*
- [RFC 5840](#) *Wrapped ESP for Traffic Visibility*
- [RFC 5879](#) *Heuristics for Detecting ESP-NULL packets*
- [RFC 5998](#) *An Extension for EAP-Only Authentication in IKEv2*
- [RFC 6027](#) *IPsec Cluster Problem Statement*
- *A Quick Crash Detection Method for IKE*⁴³
- *Protocol Support for High Availability of IKEv2/IPsec*⁴⁴

5.3.5 Security Ramifications

IPsec has a potentially significant impact on any network's performance. Security policy enforcement, security association lookup, cryptographic operations, and key management are all necessary components of any network employing IPsec. In some studies, it was shown that a security gateway implementing IPsec required more resources for managing and looking up security associations than for performing cryptographic operations. The proper implementation and configuration of these components is, therefore, critical to network performance.

IPsec can use IKE for entity authentication and key management based on pre-shared secret keys, but as the use of IPsec scales upward, this becomes unwieldy to deploy, keep secure, and update with fresh keys. To avoid this, public-key-based methods are recommended. A fully compliant X.509 PKI can be used, but this again involves expense and overhead—consider, for example, checking certificate revocation

⁴³ See the current version of the IETF Internet Draft, [A Quick Crash Detection Method for IKE](#), Work in Progress.

⁴⁴ See the current version of the IETF Internet Draft, [Protocol Support for High Availability of IKEv2/IPsec](#), Work in Progress.

lists. Therefore, lighter-weight approaches based on some simplifications (i.e., security shortcuts) in the trust model have been proposed.

IPsec is not the antidote for all security concerns with IPv6. It provides specific functionality such as cryptographic transforms to mitigate threats and vulnerabilities from the network layer up through the application layer. It does not replace other security functionality dealing with malware, spam, access controls, intrusion detection, and so forth.

IPsec may introduce a new security consideration insofar as it may thwart deep-packet inspection mechanisms between the two IPsec endpoints. Security tools such as intrusion detection systems and virus scanners cannot inspect encrypted packets. One viable approach in such cases is to distribute the intrusion detection services and virus scanners between network-based and host-based components.

Denial of service attacks and covert channels exist with IPsec, but it is not clear that these are any worse than those using IP without IPsec. Also, IPsec can protect against certain TCP-level denial of service attacks like SYN floods, whereas session layer security protocols such as SSH and SSL/TLS cannot.

The biggest security problem with IPsec seems to be faulty choices of security services or parameters. For example, vendor documentation may recommend using IPsec encryption without authentication in a way that is not consistent with up-to-date security guidelines.

5.3.6 Unknown Aspects

Although cryptographic security reduces the risks of networking, it cannot totally eliminate them. For example, a VPN implementation may have flaws in algorithms or software, it may be set up with insecure configuration settings and values, or it may be subject to denial of service attacks.

One significant concern with IPsec in IPv6 has been incomplete vendor implementations of the protocol, which may result in end-to-end incompatibility. This situation is gradually improving.

A debate exists over whether to use AH or ESP with NULL encryption for IPsec packets requiring integrity protection but not confidentiality. The standards themselves do not answer the question: IPv6 standards tend to recommend or require AH, whereas IPsec standards have downgraded AH from MUST implement to MAY implement. OSPFv3 originally specified AH, but this has been changed to ESP-NULL. Some implementations only provide ESP. AH has not been widely deployed. (The common IPv4 applications of IPsec—VPNs and secure remote access—usually provide confidentiality, so this question does not arise.) AH may also require more processing steps to compute or verify the integrity check value. On the other hand, one of the main arguments in favor of AH is that it makes it easier for devices such as packet filtering routers, firewalls, and intrusion detection systems to recognize plaintext and examine packets (because ESP-NULL transmits plaintext but gives no visible indication that the payload is unencrypted). As far as the cryptographic protection provided by AH versus ESP, it is difficult to discern any tangible difference when IPsec is used as intended. In some cases, AH may protect certain vulnerable parts of the header or extension headers, but these cases are rare, and in these few cases, ESP-NULL can achieve the same effect by being run in tunnel mode.

As IPv6 applications attempt to use IPsec for end-to-end security, more work will be needed on security APIs.

IPsec also relies on some security infrastructure outside of the IPsec protocols for authentication. This can be based on shared secret keys, locally generated certificates, or an external public key infrastructure.

Finally, some multicast issues, as noted above, remain open.

5.4 Secure Stateless Address Autoconfiguration and Neighbor Discovery

Security for autoconfiguration and neighbor discovery protocols has always been a difficult topic, because it is naturally problematical to have pre-existing trust relationships with entities one has not yet “discovered” or to use security before address configuration.

Attacks against address configuration and address resolution protocols include spoofing or hijacking addresses, denial of service, propagating inaccurate information, forging unreachability conditions, redirecting traffic, disabling routers, advertising bogus prefixes, misrepresenting network parameters, and others. Most are directed at compromising the availability of the local network, but some may also target confidentiality or integrity of information.

IPv4 networks have no means of stateless address autoconfiguration and perform neighbor discovery with the Address Resolution Protocol (ARP). ARP has no cryptographic security mechanisms, and it has always been subject to spoofing, cache poisoning, and denial-of-service attacks. Most installations cope with this by restricting ARP to local links (it is inherently a non-routable Layer 2 protocol) and policing the nodes on such local links. This has become increasingly difficult with wireless LANs. IPv6 neighbor ND is implemented with ICMPv6, and it is both possible and desirable to make sure such messages are not routed; since ND uses the “All Nodes Addresses” link local multicast group, by default these messages are not routed. It is also possible to apply cryptographic security to these messages.

The security requirements for autoconfiguration and ND vary according to the type of network used. For a more detailed discussion and examples of this, see [RFC 3756](#). For example:

- On a closed network within an enterprise, all users may trust each other to behave properly, and any malfunctioning nodes can be removed and corrected administratively. Cryptographically securing autoconfiguration and ND in this type of network may not be essential, but it could limit the damage that a corrupted (e.g. by malware) or malfunctioning node can do.
- In a college dormitory or on a network offering public access, for example, users depend on a trustworthy router to provide service, but they have little knowledge or control over what other users are doing. In this case, cryptographic authentication of autoconfiguration and ND represents a substantial improvement in robustness and security.
- In ad hoc networks, none of the nodes may start out trusting each other. In this case, it is likely that no basis for authentication exists, and there is no pre-existing key management infrastructure. Nevertheless, limited cryptographic protection can be provided.

IPv6 Autoconfiguration and ND use ICMPv6 messages to perform several different tasks. The main ICMPv6 message types are:

- NS (Neighbor Solicitation)
- NA (Neighbor Advertisement)
- RS (Router Solicitation)
- RA (Router Advertisement)
- Redirect

■ Router renumbering

This section defines and discusses two cryptographic security mechanisms for autoconfiguration and ND—IPsec and SEND.

Alternative methods for providing security to ND communications include: 802.1X⁴⁵, IEEE macsec⁴⁶, and an IETF layer 2 solution called RA-Guard⁴⁷.

5.4.1 Using IPsec to Secure Autoconfiguration and ND

IPsec can, in principle, be used to secure any IP packet, but deciding how IPsec should cover all of the different types and uses of ICMPv6 messages and actually specifying this behavior are not simple tasks. Certain ICMPv6 messages are used with mobility or multicast; others, such as the following, report error conditions or simply provide information:

- Destination unreachable
- Packet too big (also used to determine PMTU)
- Time exceeded
- Parameter problem
- Echo request
- Echo reply.

These are not used in autoconfiguration or ND; securing these messages is covered in [Section 3.5.2](#) and [3.5.3](#). This section covers:

- NS and NA messages used in stateless address autoconfiguration (SLAAC) and Duplicate Address Detection (DAD)
- RS and RA messages used for router and prefix discovery
- NS and NA messages used for address resolution and reachability detection
- Redirect and router renumbering messages sent by routers.

These messages use a variety of types of addressing. Most of them use a unicast source address, but autoconfiguration RS and NS messages also use the unspecified all-zero address (::). Both unicast and multicast destination addresses are used.

[RFCs 2461](#) and [2462](#) (which have now been replaced by [RFC 4861](#) and [RFC 4862](#), respectively) suggest that IPsec AH SAs be used to secure these messages. No specific attention was given to what SAs are needed, how they are established, or how they work (or do not work) with IPsec's key management protocols (IKEv1 and later IKEv2). Subsequent attempts to use this mechanism ran into problems:

⁴⁵ [IEEE 802.1X](#).

⁴⁶ [IEEE 802.1AE](#). See also *MACSec: A Security Solution that Protects LANS from Internal Threats*.

⁴⁷ See the current version of the IETF Internet Draft, [IPv6 Router Advertisement Guard](#), Work in Progress.

- It is impossible to use IKE without getting into an endless spiral. IKE uses UDP, which requires ND, which in turn uses the very same ICMPv6 messages IPsec was supposed to protect. Also, IKE does not work with multicast addresses.
- A large number of security associations needs to be pre-established.
- AH was downgraded to MAY in [RFC 4301](#) and is not included in many implementations.

Nevertheless, it is possible to use IPsec to protect autoconfiguration and ND if security associations can be set up beforehand. One approach that has been suggested is to use pre-established security associations from a known set of temporary IPv6 addresses to secure setting up longer-lived addresses and then new IPsec security associations for these addresses. This process needs to cover certain multicast addresses (All Nodes, Solicited Node for each unicast address, All Routers, etc.) as well as unicast. Although this has been described at a high level, little practical experience working out all of the details exists. The following limitations appear to be unavoidable:

- Manual keying must be used, and IKE cannot be used.
- All interfaces and their hardware (MAC) addresses must be known ahead of time.
- Replay detection must be turned off.
- A fixed set of IPsec parameters must be used without any negotiation.

5.4.2 Using SEND to Secure Autoconfiguration and ND

SEND was designed by operators of networks who needed to secure IPv6 autoconfiguration and ND but found IPsec to be an impractical choice. Many of these were mobile service operators incorporating IPv6 into third generation mobile handsets.

The core ideas behind SEND are:

- Use CGAs.
- Add an RSA Signature option to ICMPv6.
- Define *trust anchors* capable of attesting for public keys of routers.
- Add Nonce and Timestamp options to ICMPv6 for replay detection.

Nodes are configured with public keys of trust anchors, which let them verify signatures on RAs. By using CGAs and signatures, addresses cannot be spoofed or hijacked. Also, certain denial-of-service attacks are prevented. Replay attacks can be detected by examining timestamps on multicast messages and verifying that nonces are returned in two-way exchanges.

In ad hoc networks, using CGAs and signatures ensures that messages from the same address really came from the same entity, and no one else has stolen the address.

5.4.3 Current Status and Unknown Aspects

Cryptographic protection for autoconfiguration and ND has not been widely used, so it remains to be seen whether they will become parts of generally accepted practices for IPv6 security.

SEND also has not been widely implemented or deployed. Issues reportedly holding back some vendors

are intellectual property claims and licensing terms concerning CGAs.

6. IPv6 Deployment

This section suggests approaches to a successful and secure IPv6 deployment. Although this document uses the term deployment, rather than transition, many of the methods and mechanisms described in this section are referred to as *transition mechanisms*; to avoid confusion, this document also uses the word transition in that context. The overarching goal during the process of IPv6 deployment should be to maintain functional parity with existing networks and services. This section addresses that goal. IPv6 deployment focuses on integrating IPv6 with an existing IPv4 environment while maintaining, if not enhancing, the existing level of security. It includes the following subjects:

- Security risks
- Addressing security
- Transition mechanisms
- Transition planning

6.1 Security Risks

This section provides an overview of risks organizations may face when moving from an IPv4 to IPv4/IPv6 and eventually IPv6 environment.

The deployment of IPv6 is inevitable. The IPv4 address space is almost exhausted and the only long-term solution is to deploy IPv6. IPv6 is not backwards compatible with IPv4, which means organizations will have to change their network infrastructure and systems to deploy IPv6. Organizations should begin now to understand the risks and the risk mitigation strategies. Planning will enable an organization to make a smooth, secure and successful transition.

Some general risks an organization may face include:

- The attacker community's use of IPv6
- Unauthorized deployment of IPv6 on existing IPv4 production networks
- Vulnerabilities present in IPv6, including "day zero" vulnerabilities that are inherent in any new or revised system
- Complexity added by dual IPv4/IPv6 operations
- Immaturity of IPv6 security products and processes
- Possible lack of vendor support.

6.1.1 Attacker Community

The attacker community is adapting to the IPv6 environment, posing a serious risk to all organizations connected to the Internet. The attacker community leverages IPv6 to infiltrate and attack both IPv4 and IPv6 networks. The publication of a United States Computer Emergency Readiness Team (US-CERT) whitepaper⁴⁸ and the public release of IPv6-enabled attack tools⁴⁹ in May 2005 demonstrate how the

⁴⁸ US CERT, [IPv6 Malware Tunneling](#).

⁴⁹ One such tool is [THC-IPv6](#).

attacker community is attacking IPv6 implementations and using IPv6 to compromise systems.

The US-CERT whitepaper warned administrators that malware could use IPv6 for covert communications and tunnel traffic out of IPv4 networks. The US-CERT cited two causes in the whitepaper:

- Most current operating systems now support IPv6 by default.
- Firewall and IDS equipment not configured to recognize IPv6 traffic could be bypassed.

The warning that attackers could use IPv6 to tunnel traffic did not surprise the security community. As early as 2002⁵⁰, security researchers began documenting instances of attackers enabling IPv6 on compromised systems to evade IPv4 security controls.

Also in 2005, a security researcher released an IPv6 attack toolkit at a security conference. This toolkit includes tools to attack ICMPv6, detect IPv6 hosts, and hijack autoconfiguration.

These two events demonstrated that the attacker community is taking advantage of IPv6 and is developing techniques and tools for exploitation. If an organization's security controls are not already monitoring for IPv6 traffic, then IPv6 traffic to and from client devices may not be detectable by existing security controls. The mitigation for this risk is to reconfigure or deploy security controls to be both IPv4 and IPv6 aware.

6.1.2 Unauthorized IPv6 Clients

The Internet is actively deploying IPv6. IPv6 support is available for most operating systems. The commands to enable IPv6 on these operating systems are easily accessible and user-friendly. Using IPv6 autoconfiguration, a host can configure its own global address if it is able to find a prefix.

Recent operating systems not only support IPv6 but also often enable IPv6 by default. This exposes organizations to vulnerabilities it may not be able to detect or mitigate. The following options are available to reduce this risk:

- Locate and disable any IPv6 enabled equipment.
- Block IPv6 and IPv6 tunnel traffic at the perimeter.
- Include IPv6 use policies in the organization's security plan.

6.1.3 Vulnerabilities in IPv6

The IETF regards security as an important design constraint when developing standards for IPv6. Work to address a lack of confidentiality and integrity in IPv4 has resulted in the development of IPsec. IPv6 with IPsec resolves these issues, but IPsec has not addressed other weaknesses found in the TCP/IP protocol stack.

IPv6 does not solve many traditional layer 2 attacks like sniffing traffic, traffic flooding, man-in-the-middle attacks, rogue devices or Address Resolution Protocol (ARP) table overflow attacks. Some of these attacks are dealt with partially in IPv6 while other attacks, similar in nature, exploit different features.

Although most modern operating systems have supported IPv6 since at least 2003, the protocol stacks of

⁵⁰ Spitzner, [Honeypots: IPV6](#).

these operating systems have not been fully proven. A review of the National Vulnerabilities Database (NVD)⁵¹ shows stack vulnerabilities in most major operating systems. As new vulnerabilities are exposed, vendors will release updates. With any new code, vendors will need time to stabilize or harden their code. When deploying IPv6, an organization should understand that the code used in the IPv6 protocol stacks could be relatively new.

IPv4 Layer 2 and layer 3 attacks are possible because IPv4 assumes all network nodes will behave in a trustworthy manner. IPv4 uses ARP to associate physical addressing to logical addressing.

This assumption allowed attacks that interfered with IP address resolution and the association of physical addresses with logical addresses. As discussed in [Section 3.5.2](#), IPv6 does not use ARP to map IP addresses to physical interfaces. Instead, IPv6 uses ICMPv6. IPv6 uses ICMP for neighbor discovery and stateless address autoconfiguration processes that associate physical and logical addresses. IPv6 is still vulnerable to layer 3 attacks. A type of layer 3 attack is the *host initialization attack*. Three common host initialization attacks are:

- Neighbor Solicitation (NS). For neighbor discovery, a client node sends an NS message. The problem is that any node can claim to be any other node on the LAN. An attacker can falsify a NS message with their MAC address and the fake IP address. This technique is similar to ARP spoofing in IPv4 and only works on link-local.
- Router Solicitation (RS). With stateless address autoconfiguration, the client node sends an RS message. Like the NS attack, any node can claim to be the default router and subvert the initialization process. These attacks are link-local scoped.
- Duplicate Address Detection (DAD) Process. When a node attempts to configure its link-local address, it runs DAD. An attacker can cause a denial of service attack by responding to a DAD request. As a result, the victim's interface will fail to initialize.

The subnet size in IPv6 can present its own security challenges. These challenges are discussed in [RFC 5157](#), *IPv6 Implications for Network Scanning*. The subnet size is much larger than it was in IPv4; a default subnet can have 2⁶⁴ addresses. The issues with scanning an address space this large was discussed in [Section 2.4](#). Subnet size and resulting lengthy scan times are often presented as security features since they will slow down a malicious scanner or the propagation of malicious logic. The subnet size, however, makes it difficult for network and system administrators to manage assets. Attackers and administrators need to rely on other discovery techniques like layer 2 discovery and DNS to find hosts. The large address space makes it more difficult to find rogue hosts, possibly causing administrators to rely on passive discovery techniques. Administrators may need to resort to predictable or sequential numbering schemas for device addressing.

To mitigate most layer 3 vulnerabilities, administrators should consider using fixed addressing or DHCPv6 with DHCP rogue detection instead of stateless address autoconfiguration. Autoconfiguration and ND can be secured with SEND. More information on SEND is available in [Section 5.4.2](#). Layer 2 discovery, 802.1x based network access control, and good configuration management practices will help to mitigate layer 3 vulnerabilities.

The deployment of IPv6 reinforces the basic security lessons learned with IPv4. These security practices include defense in depth, diversity, patching, configuration management, access control, and system and network administrator best practices. Good security practices remain unchanged with the deployment of IPv6. Good security practices will reduce exposure and recovery time in case of a security event.

⁵¹ NIST, [NVD](#).

6.1.4 Dual Operations

An organization that deploys IPv6 may support IPv4 for legacy applications, services, and clients. This will result in a dual protocol environment and increased complexity. The risk is that an organization's security infrastructure may not be aware of both protocols. Dual protocols will increase the complexity of the environment. Two protocols mean twice as many things can go wrong, and twice the number of configurations are required to install new equipment or change existing equipment. Attacks against upper layer protocols could use either the IPv4 or IPv6 stack to reach the client. Administrators will need to maintain the same level of coverage for both protocols to mitigate risks.

6.1.5 Perceived Risk

A misconception about IPv6 is that IPv6 by itself introduces many more security risks than the IPv4 protocol. Perceived risk can cause an organization to delay deployment despite the fact that IPv6 enabled equipment is already on hand. As discussed in [Section 2.4](#) of this document, IPv6 is no more or less secure than IPv4. General security concepts are the same for both IP protocols. However, it will take time to acquire the level of operational experience and practical deployment solutions that have been developed for IPv4 over the years.

6.1.6 Vendor Support

Many security vendors are waiting for customer demand before implementing support for IPv6, while customers are waiting for vendors to support IPv6 before purchasing software and systems. This has resulted in a "chicken and egg" problem. While some vendors fully support IPv6, many do not support IPv6 at all or only offer limited support.

Organizations need to analyze their existing IPv4 security infrastructure and develop baseline requirements that an IPv6 infrastructure must meet to achieve the same or better security as their IPv4 environments. New base requirements may be added to address IPv6-only requirements or deficiencies in the existing environment. Some existing IPv4 equipment may support both IPv4 and IPv6 requirements, but organizations should plan to evaluate products with IPv6 support to supplement and/or replace existing equipment or bridge security gaps. Use the developed baseline security requirements to evaluate IPv6 security products. Product evaluation, selection and procurement can slow down the deployment schedule for IPv6. Organizations planning to deploy IPv6 should begin a dialog with their security vendors to support IPv6 as early as possible.

There are no industry standards that define IPv6 support. Organizations must understand they will be gaining and losing capabilities by deploying IPv6 and this will change their security posture. When vendors claim IPv6 support, it is up to the organization to evaluate whether they can function with the vendors' level of IPv6 support. Often IPv6 support is not as complete as IPv4 support. An example of this is that many IPv6 security devices require IPv4 addressing for management and configuration.

NIST has issued a special publication to define the capabilities IPv6 devices should support. NIST SP 500-267, *A Profile for IPv6 in the U. S. Government – Version 1.0*²¹ will help organizations identify and select products that support IPv6. This document has a section dedicated to Network Protection Devices that discusses the capabilities firewalls, application firewalls, and intrusion detection systems should support. When evaluating these types of security devices, organizations should ensure the equipment complies with the profile. To facilitate that goal, NIST has launched a USGv6 (US Government IPv6) test program⁵² to verify the IPv6 conformance and interoperability of three classes of IPv6 devices: hosts,

⁵² NIST [USGv6 test program](#) general information; information for users, [NIST SP 500-281](#), USGv6 Testing Program User's

routers and Network Protection Devices (NPDs). Testing is conducted by independent laboratories; the laboratories are accredited to perform the testing by ISO-certified accreditors. Vendors declare products that have qualified as USGv6-compliant through the use of a Supplier's Declaration of Conformity (SDOC), which enables potential users of the product to identify the capabilities that were tested and the laboratory that conducted the testing.

NIST SP 500-267 does not address several classes of security devices including security event/log management, vulnerability/patch management, flow (NetFlow, SFlow), forensic tools, and authentication systems. Many vendors are able to process IPv6 traffic but lack capabilities to present the information effectively, require IPv4 enabled equipment for management, or are not as feature-rich. Organizations will have to develop capability profiles for their security tools.

Organizations should accept the possibility that they may not achieve full parity in their IPv4 and IPv6 environments and take steps to mitigate associated risks.

6.2 Addressing Security

Administrators often see IP addressing as an operational issue. The numbering plan can also affect the organization's security posture. The addressing structure defines the fundamental organization and function of a network. Consider the following components in the IPv6 addressing plan to reduce threats to security and privacy:

- Numbering plan
- Hierarchical addressing to support security segmentation
- Security implications of EUI-64 addresses
- Address management
- Privacy extension.

These components are covered individually in the following sections.

6.2.1 Numbering Plan

The numbering plan describes how the organization subnets its IPv6 allocation. Most organizations would likely receive a /48 address block. This will allow the organization to support 65,000 subnets. Most of the RFCs concerned with numbering plans focus on address allocations above the site level, but the numbering plan designer should be familiar with the IETF's intent and philosophy of IPv6 addressing. Some RFCs of interest include:

- [RFC 3056](#) – Connection of IPv6 Domains via IPv4 Clouds
- [RFC 3879](#) – Deprecating Site Local Addresses
- [RFC 4007](#) – IPv6 Scoped Address Architecture
- [RFC 4193](#) – Unique Local IPv6 Unicast Addresses
- [RFC 4291](#) – IP Version 6 Addressing Architecture.

[RFC 5375](#), *IPv6 Unicast Address Assignment Considerations*, specifies best practices to follow when creating a numbering plan. Site allocations are normally provided as a /48, /56 or /64.

IPv6 Unicast Address Assignment Considerations addresses exceptions to this rule. It provides the following factors as considerations when designing a number plan:

- Prefix aggregation
- Network growth
- ULA (Unique Local Address) usage in large networks
- Sparse subnet numbering.

These design considerations are mostly operational concerns. A good numbering plan will streamline security operations by simplifying access control lists and firewall rule sets, identifying ownership of sites, links, and interfaces, and allowing for the rapid location of interfaces. *IPv6 Unicast Address Assignment Considerations* states that the lack of a numbering plan could slow down the deployment of IPv6.

6.2.2 Hierarchical Addressing to Support Security Segmentation

Network provisioning is different in IPv6 than in IPv4. With IPv4, the administrator is concerned with creating subnets that can support enough available IP addresses for current and future requirements. The more subnets created, the greater the number of lost addresses due to broadcast and subnet addresses. With IPv6, the addresses are so large that a subnet can support current and future addressing requirements. Because address space availability is not an issue with IPv6, the administrator can concentrate on creating a hierarchical address-numbering plan and not on address availability. A numbering plan should support network segmentation. Most networks are composed of different communities with differing requirements for access and protection. By segmenting a network, administrators can meet these various groups' requirements with access control rules to protect sensitive sites, links, and hosts.

IPv6 distinguishes itself from IPv4 by identifying not just a host by an address but also by identifying the regional provider based on the global hierarchy enforced by the RIRs. The network prefix of an address identifies an RIR and an ISP providing the address. IPv6 allows 16 subnet bits (/48) for the Site-Level Aggregation Identifier (SLA). Organizations can use these 16 bits to create a site hierarchy. Some suggested methods of subnetting include:

- Sequentially numbering subnets
- VLAN (Virtual LAN) number
- AS number
- IPv4 subnet number
- Physical location (building, city, county, etc.)
- Functional unit (Human Resources [HR], Operations, Finance, etc.).

The goal of address hierarchy design is to give an organization the ability to group assets logically and to simplify administration and security. Government agencies should be aware that switching from an environment in which NAT (Network Address Translation) provides IP addresses to unique global IPv6

addresses could trigger a change in the FISMA system boundaries.

6.2.3 Problems with EUI-64 Addresses

One of the more common ways to number interfaces is to use EUI-64 addresses. Using EUI-64 addressing can be a security risk. Autoconfiguration is one method that uses EUI-64 to generate an interface identifier (IID). Any machine that is trying to autoconfigure its own IP address will combine the network identifier with the EUI-64 address to create a unique address.

The physical address of the interface (MAC address) is an input to the algorithm that generates EUI-64 addresses. [RFC 4291](#) documents the methods for extracting the MAC address from the IPv6 address. Using a EUI-64 address could potentially reveal the make and model of a remote machine. An attacker could use that information to target attacks.

The use of the MAC address in EUI-64 addresses may make it easier for an attacker to scan a network. The MAC address is theoretically 48 bits in length, which yields a large number of host addresses (2^{48}). Address spaces of this size are difficult to scan. However, several factors reduce the effective size of this space. The first 24 bits of a MAC address are the Organizationally Unique Identifier (OUI), which identifies the hardware manufacturer. Two of these bits are reserved (for locally/globally administered addresses and unicast/multicast addresses), which still theoretically leaves 2^{22} possibilities. However, as of this date (December 2010), fewer than 15,000 OUIs have been assigned; of these, most current devices will only use a few hundred. Often manufacturers allocate a limited range for the second 24 bits to specific models of equipment. Evaluating specific OUI codes and bit ranges greatly reduces the number of EUI-64 addresses. Scanning a network based on the 24 unknown address bits is equivalent to scanning an IPv4 /8 network, which takes only a few seconds.

There are several strategies to mitigate the risk. The preferred mitigation strategy is to block scanning activity at the network perimeter and to limit inbound connections to only those hosts and protocols providing services beyond the gateway. Other methods include generating addresses using a cryptographic (i.e., non-predictable) algorithm or assigning addresses with DHCPv6.

6.2.4 Address Management

Address management for IPv6 has security implications. Various mechanisms are available for address management:

- Managed addresses
- Autoconfiguration
- Manual.

DHCP is widely used in IPv4 networks to allocate managed addresses. IPv4 administrators understand its configuration and use. IPv6 provides a similar mechanism called DHCPv6. DHCPv6 [RFC 3315](#) is not the same protocol as DHCP, but it does provide similar functionality. DHCPv6 is still susceptible to the same security vulnerabilities affecting DHCPv4. Using DHCPv6 has the following security advantages:

- DHCPv6 can limit smaller ranges of valid IPv6 addresses. A smaller range allows administrators to implement better access control rules.

- DHCPv6 allows administrators to identify managed clients more easily than autoconfigured or manually configured clients, and to follow their network usage through logging.

With autoconfiguration, the host machine automatically generates an IPv6 address on its own. Susceptibility to host initialization attacks poses one security risk. With autoconfiguration, generated addresses are in a wider range of valid addresses. Administrators have to create broader rules for access control lists if they are using range rules. This allows more addresses for rogue devices to connect to the network, and scanning the network is harder. Benefits of using autoconfiguration include:

- Addresses are deterministic. The same machine will always generate the same IP address. Administrators can pre-populate DNS and other logging systems with valid hosts.
- If administrators use host enumeration rules, tighter access controls are possible than with range rules.

With manual address management, administrators configure each node manually. This method is more resource intensive than the other two methods described in this section. Organizations often use manual address management with well-known services to make them easier to discover or manage.

Address management in IPv6 will probably look similar to methods of address management used for IPv4. After deploying IPv6, organizations should use a combination of methods for address management. Administrators should use manually configured addresses for servers. Client machines will use either DHCPv6 or autoconfiguration. The decision of which method to use (DHCPv6 or autoconfiguration) will depend on how administrators want to manage addressing. If administrators want to manage their addresses as pools of resources, they would opt for DHCPv6 address management. On the other hand, if administrators want to allow equipment to assign itself an IP address without additional network resources or administration, they would opt for autoconfiguration.

6.2.5 Privacy Extensions

People accessing the Internet are concerned about their privacy and having their IP addresses tracked when they are online. In response, the IETF released an RFC specifying the mechanism to create temporary addresses, thereby affording the end user a degree of anonymity when online. [RFC 4941](#), *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, specifies the mechanism for autoconfiguration to generate a temporary address that inhibits device or user tracking of the end users. Organizations that want to implement privacy extensions should consult [RFC 4941](#) for more information. The temporary interface ID is a randomly generated 64-bit value. Local policies control the regeneration schedule of temporary interface IDs.

Organizations generally should use privacy extensions for external communications but not for internal communications. Privacy extensions thwart correlation of a host with the use of a service. With internal IT systems, privacy extensions affect logging and prevent administrators from properly tracking which systems are accessing which services. Many internal resources require the ability to track the end user's use of services for correct operations. Organizations should document where and how it may intend to use privacy extensions in the numbering plan. This will allow an organization to evaluate how it will use privacy extensions and assess the security impact. Privacy extensions can be configured to reallocate addressing on a scheduled interval; this rotation policy should also be documented in the numbering plan.

6.3 Transition Mechanisms

IPv6 is not backwards compatible with IPv4, and IPv4 systems cannot use IPv6 services or communicate with IPv6 hosts. The transition from IPv4 to IPv6 is expected to take a significant amount of time. As

long as systems require interoperability between IPv4 and IPv6, transition mechanisms are needed. In the transition environment, three different types of hosts exist: IPv4 only, IPv6 only, and dual stack IPv4/IPv6. While dual stack is the recommended approach, there may be circumstances that require an organization to operate and communicate with single-stack systems.

Transition mechanisms support interoperability between IPv4 and IPv6 hosts. Multiple transition mechanisms may be deployed with any IPv6 transition. Client capabilities, the particular transition strategy chosen, time frame, and the transition stage all impact which transition mechanisms make the most sense. Transition mechanisms fall into three categories:

- Dual stack
- Tunneling
- Translation.

A typical transition will transform an organization's network from an all IPv4 environment to one where there are isolated IPv6 hosts or small islands of IPv6 hosts. The term *island* often appears in RFCs and refers to a small collection of like protocol hosts (either IPv4 or IPv6).

This example demonstrates how an organization could use the different transition mechanisms during this process:

- Early in a transition, IPv4 tunnels will connect IPv6 islands. Translation and dual stack mechanisms will allow IPv6 hosts to use IPv4 resources. Wherever possible, the use of translation mechanisms should be avoided, because of the complexities and incompatibilities associated with the translation of multiple protocols.
- As the transition progresses from IPv4 dominant to IPv6 dominant, the organization will configure the network core to either IPv6 or a dual stack. This will allow the organization to dispense with some of the tunnel mechanisms installed in the earlier stage of IPv6 transition. Translation mechanisms will be required to allow IPv4 hosts to use new IPv6 services.
- In the last phase of the migration, most equipment and services will support IPv6. Only isolated islands of IPv4 legacy services will remain. IPv4 traffic will tunnel over IPv6 and translation services will allow IPv6 clients to access legacy services.

6.4 Dual Stack IPv4/IPv6 Environments

In the dual stack method of IPv4 to IPv6 transition, each host is both IPv4 and IPv6 aware. Dual stack hosts run both IPv4 and IPv6 protocols and allocate addresses for both protocols. [RFC 4213, Basic IPv6 Transition Mechanisms](#), further explains the dual stack method. In most cases, IPv4/IPv6 dual stack deployments rely on tunneling and translation mechanisms for interoperability for parts of the network that are not dual stack.

The purpose of a dual stack method is to minimize the number of tunnels used in a transition. Organizations use dual stack when the majority of an organization's equipment is dual stack capable and they want a rapid deployment. In the following sections, we cover deployment, addressing, and security issues for a dual stack environment.

6.4.1 Deployment of a Dual Stack Environment

When considering a deployment of a dual stack environment, one must consider the following issues:

- Shared infrastructure
- Need for more resources
- Application protocol preference.

Logically, the IPv4 and IPv6 infrastructures are different. Dual stack devices require routing and switching infrastructures that are protocol aware. [RFC 4554](#), *Use of VLANs for IPv4-IPv6 Coexistence in Enterprise Networks*, describes how 802.1q tags can be used to divide a network logically into IPv4 routing and IPv6 routing domains.

Dual stack environments use more resources than a single protocol environment. The following are examples of how a dual stack environment uses more resources:

- Each protocol stack must share the available network bandwidth
- Routers need to:
 - Maintain forwarding tables for both IPv4 and IPv6
 - Run routing protocols for both protocols
 - Implement packet filtering for both protocols
 - Provide for congestion control for both protocols
 - Handle special cases (IPv4 Router Alerts and IPv6 Hop-by-Hop Options) for both
 - Forward packets for both protocols.
- Hosts must devote resources to both protocol stacks (for example, processing, memory, and network infrastructure traffic).
- Administrative and security staff must maintain concurrent environments as well.

Within a dual stack environment, some applications are IPv4 only, some are IPv6 only, and some applications may be IPv4/IPv6. The host must use the correct protocol to access each. The administrator using DNS record order or translation mechanisms can influence protocol selection.

Applications are written to query only A, only AAAA, or both A and AAAA records for name resolution. The administrator can influence the service called by ordering the records returned by DNS, giving precedence to the preferred service. Except for IP addressing, DNS is the same protocol for both IPv4 and IPv6. For example, to allow an IPv4 host to locate a service, create a DNS A record, and to allow an IPv6 host to locate a service, create a DNS AAAA record. Order the DNS records so that dual stack hosts are resolved to the preferred service. When configuring DNS in a transition environment, administrators should set the preference for IPv6.

6.4.2 Addressing in a Dual Stack Environment

Each protocol stack is responsible for configuring its own addresses. The administrator can configure static addresses or configure the host to receive dynamic addressing. If DHCP is used, then each protocol stack must access a DHCP server for address allocation. DHCP has different protocols for IPv4 and IPv6. Each protocol must access its own DHCP server.

6.4.3 Security Implications of a Dual Stack Environment

A dual stack strategy is useful in making a transition between protocols, but the approach exposes every dual stack node to the vulnerabilities of both protocols, plus any new vulnerabilities resulting from unintended interactions between them.

[RFC 4852](#), *IPv6 Enterprise Network Analysis - IP Layer 3 Focus*, contains sound general advice on securing dual stack systems. [RFC 4942](#), *IPv6 Transition/Coexistence Security Considerations*, includes many specific details:

- Organizations need to implement a consistent security policy for both IPv4 and IPv6 (including firewalls and packet filters).
- Organizations should account for new IPv6 functionality. This functionality may include mobility, stateless address autoconfiguration, neighbor discovery, privacy addresses, and end-to-end encryption with IPsec.
- Because both protocols are running, unexpected tunneling between the hosts may occur. The result may violate security policies.
- Organizations must upgrade intrusion detection or intrusion prevention systems, firewalls, monitoring, logging, and auditing to provide IPv6 protection equivalent to what was available for IPv4. If tunneled packets are allowed to enter the network, the firewall or IDS/IPS system must be able to perform deep packet inspection.
- The performance of security systems may degrade when handling IPv6 (when using the same resources compared to IPv4).

Good security practice dictates disabling unneeded services. Network administrators deploying IPv6 dual stack should configure nodes (hosts, servers, routers, etc.) to treat the IPv6 protocol as preferred and phase out remaining instances of the IPv4 protocol in a timely manner. When a given IPv4/IPv6 node no longer needs IPv4 services, administrators should disable the IPv4 protocol. Employing both protocols is useful during the early phases of IPv6 deployment, but the practice becomes a security risk because of increased complexity.

Administrators also need to watch for an unintended dual stack transition due to IPv6 being enabled prematurely. Security organizations should monitor for IPv6 traffic. Organizations should also audit router and neighbor solicitations to detect the insertion of rogue routers and devices on the network. Security and network administrators should have an incident response plan⁵³ in place for responding to violations of the configuration and security policies.

6.5 Tunneling

Tunneling is the encapsulation of one protocol inside of another. The tunneling protocol carries the tunneled protocol. The tunneled protocol is unaware of the tunnel and will not incur hop counts while in transit. This section provides an overview of tunneling either IPv6-over-IPv4 or IPv4-over-IPv6 packets.

Tunneling is a flexible transition mechanism and supports several scenarios including:

- As part of a dual stack IPv4/IPv6 transition strategy

⁵³ [NIST SP 800-61](#), *Computer Security Incident Handling Guide*,.

- As a stand-alone transition method
- Used together with protocol translation.

At a high level, tunnels are either *configured tunnels* or *automatic tunnels*. Configured tunnels require system administrators to configure the endpoints of the tunnel. With automatic tunnels, the nodes configure the endpoints themselves. Typically, an organization uses configured tunnels for infrastructure tunneling. Hosts use automatic tunnels to tunnel back to IPv4 or IPv6 islands. Configured tunnels are static in nature, and automatic tunnels are dynamic in nature.

Figure 6-1 shows a generic case in which two hosts on IPv6 networks can only reach each other through an IPv4 network. Each host can reach a dual stack IPv4/IPv6 router. This dual stack router in turn knows how to reach another IPv4/IPv6 dual stack router over an IPv4 network. The second router can forward the IPv6 packet successfully.

These routers are also set up to serve as tunnel endpoints. As tunnel endpoints, they encapsulate the IPv6 packet inside an IPv4 packet and forward it to the other tunnel endpoint. Then the tunnel endpoint decapsulates the IPv6 packet for further processing. The paths do not need to be symmetric. The end-point of the tunnel found in the left-to-right direction may be different from the starting point of the tunnel found in the right-to-left direction. From the IPv6 perspective, the IPv4 tunnel looks like a single hop.

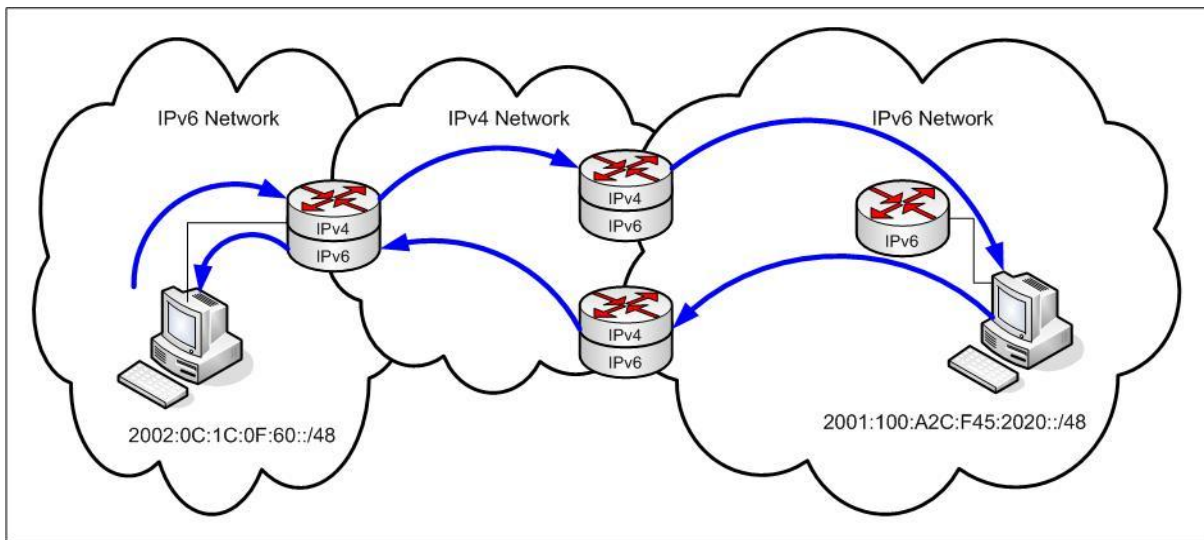


Figure 6-1. Example of Tunneling IPv6 over IPv4 Networks

Figure 6-1 is a simplified view of tunneling. It illustrates the general concept. There are, in fact, many different mechanisms for tunneling.

When an IPv4 header directly encapsulates IPv6, it uses protocol number 41. When an IPv6 header directly encapsulates IPv4, it uses Next Header 4. It is possible to use other encapsulation methods such as Generic Routing Encapsulation (GRE [[RFC 2784](#)]) or IPsec ESP ([RFC 4303](#)).

IPv6 packets can be up to four gigabytes in size. The maximum size for an IPv4 packet is 64k. Fragmentation may be required when an IPv6 packet is encapsulated in IPv4. Where possible, the IPv6 layer should discover the IPv4 path maximum transmission unit (MTU). This may not always be possible, since not all tunnels report real (reduced) MTUs. Using this information, it should create the most

efficient packets for transmission. The encapsulated protocol (IPv4 or IPv6) sees the tunnel as a single hop. The hop limit or TTL for the encapsulating protocol is independent of the TTL for the encapsulated protocol.

6.5.1 General Security Considerations for Tunneling

The following are security items to consider for tunneled environments:

- Tunnel endpoints
- Inspection
- Access control
- Termination.

Tunnel endpoints are always a focal point for security; attackers frequently target them in attacks. Potentially, traffic from anywhere can arrive at tunnel endpoints. Tunnels should be treated as an external link. The tunneled traffic should not be trusted without inspection. This requires examining IPv6 traffic within IPv4 packets and subjecting it to security controls at the same place, logically, that IPv4 is controlled. Packet filters, network ingress filtering, virus protection, application proxies, and intrusion detection systems need to apply the same security policy to what is inside tunneling protocols as they do to IPv4. Enforcing security policies at both the ingress and egress of the tunnel will protect the tunnel provider from attacks and protect hosts from traffic coming through the tunnel. Even if the traffic going through the tunnel is protected with end-to-end IPsec, additional security controls such as authorization should be applied at the tunnel endpoints.

In addition to filtering on tunnel endpoints it is important to use Unicast Reverse Path Forwarding (Unicast RPF) filtering on the tunnel endpoints ([RFC 3704](#)). An attacker can perform traffic injection or reflection on or through the tunnel endpoints, sending crafted packets toward a tunnel endpoint router that will then forward the packet toward the other tunnel destination. The far-end tunnel endpoint will assume that the packet is legitimate because it came out of the tunnel interface and forward it onward to the destination. This threat can be mitigated by using Unicast RPF and tunnel endpoint filtering at both tunnel endpoints. Another technique for protecting the tunnel endpoints is to use traffic ACLs to insure that only legitimate packets are going through the tunnel. There is also a risk that an attacker could be performing tunnel sniffing and eavesdropping on traffic going through the tunnel. This can be protected with the use of IPsec (IPv4 with GRE to protect IPv6 tunnel).

However, checking the IPv4 outer header doesn't necessarily mean the inner IPv6 packet is legitimate. Depending on the security device being used, it can be difficult to check/match the outer and the inner header. An attacker may even be using tunnels to obfuscate traffic inside because the attacker knows that many organizations lack deep packet inspection of IPv4/IPv6 packets.

Routers, firewalls, and security devices at the end of an enterprise network may not be technically capable of inspecting the IPv6 payload contained within IPv4 packets entering and exiting the network. In particular, security devices are unable to inspect encrypted tunnel traffic. By default, tunneling protocols do not encrypt traffic. Organizations should deploy security devices that can understand tunneled traffic or can inspect traffic once decapsulated.

The access control lists (ACL) on IPv4 routers may be incapable of blocking access to the network from specific IPv6 addresses or from entire blocks of IPv6 addresses. They may not be able to recognize IPv6

next header numbers, ICMPv6 message types, or port numbers for TCP or UDP over IPv6. The reason is that the IPv6 header and addresses are not visible to the router; they are part of the packet payload, which is the functional intent of any tunneling process.

Network administrators must understand that IPv6 tunneling may be taking place without their knowledge. Strategic deployment of network devices capable of detecting and filtering IPv6 traffic is essential. IPv4 can tunnel IPv6 traffic through security controls and violate normal access control policies. In effect, the IPv6 over IPv4 tunnel becomes a backdoor into the network. Figure 6-2 illustrates the relationship between IPv6 tunnels and the existing IPv4 network.

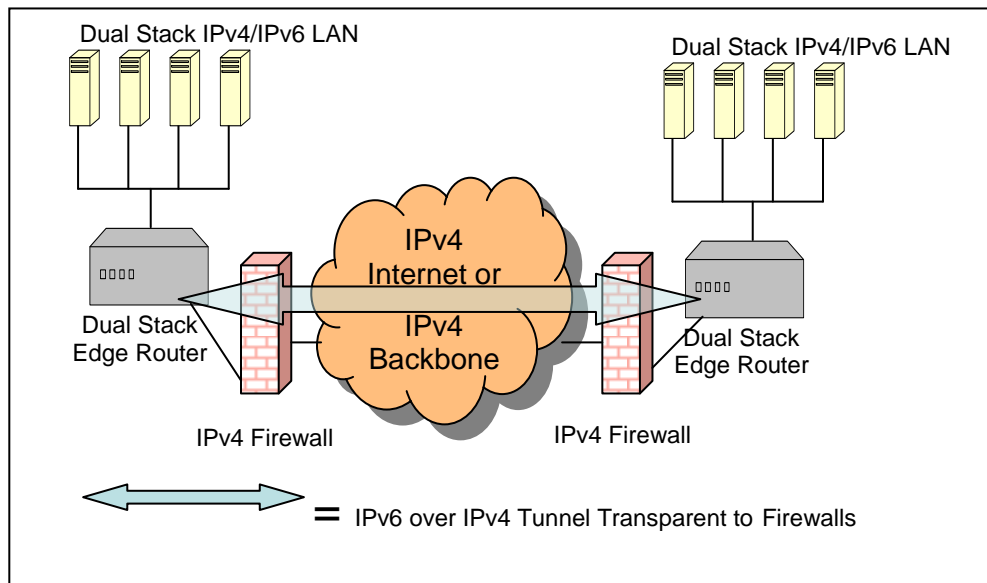


Figure 6-2. IPv6 over IPv4 Tunnels Transparent to the IPv4 Infrastructure

IPv6 over IPv4 tunneling protocols themselves are visible to edge devices such as routers and firewalls, even if those devices cannot apply security to encapsulated IPv6 traffic. For this reason, IPv4 access control lists and firewalls should block well-known tunneling protocols and ports or ensure that these protocols are destined to an IPv6-capable firewall. Of course, this raises the question of whether an attack could get *through* the IPv6 firewall with, for example, a type 0 routing header. A firewall which meets the USGv6 Profile ²¹ should be able to block type 0 routing headers. Many firewalls can block IPv4 protocol 41 (used in the 6over4, ISATAP [Intra-Site Automatic Tunnel Addressing Protocol], and 6to4 tunnel protocols). Network and security administrators should confirm that their infrastructure appropriately filters protocol 41 by verifying security device configurations and evaluating existing traffic.

Blocking protocol 41 with access control lists or firewalls does not prevent tunneling IPv6 over IPv4 UDP tunnels such as Teredo. It is hard to identify and filter all Teredo traffic. Teredo uses a well-known outbound UDP port 3544 to locate Teredo servers; simply blocking port UDP 3544 does not guarantee that Teredo traffic is blocked, because it is easy to change the port assignment. A more effective technique is to use an Intrusion Detection and Prevention System (IDPS) that can identify the protocols to block regardless of the port assignment.

Tunnels encapsulating IPv6 in SSL/TLS or IPsec are more problematic. Because the payload may be encrypted, no mechanism exists that can inspect the traffic. Organizations must possess the decryption keys on the inspection device or terminate tunnels in front of the security devices and evaluate plain text.

Better security for tunnels is available if both endpoints are under the same administrative control or have a trust relationship sufficient to set up cryptographic security. [RFC 4891](#), *Using IPsec to Secure IPv6-in-IPv4 Tunnels*, explains the easiest way to set up IPsec ESP to secure IPv6 over IPv4 encapsulation.

[RFC 4213](#), *Basic Transition Mechanisms for IPv6 Hosts and Routers*, recommends sending the same response as for “destination unreachable” when dropping a packet for security reasons. Otherwise, an attacker could probe for the existence of the tunnel endpoint in preparation for other attacks.

The following sections address specific deployment and security issues for various tunneling mechanisms. [Section 6.5.2](#) covers configured tunneling of IPv6 over IPv4. Automatic tunneling mechanisms in general are discussed in [Section 6.5.3](#), following by detailed discussions of various automatic tunneling mechanisms in [Sections 6.5.4](#) through [6.5.10](#).

6.5.2 Configured Tunneling

Configured tunnels establish connections before traffic using the tunnel arrives. Configured tunnels are set up and maintained by network administrators. Establishing remote endpoints outside the organization may require peering agreements and may not use optimal paths between peering sites. The general model for configured tunnels is to set them up and use them router-to-router for regular traffic between sites.

6.5.2.1 Using Configured Tunnels

Configured tunnels provide a straightforward way to use the IPv4 routing and forwarding infrastructure to carry IPv6 traffic to IPv6 islands. [RFC 4213](#) further describes configured tunnels. This document contains details about maintaining state information at the tunnel endpoints, constructing the IPv4 encapsulation, handling MTUs and fragmentation, running neighbor discovery through a tunnel, and dealing with ICMPv4 (Internet Control Message Protocol for IPv4) error messages.

For instance, in tunneling with IPv6 over IPv4 using protocol 41, the endpoint routers at both ends of a configured tunnel are provided with the four parameters defining the tunnel (IPv4 and IPv6 source and destination addresses of both endpoints). Commonly, IPv6 traffic on a link reaches a default router that serves as the endpoint of a configured tunnel and knows how to use IPv4 as the next hop to forward the IPv6 traffic. The tunnel will not work if protocol 41 is blocked (e.g., filtered by NAT devices). Configured tunnels can also tunnel IPv4 over IPv6. Tunnels may use GRE encapsulation or MPLS instead of protocol 41.

Using configured tunnels does not scale well, because of the need to set up and maintain each one manually. However, configured tunnels provide a much greater degree of control than using automatic tunneling.

6.5.2.2 Security Considerations for Configured Tunnels

General security considerations for tunneling are covered in [Section 6.5.1](#) and in [RFC 4942](#). The following points apply to security for configured tunnels:

- After decapsulation, packets with a multicast, loopback, IPv4 compatible or IPv4 mapped source address must be discarded. Apply normal ingress filtering, which may require manually maintaining IPv6 prefix lists.
- The tunnel endpoints must be configured to allow the encapsulated packet to get through any instances of security policy enforcement: packet filters, application layer gateways, intrusion detection or prevention systems, and so forth.

- Additional restrictions may be applied to using the tunnel. For example, the tunnel endpoints may accept encapsulated packets only from a list of approved senders.
- Tunnel endpoints handling multiple configured tunnels must treat each one as a separate interface. Treating each tunnel endpoint as a separate interface prevents violating scoping rules and isolates neighbor discovery traffic.
- If the encapsulated packet uses IPsec for integrity or confidentiality, little is gained by applying IPsec again for the same services through the tunnel, but if IPsec along the tunnel is desired, then the methods in [RFC 4891](#) may be used.

6.5.3 Automatic Tunneling

Due to the administrative overhead of maintaining configured tunnels, and because configured tunnels do not cover all transition scenarios, the IETF defined additional automatic tunneling mechanisms. As transition problems were encountered, the IETF created additional transition mechanisms to solve them.

The general automatic tunneling model consists of isolated hosts with limited communications needing dynamic tunneling capabilities. With automatic tunnels, one tunnel endpoint can find the other end without pre-configuration. One way for IPv6 over IPv4 tunnels to accomplish this is by embedding IPv4 addresses in IPv6 addresses. This may be a good method for jump-starting a transition to IPv6, but it is not recommended as a long-term solution. Besides the overhead involved with establishing the tunnels, this approach limits the use of new IPv6 capabilities, does not solve the IPv4 address space problem, and imports the IPv4 routing table into IPv6. Automatic tunneling is more likely needed in the earlier stages of IPv6 transition.

Deployment and security considerations for the following automatic tunneling mechanisms are discussed in subsequent sections:

- 6over4 - host to router or router to host
- 6to4 and 6rd - router to router
- ISATAP - intra-site tunnels
- Teredo - UDP encapsulation intended for tunneling through IPv4 NATs
- Tunnel brokers - using a server for automatic tunneling
- DSTM (Dual Stack Transition Mechanism)
- Carrier-Grade NAT (CGN) and Dual-Stack Lite

6.5.4 6over4 Protocol

The 6over4 protocol ([RFC 2529](#)) is a relatively old and simple transition mechanism that relies on IPv4 multicast as a virtual link layer. Also known as IPv4 multicast tunneling or *virtual Ethernet*, it is a host-to-host, host-to-router, and router-to-host automatic tunneling technology supplying unicast and multicast IPv6 connectivity between IPv6 nodes within an IPv4 site. It is not intended for connecting an isolated IPv6 user to the rest of the IPv6 Internet. Because of its dependence on IPv4 multicast, it has not been widely implemented or deployed.

6.5.4.1 Using the 6over4 Protocol

The 6over4 protocol assigns two IPv6 addresses per interface, a unicast address and a link-local address. Hosts use a valid 64-bit IPv6 prefix for unicast addresses and set their interface ID simply to their 32-bit IPv4 address (e.g., A.B.C.D). They also configure the link-local address FE80::A.B.C.D on each 6over4 interface.

The 6over4 protocol regards the IPv4 network as a link layer with multicast capability (like Ethernet). This means that neighbor discovery processes (such as address resolution and router discovery) work as they do over a physical link with multicast capabilities. The entire IPv4 infrastructure used for the tunneling must be multicast enabled.

6.5.4.2 Security Considerations for the 6over4 Protocol

Security considerations for the 6over4 protocol include:

- In addition to attacks against IPv6, attacks against IPv4 are also possible, so standard IPv4 security controls apply.
- Boundary routers need to apply normal ingress and egress filtering rules to the IPv4 addresses and filter incoming unicast IPv4 packets with protocol type 41 from unknown sources. Configure boundary routers to accept IPv6-in-IPv4 tunnels only from trusted sources. Boundary routers also need to reject multicast IPv4 packets with a more restricted scope than appropriate for the interface on which they arrive.
- Decapsulated IPv6 packets with a hop count of 255 must not be treated as locally generated packets.
- If IPsec is desired, it is better to run it in the IPv6 domain, end to end or nearly end to end. Little is gained (except perhaps some protection against traffic analysis) by adding IPsec in the IPv4 domain.

6.5.5 6to4 and 6rd Protocols

The 6to4 mechanism is designed to provide IPv6 site-to-site and site-to-existing-IPv6-network connectivity across an IPv4 network by embedding IPv4 addresses in IPv6 prefixes. The 6to4 protocol is intended to be a start-up technique for providing IPv6 connectivity. One example is when an ISP does not provide an IPv6 prefix. The 6to4 protocol is appropriate with an IPv6-dominant transition strategy without a global IPv6 prefix. It is not a permanent solution. More information on the 6to4 protocol is available in [RFC 3056](#), *Connection of IPv6 Domains via IPv4 Clouds*.

The 6rd mechanism is designed to allow IPv4 service providers to offer IPv6 to their customers with minimum delay and expense. The name “6rd” stands for “IPv6 rapid deployment.” Its main idea is to use the same protocol mechanisms as 6to4 but with the service provider’s IPv6 prefix. Two IETF documents on 6rd exist: (1) *IPv6 Rapid Deployment on IPv4 infrastructures (6rd)* describes the motivation for and user experience with 6rd ([RFC 5569](#)); and (2) *IPv6 via IPv4 Service Provider Networks* ([RFC 5969](#)) defines the protocol. The advantages of 6rd are that it automatically delegates IPv6 prefixes to customers’ sites, operates statelessly, allows customers to use native IPv6, and can be provisioned with a few simple steps. The problem 6rd solves that 6to4 does not is route discovery. A packet originating from any native IPv6 address needs to traverse, somewhere, a relay router to obtain IPv6 over IPv4 encapsulation. Using

the service provider's IPv6 prefix provides such a route. In all other respects, except as noted below, 6to4 and 6rd work similarly.

6.5.5.1 Using 6to4 and 6rd

Each 6to4 border router must have a globally unique IPv4 address (e.g., w.x.y.z.). The IPv6 network connected to that router uses the IPv6 prefix 2002:w.x.y.z/48⁵⁴. An entire /48 IPv6 prefix can be routed through one IPv4 address. 6to4 IPv6 sites can automatically connect to each other over the IPv4 Internet. This prefix can be advertised to the IPv6 network or sub-allocated as desired. IPv6 hosts can use autoconfiguration to generate interface IDs for 6to4 prefixes. The 6to4 border routers tunnel IPv6 over IPv4 using protocol 41.

The IPv6 destination does not need to have a 6to4 prefix. It may be any globally unique IPv6 address. Similarly, IPv6 packets can be sent from any IPv6 address to a 6to4 address. Relays, which are IPv6 routers with at least one 6to4 address and one native IPv6 address, are used to connect 6to4 domains with the native IPv6 Internet. Each 6to4 domain must have at least one relay router, which is normally reached with an (IPv4) anycast address ([RFC 3068](#), *An Anycast Prefix for 6to4 Relay Routers*).

The 6to4 mechanism has been widely implemented and is in use in both production and trial networks. Sites that have globally routable IPv6 prefixes should avoid using the 6to4 protocol. Note that 6to4 sites require a globally unique IPv4 address. The 6to4 and 6rd protocols presume that IPv4 is still being used and do not solve the IPv4 space exhaustion problem. Relay routers that serve a large number of 6to4 routers have to import that part of the IPv4 forwarding table into IPv6.

The 6to4 and 6rd mechanisms can work on networks using IPv4 NAT as long as these two conditions are met:

- The NAT boxes are fully functional IPv6 routers
- The globally unique IPv4 address of the outermost NAT box is used to construct and advertise the IPv6 6to4 prefix. (For 6rd, this condition is not necessary.)

6.5.5.2 Security Considerations for 6to4 and 6rd

In addition to the general security advice for tunneling mechanisms, [RFC 3056](#) contains an overview of 6to4 security. [RFC 3964](#), *Security Considerations for 6to4*, examines many 6to4 security concerns more thoroughly. In particular, it considers the following cases:

- 6to4 routers not being able to identify whether relays are legitimate
- Wrong or partially implemented 6to4 protocol on the router or relay security checks
- The 6to4 architecture being used to participate in DOS or reflected DOS attacks or made to participate in “packet laundering,” i.e., making another attack harder to trace
- The 6to4 relays being subject to administrative abuse, e.g., theft of service, or being seen as a source of abuse.

Complete solutions to all of these abuses do not exist, but basic guidelines for using 6to4 effectively

⁵⁴ For 6rd, the IPv4 address may be globally unique or a non-routable RFC 1918 address used only within the service provider's network. The exact layout and length of the 6rd IPv6 prefix varies according to the length of the service provider's IPv6 prefix and other tradeoffs.

follow:

- If IPsec is desired, it is better to run it in the IPv6 domain, either end to end or nearly end to end. Little is gained (except perhaps some protection against traffic analysis) by adding IPsec in the IPv4 domain.
- If IPv6 source address spoofing is a problem, check that the IPv4 and encapsulated IPv6 prefixes agree with the necessary exceptions for relay routers.
- Many address spoofing and denial of service attacks stem from the requirements that 6to4 routers must accept and decapsulate IPv4 packets from any other 6to4 router or relay, and 6to4 relays must accept traffic from any IPv6 node. Administrators should apply normal ingress and egress filtering rules to IPv4 addresses.
- Any 6to4 prefix must correspond to a globally routable unicast IPv4 address. The IPv4 address must not be private, loopback, unspecified, multicast, broadcast, DHCP link local, or a reserved IPv4 address.
- The IPv6 address must not be IPv4 compatible, IPv4 mapped, loopback, unspecified, link local, site local, or multicast.

One risk posed by 6to4 is that of internal hosts using 6to4 to reach the Internet and bypassing a corporate firewall. An attacker could deploy a rogue 6to4 relay router using the anycast IP address 192.88.99.1 in order to insert these routers along the traffic path and perform man-in-the-middle attacks. 6to4 lacks any authentication mechanism. Protection mechanisms for 6to4 include using Unicast RPF on the 6to4 relay router itself and applying ACLs to the 6to4 relay router's tunnel interface. ACLs could be created to completely block all 2002::/16 packets.

Many of the threats against 6to4 do not apply to 6rd, because the IPv6 over IPv4 tunnels exist only within a single service provider. First, the customer's end of a 6rd tunnel only needs to accept packets from a single or small set of known 6rd relay routers. Second, the tunnel endpoint within the service provider's network can ensure that tunneled traffic arriving at its IPv4 address comes from a customer and not an outsider. Third, IPv6 packets can be protected against source address forgery to the same extent that IPv4 packets are within a service provider's network.

6.5.6 Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

ISATAP allows isolated IPv6 hosts within a site running IPv4 to construct an automatic IPv6-in-IPv4 tunnel. ISATAP uses IPv6 without depending on IPv4 multicast, as required with 6over4, but ISATAP requires more setup work than 6over4. All hosts using ISATAP must be dual stack IPv4/IPv6. The network may only route IPv4. Most dual stack IPv4/IPv6 operating systems implement ISATAP. More information about ISATAP is available in [RFC 5214](#), *Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*.

ISATAP is a useful mechanism promoting early deployment of IPv6 within a site with an IPv6 prefix. Consider ISATAP a bootstrap mechanism for connecting IPv6 hosts as the other parts of the network become IPv6 enabled. Remove ISATAP when there is no longer a need. ISATAP is not appropriate for IPv6 dominant networks. Once IPv6 is available in a dual stack environment, disable all tunneling of IPv6 over IPv4.

6.5.6.1 Using ISATAP

ISATAP hosts communicate by tunneling IPv6 packets over IPv4 using protocol 41. The IPv4 addresses

are encoded in the low-order bits of the IPv6 addresses, allowing automatic tunneling. Globally routable IPv4 addresses are not needed. If using a private IPv4 address to create the ISATAP address, then that IPv6 address cannot be used out of its IPv4 scope. The local IPv4 network is represented in IPv6 as a single subnet.

ISATAP assumes that some dual stack IPv4/IPv6 router exists and that it advertises an IPv6 prefix. A host with an IPv4 address w.x.y.z then performs autoconfiguration with interface ID = ::0:5EFE:w.x.y.z.

The example in Figure 6-3 shows:

- IPv4 address: 192.168.1.10
- Global IPv6 prefix: 2001:1C3:1:200F::/64
- Link-local address: FE80::5EFE:65:C0A8:010A = FE80::5EFE:65:192.168.1.10
- Global IPv6 address: 2001:1C3:1:200F::5EFE:C0A8:010A = 2001:1C3:1:200F::5EFE: 192.168.1.10.

To make this work, one needs to ensure that:

- All IPv6 hosts run dual stack IPv4/IPv6 with support for ISATAP.
- Each ISATAP host must know at least one dual stack IPv4/IPv6 router.
- All traffic is constrained to a single administrative domain.
- There is no need for IPv4 NAT traversal.

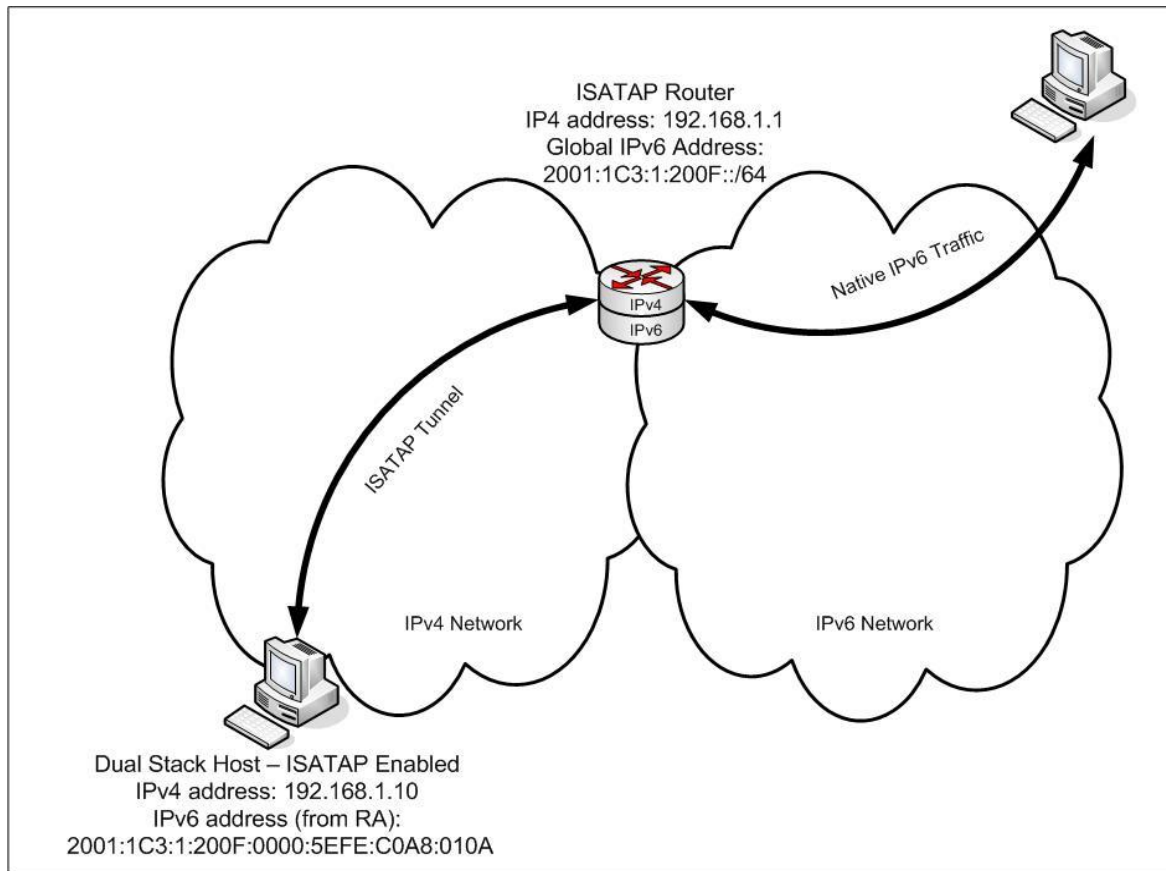


Figure 6-3. Example - Tunneling IPv6 over IPv4 Networks with ISATAP

6.5.6.2 Security Considerations for ISATAP

Security considerations for ISATAP include:

- ISATAP does not assume that multicast works. ISATAP foregoes IPv6 autoconfiguration and neighbor discovery protocols. It automatically knows the correspondence between the IPv6 and IPv4 (link layer) addresses. Administrators statically configure the address of the default router. Hence, many of the security considerations with autoconfiguration and neighbor discovery do not exist with ISATAP.
- The appropriate use of IPsec is the same as with 6to4. Implement IPsec in the IPv6 domain.
- Certain IPv6 addressing capabilities such as privacy addresses or CGA cannot be used.
- All of the usual protocol 41 concerns apply. Various spoofing attacks may try to take advantage of the protocol 41 tunneling mechanism. Firewalls need to perform ingress and egress filtering on either the encapsulating and encapsulated addresses or block protocol 41, as appropriate.
- ISATAP clients are vulnerable to an internal attack in which an imposter pretends to be an ISATAP router. The recommended defense is for administrators to ensure that the ISATAP potential routers list (PRL) is accurate and protected from tampering.

A hacker who can change the DNS A record for `isatap.example.com` to be the attacker's own IPv4 address would be able to communicate with all ISATAP-enabled computers on the network. The attacker would have created a rogue ISATAP relay router. Thus, even if an organization has not enabled ISATAP it might be useful to have a static DNS entry for `isatap.vzw.com` pointing to the loopback address `127.0.0.1`, thus disabling ISATAP.

Other risks include traffic injection into the tunnel interface on the relay router. This can be mitigated through the use of ACLs and Unicast RPF filtering on the ISATAP router. ISATAP packets use IP protocol 41, so if those packets are filtered at the perimeter this will prevent ISATAP packets from leaving the organization.

6.5.7 Teredo Protocol

The 6over4 mechanism requires IPv4 multicast. The 6to4 and 6rd protocols require public IPv4 addresses. Although ISATAP solves both of these problems, ISATAP does not work across NATs. Microsoft developed the Teredo protocol as a solution, submitting it to IETF. The IETF approved it as [RFC 4380](#), *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)*.

Teredo's functionality and usage scenarios are similar to 6to4, but 6to4 requires support from a router, which serves as the edge device connected to the Internet. IPv4 NAT devices do not usually support 6to4 router functionality. Even if the NAT device supported 6to4, the 6to4 protocol would still not work for configurations with multiple NATs between a site and the IPv4 Internet or for NATs that do not handle protocol 41 packets, since some NATs only handle TCP and UDP.

Teredo is a useful mechanism for promoting early deployment of IPv6 on hosts behind a NAT, prior to having an IPv6 network, in the case where an IPv6 prefix has been provided to the site. Production and pre-production networks have been using Teredo.

Because Teredo has more overhead than other methods, Teredo is considered a technology of last resort for IPv6 connectivity. The design of Teredo favors robustness over efficiency. If native IPv6, ISATAP, or 6to4 connectivity can be used, Teredo is not needed. The use of Teredo will decrease as more IPv4 NATs support 6to4 and IPv6 connectivity becomes more common.

Teredo provides address assignment and automatic tunneling for host-to-host unicast IPv6 traffic across an IPv4 Internet even if the dual stack IPv4/IPv6 hosts are located behind one or more IPv4 NATs. Teredo works for the various types of cone NATs, but it does not work, in general, for symmetric NATs. See [RFC 5389](#), *Session Traversal Utilities for NAT (STUN)*, for definitions of these types of NAT devices. To get through the IPv4 NATs, IPv6 packets are tunneled over IPv4 and UDP.

6.5.7.1 Using Teredo

The general approach to NAT traversal starts with detecting the presence of NAT. Then the protocol is started from inside the NAT and encapsulated in UDP. There is a need for some mechanism to keep track of the NAT mappings.

Figure 6-4 shows the components of Teredo and the protocol messages used to set up communications between a Teredo client and a remote IPv6 interface. A *Teredo client* is a dual stack IPv4/IPv6 node with an IPv4 address behind a NAT device. A *Teredo server* is also dual stack IPv4/IPv6, but it has a globally routable IPv4 address. The Teredo server listens to UDP port 3544 for client requests and uses a *Teredo service prefix* to construct an IPv6 address for the Teredo client. A *Teredo relay* is a dual stack IPv4/IPv6

router that provides the remote tunnel endpoint for the Teredo client. Together, the Teredo server and Teredo relay provide IPv6 connectivity for the client with the IPv6 Internet and with mechanisms like 6to4.

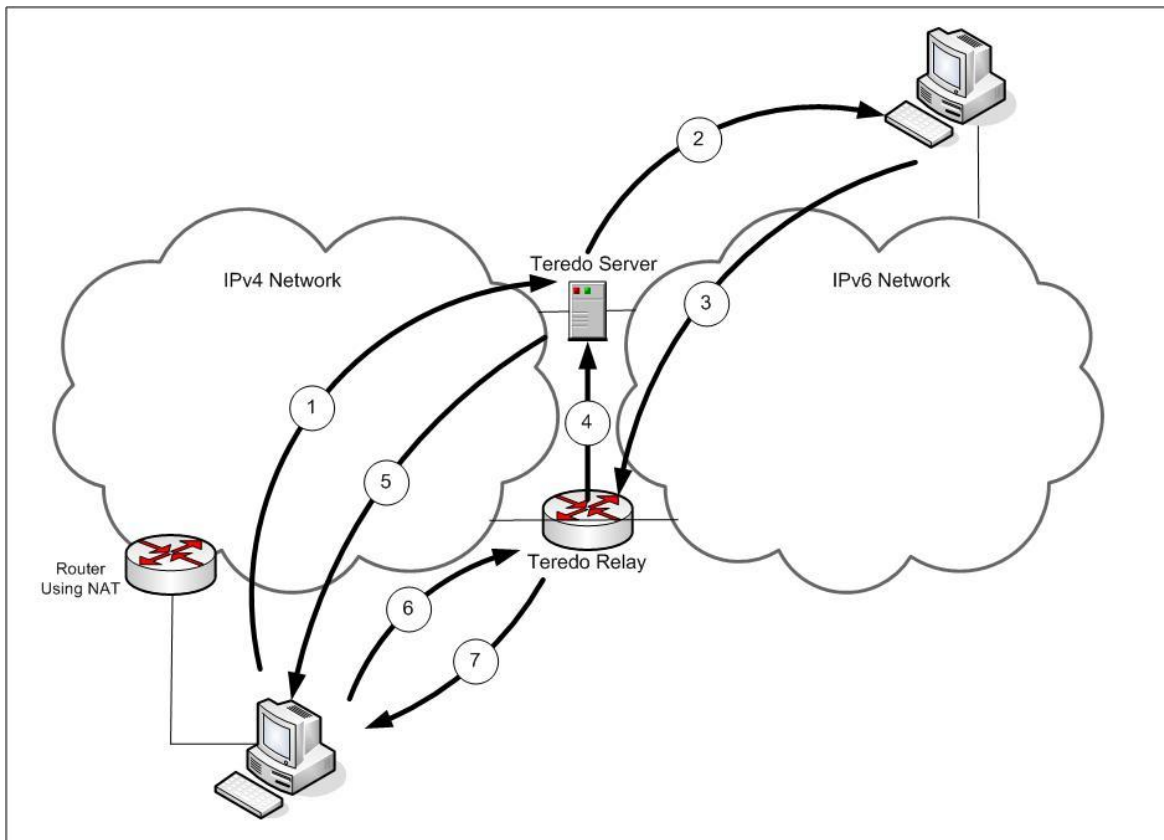


Figure 6-4. Example - Tunneling IPv6 over IPv4 Networks with Teredo

In Figure 6-4, the messages work as follows:

1. The Teredo Client sends an IPv6 *ping* to the remote interface through its Teredo server (encapsulated in UDP and IPv4).
2. The Teredo server decapsulates and forwards the *ping*.
3. The *ping reply* goes to a Teredo relay.
4. The *ping reply* goes to the Teredo server.
5. The Teredo server tells the Teredo client what Teredo relay to use.
6. The Teredo client sends normal (encapsulated) traffic to the Teredo relay.
7. The Teredo client receives normal (encapsulated) traffic from the Teredo relay.

Separating the Teredo server and Teredo relay reduces the load on servers and provides a more direct path

for tunneled traffic. Servers do not forward ordinary traffic for clients. The Teredo client and Teredo server only communicate when:

- Getting an IPv6 address for the client
- Helping the client get through the NAT with *Teredo Bubbles*, empty (protocol 59) IPv6 packets designed to create mappings through the NAT, or
- Sending an IPv6 *ping* to other IPv6 nodes to discover the best Teredo relay for that address.

A Teredo client can learn about other Teredo clients on its own IPv4 network by using the IPv4 multicast address 224.0.0.253.

IPv6 addresses for Teredo clients are comprised of the following five parts:

- Prefix: the 32-bit Teredo service prefix 2001:0000::/32
- Server IPv4: the 32-bit IPv4 address of a Teredo server
- Flags: 16 bits set to 8000 for cone NATs and 0000 otherwise
- The Teredo client's 16-bit UDP port number, inverted bit by bit
- The Teredo client's 32-bit IPv4 address (behind the NAT), inverted bit by bit.

The parts are illustrated in Figure 6-5.

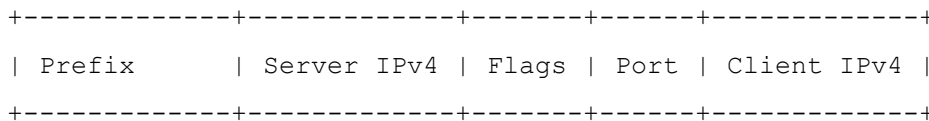


Figure 6-5. Teredo Address

6.5.7.2 Security Considerations for Teredo

The Teredo specification, [RFC 4380](#), *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)* and a subsequent Internet Draft, *Security Concerns with Tunneling*⁵⁵, contain lengthy descriptions of Teredo security considerations, including:

- Teredo allows the use of end-to-end IPsec for the encapsulated IPv6 traffic, thereby preventing the inspection of Teredo encapsulated traffic.
- No effective method exists to disable Teredo and filter all Teredo traffic. The initial communication with the Teredo server (UDP Port 3544) is easily recognizable, but the port assignment can easily be changed.

⁵⁵ See the current version of the IETF Internet Draft, [Security Concerns with IP Tunneling](#), Work in Progress.

- Subsequent traffic has no immediately distinguishing feature. Heuristic checks have been devised, but these are not likely to be practical.
- Teredo requires that external UDP ports are open and are not filtered at a NAT device or firewall. This could allow port scans to identify openings they would not see otherwise. Teredo bubbles increase this exposure.
- Teredo's IPv6 addresses are more predictable than native IPv6 addresses. Techniques like privacy addressing and CGA are not usable with Teredo.
- An attacker could deploy a rogue Teredo server or Teredo relay on the Internet to perform man-in-the-middle attacks. Securing these Teredo servers/relays are difficult because they are not network elements but servers, and those servers are on the Internet, not owned or operated by the organization. The security of those Teredo servers/relays may be in question and cause for concern over man-in-the-middle attacks is warranted.
- A third party may impersonate a Teredo server and carry out a DOS attack by sending wrong information to the Teredo client. To counter impersonation attacks, use two-way authentication between Teredo client and server.
- Teredo relays must protect themselves against being used as anonymous senders in distributed DOS attacks.
- The Teredo service must not bypass or disable ingress filtering, whatever firewall service was available in the NAT box, or other firewalls, virus checkers, or intrusion detection systems. This is a common security concern for all tunneling protocols. Ingress filtering may show up particularly with Teredo, because the system was assumed to be behind a NAT. Such security checks should also look for unexpected effects due to IPv6 type 0 routing headers.
- Services that listen to the Teredo IPv6 address become potential targets of attacks from the entire IPv6 Internet. This vulnerability can be reduced in three ways. First, it is possible to restrict some services to accept traffic only from local neighbors (for example, by using link-local addresses). Teredo does not support link-local addresses, so link-local services cannot be accessed through Teredo and are restricted to whatever other IPv6 connectivity may be available, (for example, direct traffic with neighbors on the local link, behind the NAT). Second, a local firewall can perform the same kind of filtering otherwise performed in a perimeter firewall. Third, use IPsec with Teredo to protect clients from intermediate nodes.
- An attacker may intercept a router solicitation from a Teredo client, respond with a spoofed router advertisement, and provide a Teredo client with an incorrect default router address. This may simply deny service to the Teredo client, or the attacker may try to insert itself as a relay for all client communications, effectively enabling a variety of man-in-the-middle attacks.
- A simple nonce-based verification procedure described in [RFC 4380](#) provides a first level of protection against attacks in which a third party tries to spoof the Teredo server. In practice, the nonce procedure can be defeated only if the attacker observes both the client and server communications.
- If client and server share a secret and agree on an authentication algorithm, the Teredo's secure qualification procedure provides further protection. The most likely way to obtain the shared secret is to listen to the traffic and run an offline dictionary attack. To protect against this attack, the secret shared between client and server should contain sufficient unpredictability.

- A more sophisticated attack can be carried out, but it is somewhat difficult and it does not gain too much. Actually, it works because the communications with the NAT are unprotected. It goes as follows:

1. The client (victim) prepares router solicitation, including authentication.
2. The attacker intercepts the solicitation and somehow manages to prevent it from reaching the server. One way to intercept the solicitation is by running a short-duration DOS attack against the server.
3. The attacker replaces the source IPv4 address and source UDP port of the request by one of its own addresses and port and sends the modified request to the server.
4. The server sees the IPv4 address from which the packet is received, verifies that the authentication is correct, prepares a router advertisement, signs it, and returns it to the attacker.
5. The attacker receives the advertisement, remembers the mapping, replaces the IPv4 address and UDP port by the original values in the intercepted message, and sends the response to the client.
6. The client receives the advertisement and uses the proposed prefix and the mapped addresses in the origin indication encapsulation. The attacker can now run a DOS or man-in-the-middle attack.

- Other attacks may try to spoof a Teredo relay. An attacker may try to spoof another IPv6 host or to place itself as a man-in-the-middle between a Teredo host and a native IPv6 host. The attacks work by convincing the Teredo client that packets bound to the native IPv6 host should be relayed to the IPv4 address of the attacker. These attacks are possible because there is no obvious relationship between the IPv4 address of a relay and the native IPv6 addresses it serves. A Teredo client cannot decide by looking at the encapsulating IPv4 and UDP headers whether or not a relay is legitimate. The proper relay is identified by a *direct IPv6 connectivity test*. Traffic sent to this relay will be out of reach of attackers that are not on the direct path between the Teredo client and its IPv6 peer. To stop on-path attacks, end-to-end security such as IPsec is needed. IPsec is the best defense against many of these attacks.

- A Teredo client keeps a cache of recently used peers. It is possible to provoke it to respond to packets that appear to come from a large number of Teredo peers, thus overloading this list.

- A DOS attack against the local peer discovery procedure exists if attackers can send spoofed local discovery bubbles to a Teredo client. Mitigate DOS attacks by packet filtering, restricting local peer discovery to the local link, or turning off support for discovery bubbles.

- An attacker may try to overwhelm and take down a Teredo server. Having a failover server can help, but addresses will have to be renumbered. A similar attack against a Teredo relay may cause the Teredo relay to temporarily stop announcing the reachability of the Teredo service prefix to the IPv6 network. The traffic will be picked up by the next relay. An attacker may also try to overwhelm the state table in a Teredo relay. Relays may have to be selective about whom they serve.

- Three classes of DOS attacks attempt to inject traffic at locations where it is not expected. The first uses a Teredo server as a reflector in a denial of service attack. The second uses a Teredo server to carry out a denial of service attack against IPv6 nodes. The third uses a Teredo relay to carry out a denial of service attack against IPv4 nodes. Adding appropriate filters are effective at mitigating these attacks.

- An attacker can use a Teredo server as a reflector in a DOS attack on an IPv4 target in two ways. The first is to construct a router solicitation message and post it to a Teredo server, using as the IPv4 source address the spoofed address of the target. The Teredo server will then send a router advertisement message to the target. The second is to construct a Teredo IPv6 address using the Teredo prefix, the address of a selected server, the IPv4 address of the target, and an arbitrary UDP port, and to send packets bound to that address to the Teredo server. Again, adding appropriate filters can defend against these.
- An attacker may use a Teredo server to launch a DOS attack against an arbitrary IPv6 destination. The attacker builds an IPv6 packet bound for the target. The attacker sends that packet to the IPv4 address and UDP port of a Teredo server to be relayed from there to the target over IPv6. Servers should check that the IPv4 and IPv6 source addresses are consistent, and routers or security devices should perform ingress filtering for the IPv6 addresses.
- An attacker with IPv6 connectivity may use a Teredo relay for a DOS attack against an arbitrary IPv4 destination. The attacker may compose a Teredo IPv6 address using the Teredo prefix, a “cone” flag set to 1, the IPv4 address of the target, and an arbitrary UDP port. First, relays should not allow the attacker to use multicast, broadcast, or non-routable IPv4 addresses. Second, IPv6 ingress filtering and trace back could be used (if enabled).
- Most NPDs do not have the ability to inspect IPv6 UDP traffic.

One protection mechanism against these threats is to make all enterprise computers be domain-joined to disable Teredo. If internal computer systems are joined to a Microsoft AD domain (enterprise-client) then they will not use Teredo by default. In addition, an organization could have users’ host operating systems block Teredo unless an IPv6-capable host-based firewall is being used. An even more likely strategy would be to go full dual-stack which will prevent Teredo from working because the host will already be on an IPv6 subnet.

If Teredo is not required then it can be filtered. Perimeter systems can prevent Teredo packets from going to well-known Teredo servers and relays. Firewalls can also block all Teredo addresses of the form 2001::/32 in UDP 3544 packets (use deep packet inspection [DPI] in case Teredo is being run on a non-standard TCP/UDP port). Organizations can also disallow the NAT firewall pin-hole to be created on the perimeter Internet firewall that would allow Teredo to operate through a NAT. It is also possible to create your own Teredo server on the perimeter and put it behind a firewall.

6.5.8 Tunnel Brokers

IPv6 tunnel brokers provide dual stack IPv4/IPv6 nodes on IPv4 networks with a way to obtain IPv6 connectivity without the administrative support of a large site running, for example, 6to4. It is intended for small sites or individual hosts. The IPv6 tunnel broker method requires the deployment of a tunnel broker server. The tunnel broker server can be viewed as a “virtual IPv6 ISP.” More information on IPv6 tunnel brokers is available in [RFC 3053](#), *IPv6 Tunnel Broker*.

6.5.8.1 Using IPv6 Tunnel Brokers

The client connects to a tunnel broker, which sets up and manages tunnels between clients and tunnel servers. Tunnel servers are dual stack IPv4/IPv6 routers connected to the global Internet. These proxy servers reside at well-known IPv4 addresses. Public tunnel broker servers are available. A tunnel broker’s clients can be hosts or routers. The tunnel broker performs authentication and access control. The tunnel broker assigns the IPv6 tunnel endpoints and creates records for those endpoints in the DNS. The tunnel broker allocates relatively long-lived IPv6 addresses from its own global IPv6 address space.

It does not work for clients behind NATs.

The tunnel broker description leaves many feature choices to individual implementations. Typically, clients communicate with tunnel brokers using HyperText Transfer Protocol (HTTP). A tunnel broker may help the client with the actual tunnel setup by sending automated scripts, but this raises security concerns. The client may run dynamic DNS and may communicate with tunnel servers via SNMP (Simple Network Management Protocol). These are just examples, and other implementation details are possible.

6.5.8.2 Security Considerations for IPv6 Tunnel Brokers

Security must be a consideration for all the interactions between the tunnel broker and other entities: clients, tunnel servers, and the DNS. The appropriate security depends on how each of these interfaces is implemented:

- For HTTP clients, using SSL/TLS to protect the username and password seems appropriate. HTTP is a clear text protocol. When sending confidential information, an encrypted channel should be used. Organizations should choose implementations that are inherently more secure. For example, delivering lists of tunnel parameters is safer than downloading executable scripts that run on the client.
- SNMPv1 and SNMPv2 are clear text protocols. As such, they can disclose confidential information. SNMPv3 provides encryption but lags behind SNMPv1 and SNMPv2 in support and deployment. SNMP interfaces to tunnel servers should use IPsec or SNMP over SSH if SNMPv3 is not an option. Using this option will support confidentiality and integrity of the information passed on the channel.
- For DNS updates, either Secure Dynamic Update ([RFC 3007](#), *Secure Domain Name System (DNS) Dynamic Update*) or command scripts protected by SSH or IPsec could be used.
- If a host disconnects from the Internet and its IPv4 address is reallocated, the tunnel server may not find out about this and may keep forwarding IPv6 traffic to that address.
- Tunnel broker servers maintain state for each client. As such, they are susceptible to resource exhaustion attacks and other types of DOS attacks.

6.5.9 Automatic Tunneling of IPv4 over IPv6 (Dual Stack Transition Mechanism [DSTM])

As IPv6 local networking and internetworking become more prevalent, tunneling IPv4 over IPv6 will increase in importance. One approach, called DSTM (for Dual Stack Transition Mechanism), defines an IPv6-dominant approach: convert the infrastructure to IPv6 and force IPv4 to run in tunnels over IPv6. All of the tunneling mechanisms discussed above (6over4, 6to4, 6rd, ISATAP, Teredo, and IPv6 tunnel broker) tunnel IPv6 packets within an IPv4 network. DSTM does the opposite, tunneling IPv4 packets within IPv6. DSTM has been implemented on multiple platforms; however, it has not been standardized within the IETF.

DSTM could be used where there is an IPv6 network without any IPv4 infrastructure and a relatively small number of nodes needing IPv4 support. Such networks, called IPv6 dominant, can be deployed to encourage a rapid migration to IPv6. DSTM is functionally similar to the IPv6 tunnel broker with the

roles of IPv4 and IPv6 reversed. One major difference is that DSTM presumes clients, servers, and border routers belong to the same enterprise's intranet.

[RFC 4852](#), *IPv6 Enterprise Network Analysis—IP Layer 3 Focus*, discusses the IPv6-dominant transition strategy and a research report⁵⁶ contains the details.

Running DSTM has several advantages:

- It makes IPv4 available end to end without NAT.
- It conserves IPv4 addresses by reusing them.
- It is transparent to IPv6 applications.
- It promotes a rapid transition to IPv6.

6.5.9.1 Using DSTM

The idea is to run IPv6 as the local networking protocol and still be able to access IPv4 services as needed with IPv4 over IPv6 tunneling. When an IPv6 address corresponding to an IPv4 address is needed, the IPv4 address is embedded in the IPv6 interface ID as the low order 32-bits. The high-order 16 bits are set to all ones, so the IPv4 address w.x.y.z yields the IPv6 interface ID FFFF:w.x.y.z. These are called IPv4-mapped IPv6 addresses. The IPv6 packet encapsulating the IPv4 packet has Next Header = 4.

Tunneling is set up automatically. A DSTM server receives a request from a client for a temporary, dynamically allocated IPv4 address, and it sets up the IPv4 over IPv6 tunnel to a dual stack IPv4/IPv6 DSTM border router.

The requirements for running DSTM are:

- An IPv6-only network infrastructure
- A dual stack IPv4/IPv6 client needing IPv4 connectivity and running DSTM with either DHCPv6 (the recommended choice) or Tunnel Setup Protocol (TSP) ([RFC 5572](#)).
- A DSTM server
- A DHCPv6 relay or server, or a TSP server
- An available pool of IPv4 addresses
- A dual stack IPv4/IPv6 border router capable of handling DSTM.

The dual stack IPv4/IPv6 client on the IPv6 network gets an IPv4-mapped IPv6 address and tunnel endpoint (TEP) for a DSTM router from the DSTM server. The client-server protocol is either DHCPv6 or TSP.

6.5.9.2 Security Considerations for DSTM

DSTM does not use NAT, so IPsec, SSH, and other IPv4 security services can be used end to end. DSTM can use any other security functionality that supports parts of its operation: DHCPv6 authentication, DNSSEC, or TSP over TLS.

⁵⁶ Xia, Bound, and Pouffary, [The Evaluation of DSTM: An IPv6 Transition Mechanism](#).

Tunnel endpoint security is an issue, as always. DSTM routers should be configured together with IPv4 firewalls and intrusion detection systems to secure IPv4 tunnel endpoints from IPv4-based attacks. They should also check the consistency of the IPv4 and IPv6 addresses used in tunneled traffic and apply ingress and egress filtering. Finally, they should only accept DSTM traffic on their internal interfaces to avoid being used as an open relay.

Perhaps the most serious resource exhaustion DOS attack is against the supply of IPv4 addresses. Authenticating the client-server protocol allows such attacks to be tracked down easily.

6.5.10 Carrier-Grade NAT and Dual-Stack Lite

For several years, dual stack hosts and routers have been promoted as the most practical transition plan. However, this approach faces a new problem today. Dual stack means having IPv4 and IPv6 addresses, but the world is running out of IPv4 addresses. It is expected that IPv4 will be needed to support IPv4-only devices in the home and IPv4-only content on the Internet long past the exhaustion of the pool of globally routable IPv4 addresses.

The emphasis in this transition strategy is on helping service providers plan for a situation in which not enough globally-routable IPv4 addresses are available for the WAN-facing (Wide Area Network) side of customers' NAT boxes.

Several proposals have been made to deploy native IPv6 along with some method of *sharing* globally reachable IPv4 addresses among broadband customers needing both IPv4 and IPv6 access. All of these involve some kind of centralized network address translation (NAT), which is frequently called "carrier-grade NAT." In this context, "carrier grade" essentially means centralized, and it may also imply high capacity, shared, or multiplexed.

At least four choices for how a customer's site achieves IPv4 connectivity with a carrier-grade NAT⁵⁷ have been described. One, called "IPv4->IPv4->IPv4," provides the customer with a non-routable (typically network 10.0.0.0/8) IPv4 address between the customer's gateway and the carrier-grade NAT. This can cause new problems traversing an additional layer of NAT, handling the possibility of address conflicts, and even dealing with a shortage of 10.0.0.0/8 addresses for large service providers. A second proposal, called "IPv4->IPv6->IPv4," is to run IPv6 between a customer's gateway and carrier-grade NAT. IPv4 traffic then needs to be translated twice. A third choice is to run IPv4-over-IPv6 tunnels between a customer's gateway and carrier-grade NAT. This combination of carrier-grade NAT and IPv4-over-IPv6 tunneling, called "dual stack lite," has perhaps gathered the most support, although no approved specifications exist yet. Finally, a fourth proposal tries to improve on dual-stack lite by treating portions of the port numbers effectively as extensions of the IPv4 address.⁵⁸

Carrier-grade NAT does, however, pose restrictions and present some problems for users. Applications relying on port mapping or port opening (e.g., the current UPnP [Universal Plug and Play] specification) may not work with multiple NATs. IPv4 multicast may not be supported. Finally, the total number of TCP ports for the shared IPv4 address is limited.

⁵⁷ For a discussion of the first three of these, see the expired IETF Internet Draft, [Distributed NAT for Broadband Deployments Post IPv4 Exhaustion](#).

⁵⁸ Maennel, Bush, Cittadini and Bellovin, *A Better Approach than Carrier-Grade-NAT*, [Columbia University Technical Report CUCS-041-08](#), 2008.

6.5.10.1 Using Carrier-Grade NAT and Dual-Stack Lite

Dual-stack lite⁵⁹ stretches the supply of routable IPv4 addresses by letting a service provider share them among customers. The customer is provided an IPv6 prefix and uses IPv4 over IPv6 tunnels as described in [RFC 2473](#) with Next Protocol 4 in the outer header to reach the IPv4 Internet. The service provider's carrier-grade NAT performs encapsulation and decapsulation of tunneled traffic and network address translation. It shares its routable IPv4 addresses among multiple customers, and it uses the IPv6 source address, inner IPv4 address, and port numbers of tunneled traffic to maintain a NAT mapping table and to keep track of which customer is using which TCP and UDP ports.

From the service provider's viewpoint, sharing these IPv4 addresses is an advantage. From the customer's viewpoint, only one NAT operation is performed. Transition is facilitated, because standard, native IPv6 is provided to the customer. On the other hand, IPv4 applications that require specific port assignments or specific port mapping from the NAT operation likely will not work. As user experience grows and the specifications for dual-stack lite evolve, several challenges remain:

- Keeping peer-to-peer applications, which often use only one or only a few port numbers, working.
- Providing application layer gateways for applications needing special handling.
- Defining a scheme whereby most ports are maintained by the carrier-grade NAT, but some are dedicated to and controlled by individual customers.
- Ensuring sufficient non-routable IPv4 address space is available for large service providers. The proposal asks IANA to assign new address space for this purpose.
- Handling MTU efficiently.
- Providing IPv4 multicast service.

6.5.10.2 Security Considerations for Carrier-Grade NAT and Dual-Stack Lite

Using NAT for IPv4 is likely not new, so general NAT security issues are not repeated here. (See, for example, [RFC 2663](#) and [RFC 2993](#).)

Some of the security issues with carrier-grade NAT result directly from the sharing of the routable IPv4 address. Addresses and timestamps are often used to identify a particular user, but with shared addresses, more information (i.e., protocol and port numbers) is needed. This impacts software used for logging and tracing spam, denial of service attacks, and other abuses.

Devices on the customer's side may try to carry out general attacks against systems on the global Internet or against other customers by using inappropriate IPv4 source addresses inside tunneled traffic. The carrier-grade NAT needs to protect against such abuse.

One customer may try to carry out a denial of service attack against other customers by monopolizing the available port numbers. The carrier-grade NAT needs to ensure equitable access. At a more sophisticated level, a customer may try to attack specific ports used by other customers. This may be more difficult to

⁵⁹ See the current version of the IETF Internet Draft, [Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion](#), Work in Progress.

detect and to mitigate without a complete system for authentication by port number, which would represent a huge security requirement. For a description of an analogous attack, see the Java-based attack described by Martin et al.⁶⁰

Some attack mitigation strategies block an IPv4 address (temporarily or permanently) after detecting an improper use. Thus, the carrier-grade NAT needs to protect other customers using the same address by making sure that a single user of a shared address cannot trigger such a response.

Finally, a carrier-grade NAT needs to enforce ingress filtering of IPv6 traffic to prevent IPv6 spoofing.

6.6 Translation

Translation consists of transforming IPv4 or IPv6 packets into the other protocol so they can be routed or transmitted across a network. Network Address Translation—Protocol Translation (NAT-PT) allows IPv6 and IPv4 devices to communicate via an intermediate translation device. Transport Relay Translator (TRT) is another mechanism to allow IPv6 hosts to communicate with IPv4 hosts through an intermediary. Translation mechanisms (IPv4 to IPv6 and IPv6 to IPv4) introduce new methods to construct networks and systems and thus enlarge the set of possible attacks against those networks and systems.

Protocol translation is not recommended as a strategic approach to transitioning from IPv4 to IPv6 for a variety of reasons. Translating IPv6 into IPv4 effectively negates most of the compelling reasons for transitioning to IPv6 in the first place. Hierarchical routing, expanded address space, streamlined packet headers, and IPv6 mobility are all lost when translating IPv6 into IPv4. Translation does not solve the IPv4 address space exhaustion problem. Nevertheless, an IPv6-only system cannot communicate with an IPv4-only system unless translation occurs somewhere, so translation is necessary to keep isolated IPv4 legacy applications running in an otherwise IPv6 world.

Because protocol translation is not exact, but merely the best approximation of one protocol with another, the translation of header fields, addresses, extensions, options, fragmentation, and error reporting can be exploited to try to circumvent security policies.

Translation mechanisms can work at different layers:

- Network layer translators
 - Stateless IP/ICMP Translation Algorithm (SIIT) ([RFC 2765](#))
 - NAT-PT ([RFC 2766](#) and [RFC 4966](#))
 - Bump in the Stack (BIS) ([RFC 2767](#))
- Transport layer translators
 - Transport Relay Translator (TRT) ([RFC 3142](#))
- Application layer translators
 - Bump in the API (BIA) ([RFC 3338](#))

⁶⁰ Martin, Rajagopalan and Rubin, “Blocking Java Applets at the Firewall,” *Proceedings of the Internet Society Symposium on Network and Distributed System Security*, pp. 16–26, 1997.

- SOCKS-based IPv6/IPv4 Gateway Mechanism ([RFC 3089](#))
- Application Level Gateways.

6.6.1 SIIT

SIIT stands for Stateless IP/ICMP Translation. It replaces headers, IPv4 for IPv6 and vice versa. It also translates ICMP messages and adds the ICMP pseudo-header checksum. If necessary, it fragments and reassembles IPv4 messages.

6.6.1.1 Using SIIT

SIIT uses IPv4-translated addresses to refer to IPv6-enabled nodes. When it needs to create an IPv6 address from an IPv4 address w.x.y.z, it uses 0:0:FFFF:0:0:0:w.x.y.z. SIIT uses IPv4 mapped addresses to refer to IPv4-only nodes: 0:0:0:0:FFFF:w.x.y.z. SIIT thus requires IPv6 interfaces to obtain an IPv4 address and is not a convenient transition mechanism for an IPv6 dominant network.

SIIT also adjusts the payload length appropriately and equates the following fields:

- IPv4 traffic class and IPv6 TOS
- IPv4 protocol number and IPv6 next header number
- IPv4 TTL and IPv6 hop limit.

6.6.2 NAT-PT

NAT-PT stands for Network Address Translation—Protocol Translation. It uses SIIT and performs NAT traversal for IPv4. NAT-PT allows IPv6 systems and services to communicate with IPv4 systems and vice versa. NAT-PT is the successor to IPv4 NAT, which is well understood and widely deployed.

6.6.2.1 Using NAT-PT

NAT-PT is controversial for many reasons, which are explained in detail in [RFC 4966](#). The IETF recommends that it no longer be used, and discussions of how it may be revised are ongoing. Its biggest benefit is that IPv6 connectivity becomes possible for legacy applications that may never have IPv6 support. The main objections to NAT-PT are that it reintroduces the scaling and operational issues of NAT into IPv6 and may make full migration to IPv6 more difficult. For example:

- It breaks end-to-end security: DNSSEC has to terminate on the NAT if it performs translation and IPsec cannot transit NAT-PT.
- It prevents full deployment of new applications.
- It may be a single point of failure and performance bottleneck.
- Protocols with embedded IP addresses do not work.
- Protocols like RSVP that do not use port numbers cannot be multiplexed.

In addition, translation is incompatible with multicast, mobility, and multihoming. Where possible, a

tunneling method such as Teredo or IPv6 tunnel broker may be preferable for coping with IPv4 NAT.

NAT-PT has three variations:

- Basic NAT-PT provides translation of IPv6 addresses to a pool of IPv4 addresses. It tracks and possibly alters IPv6 port numbers so that multiple IPv6 interfaces can share a single IPv4 address.
- NAPT-PT (Network Address Port Translator—Protocol Translator) is similar to NAT-PT, but it translates port numbers as well to avoid port number collisions that may break applications or even be exploited in an attack.
- DNS Application Level Gateway (DNS-ALG) is also specified, but gaps exist, a number of problems with it were indentified, the fixes are rather complicated, and it has not been widely used. The specifications are unclear about using NAT-PT without DNS-ALG.

6.6.2.2 Security Issues for NAT-PT

IPv4 NAT is often deployed as a security control, a function it was explicitly not intended to perform. This opinion is widely stated in IETF documents, and it continues to be a point of controversy. Whether deployed as a security device or to facilitate IPv4/IPv6 interoperability, NAT-PT has security implications. An Internet Draft⁶¹ documents security issues related to NAT-PT. This Internet Draft, although expired, contains lengthy descriptions of security considerations and some mitigation strategies that will influence follow-on work. These issues include:

- IPsec, in all modes except ESP tunnel mode, cannot be translated.
- If the translator attempts to reassemble fragments, this allows a DOS attack.
- If the translator can be tricked into using a multicast address, this allows an amplified DOS attack.
- NAT-PT is vulnerable to DOS attacks that deplete its address pools.
- NAT-PT is incompatible with DNSSEC.

6.6.3 Replacing NAT-PT

The IETF is defining two complementary network layer translation mechanisms to replace NAT-PT and NAPT-PT. These are described in one RFC and four Internet Drafts⁶²:

- *Framework for IPv4/IPv6 Translation*⁶³
- *IP/ICMP Translation Algorithm*⁶⁴
- *NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers*⁶⁵

⁶¹ See the expired IETF Internet Draft, [NAT-PT Security Considerations](#).

⁶² These Internet Drafts are works in progress being developed in the IETF's behave Working Group: Behave [Working Group Charter](#); Behave [Document Status](#); Behave [Presentations and Minutes](#).

⁶³ See the current version of the IETF Internet Draft, [Framework for IPv4/IPv6 Translation](#), Work in Progress.

⁶⁴ See the current version of the IETF Internet Draft, [IP/ICMP Translation Algorithm](#), Work in Progress.

⁶⁵ See the current version of the IETF Internet Draft, [Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers](#), Work in Progress.

- [RFC 6052](#) *IPv6 Addressing of IPv4/IPv6 Translators*
- *DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers*⁶⁶

The Framework recommends a dual stack coexistence and transition strategy and supports the transition plan in [RFC 5211](#), but it describes translation scenarios and guidelines for cases in which the dual stack approach may not be a practical medium-term coexistence plan. Two translation mechanisms are being defined because stateless and stateful translation have different characteristics, and each may be appropriate in certain settings.

The scenarios differ according to whether or not the IPv4 and IPv6 sides of the translator are on the global Internet or on an enterprise or ISP's network.

The mapping between IPv4 and IPv6 addresses is either maintained at the translator or conveyed by using IPv6 addresses with embedded IPv4 addresses. In the stateless translation mechanism, all address mapping is computed algorithmically. The stateful translator also uses algorithmic translation of IPv4 to IPv6 addresses, but it maintains a translation table from IPv6 addresses to, for example, a service provider's pool of IPv4 addresses. Other header fields are treated as in SIIT.

Once the address mappings are defined, perhaps the trickiest aspect of translation is handling DNS correctly. Consider, for example, an IPv6-only client accessing an IPv4-only server. The client expects a AAAA record, but the server's DNS knows nothing of this. A new DNS-ALG is being defined to handle different possible configurations securely. Two plausible choices, depending on the particular scenario, are:

- For internal use, say a legacy printer pool, simply add the necessary static AAAA records.
- In the general case, the application asks for A and AAAA records, an A record is retrieved, and, assuming none exists, the ALG generates and adds the appropriate AAAA record. Resolvers have to be careful about caching such artificially generated records.

Similar techniques with a few more wrinkles can be applied to make dynamic DNS work as well.

Some applications need special handling during translation. One good example is File Transfer Protocol (FTP) passive mode, which works differently in IPv4 and IPv6. It is anticipated that a separate Internet Draft will be written to specify how to handle each such case as it arises.

Two aspects of translation for future study are (1) handling multicast; and (2) sharing IPv4 addresses and port numbers for cases where these are in short supply.

Security has been an important concern in this new design, and security considerations have influenced the design of aspects such as the address mappings. No other new security issues have arisen.

6.6.4 TRT

Transport Relay Translator (TRT) is defined by [RFC 3142](#). Rather than simply rewriting headers, TRT keeps track of the state of TCP and UDP flows and relies on DNS translation between AAAA and A records. DNS translation is defined by [RFC 2694](#), *DNS extensions to Network Address Translators (DNS-ALG)*.

⁶⁶ See the current version of the IETF Internet Draft, [DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers](#), Work in Progress.

6.6.4.1 Using TRT

TRT works much like a proxy firewall, except that it can use IPv4 on one side and IPv6 on the other. The TRT establishes connections to the “inside” and “outside” nodes, and no IPv4 or IPv6 packets go through the TRT. The TRT does not translate DNS records; it handles DNS itself. It only works with TCP and UDP.

6.6.4.2 Security Issues for TRT

Security Considerations for TRT include:

- IPsec cannot traverse TRT.
- Because of the way TRT handles DNS, DNSSEC is incompatible with TRT.
- Protocols that base authentication on IP address (e.g., rsh, rlogin) do not work across TRT.
- The translator must retain state, so it is vulnerable to various DOS attacks.
- Attackers may use the translator to subvert address filtering or hide the true source of traffic.

6.6.5 Application Layer Translation

Application gateways provide application-specific translation, which is necessary when the application contains IP addresses. They are similar to the application gateways used in some firewalls. They enable continued operation of legacy IPv4 applications. An alternative approach is to re-write those portions of the application code that either handle IP addresses or embed those addresses in messages generated by the application.

6.6.5.1 Using Application Layer Translation

Three generic approaches to building application gateways are:

- [RFC 2767](#) covers Bump in the Stack (BIS). The translation takes place in an endpoint, so IPv4 applications can run on an IPv6 host. The translation uses SIIT and maintains address mappings from an IPv4 address pool. BIS converts DNS AAAA records to A records.
- BUMP in the API ([RFC 3338](#)) is designed to provide the same capability as BIS, but the translation occurs between IPv4 and IPv6 APIs, that is between the sockets layer calls and TCP/IP implementation. It uses the same translation conventions as SIIT, but no actual IPv4 header exists. IPv4 DNS requests are translated into IPv6 requests. An artificial pool of internal IPv4 addresses is mapped to IPv6 addresses.
- [RFC 1928](#) defines the SOCKSv5 protocol. Firewall support is part of this protocol's design. The SOCKS library is installed on the client between the application layer and the socket layer. It maintains an artificial pool of IPv4 addresses and keeps a mapping of these to domain names. SOCKS Gateways relay connections at the application layer.

6.6.5.2 Security Considerations for Application Layer Translation

Application layer translation can offer good security, even if it is of limited value in transitioning to IPv6. Host-resident translation allows use of IPv6 security mechanisms, whereas gateway-resident translation breaks end-to-end connectivity. Gateways may be subjected to DOS, address spoofing, and open relay attacks if they do not use authentication.

6.7 Other Transition Mechanisms

Dual stack operation, tunneling, and translation help to provide interoperability between IPv4 and IPv6 nodes while the transition to IPv6 is taking place. Other possible methods exist:

- Host-based translation is possible by modification of application layer software or network layer protocols. This approach is not common, perhaps because if the changes needed for translation could be made, a dual stack system could be implemented just as simply instead.
- VLANs or MPLS can carry IPv4 and IPv6 traffic. These approaches are viable ways to use Layer 2 technology instead of Layer 3 encapsulation, potentially reducing overhead. A downside is that Layer 2 technology cannot be routed effectively. All of the security concerns remain, so it is just as important in this case to ensure that security policy applies equally for IPv4 and IPv6.
- SSL/TLS encapsulation of IPv6 has been implemented and is available as a secure tunneling technology. The advantage of this approach is good security between tunnel endpoints. The disadvantage is that many of the properties of IP are lost by forcing it to run over TCP, and typical SSL/TLS mechanisms are oriented towards client-server systems rather than peer-to-peer networking.

6.8 The IPv6 Deployment Planning Process for Security

The Office of Management and Budget (OMB) developed the Enterprise Architecture Assessment Framework⁶⁷ to help organizations transitioning to IPv6. The deployment of IPv6-enabled equipment is a small part of the OMB framework. In the framework, most of the tasks deal with documenting the environment, communicating the transition both internally and externally, and making the strategic decisions. The framework outlines high-level tasks that should be accomplished during the transition effort:

- Conduct a requirements analysis to identify the current scope of IPv6 within an organization, current challenges using IPv4, and target requirements.
- Develop a sequencing plan for IPv6 implementation, integrated with the organization's Enterprise Architecture.
- Develop IPv6-related policies and enforcement mechanisms.
- Develop training material for stakeholders.
- Develop and implement a test plan for IPv6 compatibility/interoperability.
- Deploy IPv6 using a phased approach.
- Maintain and monitor networks.
- Update IPv6 requirements and target architecture on an ongoing basis.

The Architecture Assessment Framework can also be used to measure an organization's progression from an IPv4 operational environment to an IPv6 operational environment. Although the framework is not a

⁶⁷ OMB, [Enterprise Architecture Assessment Framework \(EAAF\)](#).

detailed transition plan, a transition plan should address each of the framework's objectives.

6.9 IPv6 Deployment

This document discusses IPv6 deployment, rather than IPv6 transition. The basic assumption is that, for the foreseeable future, organizations will either operate dual stack networks or accommodate both IPv4 and IPv6 networking through other means. However, the eventual goal is to transition to an IPv6 only network, or at least an IPv6-centric one. This section is written with that ultimate goal in mind.

This section covers the deployment of IPv6. The IPv6 deployment should follow the NIST guidelines for a secure information system life cycle⁶⁸ and should include the following stages:

- Initiation Phase
- Acquisition/Development Phase
- Implementation Phase
- Operations/Maintenance Phase
- Disposition Phase.

The framework calls for a phased approach or a gradual transition from IPv4 to IPv6. The use of a phased implementation will enable an organization to implement IPv6 with as little disruption to the current environment as possible. Existing users should be unaware of the new protocol until they require its use. The phased approach will minimize the affect on day-to-day operations. There are two main approaches to transition deployment:

- Pervasive IPv6 deployment
- Sparse IPv6 deployment.

In a pervasive approach, the organization enables dual (IPv4/IPv6) stack equipment rapidly throughout the entire enterprise. This scenario is appropriate when an organization has mostly new equipment that supports both IPv4 and IPv6. After the organization validates core services and translation mechanisms are functioning properly, IPv4 is disabled on all equipment, leaving an IPv6 dominant network.

Edge to core describes a sparse IPv6 deployment. In this approach, organizations enable groups or islands of IPv6 equipment in an IPv4 dominant network. After most of the edge devices transition to IPv6, the network core transitions to either dual stack or IPv6 only. A sparse IPv6 deployment requires supporting both IPv4 and IPv6 traffic throughout the duration of the deployment life cycle. This approach makes extensive use of IPv4/IPv6 and IPv6/IPv4 tunneling. This scenario is appropriate when an organization has a large installed base of older equipment or services that cannot transition to IPv6.

Software or applications may be more important than equipment when selecting a transition approach. Upgrades for hardware and embedded operating systems can be quicker than custom or off the shelf applications. The hardware vendors have been working towards IPv6 support for longer than application software vendors have. Many vendors may not be able or willing to upgrade software to support IPv6 and many organizations do not have the expertise in house to upgrade the code base. The more legacy applications and custom code an organization supports (either developed in house or highly customized off the shelf software) the greater the risk that the software will not support IPv6. Transition planners

⁶⁸ [NIST SP 800-64](#), *Security Considerations in the Information System Development Life Cycle*.

must address software in the approach to IPv6 transition.

The two main differences between an IPv6 pervasive deployment and an IPv6 sparse deployment are:

- The IPv6 pervasive deployment has a shorter lifecycle than an IPv6 sparse deployment.
- An IPv6 sparse deployment will take longer and make use of tunneling mechanisms.

Both deployment scenarios (IPv6 pervasive deployment and IPv6 sparse deployment) are covered by the same general deployment plan. All phases of the lifecycle are the same regardless of approach.

Organizations that are not yet deploying IPv6 globally should implement the following recommendations:

- Blocking all IPv6 traffic, native and tunneled, at the organization's firewall. Both incoming and outgoing traffic should be blocked.
- Disabling all IPv6-compatible ports, protocols and services on all software and hardware.
- Beginning to acquire familiarity and expertise with IPv6, through laboratory experimentation and/or limited pilot deployments.
- Making organization web servers, located outside of the organizational firewall, accessible via IPv6 connections. This will enable IPv6-only users to access the servers and aid the organization in acquiring familiarity with some aspects of IPv6 deployment.

6.9.1 Initiation Phase

The initiation phase is concerned with requirements gathering. It is important for an organization to understand its current environment before deploying IPv6. By understanding the current environment, the correct transition approach can be selected, and an organization can ensure that it maintains security parity between its IPv4 and IPv6 environment.

An organization must also begin to plan for new IPv6 features. The transition to IPv6 gives an organization an opportunity to fix problems in its current environment. IPv6 addressing will allow more opportunities to aggregate addressing and simplify routing and access control rules if requirements are collected early.

One of the key tasks in the initiation phase is to conduct an extensive inventory of the IP equipment and services. [RFC 4057](#), *IPv6 Enterprise Network Scenarios*, covers the types of questions that an organization needs to answer to plan a successful IPv6 transition. These questions are broken out by different scenarios, because each type of organization will have unique transition requirements.

What all organizations have in common are IPv4 assets that require transition to IPv6. In an effort to understand the transition requirements, an organization must perform an asset discovery. While it is feasible that asset discovery could populate a configuration management database (CMDB), the primary purpose of asset discovery in an IPv6 transition plan is to gather transition requirements.

Transition requirements determine which assets will transition to IPv6, the order assets will transition, the transition methods selected, and the security controls implemented. The type of information that can help determine if an asset or service will transition to IPv6, require some other coexistence mechanism, or be

replaced include system security categorization, operating system and applications, lifecycle replacement, and dependencies.

Each information system should designate a security categorization. FIPS 199, *Standards for Security Categorization of Federal Information and Information Systems*,⁶⁹ is the document that guides an organization in the determination of a system's category. The category of the system determines which security controls must be implemented to protect the system appropriately. The controls implemented for IPv6 will be similar to the controls implemented for IPv4.

Most organizations have a large installed base of legacy equipment and applications. Many legacy systems cannot support IPv6. IPv6 is widely supported by network equipment vendors, and desktop systems and productivity applications generally support IPv6. Firewall and IDS implementations also are beginning to have good support for IPv6. Where organizations have difficulties supporting IPv6 is with management applications, embedded systems, and legacy applications. Legacy applications that implement network protocols or process network addresses were not written to support IP agnostic addressing (IPv4/IPv6). There were no requirements for IPv6 address handling. These applications' code assumes 32-bit addressing. The application code logic and storage allocations work with IPv4 addressing. It will take a long time to update legacy applications to support IPv6. Applications that cannot support IPv6 require the evaluation and selection of co-existence mechanisms. Organizations should plan that a significant number of legacy applications will not natively support IPv6 and include requirements to support accessing IPv4 only applications and services by IPv4 and IPv6 clients.

The information captured in this inventory should populate the organization's IPv6 configuration management database and be maintained by configuration management. Configuration management data will not be perfect and will become out of date with the production environment. An organization must decide the acceptable level of accuracy needed to make good decisions and build processes to maintain the inventory data within the established parameters.

The following decisions need to be made for each piece of equipment:

- Will the equipment be replaced to support IPv6?
- Can a service be upgraded to support IPv6?
- Will a translation mechanism be necessary?

The asset inventory is the main input into the organization's decision process concerning the deployment scenario. If the inventory demonstrates mostly new equipment, an IPv6 pervasive scenario is available as a deployment option. On the other hand, if the inventory demonstrates mostly older equipment that cannot readily support IPv6, then an IPv6 sparse deployment scenario is the more likely choice.

Organizations should consider the life cycle replacement of equipment in conjunction with the IPv6 rollout schedule. When possible, the IPv6 rollout schedule should align with the lifecycle replacement to reduce costs of the overall transition. Organizations that do not align schedules should plan for additional costs.

Other requirements subject areas are:

- Security

⁶⁹ [NIST FIPS 199](#), *Standards for Security Categorization of Federal Information and Information Systems*.

- Routing
- Network Management
- Host Configuration
- Address Planning
- Application Support and Development
- Performance and Bandwidth.

6.9.2 Acquisition / Development Phase

The acquisition and development phase is concerned with taking the requirements gathered during the initiation phase and developing the IPv6 enterprise architecture. When developing the IPv6 environment, the current enterprise architecture should be considered. The acquisition/development phase will work with three different architectures: the “as is” IPv4 based enterprise architecture, the “to be” IPv6, and the transitional architecture that bridges the “as is” and “to be”.

During the development phase, an organization should plan for an IPv6 evaluation pilot. The goals of an IPv6 pilot are to test IPv6 configuration and design assumptions against existing equipment, test and evaluate new IPv6 equipment and begin training staff. The pilot is the time to test the IPv6 numbering plan and other design assumptions. It is easier and less expensive to correct design deficiencies earlier in the transition lifecycle. As the requirements are evaluated against the existing equipment, inventory gaps may be discovered. It should be expected that some equipment will not make the transition; either it does not support IPv6 or does not meet performance expectations. The pilot should be used to evaluate new equipment and its ability to coexist in the “to be” and transition architectures. The pilot is a great opportunity to allow technical staff to develop their IPv6 skills. A pilot affords the opportunity to combine classroom training with a hands-on environment.

IPv4 and IPv6 are different protocols; they behave differently on similarly configured equipment. The IPv6 pilot should include testing to validate the performance of IPv6 in the environment and to develop strategies to mitigate any problems. IPv6 generally performs worse on equipment that was designed for IPv4. This is mainly the result of IPv6 header manipulation in software instead of hardware. Other factors besides header size can degrade performance, including extension headers, large packet sizes, and differences in fragmentation characteristics. This performance difference can vary.

The pilot should establish the baseline performance of IPv4 only, IPv6 only, and dual stack equipment and services. The baseline performance will allow the enterprise architects to understand the different performance characteristics and to design capacity into the “to be” and transition architectures. When possible the testing should use production configuration to reduce the variation between the production and the test environment.

When establishing a baseline for security equipment, performance and coverage measures must be established. Security equipment evaluates traffic, which introduces latency into the network and reduces network throughput. IPv6 security equipment tends to have fewer known heuristics available for identifying potentially suspicious traffic than equivalent IPv4 equipment, which reduces the level of traffic coverage and potentially increases risk. Testing should be performed after establishing equivalent levels of coverage between IPv4 and IPv6. All security controls should be tested to include access control list, firewall rules, and IDPS signatures. The goal should be to design an IPv6 environment that is as secure as or more secure than the IPv4 environment.

Another area the pilot should address is the validation of tunnel performance. A mixed IPv4/IPv6 environment uses tunnels to connect the islands of similar protocol equipment. The performance of these tunnels affects the user experience, availability, and scalability of the network. Each transition mechanism has different performance characteristics. Before the pilot begins, the enterprise architecture team should select the appropriate mechanism based on functionality, performance, and the capabilities of the communicating peers. The testing of tunnels requires a realistic environment.

The final area the pilot should address is baseline transition mechanisms. Translation is performed in software and incurs a performance penalty.

In addition to developing the “to be” and transition enterprise architectures and running the IPv6 pilot, the other activities of the acquisition and development phase include:

- Developing a risk assessment.
- Developing a sequencing plan for IPv6 implementation.
- Developing IPv6 related policies and enforcement mechanisms.
- Developing training material for stakeholders.
- Developing and implementing a test plan for IPv6 compatibility/interoperability.

IPv6 Enterprise Architecture introduces new risks into the current environment. These risks can be the result of many different factors, including reduced security coverage, new IPv6 features, lack of IPv6 experience, degraded performance, and dual operations. A formal risk assessment must be performed with a risk mitigation strategy to address the identified risks. The risk assessment drives several other deliverables, including the security plan and certification and accreditation.

The transition to IPv6 offers a clean slate in implementing ingress and egress firewall rules. Organizations should have few IPv6 source or destination addresses that traverse the perimeter. Organizations, at this point, should be able to implement a *deny all, permit by exception* IPv6 firewall policy. In the future, as new external connections are required, either in bound or out bound, they can be documented and allowed by exception.

As the transition enterprise architecture is developed, a transition plan must also be developed. The transition plan details which equipment and servers are to be transitioned and in which order. The transition plan, which includes a coexistence plan, drives the project plan. The coexistence plan documents the type of transition mechanism used to connect islands of like protocol hosts and will also document the translation mechanisms for enterprise services.

Existing policies must be reviewed to determine whether they provide adequate coverage for IPv6. New policies should be developed to address IPv6 specific areas or areas where IPv6 has a significant impact. Senior leadership should support the new policies and sign off on them. Once policies are established, standards, procedures, and guidelines can be developed to provide guidance for equipment configuration and operation. The procurement policies should be amended early in the transition to require that all new equipment introduced to the environment support IPv6. This requirement should include both equipment that is lifecycle replacement and equipment introducing new capabilities. In many cases, existing policies can simply be amended.

IPv6 affects a broad range of support and operations personnel. During the acquisition and development

phase, individual jobs should be evaluated, and training material should be developed or identified. IPv6 skills are not readily available in the job candidate pool, and organizations should expect to increase training budgets for IPv6 training for existing staff and new hires.

Proving the connectivity and interoperability of the enterprise architecture is a key factor in the success of the IPv6 transition. To ensure a successful transition, organizations must develop measures and test procedures for key services, applications, and capabilities. These measures and procedures collectively are the test plan. After a service, application, or capability is migrated, the test plan validates that the IPv6 equivalent provides service equal to or better than the IPv4 service.

At the conclusion of the acquisition and development phase, an organization should have produced the following artifacts:

- Enterprise Architecture
- Address Allocation Plan
- Address Management Plan
- Routing Plan
- Training Plan
- Security Plan
- Coexistence Plan.

The first design decision is to determine the organization's IPv6 address allocation. The size of the address space required to support the deployment depends on the number of devices and how many networks those devices require.

At this point, the organization should request an IPv6 address allocation from their regional internet registrar (RIR).

The organization also needs to create an address management plan. The address management plan documents the following:

- How devices allocate an address (for example, clients could use autoconfiguration or DHCPv6)
- Which nodes will have fixed IP addresses.
- How to update DNS with address ranges.

The training plan documents the training requirements across the organization as related to the IPv6 deployment. Typically, core engineers, help desk support, application developers, the security team, and system administrators will need training. The training plan includes both the types of training and length of training required. Training should allow IT operations to support IPv6 in production along with their current job specifications. Poor security practices, misconfiguration, and operator error are some of the leading causes of system compromise and loss of availability. Increasing staff knowledge about security and the computing environment will increase security and availability.

The goal of the security plan is to ensure that the IPv6 environment has the same level of security or better than the existing IPv4 environment. It should document how an organization plans to maintain

security parity during the IPv6 deployment. The security plan should address the following areas:

- Equipment configuration
- Perimeter defense (firewall, ACL, IDPS)
- Content filtering
- Mail filtering
- Patch management
- Vulnerability management (scanning)
- Certification and accreditation of the new systems
- AAA (authentication, authorization, and accounting)
- Rogue detection
- Infrastructure protocol security.

The coexistence plan documents which mechanisms support IPv6/IPv4 internetworking. The coexistence plan details how IPv6 clients will access legacy IPv4 services (i.e., deployment of which translation mechanisms) and how existing IPv4 clients will access IPv6 services. By planning the coexistence mechanism in advance, an organization is able to leverage economy of scale and select technology that can be a repeatable solution. This reduces the amount of required training and increases operator familiarity with the mechanisms.

6.9.3 Implementation Phase

The implementation phase involves the secure installation and configuration of IPv6 equipment, tunnels, and translation mechanisms. The deployment stage differs depending on which deployment scenario is used (IPv6 pervasive deployment or IPv6 sparse deployment). In both scenarios, the actual IPv6 roll out involves a phased deployment. The first three steps are the same regardless of deployment scenario.

The steps for the IPv6 pervasive deployment are as follows:

1. Enabling perimeter firewall IPv6 policies and IPv6 access control lists and configuring devices in accordance with security plan, standards, and procedures.
2. Deploying external IPv6 connectivity with exterior IPv6 routing.
3. Deploying basic IPv6 services (DNS, DHCPv6, and NTPv6).
4. Deploying IPv6 interior routing.
5. Enabling management monitoring (SNMP, service monitoring, IDPS, authentication, statistical monitoring, and netflow).
6. Enabling IPv6 hosts.
7. Deploying IPv6 to IPv4 translation mechanisms.

The steps for the IPv6 sparse deployment are as follows:

1. Enabling perimeter firewall IPv6 policies and IPv6 access control lists and configuring devices in accordance with security plan, standards, and procedures.
2. Deploying external IPv6 connectivity with exterior IPv6 routing.
3. Deploying basic IPv6 services (DNS, DHCPv6, and NTPv6).
4. Enabling dual protocols on core routers. (This step can be performed at this point or after completing Step 5).
5. For each IPv6 island:
 - a. Enabling IPv6 on hosts.
 - b. Enabling transition mechanism (tunnels) to other IPv6 islands.
6. Enabling management monitoring (SNMP, service monitoring, IDPS, authentication, statistical monitoring, and netflow).
7. Deploying IPv4 to IPv6 translation mechanisms.

With Step 5 above, the process is iterative until all hosts are either dual stack or IPv6 enabled.

Organizations should have a change control board that enforces the change control process. The security manager should be a member of the change control board. An IPv6 transition should follow established change control processes. Once a system is transitioned, it should be accredited and certified using inspection and acceptance testing. The measures and test procedures developed during the acquisition and development phase are used to certify that the transition equipment performs in the environment as intended.

Organizations must validate IPv6 migrated equipment by inspecting the configuration and running test procedures before transitioning devices to production. Device configurations must comply with established security and operations standards. Device configuration settings can be validated using either manual inspection or automated inspection tools. When possible, the use of automated compliance management solutions will increase the accuracy and consistency of configuration compliance.

Transitioned equipment must integrate and interoperate with other production and migrated equipment and systems. Integration validates that equipment is able to communicate with other systems and equipment and correctly exchange data. Checklists should be consulted that list the services with which a transitioned device must interoperate. Some examples include routing neighbors, DNS, DHCPv6, NTPv6, SNMP trap servers, syslog servers, mail servers, and application gateways. Integration and interoperations testing is difficult and usually performed manually.

Transitioned equipment must perform at the levels established in the test plan. Performance testing may be validated using live traffic or load generators. Failure to comply with test plans could result in a loss of availability when the equipment is placed into production.

IPv6 migration efforts require the certification and accreditation of systems. The migration to IPv6 (either IPv6 or dual stack) has the potential to significantly change the existing security posture. While the same security controls required for IPv4 are also required for IPv6, existing controls require reworking and reconfiguration to support IPv6, and new security controls are required to mitigate new IPv6 vulnerabilities. Organizations should plan on certifying and accrediting systems that have been migrated to IPv6 and systems that interact with IPv6 systems. Organizations should evaluate their certification support tools, techniques and procedures to ensure that these support IPv6. Organizations

should ensure auditors performing the certification function are knowledgeable about IPv6 security and have the tools to look for IPv6 traffic and vulnerabilities. If two separate environments are supported (IPv4 and IPv6), then a system accreditor may require two separate certifications and accreditations; with a dual stack environment only a single certification and accreditation may be required. The goal should be to achieve security as good as or better than the IPv4 network.

6.9.4 Operations / Maintenance Phase

The operations phase often begins concurrently with the implementation phase. During operations, the focus is the secure operation of a dual stack or mixed IPv6/IPv4 environment. One of the most difficult challenges facing the operations staff in a mixed IPv6/IPv4 environment is keeping the two environments synchronized.

When operations makes changes to security controls such as firewall rule sets, access control lists, and IDS signatures, they must ensure the change occurs on both the IPv4 and IPv6 networks. Rules and signatures must be translated between IPv4 and IPv6 syntax and deployed as a single coordinated process. If strict change control is not followed, the two environments will have non-overlapping protections.

In a dual stack environment, the physical topologies of the equipment are the same, but the logical topologies can be very different. Configuration changes can have unpredictable or unforeseen consequences. Configuration management controls should be structured to prevent configuration changes on IPv4 or IPv6 networks that affect the other network. Organizations should manage and monitor their IPv4, IPv6, and dual stack environment as a single environment.

6.9.5 Disposition Phase

A migration from IPv4 to IPv6 results in displacement or retirement of equipment. Some equipment does not support IPv6 and is retired, while other equipment is transferred to IPv4 islands or to other organizations. Organizations must plan for the secure disposition of this obsolete equipment, ensuring that no confidential data is released. Organizations place themselves at great risk for exposing confidential information when disposing of obsolete equipment.

A key decision concerning sanitization is whether the equipment is planned for reuse or retirement. Often, equipment is reused to conserve an organization's resources. Equipment should be sanitized before it is reused or retired. Most organizations have policies, regulations or standards that require the proper disposition of obsolete equipment. It is important that these processes be followed.

Failure to follow proper disposition procedures could result in the disclosure of confidential information. Organizations store and process information such as personal records of employees, personal records of clients, credit card numbers, social security numbers, medical information, trade secrets, classified data, system passwords, system configuration, network documentation, and many other examples of data that must be kept confidential. Network equipment can expose administrative accounts, passwords, authentication credentials, certificates, pre-shared keys, and community strings.

Failure to maintain confidentiality could result in embarrassment to the organization, system intrusions, software and hardware license violations, legal fees and fines, and other avoidable consequences. Some data requires protection under State law or Federal law or both (e.g., the Health Insurance Portability and Accountability Act [HIPAA], the Family Education Rights Privacy Act [FERPA]) or industry governance boards like the Payment Card Industry Security Standards Council).

There are several methods available for sanitizing storage media. Equipment should be carefully

evaluated to determine the types of media employed for persistent information. Some equipment (e.g. personal computers) employs a single persistent storage mechanism (hard drive); other equipment (e.g. routers) could employ multiple methods including hard drives, flash memory, Electronically Erasable Programmable Read-Only Memory (EEPROM), and other non-volatile memory. It is important to address the sanitation of solid-state memory. The method of media destruction should insure the destruction of the information or media so that the information is unrecoverable. NIST SP 800-88, *Guidelines for Media Sanitization*,⁷⁰ provides information about the proper sanitation or destruction of equipment. SP 800-88 includes pointers to other resources such as the U.S. Department of Defense 5220.22-M Clearing and Sanitization Matrix.⁷¹

One overlooked aspect of disposition is the management of software and hardware license and service agreements. Licensing agreements often dictate the terms under which software or hardware is transferable. Once equipment is at end-of-life, it may be possible to remove that equipment from support agreements, potentially saving money.

Organizations should adopt a proactive approach to media sanitation. Equipment removed from the network should be sanitized before the replacement equipment is installed. This reduces the window of vulnerability and greatly decreases the risk of information disclosure. Installers should provide proof of sanitation, including a certificate of destruction with serial numbers and asset tags.

6.10 Summary

Security risks are inherent during the initial deployment of a new protocol such as IPv6, but mitigation strategies exist and many of the residual risks are no different from those that challenge existing IPv4 networks.

IPsec is a major component of IPv6 security and should be deployed, wherever possible, to secure IPv6 networks. Transition mechanisms allow existing IPv4 networks to coexist and interoperate with IPv6 networks, systems, and services. These transition mechanisms cover a wide range of technologies and transition scenarios. Organizations should plan their deployment and account for the full lifecycle of equipment from inception to disposal.

⁷⁰ [NIST SP 800-88](#), *Guidelines for Media Sanitization*.

⁷¹ [DOD 5220.22-M](#), *Clearing and Sanitization Matrix*.

Appendix A—Acronyms and Abbreviations

Selected acronyms and abbreviations used in the guide are defined below.

AAA	Authentication, Authorization, and Accounting
AAAA	Quad-A DNS Resource Record
ACL	Access Control List
AES	Advanced Encryption Standard
AfriNIC	Africa Network Information Centre
AH	Authentication Header
ALG	Application Layer Gateway
API	Application Program Interfaces
APNIC	Asia Pacific Network Information Centre
ARIN	American Registry of Internet Numbers
ARP	Address Resolution Protocol
AS	Autonomous System
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
BIA	Bump in the API
BIS	Bump in the Stack
BTNS	Better Than Nothing Security
BU	Binding Update
BUA	Binding Update Acknowledgement
CBC	Cipher Block Chaining
CCM	Counter with CBC-MAC
CERT	Computer Emergency Readiness Team
CGA	Cryptographically Generated Address
CGN	Carrier-Grade NAT
CIDR	Classless Inter-Domain Routing
CMDB	Configuration Management Database
CN	Correspondent Node
CoA	Care of Address
CPU	Central Processing Unit
DAD	Duplicate Address Detection
DDOS	Distributed Denial of Service
DES	Data Encryption Standard
DHAAD	Dynamic Home Agent Address Discovery
DHCP	Dynamic Host Configuration Protocol
DHCPv4	Dynamic Host Configuration Protocol for IPv4
DHCPv6	Dynamic Host Configuration Protocol for IPv6
DMZ	Demilitarized Zone
DNS	Domain Name System
DNS-ALG	Domain Name System-Application Layer Gateway
DNSKEY	Domain Name System Key
DNSSEC	DNS Security Extensions
DoD	Department of Defense
DOI	Domain of Interpretation

DOS	Denial of Service
DPI	Deep Packet Inspection
DS	Delegation Signer; Differentiated Services
DSTM	Dual Stack Transition Mechanism
DUID	DHCP Unique Identifier
EAP	Extensible Authentication Protocol
ECDSA	Elliptic Curve Digital Signature Algorithm
ECN	Explicit Congestion Notification
ECP	Elliptic Curve Group Modulo a Prime
EEPROM	Electrically Erasable Programmable Read-Only Memory
EGP	Exterior Gateway Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
ESN	Extended Sequence Number
ESP	Encapsulating Security Payload
EUI-64	Extended Unique Identifier 64 bit Addresses
FERPA	Family Education Rights Privacy Act
FIPS	Federal Information Processing Standard
FISMA	Federal Information Security Management Act
FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GCM	Galois/Counter Mode
GMAC	Galois Message Authentication Code
GRE	Generic Routing Encapsulation
GSEC	Group Security Research Group
GTSM	Generalized TTL Security Mechanism
HA	Home Agent
HIP	Host Identity Protocol
HIPAA	Health Insurance Portability and Accountability Act
HMAC	Hashed Message Authentication Code
HoA	Home Address
HR	Human Resources
HTTP	HyperText Transfer Protocol
IAB	Internet Architecture Board
IANA	Internet Assigned Number Authority
ICANN	Internet Corporation for Assigned Names and Numbers
ICMP	Internet Control Message Protocol
ICMPv4	Internet Control Message Protocol for IPv4
ICMPv6	Internet Control Message Protocol for IPv6
ID	Identifier
IDPS	Intrusion Detection and Prevention System
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IGMPv3	Internet Group Management Protocol version 3

IGP	Interior Gateway Protocol
IID	Interface Identifier
IKE	Internet Key Exchange
IKEv1	Internet Key Exchange version 1
IKEv2	Internet Key Exchange version 2
IP	Internet Protocol
IPComp	IP Payload Compression Protocol
IPng	Internet Protocol Next Generation
IPS	Intrusion Prevention System
IPsec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IPX	Internetwork Packet Exchange
IRTF	Internet Research Task Force
ISAKMP	Internet Security Association and Key Management Protocol
ISATAP	Intra-Site Automatic Tunnel Addressing Protocol
IS-IS	Intermediate System to Intermediate System
ISO	International Organization for Standardization
ISP	Internet Service Provider
IT	Information Technology
 KINK	 Kerberized Internet Negotiation of Keys
 LACNIC	 Latin America and the Caribbean Network Information Centre
LAN	Local Area Network
LIR	Local Internet Registry
LISP	Locator/ID Separation Protocol
 MAC	 Message Authentication Code; Media Access Control (Address)
MD5	Message-Digest algorithm 5
MH	Mobility Header
MIB	Management Information Base
MIPv4	Mobile IPv4
MIPv6	Mobile IPv6
MLD	Multicast Listener Discovery
MLDv1	Multicast Listener Discovery version 1
MLDv2	Multicast Listener Discovery version 2
MN	Mobile Node
MOBIKE	IKEv2 Mobility and Multihoming Protocol
MODP	Modular Exponential Diffie-Hellman Group
MPA	Mobile Prefix Advertisement
MPLS	Multiprotocol Label Switching
MPS	Mobile Prefix Solicitation
MSDP	Multi-source Discovery Protocol
MSEC	Multicast Security
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
MX	Mail Exchange
 NA	 Neighbor Advertisement
NAPT-PT	Network Address Port Translator—Protocol Translator

NAT	Network Address Translation
NAT-PT	Network Address Translation—Protocol Translation
ND	Neighbor Discovery
NH	Next Header
NIS	Network Information Service
NIST	National Institute of Standards and Technology
NPD	Network Protection Device
NS	Neighbor Solicitation; Name Server
NSA	National Security Agency
NSEC	Next Secure
NTP	Network Time Protocol
NTPv6	Network Time Protocol for IPv6
NUD	Neighbor Unreachability Detection
NVD	National Vulnerabilities Database
OMB	Office of Management and Budget
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OSPFv2	Open Shortest Path First version 2 (for IPv4)
OSPFv3	Open Shortest Path First version 3 (for IPv6)
OUI	Organizationally Unique Identifier
PAD	Peer Authorization Database
PDA	Personal Digital Assistant
PI	Provider Independent
PIM	Protocol Independent Multicast
PIM-SM	Protocol Independent Multicast—Sparse Mode
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure for X.509 Certificates
PMTU	Path Maximum Transmission Unit
PPP	Point-to-Point Protocol
PRF	Pseudo-Random Function
PRL	Potential Routers List
QoS	Quality of Service
RA	Router Advertisement
RADIUS	Remote Authentication Dial-In User Service
RFC	Request for Comment
RIP	Routing Information Protocol
RIPng	Routing Information Protocol next generation
RIPv2	Routing Information Protocol version 2
RIPE NCC	Réseaux IP Européens Network Coordination Centre
RIR	Regional Internet Registry
RPF	Reverse Path Forwarding
RR	Resource Record
RRSIG	Resource Record Signature
RS	Router Solicitation
RSA	Rivest-Shamir-Adleman
RSVP	Resource Reservation Protocol

SA	Security Association
SAD	Security Association Database
SCTP	Stream Control Transmission Protocol
SDOC	Supplier's Declaration of Conformity
SEND	Secure Neighbor Discovery
SHA	Secure Hash Algorithm
SHA-1	Secure Hash Algorithm version 1
SHA-2	Secure Hash Algorithm version 2 (SHA-224, SHA-256, SHA-384, SHA-512)
SHIM6	Site Multihoming by IPv6 Intermediation
SIIT	Stateless IP/ICMP Translation Algorithm
SIP	Session Initiation Protocol
SLA	Site-Level Aggregation Identifier
SLAAC	Stateless Address Autoconfiguration
SNMP	Simple Network Management Protocol
SNMPv1	Simple Network Management Protocol version 1
SNMPv2	Simple Network Management Protocol version 2
SNMPv3	Simple Network Management Protocol version 3
SNTP	Simple Network Time Protocol
SOCKS	Sockets Protocol
SOCKSv5	Sockets Protocol version 5
SP	Special Publication
SPD	Security Policy Database
SPI	Security Parameters Index
SRV	Service Location
SSH	Secure Shell
SSL	Secure Sockets Layer
STUN	Session Traversal Utilities for NAT
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TEP	Tunnel Endpoint
TKEY	Transaction Key
TLD	Top Level Domain
TLS	Transport Layer Security
TOS	Type of Service
TRT	Transport Relay Translator
TSIG	Transaction Signature
TSP	Tunnel Setup Protocol
TTL	Time to Live
UDP	User Datagram Protocol
ULA	Unique Local Address
ULP	Upper Layer Protocol
UPnP	Universal Plug and Play
URL	Uniform Resource Locator
US-CERT	United States Computer Emergency Readiness Team
USGv6	United States Government IPv6
VLAN	Virtual LAN
VoIP	Voice over IP

VPN	Virtual Private Network
WAN	Wide Area Network
WESP	Wrapped ESP
WiFi	Wireless Fidelity
XCBC	Extended CBC
3DES	Triple Data Encryption Standard
3G	Third Generation Wireless Technologies
3GPP2	Third Generation Partnership Project 2

Appendix B—References and Other IPv6 Resources

IETF Standards Documents

[RFC 0791](#), *Internet Protocol*, J. Postel, September 1981.

[RFC 0950](#), *Internet Standard Subnetting Procedure*, J.C. Mogul, J. Postel, August 1985.

[RFC 1752](#), *The Recommendation for the IP Next Generation Protocol*, S. Bradner, A. Mankin, January 1995.

[RFC 1809](#), *Using the Flow Label Field in IPv6*, C. Partridge, June 1995.

[RFC 1883](#), *Internet Protocol, Version 6 (IPv6) Specification*, S. Deering, R. Hinden, December 1995 (replaced by [RFC 2460](#)).

[RFC 1918](#), *Address Allocation for Private Internets*, Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, February 1996.

[RFC 1928](#), *SOCKS Protocol Version 5*, M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, L. Jones, March 1996.

[RFC 1981](#), *Path MTU Discovery for IP version 6*, J. McCann, S. Deering, J. Mogul, August 1996.

[RFC 2072](#), *Router Renumbering Guide*, H. Berkowitz, January 1997.

[RFC 2080](#), *RIPng for IPv6*, G. Malkin, R. Minnear, January 1997.

[RFC 2207](#), *RSVP Extensions for IPSEC Data Flows*, L. Berger, T. O'Malley, September 1997.

[RFC 2328](#), *OSPF Version 2*, J. Moy, April 1998.

[RFC 2375](#), *IPv6 Multicast Address Assignments*, R. Hinden, S. Deering, July 1998.

[RFC 2394](#), *IP Payload Compression Using DEFLATE*, R. Pereira, December 1998.

[RFC 2395](#), *IP Payload Compression Using LZS*, R. Friend, R. Monsour, December 1998.

[RFC 2404](#), *The Use of HMAC-SHA-1-96 within ESP and AH*, C. Madson, R. Glenn, November 1998.

[RFC 2406](#), *IP Encapsulating Security Payload (ESP)*, S. Kent, R. Atkinson, November 1998 (replaced by [RFC 4303](#)).

[RFC 2410](#), *The NULL Encryption Algorithm and Its Use With IPsec*, R. Glenn, S. Kent, November 1998.

[RFC 2451](#), *The ESP CBC-Mode Cipher Algorithms*, R. Pereira, R. Adams, November 1998.

[RFC 2453](#), *RIP Version 2*, G. Malkin, November 1998.

[RFC 2460](#), *Internet Protocol, Version 6 (IPv6) Specification*, S. Deering, R. Hinden, December 1998.

[RFC 2461](#), *Neighbor Discovery for IP Version 6 (IPv6)*, T. Narten, E. Nordmark, W. Simpson, December 1998 (replaced by [RFC 4861](#)).

[RFC 2462](#), *IPv6 Stateless Address Autoconfiguration*, S. Thomson, T. Narten, December 1998 (replaced by [RFC 4862](#)).

[RFC 2473](#), *Generic Packet Tunneling in IPv6 Specification*, A. Conta, S. Deering, December 1998.

[RFC 2474](#), *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, K. Nichols, S. Blake, F. Baker, D. Black, December 1998.

[RFC 2529](#), *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels*, B. Carpenter, C. Jung, March 1999.

[RFC 2663](#), *IP Network Address Translator (NAT) Terminology and Considerations*, P. Srisuresh, M. Holdrege, August 1999.

[RFC 2675](#), *IPv6 Jumbograms*, D. Borman, S. Deering, R. Hinden, August 1999.

[RFC 2694](#), *DNS extensions to Network Address Translators (DNS_ALG)*, P. Srisuresh, G. Tsirtsis, P. Akkiraju, A. Heffernan, September 1999.

[RFC 2710](#), *Multicast Listener Discovery (MLD) for IPv6*, S. Deering, W. Fenner, B. Haberman, October 1999.

[RFC 2711](#), *IPv6 Router Alert Option*, C. Partridge, A. Jackson, October 1999.

[RFC 2765](#), *Stateless IP/ICMP Translation Algorithm (SIIT)*, E. Nordmark, February 2000.

[RFC 2766](#), *Network Address Translation - Protocol Translation (NAT-PT)*, G. Tsirtsis, P. Srisuresh, February 2000 (replaced by [RFC 4966](#)).

[RFC 2767](#), *Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)*, K. Tsuchiya, H. Higuchi, Y. Atarashi, February 2000.

[RFC 2784](#), *Generic Routing Encapsulation (GRE)*, D. Farinacci, T. Li, S. Hanks, D. Meyer, P. Traina, March 2000.

[RFC 2827](#), *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*, P. Ferguson, D. Senie, May 2000.

[RFC 2845](#), *Secret Key Transaction Authentication for DNS (TSIG)*, P. Vixie, O. Gudmundsson, D. Eastlake 3rd, B. Wellington, May 2000.

[RFC 2930](#), *Secret Key Establishment for DNS (TKEY RR)*, D. Eastlake 3rd, September 2000.

[RFC 2931](#), *DNS Request and Transaction Signatures (SIG(0)s)*, D. Eastlake 3rd, September 2000.

[RFC 2993](#), *Architectural Implications of NAT*, T. Hain, November 2000.

[RFC 3007](#), *Secure Domain Name System (DNS) Dynamic Update*, B. Wellington, November 2000.

[RFC 3022](#), *Traditional IP Network Address Translator (Traditional NAT)*, P. Srisuresh, K. Egevang, January 2001.

[RFC 3041](#), *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, T. Narten, R. Draves, January 2001 (replaced by [RFC 4941](#)).

[RFC 3053](#), *IPv6 Tunnel Broker*, A. Durand, P. Fasano, I. Guardini, D. Lento, January 2001.

[RFC 3056](#), *Connection of IPv6 Domains via IPv4 Clouds*, B. Carpenter, K. Moore, February 2001.

[RFC 3068](#), *An Anycast Prefix for 6to4 Relay Routers*, C. Huitema, June 2001.

[RFC 3089](#), *A SOCKS-based IPv6/IPv4 Gateway Mechanism*, H. Kitamura, April 2001.

[RFC 3118](#), *Authentication for DHCP Messages*, R. Droms, W. Arbaugh, Eds., June 2001.

[RFC 3142](#), *An IPv6-to-IPv4 Transport Relay Translator*, J. Hagino, K. Yamamoto, June 2001.

[RFC 3168](#), *The Addition of Explicit Congestion Notification (ECN) to IP*, K. Ramakrishnan, S. Floyd, D. Black, September 2001.

[RFC 3173](#), *IP Payload Compression Protocol (IPComp)*, A. Shacham, B. Monsour, R. Pereira, M. Thomas, September 2001.

[RFC 3177](#), *IAB/IESG Recommendations on IPv6 Address Allocations to Sites*. IAB, IESG, September 2001.

[RFC 3306](#), *Unicast-Prefix-based IPv6 Multicast Addresses*, B. Haberman, D. Thaler, August 2002.

[RFC 3315](#), *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, R. Droms, Ed., J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney, July 2003.

[RFC 3319](#), *Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers*, H. Schulzrinne, B. Volz, July 2003.

[RFC 3338](#), *Dual Stack Hosts Using "Bump-in-the-API" (BIA)*, S. Lee, M-K. Shin, Y-J. Kim, E. Nordmark, A. Durand, October 2002.

[RFC 3344](#), *IP Mobility Support for IPv4*, C. Perkins, Ed., August 2002.

[RFC 3447](#), *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*, J. Jonsson, B. Kaliski, February 2003.

[RFC 3484](#), *Default Address Selection for Internet Protocol version 6 (IPv6)*, R. Draves, February 2003.

[RFC 3526](#), *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)*, T. Kivinen, M. Kojo, May 2003.

[RFC 3531](#), *A Flexible Method for Managing the Assignment of Bits of an IPv6 Address Block*, M. Blanchet, April 2003.

[RFC 3566](#), *The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec*, S. Frankel, H. Herbert, September 2003.

[RFC 3596](#), *DNS Extensions to Support IP Version 6*, S. Thomson, C. Huitema, V. Ksinant, M. Souissi, October 2003.

[RFC 3602](#), *The AES-CBC Cipher Algorithm and Its Use with IPsec*, S. Frankel, R. Glenn, S. Kelly, September 2003.

[RFC 3633](#), *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6*, O. Troan, R. Droms, December 2003.

[RFC 3646](#), *DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, R. Droms, Ed., December 2003.

[RFC 3686](#), *Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)*, R. Housley, January 2004.

[RFC 3697](#), *IPv6 Flow Label Specification*, J. Rajahalme, A. Conta, B. Carpenter, S. Deering, March 2004.

[RFC 3701](#), *6bone (IPv6 Testing Address Allocation) Phaseout*, R. Fink, R. Hinden, March 2004.

[RFC 3704](#), *Ingress Filtering for Multihomed Networks*, F. Baker, P. Savola, March 2004.

[RFC 3706](#), *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*, G. Huang, S. Beaulieu, D. Rochefort, February 2004.

[RFC 3736](#), *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*, R. Droms, April 2004.

[RFC 3756](#), *IPv6 Neighbor Discovery (ND) Trust Models and Threats*, P. Nikander, Ed., J. Kempf, E. Nordmark, May 2004.

[RFC 3775](#), *Mobility Support in IPv6*, D. Johnson, C. Perkins, J. Arkko, June 2004.

[RFC 3776](#), *Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents*, J. Arkko, V. Devarapalli, F. Dupont, June 2004.

[RFC 3810](#), *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*, R. Vida, Ed., L. Costa, Ed., June 2004.

[RFC 3833](#), *Threat Analysis of the Domain Name System (DNS)*, D. Atkins, R. Austein, August 2004.

[RFC 3849](#), *IPv6 Address Prefix Reserved for Documentation*, G. Huston, A. Lord, P. Smith, July 2004.

[RFC 3879](#), *Deprecating Site Local Addresses*, C. Huitema, B. Carpenter, September 2004.

[RFC 3898](#), *Network Information Service (NIS) Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, V. Kalusivalingam, October 2004.

- [RFC 3901](#), *DNS IPv6 Transport Operational Guidelines*, A. Durand, J. Ihen, September 2004.
- [RFC 3947](#), *Negotiation of NAT-Traversal in the IKE*, T. Kivinen, B. Swander, A. Huttunen, V. Volpe, January 2005.
- [RFC 3956](#), *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*, P. Savola, B. Haberman, November 2004.
- [RFC 3964](#), *Security Considerations for 6to4*, P. Savola, C. Patel, December 2004.
- [RFC 3971](#), *SEcure Neighbor Discovery (SEND)*, J. Arkko, Ed., J. Kempf, B. Zill, P. Nikander, March 2005.
- [RFC 3972](#), *Cryptographically Generated Addresses (CGA)*, T. Aura, March 2005.
- [RFC 4007](#), *IPv6 Scoped Address Architecture*, S. Deering, B. Haberman, T. Jinmei, E. Nordmark, B. Zill, March 2005.
- [RFC 4014](#), *Remote Authentication Dial-In User Service (RADIUS) Attributes Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Information Option*, R. Droms, J. Schnizlein, February 2005.
- [RFC 4033](#), *DNS Security Introduction and Requirements*, R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, March 2005.
- [RFC 4034](#), *Resource Records for the DNS Security Extensions*, R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, March 2005.
- [RFC 4035](#), *Protocol Modifications for the DNS Security Extensions*, R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, March 2005.
- [RFC 4057](#), *IPv6 Enterprise Network Scenarios*, J. Bound, Ed., June 2005.
- [RFC 4074](#), *Common Misbehavior Against DNS Queries for IPv6 Addresses*, Y. Morishita, T. Jinmei, May 2005.
- [RFC 4075](#), *Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6*, V. Kalusivalingam, May 2005.
- [RFC 4086](#), *Randomness Requirements for Security*, D. Eastlake 3rd, J. Schiller, S. Crocker, June 2005.
- [RFC 4106](#), *The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)*, J. Viega, D. McGrew, June 2005.
- [RFC 4177](#), *Architectural Approaches to Multi-homing for IPv6*, G. Huston, September 2005.
- [RFC 4192](#), *Procedures for Renumbering an IPv6 Network without a Flag Day*, F. Baker, E. Lear, R. Droms, September 2005.
- [RFC 4193](#), *Unique Local IPv6 Unicast Addresses*, R. Hinden, B. Haberman, October 2005.

[RFC 4213](#), *Basic Transition Mechanisms for IPv6 Hosts and Routers*, E. Nordmark, R. Gilligan, October 2005.

[RFC 4218](#), *Threats Relating to IPv6 Multihoming Solutions*, E. Nordmark, T. Li, October 2005.

[RFC 4225](#), *Mobile IP Version 6 Route Optimization Security Design Background*, P. Nikander, J. Arkko, T. Aura, G. Montenegro, E. Nordmark, December 2005.

[RFC 4230](#), *RSVP Security Properties*, H. Tschofenig, R. Graveman, December 2005.

[RFC 4285](#), *Authentication Protocol for Mobile IPv6*, A. Patel, K. Leung, M. Khalil, H. Akhtar, K. Chowdhury, January 2006.

[RFC 4291](#), *IP Version 6 Addressing Architecture*, R. Hinden, S. Deering, February 2006.

[RFC 4301](#), *Security Architecture for the Internet Protocol*, S. Kent, K. Seo, December 2005.

[RFC 4302](#), *IP Authentication Header*, S. Kent, December 2005.

[RFC 4303](#), *IP Encapsulating Security Payload (ESP)*, S. Kent, December 2005.

[RFC 4304](#), *Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)*, S. Kent, December 2005.

[RFC 4306](#), *Internet Key Exchange (IKEv2) Protocol*, C. Kaufman, Ed., December 2005 (replaced by [RFC 5996](#)).

[RFC 4307](#), *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*, J. Schiller, December 2005.

[RFC 4308](#), *Cryptographic Suites for IPsec*, P. Hoffman, December 2005.

[RFC 4309](#), *Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)*, R. Housley, December 2005.

[RFC 4311](#), *IPv6 Host-to-Router Load Sharing*, R. Hinden, D. Thaler, November 2005.

[RFC 4359](#), *The Use of RSA/SHA-1 Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH)*, B. Weis, January 2006.

[RFC 4361](#), *Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)*, T. Lemon, B. Sommerfeld, February 2006.

[RFC 4380](#), *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)*, C. Huitema, February 2006.

[RFC 4388](#), *Dynamic Host Configuration Protocol (DHCP) Leasequery*, R. Woundy, K. Kinnear, February 2006.

[RFC 4430](#), *Kerberized Internet Negotiation of Keys (KINK)*, S. Sakane, K. Kamada, M. Thomas, J.

Vilhuber, March 2006.

[RFC 4434](#), *The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)*, P. Hoffman, February 2006.

[RFC 4443](#), *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, A. Conta, S. Deering, M. Gupta, Ed., March 2006.

[RFC 4449](#), *Securing Mobile IPv6 Route Optimization Using a Static Shared Key*, C. Perkins, June 2006.

[RFC 4472](#), *Operational Considerations and Issues with IPv6 DNS*, A. Durand, J. Ihren, P. Savola, April 2006.

[RFC 4477](#), *Dynamic Host Configuration Protocol (DHCP): IPv4 and IPv6 Dual-Stack Issues*, T. Chown, S. Venaas, C. Strauf, May 2006.

[RFC 4487](#), *Mobile IPv6 and Firewalls: Problem Statement*, F. Le, S. Faccin, B. Patil, H. Tschofenig, May 2006.

[RFC 4543](#), *The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH*, D. McGrew, J. Viega, May 2006.

[RFC 4552](#), *Authentication/Confidentiality for OSPFv3*, M. Gupta, N. Melam, June 2006.

[RFC 4554](#), *Use of VLANs for IPv4-IPv6 Coexistence in Enterprise Networks*, T. Chown, June 2006.

[RFC 4555](#), *IKEv2 Mobility and Multihoming Protocol (MOBIKE)*, P. Eronen, June 2006.

[RFC 4581](#), *Cryptographically Generated Addresses (CGA) Extension Field Format*, M. Bagnulo, J. Arkko, October 2006.

[RFC 4601](#), *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*, B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, August 2006.

[RFC 4604](#), *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast*, H. Holbrook, B. Cain, B. Haberman, August 2006.

[RFC 4609](#), *Protocol Independent Multicast - Sparse Mode (PIM-SM) Multicast Routing Security Issues and Enhancements*, P. Savola, R. Lehtonen, D. Meyer, October 2006.

[RFC 4621](#), *Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol*, T. Kivinen, H. Tschofenig, August 2006.

[RFC 4632](#), *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*, V. Fuller, T. Li, August 2006.

[RFC 4635](#), *HMAC SHA (Hashed Message Authentication Code, Secure Hash Algorithm) TSIG Algorithm Identifiers*, D. Eastlake 3rd, August 2006.

[RFC 4640](#), *Problem Statement for bootstrapping Mobile IPv6 (MIPv6)*, A. Patel, Ed., G. Giarretta, Ed.,

September 2006.

[RFC 4718](#), *IKEv2 Clarifications and Implementation Guidelines*, P. Eronen, P. Hoffman, October 2006 (replaced by [RFC 5996](#)).

[RFC 4754](#), *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*, D. Fu, J. Solinas, January 2007.

[RFC 4760](#), *Multiprotocol Extensions for BGP-4*, T. Bates, R. Chandra, D. Katz, Y. Rekhter, January 2007.

[RFC 4807](#), *IPsec Security Policy Database Configuration MIB*, M. Baer, R. Charlet, W. Hardaker, R. Story, C. Wang, March 2007.

[RFC 4809](#), *Requirements for an IPsec Certificate Management Profile*, C. Bonatti, Ed., S. Turner, Ed., G. Lebovitz, Ed., February 2007.

[RFC 4835](#), *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*, V. Manral, April 2007.

[RFC 4852](#), *IPv6 Enterprise Network Analysis - IP Layer 3 Focus*, J. Bound, Y. Pouffary, S. Klynsma, T. Chown, D. Green, April 2007.

[RFC 4861](#), *Neighbor Discovery for IP version 6 (IPv6)*, T. Narten, E. Nordmark, W. Simpson, H. Soliman, September 2007.

[RFC 4862](#), *IPv6 Stateless Address Autoconfiguration*, S. Thomson, T. Narten, T. Jinmei, September 2007.

[RFC 4868](#), *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*, S. Kelly, S. Frankel, May 2007.

[RFC 4869](#), *Suite B Cryptographic Suites for IPsec*, L. Law, J. Solinas, May 2007.

[RFC 4877](#), *Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture*, V. Devarapalli, F. Dupont, April 2007.

[RFC 4882](#), *IP Address Location Privacy and Mobile IPv6: Problem Statement*, R. Koodli, May 2007.

[RFC 4890](#), *Recommendations for Filtering ICMPv6 Messages in Firewalls*, E. Davies, J. Mohacsi, May 2007.

[RFC 4891](#), *Using IPsec to Secure IPv6-in-IPv4 Tunnels*, R. Graveman, M. Parthasarathy, P. Savola, H. Tschofenig, May 2007.

[RFC 4894](#), *Use of Hash Algorithms in Internet Key Exchange (IKE) and IPsec*, P. Hoffman, May 2007.

[RFC 4941](#), *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, T. Narten, R. Draves, S. Krishnan, September 2007.

[RFC 4942](#), *IPv6 Transition/Co-existence Security Considerations*, E. Davies, S. Krishnan, P. Savola,

September 2007.

[RFC 4945](#), *The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX*, B. Korver, August 2007.

[RFC 4966](#), *Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status*, C. Aoun, E. Davies, July 2007.

[RFC 4994](#), *DHCPv6 Relay Agent Echo Request Option*, S. Zeng, B. Volz, K. Kinneer, J. Brzozowski, September 2007.

[RFC 5026](#), *Mobile IPv6 Bootstrapping in Split Scenario*, G. Giaretta, Ed., J. Kempf, V. Devarapalli, Ed., October 2007.

[RFC 5082](#), *The Generalized TTL Security Mechanism (GTSM)*, V. Gill, J. Heasley, D. Meyer, P. Savola, Ed., C. Pignataro, October 2007.

[RFC 5095](#), *Deprecation of Type 0 Routing Headers in IPv6*, J. Abley, P. Savola, G. Neville-Neil, December 2007.

[RFC 5114](#), *Additional Diffie-Hellman Groups for Use with IETF Standards*, M. Lepinski, S. Kent, January 2008.

[RFC 5157](#), *IPv6 Implications for Network Scanning*, T. Chown, March 2008.

[RFC 5211](#), *An Internet Transition Plan*, J. Curran, July 2008.

[RFC 5214](#), *Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*, F. Templin, T. Gleeson, D. Thaler, March 2008.

[RFC 5282](#), *Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol*, D. Black, D. McGrew, August 2008.

[RFC 5294](#), *Host Threats to Protocol Independent Multicast (PIM)*, P. Savola, J. Lingard, August 2008.

[RFC 5340](#), *OSPF for IPv6*, R. Coltun, D. Ferguson, J. Moy, A. Lindem, July 2008.

[RFC 5358](#), *Preventing Use of Recursive Nameservers in Reflector Attacks*, J. Damas, F. Neves, October 2008.

[RFC 5374](#), *Multicast Extensions to the Security Architecture for the Internet Protocol*, B. Weis, G. Gross, D. Ignjatic, November 2008.

[RFC 5375](#), *IPv6 Unicast Address Assignment Considerations*, G. Van de Velde, C. Popoviciu, T. Chown, O. Bonness, C. Hahn, December 2008.

[RFC 5386](#), *Better-Than-Nothing Security: An Unauthenticated Mode of IPsec*, N. Williams, M. Richardson, November 2008.

[RFC 5387](#), *Problem and Applicability Statement for Better-Than-Nothing Security (BTNS)*, J. Touch, D. Black, Y. Wang, November 2008.

[RFC 5389](#), *Session Traversal Utilities for NAT (STUN)*, J. Rosenberg, R. Mahy, P. Matthews, D. Wing, October 2008.

[RFC 5406](#), *Guidelines for Specifying the Use of IPsec Version 2*, S. Bellovin, February 2009.

[RFC 5452](#), *Measures for Making DNS More Resilient against Forged Answers*, A. Hubert, R. van Mook, January 2009.

[RFC 5533](#), *Shim6: Level 3 Multihoming Shim Protocol for IPv6*, E. Nordmark, M. Bagnulo, June 2009.

[RFC 5534](#), *Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming*, J. Arkko, I. van Beijnum, June 2009.

[RFC 5535](#), *Hash-Based Addresses (HBA)*, M. Bagnulo, June 2009.

[RFC 5555](#), *Mobile IPv6 Support for Dual Stack Hosts and Routers*, H. Soliman, Ed., June 2009.

[RFC 5569](#), *IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)*, R. Despres, January 2010.

[RFC 5572](#), *IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP)*, M. Blanchet, F. Parent, February 2010.

[RFC 5660](#), *IPsec Channels: Connection Latching*, N. Williams, October 2009.

[RFC 5685](#), *Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)*, V. Devarapalli, K. Weniger, November 2009.

[RFC 5723](#), *Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption*, Y. Sheffer, H. Tschofenig, January 2010.

[RFC 5739](#), *IPv6 Configuration in Internet Key Exchange Protocol Version 2 (IKEv2)*, P. Eronen, J. Laganier, C. Madson, February 2010.

[RFC 5796](#), *Authentication and Confidentiality in Protocol Independent Multicast Sparse Mode (PIM-SM) Link-Local Messages*, W. Atwood, S. Islam, M. Siami, March 2010.

[RFC 5840](#), *Wrapped Encapsulating Security Payload (ESP) for Traffic Visibility*, K. Grewal, G. Montenegro, M. Bhatia, April 2010.

[RFC 5879](#), *Heuristics for Detecting ESP-NULL Packets*, T. Kivinen, D. McDonald, May 2010.

[RFC 5903](#), *Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2*, D. Fu, J. Solinas, June 2010.

[RFC 5930](#), *Using Advanced Encryption Standard Counter Mode (AES-CTR) with the Internet Key Exchange version 02 (IKEv2) Protocol*, S. Shen, Y. Mao, NSS. Murthy, July 2010.

[RFC 5969](#), *IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) – Protocol Specification*, W. Townsley, O. Troan, August 2010.

[RFC 5996](#), *Internet Key Exchange Protocol Version 2 (IKEv2)*, C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, September 2010.

[RFC 5998](#), *An Extension for EAP-Only Authentication in IKEv2*, P. Eronen, H. Tschofenig, Y. Sheffer, September 2010.

[RFC 6027](#), *IPsec Cluster Problem Statement*, Y. Nir, October 2010.

[RFC 6052](#), *IPv6 Addressing of IPv4/IPv6 Translators*, C. Bao, C. Huitema, M. Bagnulo, M. Boucadair, X. Li, October 2010.

[RFC 6071](#), *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*, S. Frankel, S. Krishnan, December 2010.

IETF Internet Drafts

IETF Internet Draft (expired), [Distributed NAT for Broadband Deployments Post IPv4 Exhaustion](#).

IETF Internet Draft, [DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers](#), Work in Progress.

IETF Internet Draft, [Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion](#), Work in Progress.

IETF Internet Draft, [Framework for IPv4/IPv6 Translation](#), Work in Progress.

IETF Internet Draft, [Home Agent Reliability Protocol \(HARP\)](#), Work in Progress.

IETF Internet Draft, [IP/ICMP Translation Algorithm](#), Work in Progress.

IETF Internet Draft, [IPv6 Router Advertisement Guard](#), Work in Progress.

IETF Internet Draft (expired), [Mobile IPv6 Residual Threats](#).

IETF Internet Draft, [Mobility Support in IPv6](#), Work in Progress.

IETF Internet Draft, [Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers](#), Work in Progress.

IETF Internet Draft (expired), [NAT-PT Security Considerations](#).

IETF Internet Draft, [Protocol Support for High Availability of IKEv2/IPsec](#), Work in Progress.

IETF Internet Draft, [A Quick Crash Detection Method for IKE](#), Work in Progress.

IETF Internet Draft, [Security Concerns with IP Tunneling](#), Work in Progress

IETF Internet Draft, [Transport Layer Security-based Mobile IPv6 Security Framework for Mobile Node to Home Agent Communication](#), Work in Progress.

Other Sources of IPv6 Information

Handbook of IPv4 to IPv6 Transition: Methodologies for Institutional and Corporate Networks, J. Amos and D. Mimili, Auer Bach, 2007.

IP Addressing & Subnetting including IPv6, J.D. Wegner and R. Rockwell, Singers, 1999.

IPv6 Advanced Protocols Implementation, Q. Li, T. Jinmei and K. Shima, Morgan Kaufmann, 2007.

IPv6 Core Protocols Implementation, Q. Li, T. Jinmei and K. Shima, Morgan Kaufmann, 2006.

IPv6 Essentials (2nd edition), S. Hagen, O'Reilly, 2006.

IPv6 Network Administration, D. Malone and N. Murphy, O'Reilly, 2005.

IPv6 Security, S. Hogg and E. Vyncke, Cisco, 2008.

Migrating to IPv6, M. Blanchet, Wiley, 2006.

“Mobile IPv6 Security,” J. Kemp, J. Arkko and P. Nikander, *Wireless Personal Communications*, Vol. 29, (2004): 389-414.

Mobile IPv6, H. Soliman, Addison-Wesley, 2004.

Running IPv6, I. van Beijnum, Apress, 2005.

Online IPv6 Resources

[ARIN Number Resource Policy Manual](#) (current version).

[A Better Approach than Carrier-Grade-NAT](#), O. Maennel, R. Bush, L. Cittadini and S. Bellovin, Columbia University Technical Report CU-CS-041-08, 2008.

[The Evaluation of DSTM: An IPv6 Transition Mechanism](#), H. Xia, J. Bound and Y. Pouffary.

[A Filtering Strategy for Mobile IPv6](#), NSA Report, September 2007.

[Firewall Design Considerations for IPv6](#), NSA Report, October 2007.

[Internet Protocol Version 6: Federal Agencies Need to Plan for Transition and Manage Security Risks](#), GAO Report, May 2005.

[IPv4 and IPv6 Threat Comparison and Best Practice Evaluation \(v 1.0\)](#), Cisco White Paper, 2004.

[IPv6 Security Technology Paper](#), North American IPv6 Task Force (NAv6TF) Technology Report, July 2006.

[Malware Tunneling in IPv6](#), US CERT Report, May 2005.

[Router Security Configuration Guide Supplement – Security for IPv6 Routers](#), NSA Report, May 2006.

[*Transition Planning for Internet Protocol Version 6 \(IPv6\)*](#), OMB Memo M-05-22, August 2005.

[*Transition to IPv6*](#), OMB Memo, September 2010.

[*Worm propagation strategies in an IPv6 Internet*](#), S. Bellovin, B. Cheswick and A. Keromytis, :Login, Vol. 31, No. 1.

NIST Publications⁷²

[*NIST FIPS 199, Standards for Security Categorization of Federal Information and Information System*](#), February 2004.

[*NIST SP 500-267, A Profile for IPv6 in the U.S. Government – Version 1.0*](#), July 2008.

[*NIST SP 500-273, USGv6 Test Methods: General Description and Validation*](#), November 2009.

[*NIST SP 500-281, USGv6 Testing Program User's Guide*](#), December 2009.

[*NIST SP 800-54, Border Gateway Protocol Security*](#), July 2007.

[*NIST SP 800-61, Computer Security Incident Handling Guide*](#), March 2008.

[*NIST SP 800-64, Security Considerations in the System Development Life Cycle*](#), October 2008.

[*NIST SP 800-77, Guide to IPsec VPNs*](#), December 2005.

[*NIST SP 800-81, Secure Domain Name System \(DNS\) Deployment Guide*](#), April 2010.

[*NIST SP 800-88, Guidelines for Media Sanitization*](#), September 2006.

Other non-IPv6 Resources

“Blocking Java Applets at the Firewall,” D. Martin, S. Rajagopalan, and A. Rubin, *Proceedings of the Internet Society Symposium on Network and Distributed System Security*, pp. 16–26, 1997.

[*DOD 5220.22-M, Clearing and Sanitization Matrix*](#).

[*Domain Name System – Security Technical Implementation Guide version 4 release 1*](#), DISA Report, October 2007.

[*MACSec: A Security Solution that Protects LANS from Internal Threats*](#), IXIA Report, July 2008.

⁷² For the most recent versions of NIST publications, see: FIPS, <http://csrc.nist.gov/publications/PubsFIPS.html>; USGv6 500-series SPs, <http://www.antd.nist.gov/usgv6>; 800-series SPs, <http://csrc.nist.gov/publications/PubsSPs.html>.