ROUTING-SWITCHING TECHNOLOGIES AT COMPUTER NETWORKS AND IP v6

A Thesis Submitted to the

Graduate School of Natural and Applied Sciences of

Dokuz Eylül University

In Partial Fulfillment of the Requirements for

the Degree of Master of Science in Electrical & Electronics Engineering

by

Yaşar SARCAN

July, 2002

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We certify that we have read this thesis and **"ROUTING-SWITCHING TECHNOLOGIES AT COMPUTER NETWORKS AND IPv6"** completed by **Yaşar SARCAN** under supervision of **Asst.Prof.Dr. Zafer DİCLE** and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Supervisor

Yrd. Doç. Dr. Zafer Dicle

Prof. Dr. M. Gündüzalp

(Committee Member)

Doç. Dr. Yalçın Çebi

(Committee Member)

Approved by the
Graduate School of Natural and Applied Sciences

Prof.Dr. Cahit HELVACI
Director

ACKNOWLEDGMENTS

I would like to thank my supervisor Asst.Prof.Dr. Zafer DİCLE for his valuable guidance and support during the course of this thesis.

Yaşar SARCAN

ABSTRACT

The purpose of this thesis was to investigate the new Internet Protocol version 6 (IPv6), differences between current Internet Protocol version 4 (IPv4) and IPv6 and transition plans for migrating from IPv4 to IPv6. IPv6 is a new version of the Internet Protocol, designed as a successor to IPv4, the predominant protocol in use today. The changes from IPv4 to IPv6 are primarily in the following areas: expanded addressing capabilities; header format simplification; improved support for extensions and options; and consolidated authentication/privacy capabilities. There are different plans for transition to IPv6. All of this plans and strategies are investigated in this thesis. Current routing and switching technologies for internetworks and changes of these technologies in IPv6 were also investigated.

ÖZET

Bu tezin amacı, yeni internet protokolü versiyon 6 (IPv6) özelliklerinin, şu andaki internet protokolü versiyon 4 (IPv4) ile farklılıklarının ve IPv4 den IPv6 ya geçiş planlarının araştırılmasıdır. IPv6 internet protokolünün yeni versiyonudur. IPv6, günümüzde çok yaygın olarak kullanılan IPv4'ün yerine geçmesi için tasarlanmıştır. IPv6 ile IPv4 arasındaki öncelikli ve temel değişiklikler şu alanlarda olmuştur: daha geniş adresleme kapasitesi; paket önek yapısının basitleştirilmesi; daha gelişmiş eklenti ve opsiyonlar ve güçlendirilmiş güvenlik özellikleri. IPv6 ya geçmek için değişik planlar bulunmaktadır. Tüm bu planlar ve stratejiler bu tez içinde incelenmiştir. Günümüzde bilgisayar ağları arasında kullanılan yönlendirme ve anahtarlama teknolojileri ve IPv6 içinde bu teknolojilerde yapılan değişiklikler ayrıca incelenmiştir.

CONTENTS

Chapter One

INTRODUCTION

Chapter Two

INTERNETWORKING FUNDAMENTALS

Chapter Three

INTERNET PROTOCOL AND INTERNETWORK ADDRESS

Chapter Four
INTERNET PROTOCOL v6 (IPv6)

Chapter Five
IPv6 TRANSITION PLANS

Chapter Six

CONCLUSION

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1. Introduction

According to experts, the Internet as we know it will face a serious problem in a few years. Due to its rapid growth and the limitations in its design, there will be a point when no more free addresses are available for connecting to new hosts. At that point, no more new web servers can be set up, no more users can sign up for accounts at ISPs, and no more new machines can be set up to access the web or participate in online games.

Several approaches have been made to solve the problem. A very popular approach is to not assign a worldwide unique address to every user's machine, but rather to assign them "private" addresses, and hide several machines behind one official, globally unique address. This approach is called "Network Address Translation" (NAT, also known as "IP masquerading"). It has problems, as the machines hidden behind the global address can't be addressed, and as a result of this, opening connections to them peer to peer networking is not possible.

A different approach to the problem of Internet addresses getting scarce is to abandon the old Internet protocol with its limited addressing capabilities, and use a new protocol that does not have these limitations. The protocol, or actually, a set of protocols, used by machines connected to form today's Internet is known as the TCP/IP (Transmission Control Protocol, Internet Protocol) suite, and version 4 currently in use has all the problems described above. [Gai, S. (1999)]

Switching to a different protocol version that does not have these problems of course requires for a "better" version to be available. And actually, there is a better version. Version 6 of the Internet Protocol (IPv6) fulfills future demands on address space, and also addresses other features such as privacy, encryption, and better support of mobile computing. [Swartz,J. & Lammle,T.(2001)]

IP version 6 (IPv6) is a new version of the Internet Protocol, designed as a successor to IP version 4 (IPv4), the predominant protocol in use today. The changes from IPv4 to IPv6 are primarily in the following areas: expanded addressing capabilities; header format simplification; improved support for extensions and options; flow labeling capability; and consolidated authentication/privacy capabilities.

The current version of IP (IPv4) has not been substantially changed since RFC 791 was published in 1981. IPv4 has proven to be robust, easily implemented and interoperable, and has stood the test of scaling an internetwork to a global utility the size of today's Internet. This is a tribute to its initial design. However, the initial design did not anticipate the following:

- The recent exponential growth of the Internet and the impending exhaustion of the IPv4 address space.
- The growth of the Internet and the ability of Internet backbone routers to maintain large routing tables.
- The need for simpler configuration.
- The requirement for security at the IP level.
- The need for better support for real-time delivery of data-also called quality of service (QoS).

According to population estimates from the US Census Bureau, the world will be home to about 9 billion people in 2050. Whatever the economic constraints may be, we must clearly plan technically for all of these people to have potential Internet access. It would not be acceptable to produce a technology that simply could not scale to be accessible by the whole human population, under appropriate economic conditions. Furthermore, pervasive use of networked devices will probably mean many devices per person, not just one. Simple arithmetic tells us that the maximum of 4 billion public addresses allowed by the current IP version 4, even if backed up by the inconvenient techniques of private addresses and address translation, will simply be inadequate in the future. If the Internet is truly for everyone, we need more addresses, and IP version 6 is the only way to get them.

IPv6 has other benefits, such as provision for "plug and play" automatic configuration, which promises reduced complexity of network deployment and administration. Still, the principal benefit of IPv6 is that of having enough addresses thereby assisting in restoration of the end-to-end model on which the Internet was based. [Gai, S. (1999)]

In this thesis, the routing and switching technologies at computer networks and the new protocol IPv6 are investigated. And the transition plans for migrating from IPv4 to IPv6 are discussed.

In chapter 2, basic internetworking terms and fundamentals are examined. Open Sytem Interconnection Model and its layers are also studied. Internet protocols and IPv6's benefits are given in this chapter.

In chapter 3, Internet Protocol version 4 and today's internetwork addresses are investigated. Classless Inter Domain Routing (CIDR) and Network Address Translation (NAT) for IPv4 are also studied in this chapter.

The properties and benefits of IPv6 are investigated in chapter 4. Stateless autoconfiguration, Neighbour discovery protocol, routing protocols for IPv6 and address types of IPv6 are studied in this chapter.

In chapter 5, transition plans for migrating from IPv4 to IPv6 are investigated and compared.Tunneling methods for transition are studied also.

Finally, a conclusion is given in Chapter 6.

CHAPTER TWO

INTERNETWORKING FUNDAMENTALS

2.1. An Introduction to Internetworking

Internetworking is the functional interconnection of two or more networks; the resources of each individual network become available to the users and machines connected to the other networks.

Internetworking requires a combination of technologies, addressing, and communications protocols. All these must be understood and adhered to universally throughout the internetwork. Many different devices can be used to build internetworks, including switches, bridges, and routers. Although the boundaries between these devices had historically been very distinct, technological advances have blurred these distinctions. Routers offered the unique capability to discover paths (or routes) through large and complex internetworks. More importantly, routers could compare different routes through a network to find the most efficient one between any given points in the network. Routing is still critical to internetworking. Routing is no longer a function of just standalone routers, however. Routing can be performed by computers attached to local area networks (LANs) or even by LAN switches. [Downes, K. (2000)]

The International Organization for Standardization (ISO) developed the OSI reference model to facilitate the open interconnection of computer systems. An open interconnection is one that can be supported in a multivendor environment. The reference model identifies and stratifies into logically ordered layers all the functions required to establish, use, define, and dismantle a communications session between two computers without regard for those computers' manufacturer or architecture. Implicit in this definition of the OSI reference model is the assumption that an unknown quantity of distance and networking gear separate the two communicating devices. Consequently, the model defines mechanisms for passing data between two machines that share the same LAN or WAN. More importantly, the model identifies functions that allow two machines that are halfway around the world from each other with no direct network connections to pass data between themselves. [Downes, K. (2000)]

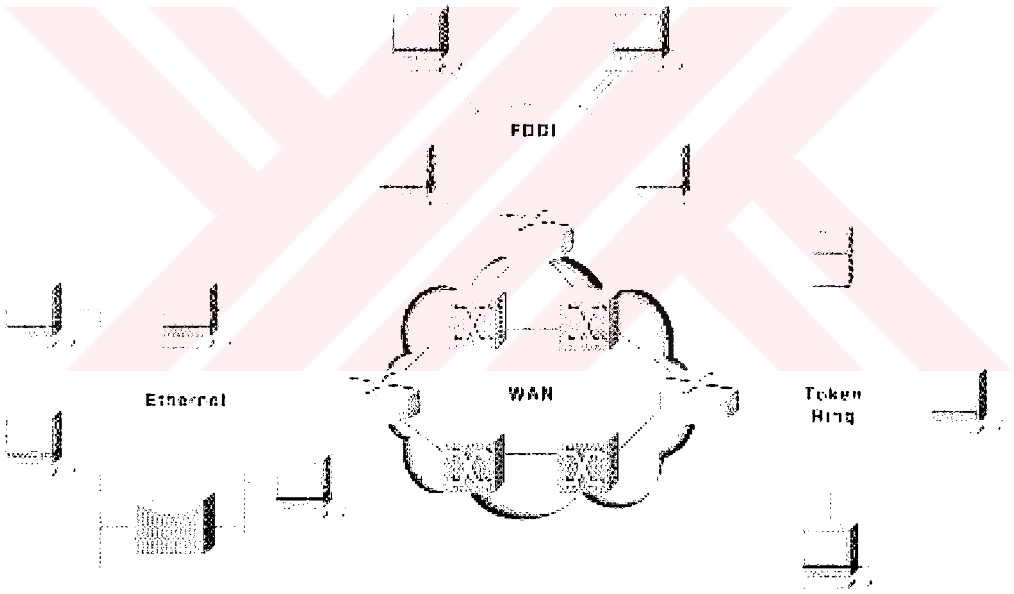


Figure 2. 1 Different Network Technologies Can Be Connected to Create an Internetwork

2.2.Open System Interconnection Reference Model

The *Open System Interconnection (OSI) reference model* describes how information from a software application in one computer moves through a network medium to a software application in another computer. The OSI reference model is a conceptual model composed of seven layers, each specifying particular network functions. The model was developed by the International Organization for Standardization (ISO) in 1984, and it is now considered the primary architectural model for intercomputer communications. The OSI model divides the tasks involved with moving information between networked computers into seven smaller, more manageable task groups. A task or group of tasks is then assigned to each of the seven OSI layers. Each layer is reasonably self-contained so that the tasks assigned to each layer can be implemented independently. This enables the solutions offered by one layer to be updated without adversely affecting the other layers. The following list details the seven layers of the Open System Interconnection (OSI) reference model:

- Layer 7—Application
- Layer 6—Presentation
- Layer 5—Session
- Layer 4—Transport
- Layer 3—Network
- Layer 2—Data link
- Layer 1—Physical

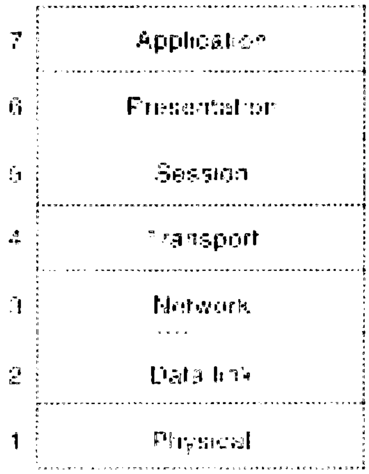


Figure 2. 2 The OSI Reference Model Contains Seven Independent Layers

Information being transferred from a software application in one computer system to a software application in another must pass through the OSI layers. For example, if a software application in System A has information to transmit to a software application in System B, the application program in System A will pass its information to the application layer (Layer 7) of System A. The application layer then passes the information to the presentation layer (Layer 6), which relays the data to the session layer (Layer 5), and so on down to the physical layer (Layer 1). At the physical layer, the information is placed on the physical network medium and is sent across the medium to System B. The physical layer of System B removes the information from the physical medium, and then its physical layer passes the information up to the data link layer (Layer 2), which passes it to the network layer (Layer 3), and so on, until it reaches the application layer (Layer 7) of System B. Finally, the application layer of System B passes the information to the recipient application program to complete the communication process.[Downes, K. (2000)]

A given layer in the OSI model generally communicates with three other OSI layers: the layer directly above it, the layer directly below it, and its peer layer in other networked computer systems. The data link layer in System A, for example, communicates with the network layer of System A, the physical layer of System A, and the data link layer in System B. Figure 2.3 illustrates this example.

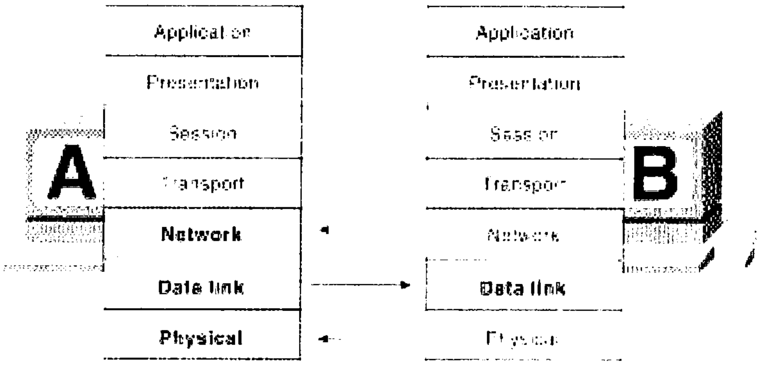


Figure 2. 3 OSI Model Layers Communicate with Other Layers

2.3. OSI Model Layers and Information Exchange

The seven OSI layers use various forms of control information to communicate with their peer layers in other computer systems. This *control information* consists of specific requests and instructions that are exchanged between peer OSI layers.

Control information typically takes one of two forms: headers and trailers. *Headers* are added to data that has been passed down from upper layers. *Trailers* are appended to data that has been passed down from upper layers. An OSI layer is not required to attach a header or a trailer to data from upper layers.

Headers, trailers, and data are relative concepts, depending on the layer that analyzes the information unit. At the network layer, for example, an information unit consists of a Layer 3 header and data. At the data link layer, however, all the information passed down by the network layer (the Layer 3 header and the data) is treated as data.

In other words, the data portion of an information unit at a given OSI layer potentially can contain headers, trailers, and data from all the higher layers. This is known as *encapsulation*. Figure 2.4 shows how the header and data from one layer are encapsulated into the header of the next lowest layer.

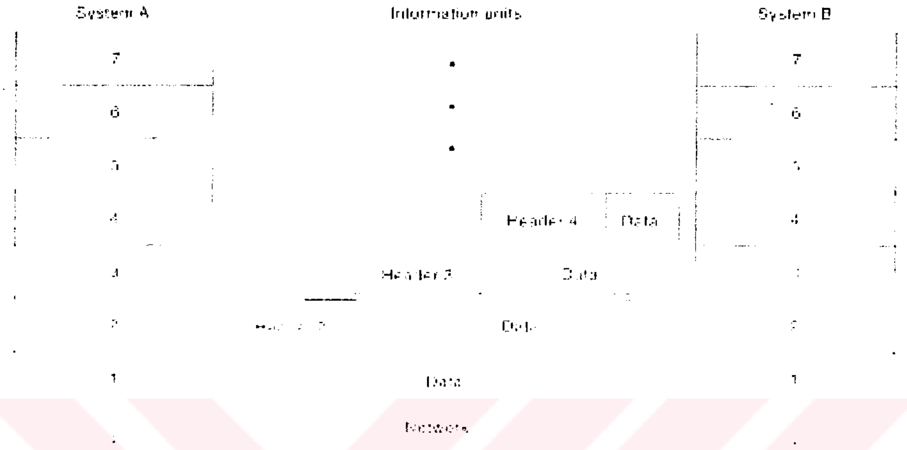


Figure 2. 4 Headers and Data Can Be Encapsulated During Information Exchange

The information exchange process occurs between peer OSI layers. Each layer in the source system adds control information to data, and each layer in the destination system analyzes and removes the control information from that data.

If System A has data from a software application to send to System B, the data is passed to the application layer. The application layer in System A then communicates any control information required by the application layer in System B by prepending a header to the data. The resulting information unit (a header and the data) is passed to the presentation layer, which prepends its own header containing control information intended for the presentation layer in System B. The information unit grows in size as each layer prepends its own header (and, in some cases, a trailer) that contains control information to be used by its peer layer in System B. At the physical layer, the entire information unit is placed onto the network medium.

The physical layer in System B receives the information unit and passes it to the data link layer. The data link layer in System B then reads the control information contained in the header prepended by the data link layer in System A. The header is then removed, and the remainder of the information unit is passed to the network layer. Each layer performs the same actions: The layer reads the header from its peer layer, strips it off, and passes the remaining information unit to the next highest layer. After the application layer performs these actions, the data is passed to the recipient software application in System B, in exactly the form in which it was transmitted by the application in System A.

2.4. OSI Model Layers

Physical layer

The physical layer defines the electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link between communicating network systems. Physical layer specifications define characteristics such as voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, and physical connectors. Physical layer implementations can be categorized as either LAN or WAN specifications. Figure 1-7 illustrates some common LAN and WAN physical layer implementations. [Feibel,W. (1996)]

Data link layer

The data link layer provides reliable transit of data across a physical network link. Different data link layer specifications define different network and protocol characteristics, including physical addressing, network topology, error notification, sequencing of frames, and flow control. Physical addressing (as opposed to network addressing) defines how devices are addressed at the data link layer. Network topology consists of the data link layer specifications that often define how devices are to be physically connected, such as in a bus or a ring topology. Error notification alerts upper-layer protocols that a transmission error has occurred, and the sequencing of data frames reorders frames that are transmitted out of sequence. Finally, flow control moderates the transmission of data so that the receiving device is not overwhelmed with more traffic than it can handle at one time. The Institute of Electrical and Electronics Engineers (IEEE) has subdivided the data link layer into two sublayers: Logical Link Control (LLC) and Media Access Control (MAC).

The *Logical Link Control (LLC)* sublayer of the data link layer manages communications between devices over a single link of a network. LLC is defined in the IEEE 802.2 specification and supports both connectionless and connection-oriented services used by higher-layer protocols. IEEE 802.2 defines a number of fields in data link layer frames that enable multiple higher-layer protocols to share a single physical data link. The *Media Access Control (MAC)* sublayer of the data link layer manages protocol access to the physical network medium. The IEEE MAC specification defines MAC addresses, which enable multiple devices to uniquely identify one another at the data link layer.[Feibel,W. (1996)]

Network layer

The network layer defines the network address, which differs from the MAC address. Some network layer implementations, such as the Internet Protocol (IP), define network addresses in a way that route selection can be determined systematically by comparing the source network address with the destination network address and applying the subnet mask. Because this layer defines the logical network layout, routers can use this layer to determine how to forward packets. Because of this, much of the design and configuration work for internetworks happens at Layer 3, the network layer.

Transport layer

The transport layer accepts data from the session layer and segments the data for transport across the network. Generally, the transport layer is responsible for making sure that the data is delivered error-free and in the proper sequence. Flow control generally occurs at the transport layer.

Flow control manages data transmission between devices so that the transmitting device does not send more data than the receiving device can process. Multiplexing enables data from several applications to be transmitted onto a single physical link. Virtual circuits are established, maintained, and terminated by the transport layer. Error checking involves creating various mechanisms for detecting transmission errors, while error recovery involves acting, such as requesting that data be retransmitted, to resolve any errors that occur. The transport protocols used on the Internet are TCP and UDP.

Session layer

The session layer establishes, manages, and terminates communication sessions. Communication sessions consist of service requests and service responses that occur between applications located in different network devices. These requests and responses are coordinated by protocols implemented at the session layer. Some examples of session-layer implementations include Zone Information Protocol (ZIP), the AppleTalk protocol that coordinates the name binding process; and Session Control Protocol (SCP), the DECnet Phase IV session layer protocol.

Presentation layer

The presentation layer provides a variety of coding and conversion functions that are applied to application layer data. These functions ensure that information sent from the application layer of one system would be readable by the application layer of another system. Some examples of presentation layer coding and conversion schemes include common data representation formats, conversion of character representation formats, common data compression schemes, and common data encryption schemes.

Common data representation formats, or the use of standard image, sound, and video formats, enable the interchange of application data between different types of computer systems. Conversion schemes are used to exchange information with systems by using different text and data representations, such as EBCDIC and ASCII. Standard data compression schemes enable data that is compressed at the source device to be properly decompressed at the destination. Standard data encryption schemes enable data encrypted at the source device to be properly deciphered at the destination.

Presentation layer implementations are not typically associated with a particular protocol stack. Some well-known standards for video include QuickTime and Motion Picture Experts Group (MPEG). QuickTime is an Apple Computer specification for video and audio, and MPEG is a standard for video compression and coding.

Among the well-known graphic image formats are Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), and Tagged Image File Format (TIFF). GIF is a standard for compressing and coding graphic images. JPEG is another compression and coding standard for graphic images, and TIFF is a standard coding format for graphic images. [Downes, K. (2000).]

Application layer

The application layer is the OSI layer closest to the end user, which means that both the OSI application layer and the user interact directly with the software application. This layer interacts with software applications that implement a communicating component. Such application programs fall outside the scope of the OSI model. Application layer functions typically include identifying communication partners, determining resource availability, and synchronizing communication.

When identifying communication partners, the application layer determines the identity and availability of communication partners for an application with data to transmit. When determining resource availability, the application layer must decide whether sufficient network resources for the requested communication exist. In synchronizing communication, all communication between applications requires cooperation that is managed by the application layer. Some examples of application layer implementations include Telnet, File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP).

2.5. Routing Basics

Routing is the act of moving information across an internetwork from a source to a destination. Along the way, at least one intermediate node typically is encountered. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer. The primary difference between the two is that bridging occurs at Layer 2 (the link layer) of the OSI reference model, whereas routing occurs at Layer 3 (the network layer). This distinction provides routing and bridging with different information to use in the process of moving information from source to destination, so the two functions accomplish their tasks in different ways. [Lammle,T. & Wallace,K. (2001)]

Routing involves two basic activities: determining optimal routing paths and transporting information groups (typically called packets) through an internetwork. In the context of the routing process, the latter of these is referred to as packet switching. Although packet switching is relatively straightforward, path determination can be very complex.

Routing protocols use metrics to evaluate what path will be the best for a packet to travel. A metric is a standard of measurement, such as path bandwidth, that is used by routing algorithms to determine the optimal path to a destination. To aid the process of path determination, routing algorithms initialize and maintain routing tables, which contain route information. Route information varies depending on the routing algorithm used. [Lammle,T. & Wallace,K. (2001)]

Routing algorithms fill routing tables with a variety of information. Destination/next hop associations tell a router that a particular destination can be reached optimally by sending the packet to a particular router representing the "next hop" on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with a next hop. Figure 2.5 depicts a sample destination/next hop routing table.

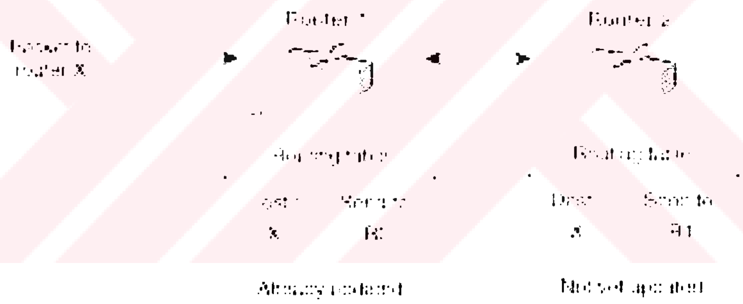


Figure 2. 5 Destination/Next Hop Associations Determine the Data's Optimal Path

Routing tables also can contain other information, such as data about the desirability of a path. Routers compare metrics to determine optimal routes, and these metrics differ depending on the design of the routing algorithm used. A variety of common metrics will be introduced and described later in this chapter.

Routers communicate with one another and maintain their routing tables through the transmission of a variety of messages. The routing update message is one such message that generally consists of all or a portion of a routing table. By analyzing routing updates from all other routers, a router can build a detailed picture of network topology. A link-state advertisement, another example of a message sent between routers, informs other routers of the state of the sender's links. Link information also can be used to build a complete picture of network topology to enable routers to determine optimal routes to network destinations.

2.6. Switching

Switching algorithms is relatively simple; it is the same for most routing protocols. In most cases, a host determines that it must send a packet to another host. Having acquired a router's address by some means, the source host sends a packet addressed specifically to a router's physical (Media Access Control [MAC]-layer) address, this time with the protocol (network layer) address of the destination host. [Lammle,T & Hales,K. (2000)]

As it examines the packet's destination protocol address, the router determines that it either knows or does not know how to forward the packet to the next hop. If the router does not know how to forward the packet, it typically drops the packet. If the router knows how to forward the packet, however, it changes the destination physical address to that of the next hop and transmits the packet.

The next hop may be the ultimate destination host. If not, the next hop is usually another router, which executes the same switching decision process. As the packet moves through the internetwork, its physical address changes, but its protocol address remains constant, as illustrated in Figure 2.6.

The preceding discussion describes switching between a source and a destination end system. The International Organization for Standardization (ISO) has developed a hierarchical terminology that is useful in describing this process. Using this terminology, network devices without the capability to forward packets between subnetworks are called *end systems (ESs)*, whereas network devices with these capabilities are called *intermediate systems (ISs)*. ISs are further divided into those that can communicate within routing domains (*intradomain ISs*) and those that communicate both within and between routing domains (*interdomain ISs*). A routing domain generally is considered a portion of an internetwork under common administrative authority that is regulated by a particular set of administrative guidelines. Routing domains are also called autonomous systems. With certain protocols, routing domains can be divided into routing areas, but intradomain routing protocols are still used for switching both within and between areas.[Lammle,T & Hales,K. (2000)]

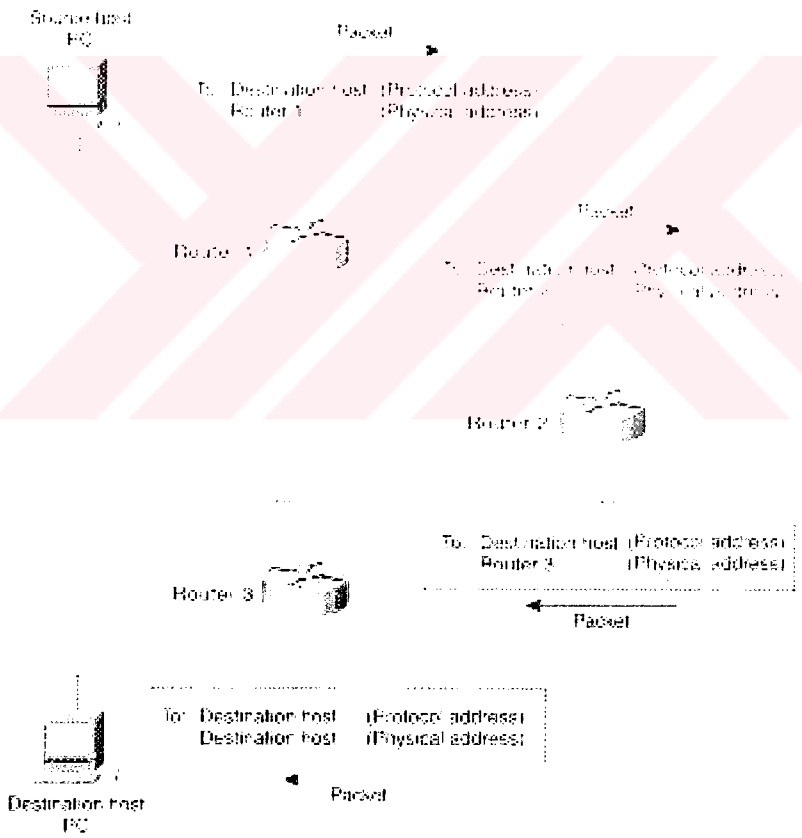

Source host
PC
Packet
To: Destination host (Protocol address)
Router 1 (Physical address)
Router 1
Packet
To: Destination host (Protocol address)
Router 2 (Physical address)
Router 2
To: Destination host (Protocol address)
Router 3 (Physical address)
Router 3
Packet
To: Destination host (Protocol address)
Destination host (Physical address)
Packet
Destination host
PC

Figure 2. 6 Numerous Routers May Come into Play During the Switching Process

2.7. Internet Protocols

The Internet protocols are the world's most popular open-system (nonproprietary) protocol suite because they can be used to communicate across any set of interconnected networks and are equally well suited for LAN and WAN communications. The Internet protocols consist of a suite of communication protocols, of which the two best known are the Transmission Control Protocol (TCP) and the Internet Protocol (IP). The Internet protocol suite not only includes lower-layer protocols (such as TCP and IP), but it also specifies common applications such as electronic mail, terminal emulation, and file transfer.

Internet protocols were first developed in the mid-1970s, when the Defense Advanced Research Projects Agency (DARPA) became interested in establishing a packet-switched network that would facilitate communication between dissimilar computer systems at research institutions. With the goal of heterogeneous connectivity in mind, DARPA funded research by Stanford University and Bolt, Beranek, and Newman (BBN). The result of this development effort was the Internet protocol suite, completed in the late 1970s. TCP/IP later was included with Berkeley Software Distribution (BSD) UNIX and has since become the foundation on which the Internet and the World Wide Web (WWW) are based. To illustrate the scope of the Internet protocols, Figure 2.7 maps many of the protocols of the Internet protocol suite and their corresponding OSI layers.

The Internet Protocol (IP) is a network-layer (Layer 3) protocol that contains addressing information and some control information that enables packets to be routed. IP is documented in RFC 791 and is the primary network-layer protocol in the Internet protocol suite. Along with the Transmission Control Protocol (TCP), IP represents the heart of the Internet protocols. IP has two primary responsibilities: providing connectionless, best-effort delivery of datagrams through an internetwork; and providing fragmentation and reassembly of datagrams to support data links with different maximum-transmission unit (MTU) sizes.[Lammle,T. & Wallace,K. (2001)]

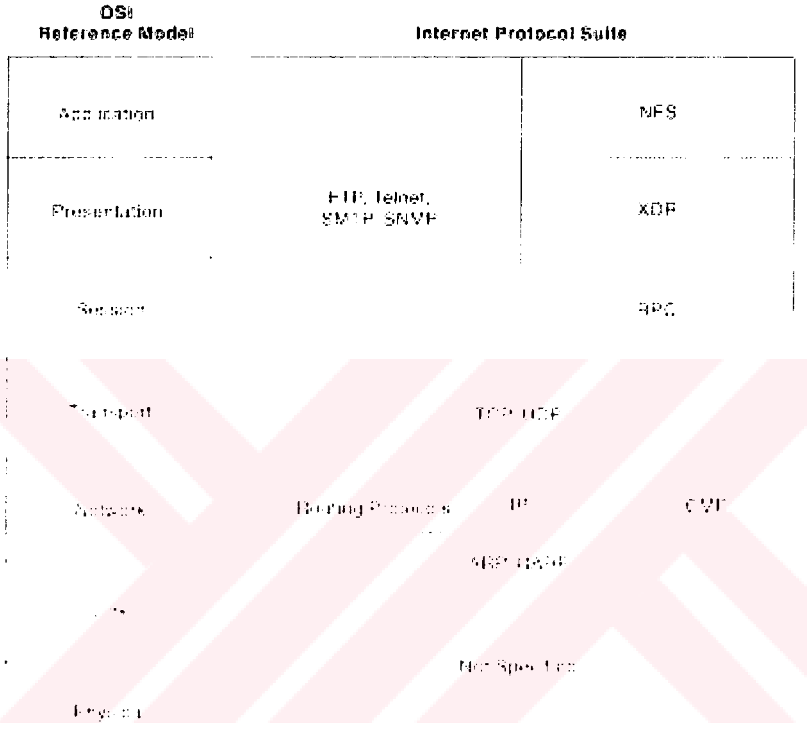


Figure 2. 7 Internet protocols span the complete range of OSI model layers.

2.8. IP Addressing

An IP (Internet Protocol) address is a unique identifier for a node or host connection on an IP network. An IP address is a 32 bit binary number usually represented as 4 decimal values, each representing 8 bits, in the range 0 to 255 (known as octets) separated by decimal points. This is known as "dotted decimal" notation. [Downes, K. (2000)]

Example: 140.179.220.200

It is sometimes useful to view the values in their binary form.

140 .179 .220 .200

10001100.10110011.11011100.11001000

Every IP address consists of two parts, one identifying the network and one identifying the node. The Class of the address and the subnet mask determine which part belongs to the network address and which part belongs to the node address.

There are 5 different address classes. We can determine which class any IP address is in by examining the first 4 bits of the IP address.

- Class A addresses begin with 0xxx, or 1 to 126 decimal.
- Class B addresses begin with 10xx, or 128 to 191 decimal.
- Class C addresses begin with 110x, or 192 to 223 decimal.
- Class D addresses begin with 1110, or 224 to 239 decimal.
- Class E addresses begin with 1111, or 240 to 254 decimal.

Addresses beginning with 01111111, or 127 decimal, are reserved for loopback and for internal testing on a local machine. Class D addresses are reserved for multicasting. Class E addresses are reserved for future use. They should not be used for host addresses. Now we can see how the Class determines, by default, which part of the IP address belongs to the network (N) and which part belongs to the node (n).

- Class A -- NNNNNNNN.nnnnnnnn.nnnnnnn.nnnnnnn
- Class B -- NNNNNNNN.NNNNNNNN.nnnnnnnn.nnnnnnnn
- Class C -- NNNNNNNN.NNNNNNNN.NNNNNNNN.nnnnnnnn

In the example, 140.179.220.200 is a Class B address so by default the Network part of the address (also known as the Network Address) is defined by the first two octets (140.179.x.x) and the node part is defined by the last 2 octets (x.x.220.200). In order to specify the network address for a given IP address, the node section is set to all "0"s. In our example, 140.179.0.0 specifies the network address for 140.179.220.200. When the node section is set to all "1"s, it specifies a broadcast that is sent to all hosts on the network. 140.179.255.255 specifies the example broadcast address.

2.9. IPv6 (IP The Next Generation)

The current version of IP (IPv4) has not been substantially changed since RFC 791 was published in 1981. IPv4 has proven to be robust, easily implemented and interoperable, and has stood the test of scaling an internetwork to a global utility the size of today's Internet. This is a tribute to its initial design. However, the initial design did not anticipate the following:

- The recent exponential growth of the Internet and the impending exhaustion of the IPv4 address space.
- The growth of the Internet and the ability of Internet backbone routers to maintain large routing tables.
- The need for simpler configuration.
- The requirement for security at the IP level.
- The need for better support for real-time delivery of data-also called quality of service (QoS).

To address these concerns, IP-The Next Generation (IPng) or IPv6, incorporates the concepts of many proposed methods for updating the IPv4 protocol. The design of IPv6 is intentionally targeted for minimal impact on upper and lower layer protocols by avoiding the random addition of new features. [Gai, S. (1999)]

IPv6 is designed as an evolution from IPv4 rather than as a radical change. Ease of transition is a key point in the design of IPv6. It is not something was added in at the end. IPv6 is designed to interoperate with IPv4. Specific mechanisms (embedded IPv4 addresses, pseudo- checksum rules etc.) were built into IPv6 to support transition and compatibility with IPv4. It was designed to permit a gradual and piecemeal deployment with a minimum of dependencies. Useful features of IPv4 were carried over in IPv6 and less useful features were dropped. According to the IPv6 specification, the changes from IPv4 to IPv6 fall primarily into the following categories:

- *Large Address Space*

The IP address size is increased from 32 bits to 128 bits in IPv6, supporting a much greater number of addressable nodes. Although 128 bits can express over 3.4x1038 possible combinations, the large address space of IPv6 has been designed to allow for multiple levels of subnetting and address allocation from the Internet backbone to the individual subnets within an organization. Even though only a small number of the possible addresses are currently allocated for use by hosts, there are plenty of addresses available for future use. With a much larger number of available addresses, address-conservation techniques, such as the deployment of NATs, are no longer necessary.

- *Efficient and hierarchical addressing and routing infrastructure*

IPv6 supports large hierarchical addresses which will allow the Internet to continue to grow and provide new routing capabilities not built into IPv4. It has anycast addresses which can be used for policy route selection and has scoped multicast addresses which provide improved scalability over IPv4 multicast. It also has local use address mechanisms which provide the ability for "plug and play" installation.

- *New Header Format*

Some IPv4 header fields have been dropped or made optional to reduce the necessary amount of packet processing and to limit the bandwidth cost of the IPv6 header.
IPv4 headers and IPv6 headers are not interoperable. A host or router must use an implementation of both IPv4 and IPv6 in order to recognize and process both header formats. The new IPv6 header is only twice as large as the IPv4 header, even though IPv6 addresses are four times as large as IPv4 addresses.

- *Improved Support for Extensions and Options*

IPv6 header options are encoded in such a way to allow for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing n ew options in the future. Some fields of an IPv4 header have been made optional in IPv6.

- *Better support for QoS*

A new quality-of-service (QOS) capability has been added to enable the labeling of packets belonging to particular traffic "flows" for which the sender requests special handling, such as real-time service.

- *Built-in security*

Extensions to support security options, such as authentication, data integrity, and data confidentiality, are built-in to IPv6.

- *Stateless and stateful address configuration*

To simplify host configuration, IPv6 supports both stateful address configuration, such as address configuration in the presence of a DHCP server, and stateless address configuration (address configuration in the absence of a DHCP server). With stateless address configuration, hosts on a link automatically configure themselves with IPv6 addresses for the link (called link-local addresses) and with addresses derived from prefixes advertised by local routers. Even in the absence of a router, hosts on the same link can automatically configure themselves with link-local addresses and communicate without manual configuration.

- *New protocol for neighboring node interaction*

The Neighbor Discovery protocol for IPv6 is a series of Internet Control Message Protocol for IPv6 (ICMPv6) messages that manage the interaction of neighboring nodes (nodes on the same link). Neighbor Discovery replaces the broadcast-based Address

Resolution Protocol (ARP), ICMPv4 Router Discovery, and ICMPv4 Redirect messages with efficient multicast and unicast Neighbor Discovery messages. [Gai, S. (1999)]

CHAPTER THREE

INTERNET PROTOCOL AND INTERNETWORK ADDRESSES

3.1. The Network Layer Protocols

There are two main reasons for the Network layer's existence: routing, and providing a single network interface to the upper layers. None of the upper-layer protocols, and none of the ones on the lower layer, have any functions relating to routing. The complex and important task of routing is the job of the Network layer. The second reason for the Internet layer is to provide a single network interface to the upper-layer protocols. Without this layer, application programmers would need to write “hooks” into every one of their applications for each different Network Access protocol. It would lead to different versions of each application, one for Ethernet, another one for Token Ring, and so on. To prevent this, IP provides one single network interface for the upper-layer protocols. Once that's accomplished, it's then the job of IP and the various Network Access protocols to get along and work together. [Sportack, M. (1999)]

All network roads lead to IP. And all the other protocols at this layer, as well as all those at the upper layers, use it. All paths through the model go through IP. The protocols that work at the Internet layer are:

- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- Address Resolution Protocol (ARP)

- Reverse Address Resolution Protocol (RARP)

There are actually two types of networking protocols that operate at Layer 3: routed protocols and routing protocols. Routing protocols are used in communications between routers. Routers need to communicate with each other about routes, their status, and availability.

Routed protocols are those that encapsulate user information and data into packets, and transport packets of data to their destinations. These protocols provide addressing that can be accessed and interpreted by routers. The routers then forward that data across unspecified distances, beyond the domain of the sender's LAN, to wherever the destination may be. Nonrouted protocols also exist; they perform a similar function to routed protocols. Unlike routed protocols, routers aren't designed to access and interpret their header information. For that matter, nonrouted protocols aren't designed to be forwarded across wide-area networks. [Sportack, M. (1999)]

3.2. The Internet Protocol (IP)

IP was developed approximately 20 years ago for the U.S. Department of Defense (DoD). The DoD needed a way to interconnect the various brands of proprietary computers and their equally proprietary support networks across a common internetwork. This was achieved by way of a layered protocol that insulated applications from networking hardware.

The Internet Protocol (IP) essentially is the Network layer. The other protocols found here merely exist to support it. IP looks at each packet's IP address. Then, using a routing table, it decides where a packet is to be sent next, choosing the best path. The Network Access–layer protocols at the bottom of the model don't possess IP's enlightened scope of the entire network; they deal only with physical links (local networks). [Wegner, J.D. (1999)]

Identifying devices on networks requires answering these two questions: Which network is it on? And what is its ID on that network? The first answer is the software, or logical address. The second answer is the hardware address. All hosts on a network have a logical ID called an IP address. This is the software, or logical, address and it contains valuable encoded information greatly simplifying the complex task of routing. IP receives segments from the transport layer and fragments them into datagrams (packets). IP then reassembles datagrams back into segments on the receiving side. Each datagram is assigned the IP address of the sender and the IP address of the recipient. Each router (layer-3 device) that receives a datagram makes routing decisions based upon the packet's destination IP address. Figure 3.1 shows an IP header.

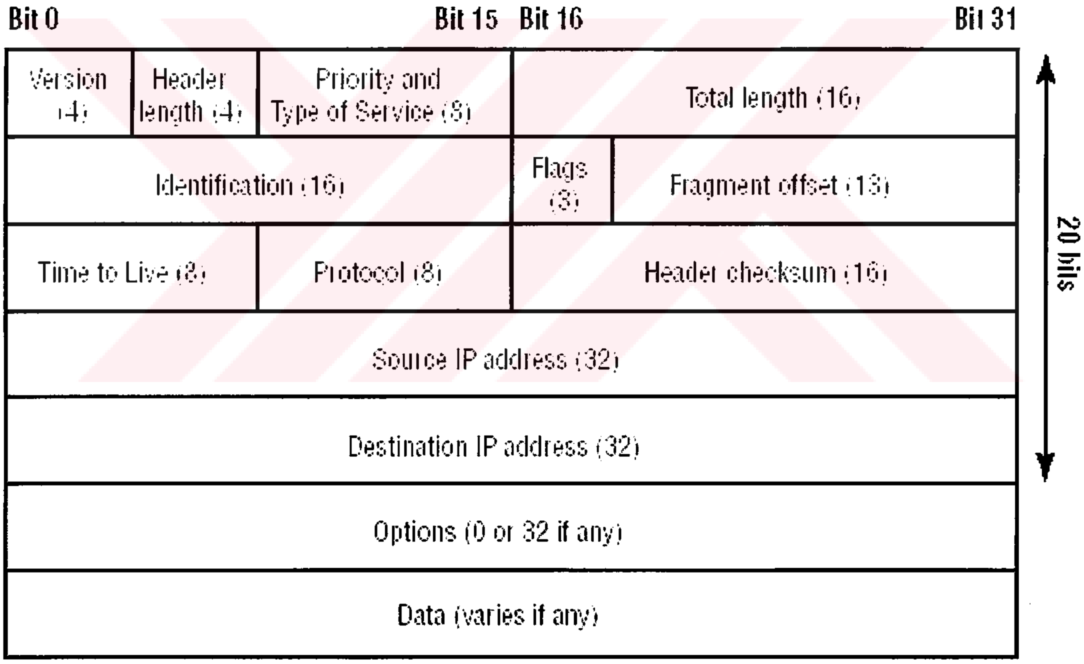


Figure 3. 1 IP Header

The following fields make up the IP header:

Version: IP version number.

HLEN: Header length in 32-bit words.

Priority or TOS: Type of Service tells how the datagram should be handled. The first three bits are the priority bits.

Total Length: The length of the packet including header and data.

Identification: Unique IP packet value.

Flags Specifies whether fragmentation should occur.

Frag Offset: These provide fragmentation and reassembly if the packet is too large to put in a frame. Allows different Maximum Transmission Sizes (MTU) on the Internet.

TTL: Time to Live. This is set into a packet when it is originally generated. It gives it a time to live. If it doesn't get to where it wants to go before the TTL expires, boom, it's gone. This stops IP packets from continuously circling the network looking for a home.

Protocol: Port of upper-layer protocol (TCP is port 6, or UDP is port 17 [hex]).

Header checksum: Cyclic Redundancy Check on header only.

Source IP Address: 32-bit IP address of sending station.

Destination IP address: The 32-bit IP address of the station this packet is destined for.

IP Option: Used for network testing, debugging, security, and more.

Data: Upper-layer data.

3.3. IP Addressing

One of the most important topics in any discussion of Internet Protocol is IP addressing. An *IP address* is a numeric identifier assigned to each machine on an IP network. It designates the location of a device on the network. An IP address is a software address, not a hardware address—the latter is hardcoded on a network interface card (NIC) and used for finding hosts on a local network. IP addressing was designed to allow a host on one network to communicate with a host on a different network, regardless of the type of LANs the hosts are participating in.

An IP address is made up of 32 bits of information. These bits are divided into four sections, referred to as octets or bytes, each containing 1 byte (8 bits). You can depict an IP address using one of three methods:

- Dotted-decimal, as in 172.16.30.56
- Binary, as in 10101100.00010000.00011110.00111000
- Hexadecimal, as in AC 10 1E 38

All of these examples represent the same IP address. Although hexadecimal is not used as often as dotted-decimal or binary when IP addressing is discussed, we still might find an IP address stored in hexadecimal in some programs For example, Windows Registry stores a machine's IP address in hex. The 32-bit IP address is a structured, or hierarchical, address, as opposed to a flat, or nonhierarchical, address. Although either type of addressing scheme could have been used, the hierarchical variety was chosen for a good reason.

The advantage of this scheme is that it can handle a large number of addresses, namely 4.2 billion (a 32-bit address space with two possible values for each position-either 0 or 1-gives you 2^{32} , or approximately 4.2 billion).

The disadvantage of this scheme, and the reason it's not used for IP addressing, relates to routing. If every address were unique, all routers on the Internet would need to store the address of each and every machine on the Internet. This would make efficient routing impossible, even if only a fraction of the possible addresses were used. The solution to this dilemma is to use a two- or three-level, hierarchical addressing scheme that is structured by network and host, or network, subnet, and host.

This two or three-level scheme is comparable to a telephone number. The first section, the area code, designates a very large area. The second section, the prefix, narrows the scope to a local calling area. The final segment, the customer number, zooms in on the specific connection. IP addresses use the same type of layered structure. Rather than all 32 bits being treated as a unique identifier, as in flat addressing, a part of the address is designated as the network address, and the other part is designated as either the subnet and host or just the node address. [Wegner, J.D. (1999)]

3.4. Network Addressing

The network address uniquely identifies each network. Every machine on the same network shares that network address as part of its IP address. In the IP address 172.16.30.56, for example, 172.16 is the network address. The node address is assigned to, and uniquely identifies, each machine on a network. This part of the address must be unique because it identifies a particular machine an individual, as opposed to a network, which is a group. This number can also be referred to as a host address. In the sample IP address 172.16.30.56, the node address is 30.56. The designers of the Internet decided to create classes of networks based on network size. For the small number of networks possessing a very large number of nodes, they created the rank Class A network. At the other extreme is the Class C network, which is reserved for the numerous networks with a small number of nodes. The class distinction for networks between very large and very small is predictably called the Class B network. Subdividing an IP address into a network and node address is determined by the class designation of one's network. Figure 3.2 provides us with a summary of the three classes of networks, plus the D and E class addresses not used for assigning hosts in production networks. While Class D addresses are not assigned as the primary IP number of a node, these addresses are used for multicast applications,

	8 bits	8 bits	8 bits	8 bits
Class A:	Network	Host	Host	Host
Class B:	Network	Network	Host	Host
Class C:	Network	Network	Network	Host
Class D:	Multicast			
Class E:	Research			

Figure 3. 2 Summary of the Five Classes of Networks

To ensure efficient routing, Internet designers defined a mandate for the leading bits section of the address for each different network class. For example, since a router knows that a Class A network address always starts with a 0, the router might be able to speed a packet on its way after reading only the first bit of its address. This is where the address schemes define the difference between a Class A, a Class B, and a Class C address.

3.4.1. Network Address Range: Class A

The designers of the IP address scheme said that the first bit of the first byte in a Class A network address must always be off. This means a Class A address must be between 0 and 127. Here is how those numbers are defined:

0xxxxxxx If we turn the other 7 bits all off and then turn them all on, we will find your Class A range of network addresses.

00000000=0

01111111=127

So, a Class A network is defined in the first octet between 0 and 127. It can't be less or more.

3.4.2. Network Address Range: Class B

In a Class B network, the RFCs state that the first bit of the first byte must always be turned on, but the second bit must always bit turned off. If we turn the other 6 bits all off and then all on, you will find the range for a Class B network:

10000000=128

10111111=191

This means that a Class B network can be defined when the first byte is configured from 128 to 191.

3.4.3. Network Address Range: Class C

For Class C networks, the RFCs define the first two bits of the first octet always turned on, but the third bit can never be on. Following the same process as the previous classes, convert from binary to decimal to find the range.

Here is the range for a Class C network:

11000000=192

11011111=223

This means that a Class C network can be defined when the first byte is configured from 192 to 223.

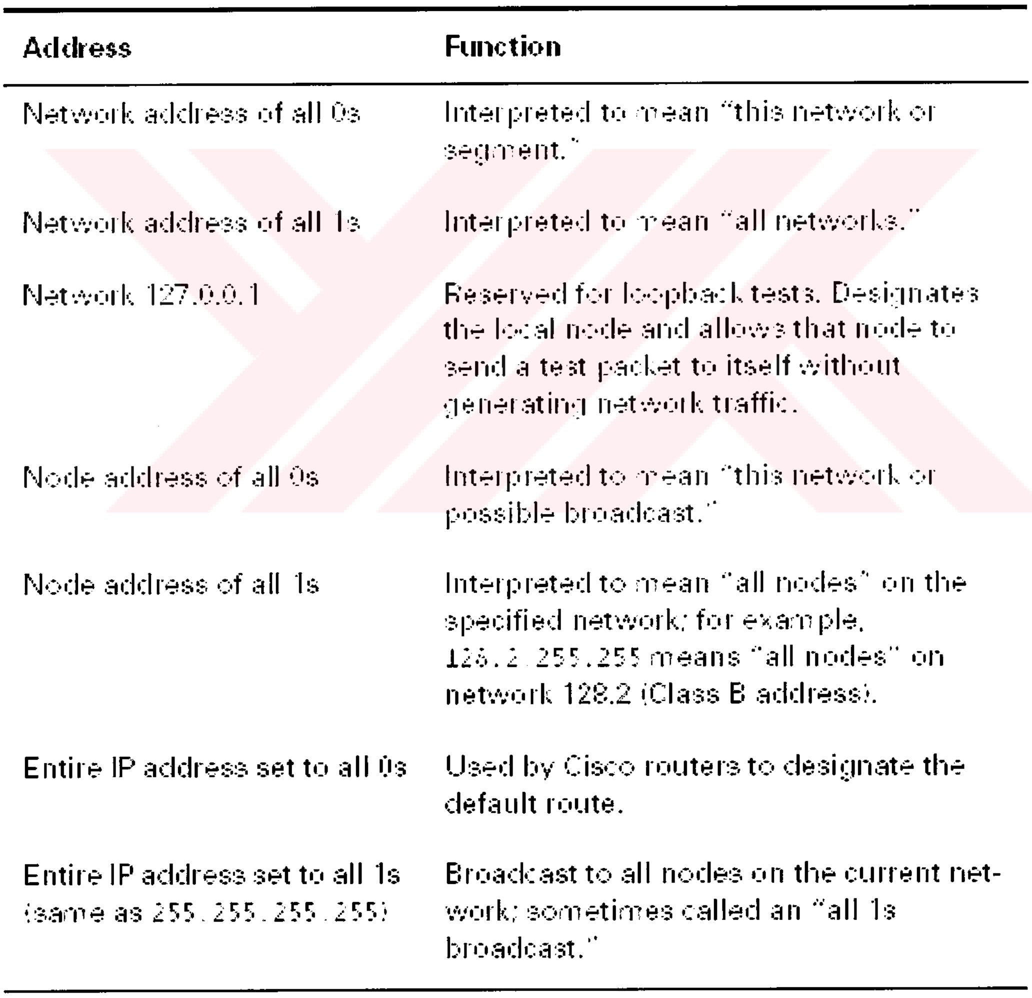
3.4.5. Network Address Ranges: Classes D and E

The addresses between 224 and 255 are reserved for Class D and E networks. Class D is used for multicast addresses and Class E for scientific purposes.

3.4.6. Network Addresses: Special Purpose

Some IP addresses are reserved for special purposes, and network administrators shouldn't assign these addresses to nodes. Table 3.1 lists the members of this exclusive little club and why they're included in it.

Table 3. 1 Reserved IP Addresses



Address	Function
Network address of all 0s	Interpreted to mean "this network or segment."
Network address of all 1s	Interpreted to mean "all networks."
Network 127.0.0.1	Reserved for loopback tests. Designates the local node and allows that node to send a test packet to itself without generating network traffic.
Node address of all 0s	Interpreted to mean "this network or possible broadcast."
Node address of all 1s	Interpreted to mean "all nodes" on the specified network; for example, 128.2.255.255 means "all nodes" on network 128.2 (Class B address).
Entire IP address set to all 0s	Used by Cisco routers to designate the default route.
Entire IP address set to all 1s (same as 255.255.255.255)	Broadcast to all nodes on the current network; sometimes called an "all 1s broadcast."

3.4.7. Private Networks

If we need large numbers of addresses on our network, and they do not need to be routed on the Internet, we can use private IP addresses that the Internet Assigned Numbers Authority (IANA) recommends. The following address ranges are designated as private networks that should not be advertised:

- 10.0.0.0 through 10.255.255.255
- 172.16.0.0 through 172.31.255.255
- 192.168.0.0 through 192.168.255.255

3.5. Address Classes

3.5.1. Class A Addresses

In a Class A network address, the first byte is assigned to the network address, and the three remaining bytes are used for the node addresses. The Class A format is:
Network.Node.Node.Node

For example, in the IP address 49.22.102.70, 49 is the network address, and 22.102.70 is the node address. Every machine on this particular network would have the distinctive network address of 49. Class A network addresses are one byte long, with the first bit of that byte reserved and the seven remaining bits available for manipulation. As a result, the maximum number of Class A networks that can be created is 128. Because each of the seven bit positions can either be a 0 or a 1, thus 2^7 or 128. To complicate matters further, the network address of all 0s (0000 0000) is reserved to designate the default route. Additionally, the address 127, which is reserved for diagnostics, can't be used either, which means that you can only use the numbers 1 to 126 to designate Class A network addresses. This means the actual number of usable Class A network addresses is 128 minus 2, or 126. Each Class A address has three bytes (24-bit positions) for the node address of a machine. Thus, there are 2^{24} or 16,777,216 unique combinations and, therefore, precisely that many possible unique node addresses for each Class A network. Because addresses with the two patterns of all 0s and all 1s are reserved, the actual maximum usable number of nodes for a Class A network is 2^{24} minus 2, which equals 16,777,214.

Here is an example of how to figure out the valid host IDs in a Class A network address:
10.0.0.0 All host bits off is the network address.
10.255.255.255 All host bits on is the broadcast address.
The valid hosts are the numbers in between the network address and the broadcast address: 10.0.0.1 through 10.255.255.254. Notice that 0s and 255s are valid host IDs. All you need to remember when trying to find valid host addresses is that the host bits cannot all be turned off or on at the same time. [Downes, K. (2000)]

3.5.2. Class B Addresses

In a Class B network address, the first two bytes are assigned to the network address, and the remaining two bytes are used for node addresses. The format is:
Network.Network.Node.Node

For example, in the IP address 172.16.30.56, the network address is 172.16, and the node address is 30.56. With a network address being two bytes of eight bits each, there would be 2^{16} unique combinations. But the Internet designers decided that all Class B network addresses should start with the binary digit 1 followed by 0. This leaves 14 bit positions to manipulate; therefore 16,384 (2^{14}) unique Class B network addresses. A Class B address use two bytes for node addresses. This is 216 minus the two reserved patterns (all 0s and all 1s), for a total of 65,534 possible node addresses for each Class B network. [Downes, K. (2000)]

Here is an example of how to find the valid hosts in a Class B network: 172.16.0.0 All host bits turned off is the network address. 172.16.255.255 All host bits turned on is the broadcast address. The valid hosts would be the numbers in between the network address and the broadcast address: 172.16.0.1 through 172.16.255.254.

3.5.3. Class C Addresses

The first three bytes of a Class C network address are dedicated to the network portion of the address, with only one measly byte remaining for the node address. The format is
Network.Network.Network.Node
Using the example IP address 192.168.100.102, the network address is 192.168.100, and the node address is 102. In a Class C network address, the first three bit positions are always the binary 110. The calculation is such: 3 bytes, or 24 bits, minus 3 reserved positions, leaves 21 positions. There are therefore 2^{21} or 2,097,152 possible Class C networks. Each unique Class C network has one byte to use for node addresses. This leads to 2^{8} or 256, minus the two reserved patterns of all 0s and all 1s, for a total of 254 node addresses for each Class C network. [Downes, K. (2000)]

Here is an example of how to find a valid host ID in a Class C network: 192.168.100.0 All host bits turned off is the network ID.
192.168.100.1 The first host.
192.168.100.254 The last host.
192.168.100.255 All host bits turned on is the broadcast address.
The valid hosts would be the numbers in between the network address and the broadcast address: 192.168.100.1 through 192.168.100.254.

Table 3. 2 IP Address Classes

Class	Address range	No. of network bits [No. of networks]	No. of host bits [No. of hosts per network]
A	1 - 126	8 [126]	24 [16.777,214]
B	128 - 191	16 [65,534]	16 [65.534]
C	192 - 223	24 [16,777.214]	8 [254]

This table shows the difference between the three main classes:

- Each individual class A network can have 16 million hosts! But there are only 126 networks in the class A range, so only the very largest of companies will ever get one.
- Class B addresses were used quite frequently but are now extremely difficult to obtain due to the rapidly diminishing address space. As shown above, a class B address can have approximately 65,000 hosts and there are about 65,000 class B networks.
- Class C network addresses are now the most frequently assigned as there are around 16,000,000 networks. Unfortunately, each class C network can only contain 254 hosts.

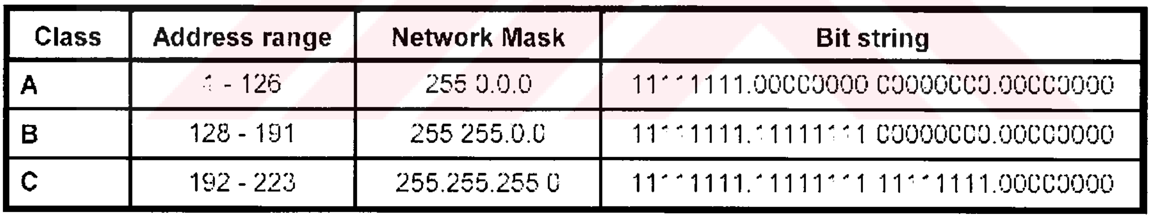
3.6. Subnetting

Subnetting an IP Network can be done for a variety of reasons, including organization, use of different physical media (such as Ethernet, FDDI, WAN, etc.), preservation of address space, and security. The most common reason is to control network traffic. In an Ethernet network, all nodes on a segment see all the packets transmitted by all the other nodes on that segment. Performance can be adversely affected under heavy traffic loads, due to collisions and the resulting retransmissions. A router is used to connect IP networks to minimize the amount of traffic each segment must receive.

3.6.1. Subnet Masking

Masking is the mechanism routers use to know which parts of the IP address are significant. Without masking, a router would have to look at every bit of every packet to determine what it should do. By using a mask, the router can ignore the part of the host address that only needs to work on network addresses. Every IP address implies its network mask in the first octet. The mask is also written in dotted decimal notation, and it indicates which bits of the address contain network address and which bits contain host address information.

Table 3. 3 Default Network Masks



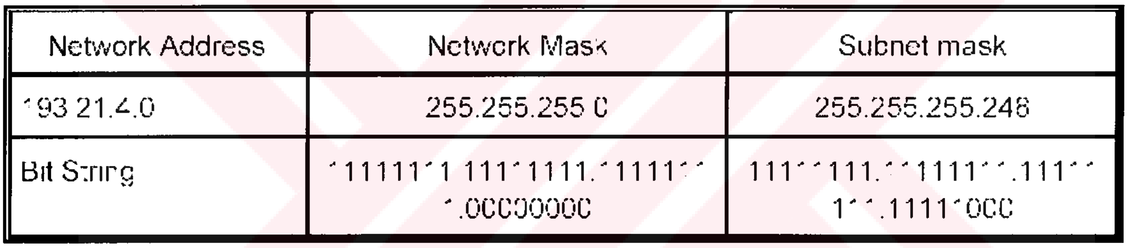
Class	Address range	Network Mask	Bit string
A	1 - 126	255 0.0.0	11111111.00000000 00000000.00000000
B	128 - 191	255 255.0.0	11111111.11111111 00000000.00000000
C	192 - 223	255.255.255 0	11111111.11111111 11111111.00000000

In table 3.3 the class C address, for example, has a network mask of 255.255.255.0. This tells the router that the first three octets (24 bits) contain network address information and the final octet only describes the host address. When a router has to decide what to do with a packet, it checks the masked address with its table, and then decides if it should forward the packet or not.

The other aspect of masking which has been used within a private network is s*ubnet masking.* This lets the administrator of the network break the host part of the network address into even smaller fragments which can be used to identify a particular geographic or departmental location.

A good example is of a company that has a class B address. They can use a subnet mask of one octet which effectively gives the organization 255 subnetworks, they have created what looks like a class B address with a class C mask. This is how network address space is typically managed. The network address is masked into smaller segments. With a class C address however, there is not much room for subnet masking as there are only eight bits (255 addresses) to play with. It is possible however to create a subnet on a class C address:

Table 3. 4 Subnet mask example



Network Address	Network Mask	Subnet mask
193 21.4.0	255.255.255 0	255.255.255.248
Bit String	11111111 11111111.11111111.00000000	11111111.11111111.11111111.11111000

In table 3.4, with this subnet mask, we can see that there are only three bits available for host addresses. The mask indicates how much of the address is to be used for network addressing. This mask indicates that each subnetwork can contain seven (2^3 - 1) hosts. There are 32 sub-networks with this mask (2^5).

Here is another, more detailed, example. We are assigned a Class C network number of 200.133.175.0 . We want to utilize this network across multiple small groups within an organization. We can do this by subnetting that network with a subnet address. We will break this network into 14 subnets of 14 nodes each. This will limit us to 196 nodes on the network instead of the 254 we would have without subnetting, but gives us the advantages of traffic isolation and security. To accomplish this, we need to use a subnet mask 4 bits long. Recall that the default Class C subnet mask is ;
255.255.255.0 (11111111.11111111.11111111.00000000 binary)

Extending this by 4 bits yields a mask of

255.255.255.240 (11111111.11111111.11111111.11110000 binary)

This gives us 16 possible network numbers, 2 of which cannot be used. Table 3.5 shows all possible subnetworks.

Table 3. 5 Subnetting example

Subnet bits	Network Number	Node Addresses	Broadcast Address
0000	200.133.175.0	Reserved	None
0001	200.133.175.16	.17 through .30	200.133.175.31
0010	200.133.175.32	.33 through .46	200.133.175.47
0011	200.133.175.48	.49 through .62	200.133.175.63
0100	200.133.175.64	.65 through .78	200.133.175.79
0101	200.133.175.80 .81	through .94	200.133.175.95
0110	200.133.175.96 .97	through .110	200.133.175.111
0111	200.133.175.112 .113	through .126	200.133.175.127
1000	200.133.175.128 .129	through .142	200.133.175.143
1001	200.133.175.144 .145	through .158	200.133.175.159
1010	200.133.175.160 .161	through .174	200.133.175.175
1011	200.133.175.176 .177	through .190	200.133.175.191
1100	200.133.175.192 .193	through .206	200.133.175.207
1101	200.133.175.208 .209	through .222	200.133.175.223
1110	200.133.175.224 .225	through .238	200.133.175.239
1111	200.133.175.240	Reserved	None

3.7. Classless Inter-Domain Routing (CIDR)

The idea of using classes was quite a good one when the Internet was small. For example, largecompanies like IBM could be given a class A address; medium-sized companies could receive a class B address; and very small organizations, a class C address.

Now that the Internet has grown so much, and many organizations that have only a few hundred employees have been given class B addresses (very wasteful) so that there are very few left. What happens to an organization which would like a registered Internet address and has a staff of one thousand? The answer is they are given multiple class C addresses (normally, multiples of eight). This is okay in that it gives the organization about 2,000 addresses, but it does mean that the Internet connection now has to advertise multiple network numbers and not just one. [Sportack, M. (1999)]

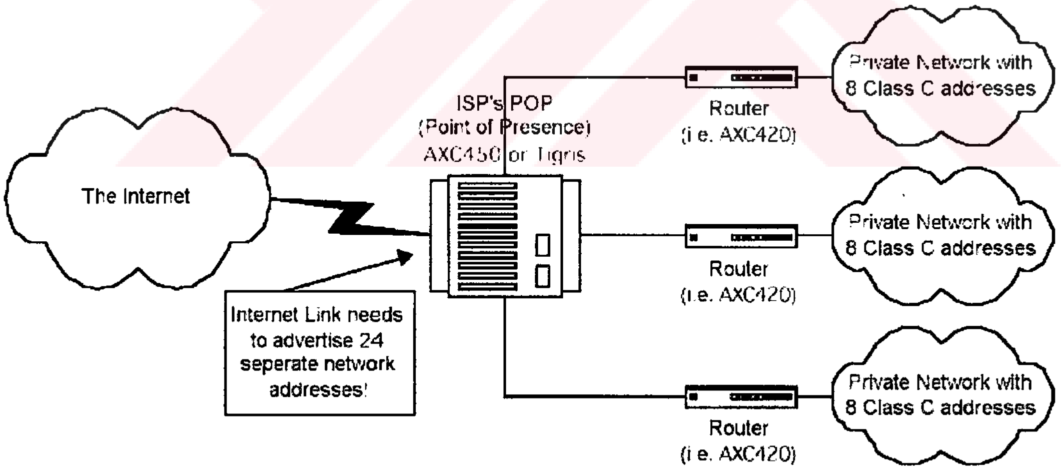

The Internet
ISP's POP
(Point of Presence)
AXC450 or Tigris
Internet Link needs to advertise 24 seperate network addresses!
Router
(i.e. AXC420)
Private Network with 8 Class C addresses
Router
(i.e. AXC420)
Private Network with 8 Class C addresses
Router
(i.e. AXC420)
Private Network with 8 Class C addresses

Figure 3. 3 ISP (Internet Service Provider) providing service to three companies,

In the figure 3.3, you can see that an ISP (Internet Service Provider) providing service to three companies, each one having eight contiguous class C network addresses, would have to advertise 24 separate network addresses. When this scenario gets multiplied by the real numbers in use in the Internet today, ISPs would have to advertise thousands of network addresses even though they may only have a limited number of customers. And in turn, the routers which connect the ISP to the Internet itself are connecting to many ISPs so they are seeing tens of thousands of network addresses. This situation has already started to happen and it is mainly for this reason that the Internet community came up with CIDR.

So we have seen two significant problems caused by the Internet's growth:

- By the use of class-based addresses many thousands, if not millions, of IP addresses have been wasted. (Think of a company with 500 employees which, five years ago, was assigned a class B address. They now have 65,000 addresses which no other organization can use.)
- The need to supply addresses based on multiple class C networks caused a massive growth in the size of routing tables and directly led to Internet network failures.

CIDR has helped the Internet community by getting around the problems of having to advertise routes for multiple IP network addresses. Basically, CIDR allows the router to aggregate multiple contiguous network addresses into a single route advertisement.

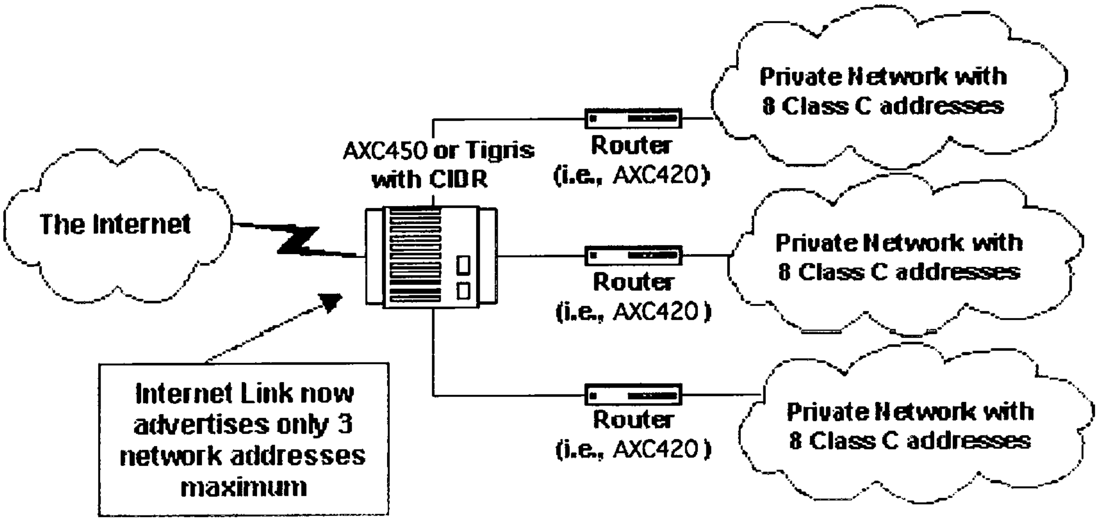

The Internet
AXC450 or Tigris with CIDR
Router (i.e., AXC420)
Router (i.e., AXC420)
Router (i.e., AXC420)
Private Network with 8 Class C addresses
Private Network with 8 Class C addresses
Private Network with 8 Class C addresses
Internet Link now advertises only 3 network addresses maximum

Figure 3. 4 ISP's concentrating router is using CIDR to group the individual addresses

In the figure 3.4, the concentrating router is using CIDR to group the individual addresses together before advertising to the Internet. If the service provider has supplied the three private networks with contiguous blocks of addresses, then the route advertisement would only have to contain a single address because CIDR could be used to aggregate all 24 network addresses.

CIDR removes the imposition of the A, B and C network address masks and allows the owner of the network to "super-net" multiple addresses together. As with subnet masking, CIDR uses a mask, but the mask used to indicate a CIDR address is less than the value of the network mask.

Table 3. 6 CIDR (supernet) mask

Class	Address range	Network Mask	CIDR (supernet) mask
C(x8)	193.21.4.0 to 193.21.11.0	255.255.255.0	255.255.248.0

With a router that supports CIDR, a route advertisement would be made which contains both the starting address (193.21.4.0) and the CIDR mask 255.255.248.0. This tells the network that there are eight contiguous network addresses starting at 193.21.4.0 and ending at 193.21.11.0. This facility can equally be applied to any type of network address, but is most commonly used with class C addresses.

For routers to be able to support CIDR, they must implement a routing protocol capable of advertising masks as well as addresses. This is one of the driving forces behind producing RIPv2, as the original version of RIP did not support this. OSPF can also be used to carry mask information. The other main issue is that original router software was often written to deliberately not allow users to enter a mask which was less than the network mask. Routing code and lookup tables need to be modified to ensure that they can generate and respond correctly to supernet masks. [Sportack, M. (1999)]

3.8. Network Address Translation (NAT)

NAT is a protocol that gives you the ability to map an inside IP address that is used in the local network environment to the outside network environment, and vice versa. There are many reasons for using NAT in network environment. Some of the benefits you will receive from NAT include the following:

- Enabling a private IP network to use nonregistered IP addresses to access an outside network such as the Internet
- Providing the ability to reuse assigned IP addresses that are already in use on the Internet
- Providing Internet connectivity in networks where there are not enough Internet-registered individual IP addresses

- Translating internal IP addresses assigned by old Internet Service Providers (ISPs) to a new ISP's newly assigned addresses without manually configuring the local network interfaces

NAT is configured on the router or route processor that is closest to the border of a stub domain, between the inside network (local network) and the outside network (public network such as an ISP or the Internet). (The outside network can also be another company, such as when two networks merge after an acquisition.) This is shown in Figure 3.5. The router separates the inside and outside network. NAT translates the internal local addresses into globally unique IP addresses, allowing data to flow into the outside network. [Lammle,T. & Wallace,K. (2001)]

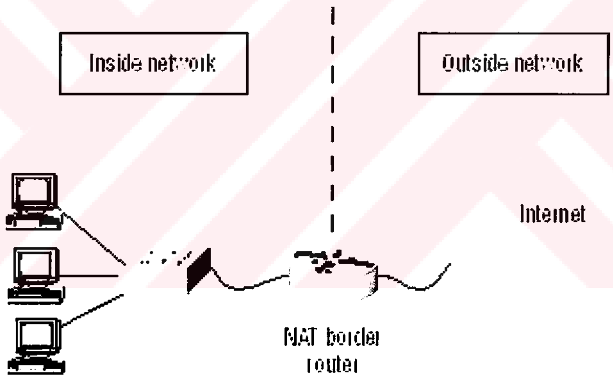


Figure 3. 5 The NAT Router on the Border of an Inside Network and an Outside Network Such

NAT takes advantage of the fact that there are relatively few network users using the outside network at any given time. NAT does this by using process switching to change the source address on the outbound packets, directing them back to the appropriate router. This allows for fewer IP addresses to be used than the number of hosts in the inside network.

NAT supports many traffic types. Let's take a look at these types in the following two sections. NAT supports the following traffic types:

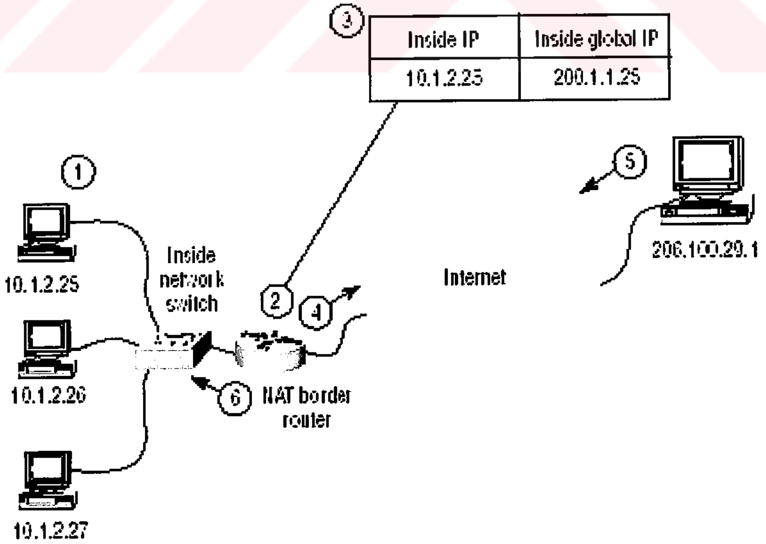
- TCP traffic that does not carry source and destination addresses in an application stream
- UDP traffic that does not carry source and destination addresses in an application stream
- Hypertext Transfer Protocol (HTTP)
- Trivial File Transfer Protocol (TFTP)
- File Transfer Protocol (FTP)
- Archie, which provides lists of anonymous FTP archives
- Finger, a software tool for determining whether a person has an account at a particular Internet site
- Network Time Protocol (NTP)
- Network File System (NFS)
- rlogin, rsh, rcp (TCP, Telnet, and UNIX entities to ensure the reliable delivery of data)
- Internet Control Message Protocol (ICMP)
- NetBIOS over TCP (datagram and name service only)
- Progressive Networks RealAudio
- White Pines CuSeeMe
- Xing Technologies StreamWorks
- DNS "A" and "PTR" queries
- H.323 (versions 12.0(1)/12.0(1)T or later)
- NetMeeting (versions 12.0(1)/12.0(1)T or later)
- VDOLive (versions 11.3(4)/11.3(4)T or later)
- Vxtreme (11.3(4)/11.3(4)T or later)
- Telnet
- Domain Name Service (DNS)

NAT does not support some traffic types, including the following:

- IP Multicast
- Routing table updates
- DNS zone transfers
- BOOTP
- Talk
- Ntalk
- Simple Network Management Protocol (SNMP)
- Netshow

3.8.1. Network Address Translation (NAT) Operation

NAT operates on a router and usually connects two networks. NAT translates the local non-unique (illegal to use on the Internet) IP addresses into legal, registered Internet IP addresses before placing packets from the local network to the Internet or other outside network. To do this, NAT uses a six-step process, as shown in Figure 3.6.



Inside IP	Inside global IP
10.1.2.25	200.1.1.25

Figure 3. 6 The Process of Translating Inside Local Addresses

The six-step process, as Figure 11.2 shows, is as follows:

1. User 10.1.2.25 sends a packet and attempts to open a connection to 206.100.29.1.

2. When the first packet arrives at the NAT border router, the router then checks to see if there is an entry for the source address that matches an outside address in the NAT table.

3. If a match is found in the NAT table, it continues to step 4. If a match is not found, the NAT router uses what is called a simple entry from its pool of legal Internet addresses. A simple entry occurs when the NAT router matches an illegal Internal IP address (such as the one we are using) to a registered legal Internet usable IP address. In this example, the NAT router will match the address of 10.1.2.25 to 200.1.1.25.

4. The NAT border router then replaces the local illegal address of 10.1.2.25 (listed as the packet's source address) with 200.1.1.25. This makes the destination host believe that the sending interface's IP address is 200.1.1.25.

5. When the host on the Internet using the IP address 206.100.29.1 replies, it lists the NAT router–assigned IP address of 200.1.1.25 as the destination address.

6. When the NAT border router receives the reply from 206.100.29.1 with the packet destined for 200.1.1.25, the NAT border router then checks its NAT table again. The NAT table shows that the internal address of 10.1.2.25 should receive the packet destined for 200.1.1.25 and replaces the destination address with the internal interface's IP address.

Steps 2 through 6 are repeated for each individual packet.

The following is a list of some of the disadvantages of using NAT compared to using individually configured, registered IP addresses on each network host:

- NAT increases latency (delay). Delays are introduced in the switching paths due to the sheer number of translations of each IP address contained in the packet headers. The router's CPU must be used to process every packet to decide whether the router needs to translate and change the IP header.
- NAT hides end-to-end IP addresses which renders some applications unusable. Some applications that require the use of physical addresses instead of a qualified domain name will not reach destinations when NAT translates the IP addresses across the NAT border router.

- Since NAT changes the IP address, there is a loss of IP end-to-end traceability. The multiple-packet address changes confuse IP tracing utilities. This provides one advantage from a security standpoint: it eliminates some of a hacker's ability to identify a packet's source.

CHAPTER FOUR

INTERNET PROTOCOL v6

(IPv6)

4.1.Introduction to IPv6

The Internet Protocol was introduced in the ARPANET in the mid-1970s. The version of IP in common use today is IP version 4 (IPv4), described in Request for Comments (RFC) 791 (September 1981). Although several protocol suites (including Open System Interconnection) have been proposed over the years to replace IPv4, none have succeeded because of IPv4's large, and continually growing, installed base. Nevertheless, IPv4 was never intended for the Internet that we have today, either in terms of the number of hosts, types of applications, or security concerns.

In the early 1990s, the Internet Engineering Task Force (IETF) recognized that the only way to cope with these changes was to design a new version of IP to become the successor to IPv4. The IETF formed the IP next generation (IPng) Working Group to define this transitional protocol to ensure long-term compatibility between the current and new IP versions, and support for current and emerging IP-based applications.

The communication networks and services are changing rapidly. The conventional circuit and packet switched networks are being replaced by next generation networks, primarily based on Internet Protocol. The rapid growth of web based services has lead to the explosive growth of the internet. However, the current internet protocol (IPv4), which is the backbone of transmission control protocol (TCP/IP) networking, is rapidly becoming obsolete, with the inherent problems related with limited address space, security and QoS features. The new protocol IPv6 has been developed to overcome all these problems and to provide solutions for the next generation networks. [Downes, K. (2000)]

IP version 6 (IPv6) is a new version of the Internet Protocol, designed as a successor to IP version 4 (IPv4), the predominant protocol in use today. The changes from IPv4 to IPv6 are primarily in the following areas: expanded addressing capabilities; header format simplification; improved support for extensions and options and consolidated authentication/privacy capabilities.

4.2. Why IPv6?

The current version of IP (IPv4) has not been substantially changed since RFC 791 was published in 1981. IPv4 has proven to be robust, easily implemented and interoperable, and has stood the test of scaling an internetwork to a global utility the size of today's Internet. This is a tribute to its initial design. However, the initial design did not anticipate the following:

- The recent exponential growth of the Internet and the impending exhaustion of the IPv4 address space.
- The growth of the Internet and the ability of Internet backbone routers to maintain large routing tables.
- The need for simpler configuration.
- The requirement for security at the IP level.
- The need for better support for real-time delivery of data-also called quality of service (QoS).

According to population estimates from the US Census Bureau, the world will be home to about 9 billion people in 2050. Whatever the economic constraints may be, we must clearly plan technically for all of these people to have potential Internet access. It would not be acceptable to produce a technology that simply could not scale to be accessible by the whole human population, under appropriate economic conditions. Furthermore, pervasive use of networked devices will probably mean many devices per person, not just one. Simple arithmetic tells us that the maximum of 4 billion public addresses allowed by the current IP version 4, even if backed up by the inconvenient techniques of private addresses and address translation, will simply be inadequate in the future. If the Internet is truly for everyone, we need more addresses, and IP version 6 is the only way to get them. IPv6 has other benefits, such as provision for "plug and play" automatic configuration, which promises reduced complexity of network deployment and administration. Still, the principal benefit of IPv6 is that of having enough addresses thereby assisting in restoration of the end-to-end model on which the Internet was based.

IPv6 provides:

- An expanded address space (32-bits to 128-bits)
- Required support for network layer security
- Improved real-time/Quality-of-Service (QoS) service support
- Required support for mobility
- New routing and addressing concepts
- Autoconfiguration

4.3. IPv6 Features

The Internet Protocol (IP) is a packet-based protocol used to exchange data, voice, and video traffic over digital networks. IP handles addressing, fragmentation, reassembly, and protocol demultiplexing. It is the foundation on which all other IP protocols (collectively referred to as the IP Protocol suite) are built. As a network-layer protocol, IP contains addressing and control information that allows data packets to be routed. IPv6, formerly called IPng (next generation), is a replacement for the current version of IP (version 4). [Swartz,J. & Lammle,T.(2001)]

4.3.1. Larger Address Space

The primary motivation for IPv6 is the need to meet the anticipated future demand for globally unique IP addresses. Applications such as mobile Internet-enabled devices (such as personal digital assistants (PDAs), telephones, and cars), home-area networks (HANs), and wireless data services are driving the demand for globally unique IP addresses. IPv6 quadruples the number of network address bits from 32 bits (in IPv4) to 128 bits, which provides more than enough globally unique IP addresses for every networked device on the planet. By being globally unique, IPv6 addresses inherently enable global reachability and end-to-end security for networked devices, functionality that is crucial to the applications and services that are driving the demand for the addresses. Additionally, the flexibility of the IPv6 address space reduces the need for private addresses and the use of Network Address Translation (NAT); therefore, IPv6 enables new application protocols that do not require special processing by border routers at the edge of networks. [Gai, S. (1999)]

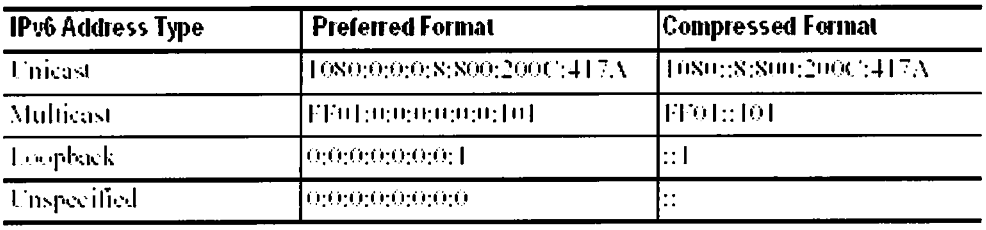
4.3.1.1. IPv6 Address Formats

IPv6 addresses are represented as a series of 16-bit hexadecimal fields separated by colons (:) in the format: x:x:x:x:x:x:x:x. Following are two examples of IPv6 addresses:
FEDC:BA98:7654:3210:FEDC:BA98:7654:3210
1080:0:0:0:8:800:200C:417A

It is common for IPv6 addresses to contain successive hexadecimal fields of zeros. To make IPv6 addresses less cumbersome, two colons (::) may be used to compress successive hexadecimal fields of zeros at the beginning, middle, or end of an IPv6 address (the colons represent successive hexadecimal fields of zeros). Table 4.1 lists compressed IPv6 address formats. A double colon may be used as part of the ipv6-

address argument when consecutive 16-bit values are denoted as zero. We can configure multiple IPv6 addresses per interfaces, but only one link-local address.

Table 4. 1 ***Compressed IPv6 Address Formats***



IPv6 Address Type	Preferred Format	Compressed Format
Unicast	1080:0:0:0:8:800:200C:417A	1080::8:800:200C:417A
Multicast	FF01:0:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

The loopback address listed in table 4.1 may be used by a node to send an IPv6 packet to itself. The loopback address in IPv6 functions the same as the loopback address in IPv4 (127.0.0.1).

The unspecified address listed in table 4.1 indicates the absence of an IPv6 address. For example, a newly initialized node on an IPv6 network may use the unspecified address as the source address in its packets until it receives its IPv6 address.

An IPv6 address prefix, in the format *ipv6-prefix/prefix-length*, can be used to represent bit-wise contiguous blocks of the entire address space. The *ipv6-prefix* variable must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The */prefix-length* variable is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example,
1080:6809:8086:6502::/64 is an acceptable IPv6 prefix.

4.3.1.2. IPv6 Address Types

Following are the three types of IPv6 addresses:

Unicast : An address for a single interface. A packet that is sent to a unicast address is delivered to the interface identified by that address. The Cisco IOS software supports the following IPv6 unicast address types:

- Global aggregatable address
- Site-local address
- Link-local address
- IPv4-compatible IPv6 address

Anycast : An address for a set of interfaces that typically belong to different nodes. A packet sent to an anycast address is delivered to the closest interface—as defined by the routing protocols in use—identified by the anycast address.

Multicast : An address for a set of interfaces (in a given scope) that typically belong to different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address (in a given scope).

4.3.1.2.1. Aggregatable Global Address

An aggregatable global address is an IPv6 address from the aggregatable global unicast prefix. The structure of aggregatable global unicast addresses enables strict aggregation of routing prefixes that limits the number of routing table entries in the global routing table. Aggregatable global addresses used on links are aggregated upward through organizations, then to intermediate-level Internet service providers (ISPs), and eventually to top-level ISPs. Figure 4.1 shows the structure of an aggregatable global address. [Downes, K. (2000)]

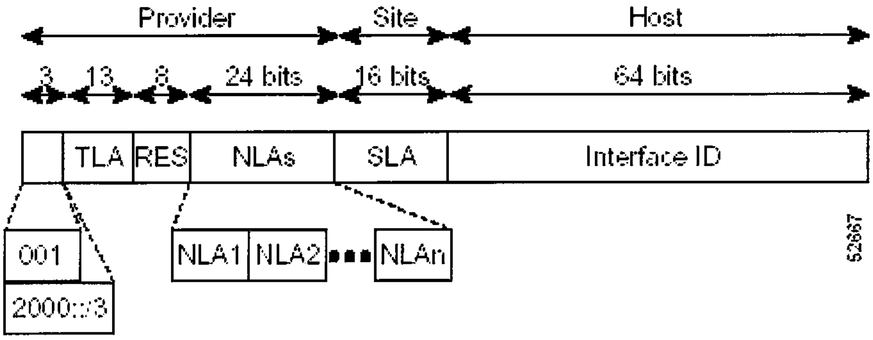


Figure 4. 1 Aggregatable Global Address Format

A fixed prefix of 2000::/3 (001) indicates an aggregatable global IPv6 address. Addresses with a prefix of 2000::/3 (001) through E000::/3 (111), excluding the FF00::/8 (1111 1111) multicast addresses, are required to have 64-bit interface identifiers in the modified EUI-64 format.

A Top-Level Aggregator (TLA) identifies tier 1 ISPs. TLAs are connected in a default-free zone. Routers in the default-free zone must have a default-free routing table entry for every active TLA identifier.

A field of 8 bits is a reserved field for the growth of the TLA and Next-Level Aggregator (NLA) fields. The reserved field must always be equal to zero.

An NLA identifies intermediate service providers assigned an NLA to create an addressing hierarchy and to identify sites. An organization can assign the top part of the NLA in a manner to create an addressing hierarchy appropriate to its network. It can use the remainder of the bits in the field to identify sites it wants to serve.

A Site-Level Aggregator (SLA) is used by individual organizations to create their own local addressing hierarchy and to identify subnets. An SLA is similar to a subnet in IPv4, except that an organization with an SLA has a much greater number of subnets to utilize; the 16-bit SLA field supports 65,535 individual subnets.

An interface identifier is used to identify interfaces on a link. The interface identifier must be unique to the link. They may also be unique over a broader scope. In many cases, an interface identifier will be the same as or based on the link-layer address of an interface. Interface identifiers used in aggregatable global unicast and other IPv6 address types must be 64 bits long and constructed in the modified EUI-64 format.

Interface identifiers are constructed in the modified EUI-64 format in one of the following ways:

- For all IEEE 802 interface types (for example, Ethernet, and FDDI interfaces), the first three octets (24 bits) are taken from the Organizationally Unique Identifier (OUI) of the 48-bit link-layer address (MAC address) of the interface, the fourth and fifth octets (16 bits) are a fixed hexadecimal value of FFFE, and the last three octets (24 bits) are taken from the last three octets of the MAC address. The construction of the interface identifier is completed by setting the Universal/Local (U/L) bit—the seventh bit of the first octet—to a value of 0 or 1. A value of 0 indicates a locally administered identifier; a value of 1 indicates a globally unique IPv6 interface identifier.
- For all other interface types (for example, serial, loopback, ATM, Frame Relay, and tunnel—except tunnel interfaces used with IPv6 overlay tunnels—interface types), the interface identifier is constructed in the same way as the interface identifier for IEEE 802 interface types; however, the first MAC address from the pool of MAC addresses in the router is used to construct the identifier (the MAC address from the interface is not used).
- For tunnel interface types that are used with IPv6 overlay tunnels, the interface identifier is the IPv4 address assigned to the tunnel interface with all zeros in the high-order 32 bits of the identifier.

If no IEEE 802 interface types are in the router, link-local IPv6 addresses are generated on the interfaces in the router in the following sequence:

1. The router is queried for MAC addresses (from the pool of MAC addresses in the router).

2. If no MAC addresses are available in the router, the serial number of the router is used to form the link-local addresses.

3. If the serial number of the router cannot be used to form the link-local addresses, the router uses a Message Digest 5 (MD5) hash (a "message digest") to determine the MAC address of the router from the host name of the router.

4.3.1.2.2. Site-Local Address

A site-local address is an IPv6 unicast address that uses the prefix FEC0::/10 (1111 111011) and concatenates the subnet identifier (the 16-bit SLA field) with the interface identifier in the modified EUI-64 format. Site-local addresses can be used to number a complete site without using a globally unique prefix. Site-local addresses can be considered private addresses because they can be used to restrict communication to a limited domain. Figure 4.2 shows the structure of a site-local address. [Downes, K. (2000)]

IPv6 routers must not forward packets that have site-local source or destination addresses outside of the site.

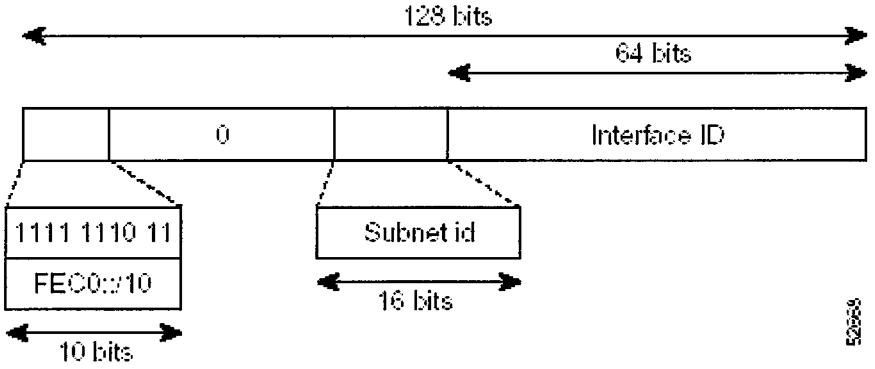


Figure 4. 2 Site-local Address Format

4.3.1.2.3. Link-Local Address

A link-local address is an IPv6 unicast address that can be automatically configured on any interface by using the link-local prefix FE80::/10 (1111 111010) and the interface identifier in the modified EUI-64 format. Link-local addresses are used in the neighbor discovery protocol and the stateless autoconfiguration process. Nodes on a local link can use link-local addresses to communicate; the nodes do not need site-local or globally unique addresses to communicate. Figure 4.3 shows the structure of a link-local address. IPv6 routers must not forward packets that have link-local source or destination addresses to other links. [Downes, K. (2000)]

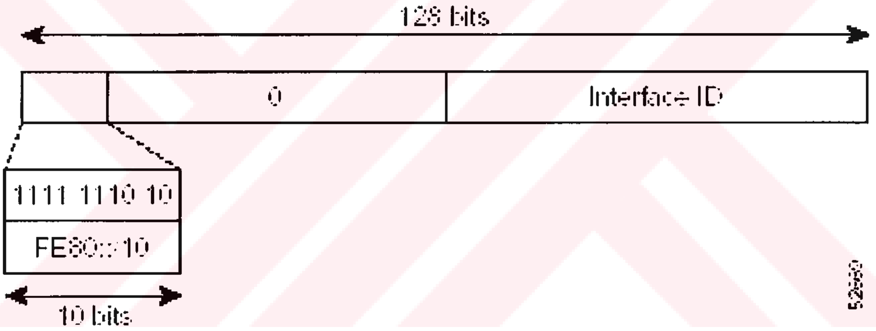


Figure 4. 3 Link-local Address Format

4.3.1.2.4. IPv4-compatible IPv6 Address

An IPv4-compatible IPv6 address is an IPv6 unicast address that has zeros in the high-order 96 bits of the address and an IPv4 address in the low-order 32 bits of the address. The format of an IPv4-compatible IPv6 address is 0:0:0:0:0:0:A.B.C.D or ::A.B.C.D. The entire 128-bit IPv4-compatible IPv6 address is used as the IPv6 address of a node and the IPv4 address embedded in the low-order 32-bits is used as the IPv4 address of the node. IPv4-compatible IPv6 addresses are assigned to nodes that support both the IPv4 and IPv6 protocol stacks and are used in automatic tunnels. Figure 4.4 shows the structure of an IPv4-compatible IPv6 address and a few acceptable formats for the address.

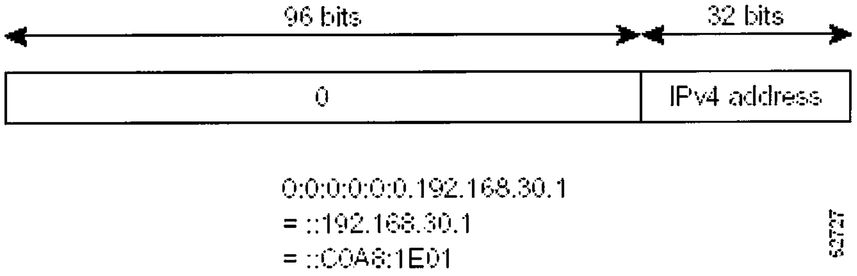


Figure 4. 4 IPv4-compatible IPv6 Address Format

4.3.1.2.5. Anycast Address

An anycast address is an address that is assigned to a set of interfaces that typically belong to different nodes. A packet sent to an anycast address is delivered to the closest interface—as defined by the routing protocols in use—identified by the anycast address. Anycast addresses are syntactically indistinguishable from unicast addresses because anycast addresses are allocated from the unicast address space. Assigning a unicast address to more than one interface makes a unicast address an anycast address. Nodes to which the anycast address is assigned must be explicitly configured to recognize that the address is an anycast address. Figure 4.5 shows the format of the subnet router anycast address; the address has a prefix concatenated by a series of zeros (the interface ID). The subnet router anycast address can be used to reach a router on the link that is identified by the prefix in the subnet router anycast address.

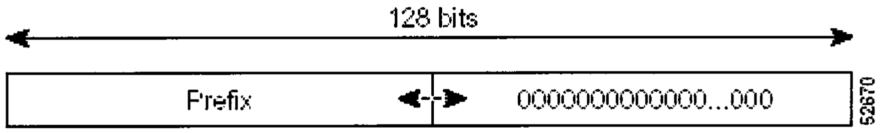


Figure 4. 5 Subnet Router Anycast Address Format

4.3.1.2.6. IPv6 Multicast Address

An IPv6 multicast address is an IPv6 address that has a prefix of FF00::/8 (1111 1111). An IPv6 multicast address is an identifier for a set of interfaces that typically belong to different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. The second octet following the prefix defines the lifetime and scope of the multicast address. A permanent multicast address has a lifetime parameter equal to 0; a temporary multicast address has a lifetime parameter equal to 1. A multicast address that has the scope of a node, link, site, or organization, or a global scope has a scope parameter of 1, 2, 5, 8, or E, respectively. For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link scope. Figure 4.6 shows the format of the IPv6 multicast address.

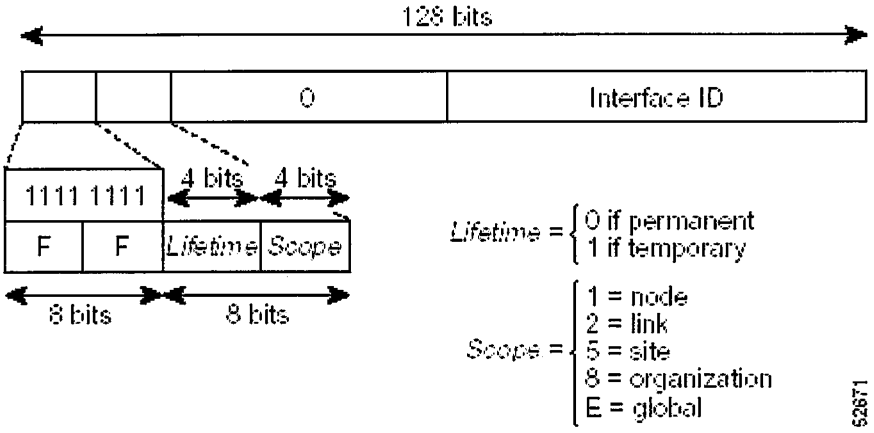


Figure 4. 6 IPv6 Multicast Address Format

IPv6 nodes (hosts and routers) are required to join (receive packets destined for) the following multicast groups:

- All-nodes multicast group FF02:0:0:0:0:0:0:1 (scope is link-local)
- Solicited-node multicast group FF02:0:0:0:0:1:FF00:0000/104 for each of its assigned unicast and anycast addresses

IPv6 routers must also join the all-routers multicast group FF02:0:0:0:0:0:0:2 (scope is link-local). The solicited-node multicast address is a multicast group that corresponds to an IPv6 unicast or anycast address. IPv6 nodes must join the associated solicited-node multicast group for every unicast and anycast address to which it is assigned. The IPv6 solicited-node multicast address has the prefix FF02:0:0:0:0:1:FF00:0000/104 concatenated with the 24 low-order bits of a corresponding IPv6 unicast or anycast address. For example, the solicited-node multicast address corresponding to the IPv6 address 2037::01:800:200E:8C6C is FF02::1:FF0E:8C6C. Solicited-node addresses are used in neighbor solicitation messages.

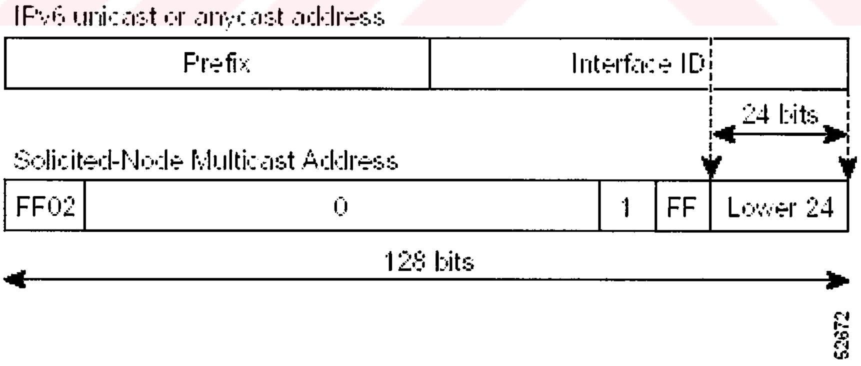

IPv6 unicast or anycast address
Prefix
Interface ID
24 bits
Solicited-Node Multicast Address
FF02
0
1
FF
Lower 24
128 bits
52672

Figure 4. 7 IPv6 Solicited-Node Multicast Address Format

4.3.2. Simplified Packet Header

The basic IPv4 packet header has 12 fields with a total size of 20 octets (160 bits). The 12 fields may be followed by an Options field, which is followed by a data portion that is usually the transport-layer packet. The variable length of the Options field adds to the total size of the IPv4 packet header. The shaded fields of the IPv4 packet header shown in Figure 4.8 are not included in the IPv6 packet header.

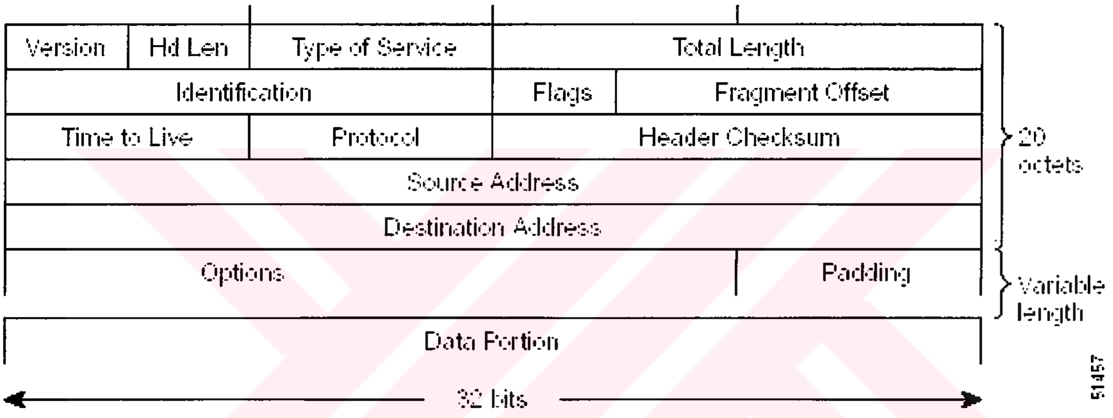


Figure 4. 8 IPv4 Packet Header Format

The basic IPv6 packet header has 8 fields with a total size of 40 octets (320 bits). Fields were removed from the IPv6 header because, in IPv6, fragmentation is not handled by routers and checksums at the network layer are not used. Instead, fragmentation in IPv6 is handled by the source of a packet and checksums at the data link layer and transport layer are used. (In IPv4, the User Datagram Protocol (UDP) transport layer uses an optional checksum. In IPv6, use of the UDP checksum is required to check the integrity of the inner packet.) Additionally, the basic IPv6 packet header and Options field are aligned to 64 bits, which can facilitate the processing of IPv6 packets.

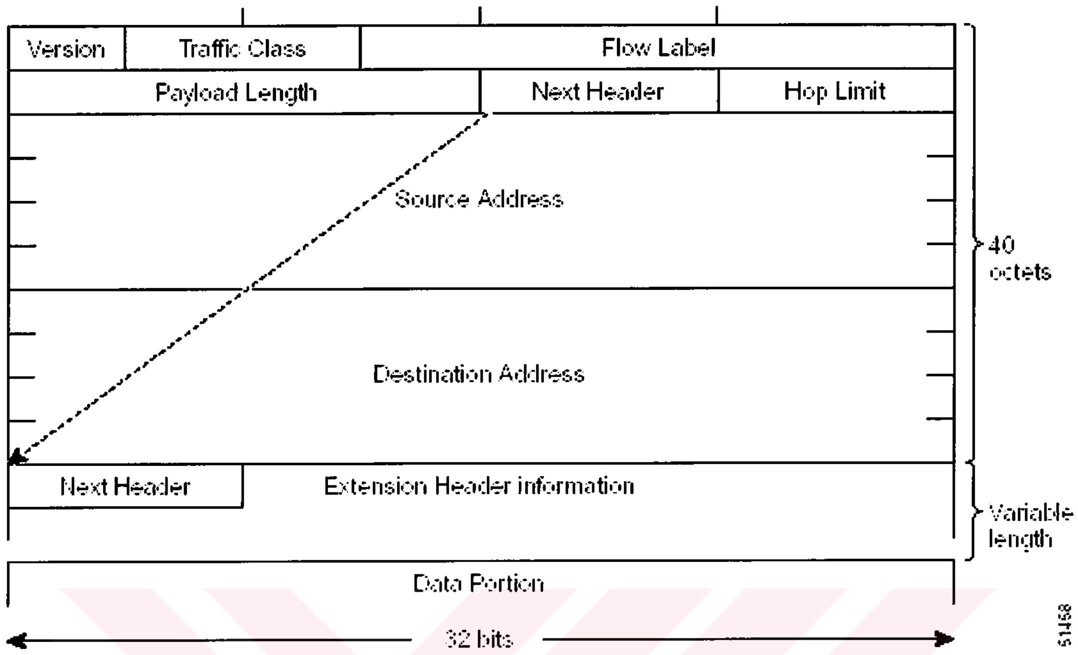


Figure 4. 9 IPv6 Packet Header Format

Table 4. 2 Basic IPv6 Packet Header Fields

Field	**Description**
Version	Similar to the Version field in the IPv4 packet header, except that the field lists number 6 for IPv6 instead of number 4 for IPv4.
Traffic Class	Similar to the Type of Service field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services.
Flow Label	A new field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer.
Payload Length	Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet.

Next Header

Similar to the Protocol field in the IPv4 packet header. The value of the Next Header field determines the type of information following the basic IPv6 header. The type of information following the basic IPv6 header can be a transport-layer packet, for example, a TCP or UDP packet, or an Extension Header, as shown in Figure 4.9.

Hop Limit

Similar to the Time to Live field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of routers that an IPv6 packet can pass through before the packet is considered invalid. Each router decrements the value by one. Because no checksum is in the IPv6 header, the router can decrement the value without needing to recalculate the checksum, which saves processing resources.

Source Address

Similar to the Source Address field in the IPv4 packet header, except that the field contains a 128-bit source address for IPv6 instead of a 32-bit source address for IPv4.

Destination Address

Similar to the Destination Address field in the IPv4 packet header, except that the field contains a 128-bit destination address for IPv6 instead of a 32-bit destination address for IPv4.

Following the eight fields of the basic IPv6 packet header are optional extension headers and the data portion of the packet. If present, each extension header is aligned to 64 bits. There is no fixed number of extension headers in an IPv6 packet. Together, the extension headers form a chain of headers. Each extension header is identified by the Next Header field of the previous header. Typically, the final extension header has a Next Header field of a transport-layer protocol, such as TCP or UDP. Figure 4.10 shows the IPv6 extension header format. [Gai, S. (1999)]

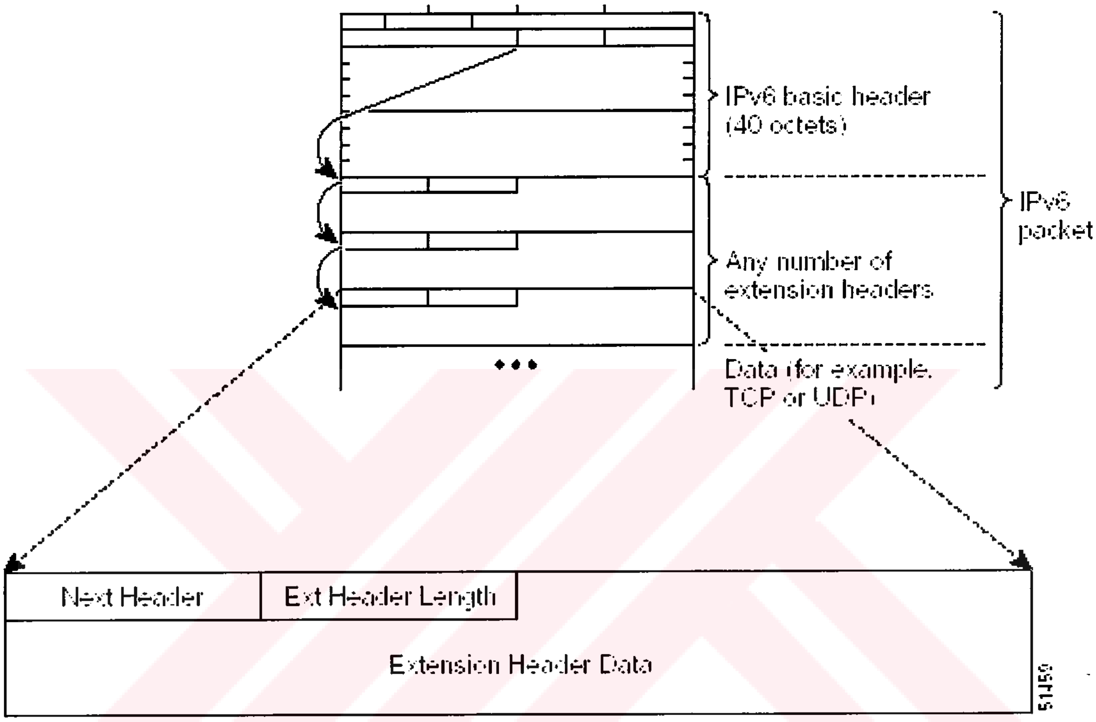


Figure 4. 10 IPv6 Extension Header Format

Table 4.3 lists the extension header types and their Next Header field values.

Table 4. 3 IPv6 Extension Header Types

Header Type	Next Header Value	Description
Hop-by-hop options header	0	This header is processed by all hops in the path of a packet. When present, the hop-by-hop options header always follows immediately after the basic IPv6 packet header.
Destination options header	60	The destination options header can follow any hop-by-hop options header, in which case the destination options header is processed at the final destination and also at each visited address specified by a routing header. Alternatively, the destination options header can follow any Encapsulating Security Payload (ESP) header, in which case the destination options header is processed only at the final destination.
Routing header	43	The routing header is used for source routing.
Fragment header	44	The fragment header is used when a source must fragment a packet that is larger than the Maximum Transmission Unit (MTU) for the path between itself and a destination. The Fragment header is used in each fragmented packet.
Authentication header and ESP header	51 50	The Authentication header and the ESP header are used within IP Security Protocol (IPSec) to provide authentication, integrity, and confidentiality of a packet. These headers are identical for both IPv4 and IPv6.
Upper-layer header	6 (TCP) 17 (UDP)	The upper-layer (transport) headers are the typical headers used inside a packet to transport the data. The two main transport protocols are TCP and UDP.

4.3.3. ICMP for IPv6

Internet Control Message Protocol (ICMP) in IPv6 functions the same as ICMP in IPv4, ICMP generates error messages, such as ICMP destination unreachable messages and informational messages like ICMP echo request and reply messages. Additionally, ICMP packets in IPv6 are used in the IPv6 neighbor discovery process, path MTU discovery, and the Multicast Listener Discovery (MLD) protocol for IPv6. MLD is used by IPv6 routers to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. MLD is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4.

A value of 58 in the Next Header field of the basic IPv6 packet header identifies an IPv6 ICMP packet. ICMP packets in IPv6 are like a transport-layer packet in the sense that the ICMP packet is after all the extension headers and is the last piece of information in the IPv6 packet.Within IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is derived (computed by the sender and checked by the receiver) from the fields in the IPv6 ICMP packet and the IPv6 pseudoheader. The ICMPv6 Data field contains error or diagnostic information relevant to IP packet processing. Figure 4.11 shows the IPv6 ICMP packet header format. [Sportack, M. (1999), Downes, K. (2000)]

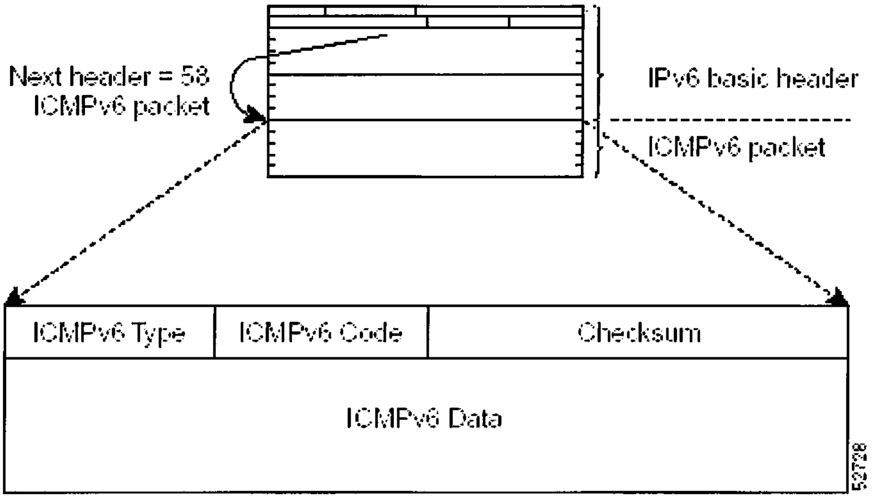


Figure 4. 11 IPv6 ICMP Packet Header Format

4.3.4. Neighbour Discovery

The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and keep track of neighbor routers.

The Static Cache Entry for IPv6 Neighbor Discovery feature enables the configuring of static entries in the IPv6 neighbor discovery cache, which provides functionality in IPv6 that is equivalent to static Address Resolution Protocol (ARP) entries in IPv4. Static entries in the IPv6 neighbor discovery cache are not modified by the neighbor discovery process.The Cisco IOS software uses static ARP entries in IPv4 to translate 32-bit IP addresses into 48-bit hardware addresses. In IPv6, the Cisco IOS software uses static entries in the IPv6 neighbor discovery cache to translate 128-bit IPv6 addresses into 48-bit hardware addresses. [Sportack, M. (1999)]

A value of 135 in the Type field of the ICMP packet header identifies a neighbor solicitation message. Neighbor solicitation messages are sent on the local link when a node wants to determine the link-layer address of another node on the same local link. (See Figure 4.12) When a node wants to determine the link-layer address of another node, the source address in a neighbor solicitation message is the IPv6 address of the node sending the neighbor solicitation message. The destination address in the neighbor solicitation message is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

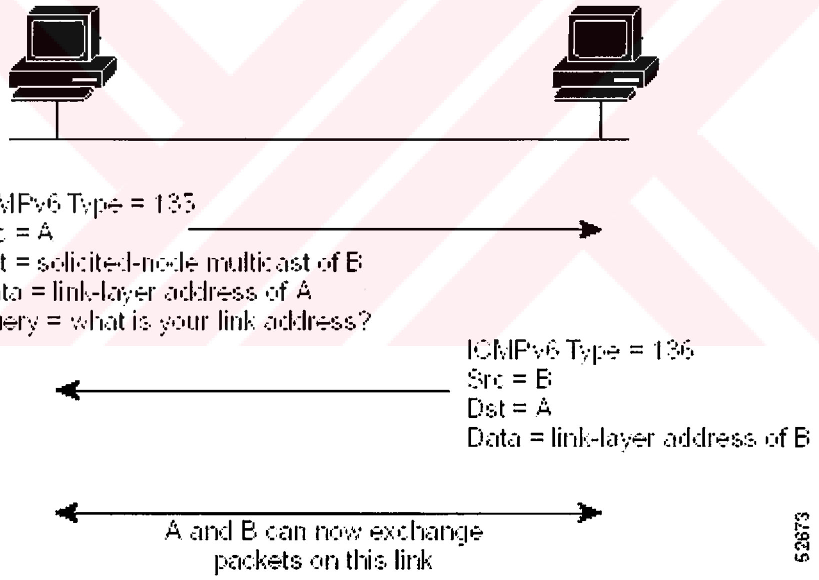

ICMPv6 Type = 135
Src = A
Dst = solicited-node multicast of B
Data = link-layer address of A
Query = what is your link address?
ICMPv6 Type = 136
Src = B
Dst = A
Data = link-layer address of B
A and B can now exchange packets on this link
52673

Figure 4. 12 IPv6 Neighbor Discovery—Neighbor Solicitation Message

After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address in the neighbor advertisement message is the IPv6 address of the node (more specifically, the IPv6 address of the node's interface) sending the neighbor advertisement message. The destination address in the neighbor advertisement message is the IPv6 address of the node

that sent the neighbor solicitation message. The data portion of the neighbor solicitation message includes the link-layer address of the node sending the neighbor advertisement message.

After source node receives the neighbor advertisement, the source node and destination node can communicate.

Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. When a node wants to verifying the reachability of a neighbor, the destination address in a neighbor solicitation message is the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is such a change, the destination address for the neighbor advertisement is the all-nodes multicast address.

Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. Neighbor unreachability detection identifies the failure of a neighbor or the failure of the forward path to the neighbor, and is used for all paths between hosts and neighboring nodes (hosts or routers). Neighbor unreachability detection is performed for neighbors to which only unicast packets are being sent and is not performed for neighbors to which multicast packets are being sent.

A neighbor is considered reachable when a positive acknowledgment is returned from the neighbor (indicating that packets previously sent to the neighbor have been received and processed). A positive acknowledgment—from an upper-layer protocol (such as TCP)—indicates that a connection is making forward progress (reaching its destination) or the receipt of a neighbor advertisement message in response to a neighbor solicitation message. If packets are reaching the peer, they are also reaching the next hop neighbor of the source. Therefore, forward progress is also a confirmation that the next hop neighbor is reachable.

For destinations that are not on the local link, forward progress implies that the first hop router is reachable. When acknowledgments from an upper-layer protocol are not available, a node probes the neighbor using unicast neighbor solicitation messages to verify that the forward path is still working. The return of a solicited neighbor advertisement message from the neighbor is a positive acknowledgment that the forward path is still working (neighbor advertisement messages that have the solicited flag set to a value of 1 are sent only in response to a neighbor solicitation message). Unsolicited messages confirm only the one-way path from the source to the destination node; solicited neighbor advertisement messages indicate that a path is working in both directions.

Neighbor solicitation messages are also used in the stateless autoconfiguration process to verify the uniqueness of unicast IPv6 addresses before the addresses are assigned to an interface. Duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed). Specifically, a node sends a neighbor solicitation message with an unspecified source address and a tentative link-local address in the body of the message. If another node is already using that address, the node returns a neighbor advertisement message that contains the tentative link-local address. If another node is simultaneously verifying the uniqueness of the same address, that node also returns a neighbor solicitation message. If no neighbor advertisement messages are received in response to the neighbor solicitation message and no neighbor solicitation messages are received from other nodes that are attempting to verify the same tentative address, the node that sent the original neighbor solicitation message considers the tentative link-local address to be unique and assigns the address to the interface. Every IPv6 unicast address (global or link-local) must be checked for uniqueness on the link; however, until the uniqueness of the link-local address is verified, duplicate address detection is not performed on any other IPv6 addresses associated with the link-local address. The Cisco implementation of duplicate address detection in the Cisco IOS software does not check the uniqueness of anycast or global addresses that are generated from 64-bit interface identifiers. Router advertisement messages, which have a value of 134 in the Type field of the ICMP packet header, are periodically sent out each configured interface of an IPv6 router. The router advertisement messages are sent to the all-nodes multicast address. (Figure 4.13)

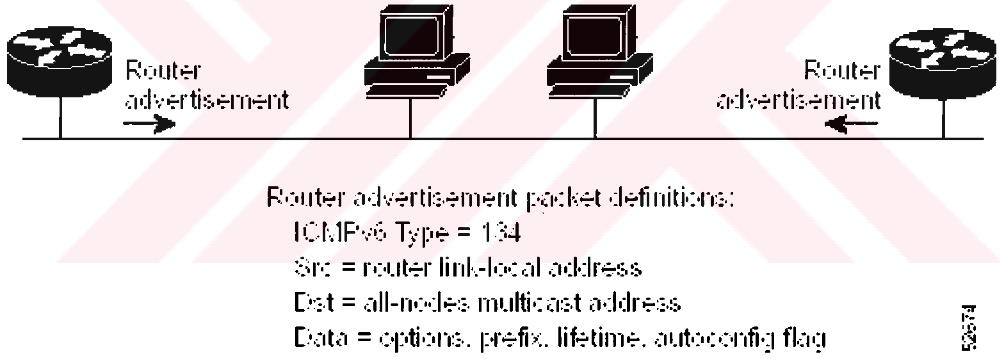

Router advertisement
Router advertisement
Router advertisement packet definitions:
ICMPv6 Type = 134
Src = router link-local address
Dst = all-nodes multicast address
Data = options, prefix, lifetime, autoconfig flag
52874

Figure 4. 13IPv6 IPv6 Neighbor Discovery-Router Advertisement Message

Router advertisement messages typically include the following information:

- One or more on-link IPv6 prefixes that nodes on the local link can use to automatically configure their IPv6 addresses
- Lifetime information for each prefix included in the advertisement
- Sets of flags that indicate the type of autoconfiguration (stateless or statefull) that can be completed

- Default router information (whether the router sending the advertisement should be used as a default router and, if so, the amount of time (in seconds) the router should be used as a default router)
- Additional information for hosts, such as the hop limit and MTU a host should use in packets that it originates

Router advertisements are also sent in response to router solicitation messages. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled router advertisement message. Given that router solicitation messages are usually sent by hosts at system startup (the host does not have a configured unicast address), the source address in router solicitation messages is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a configured unicast address, the unicast address of the interface sending the router solicitation message is used as the source address in the message. The destination address in router solicitation messages is the all-routers multicast address with a scope of the link. When a router advertisement is sent in response to a router solicitation, the destination address in the router advertisement message is the unicast address of the source of the router solicitation message. The following router advertisement message parameters can be configured:

- The time interval between periodic router advertisement messages
- The "router lifetime" value, which indicates the usefulness of a router as the default router (for use by all nodes on a given link)
- The network prefixes in use on a given link
- The time interval between neighbor solicitation message retransmissions (on a given link)
- The amount of time a node considers a neighbor reachable (for use by all nodes on a given link)

A value of 137 in the Type field of the ICMP packet header identifies an IPv6 neighbor redirect message. Routers send neighbor redirect messages to inform hosts of better first hop nodes on the path to a destination. (Figure 4.14)

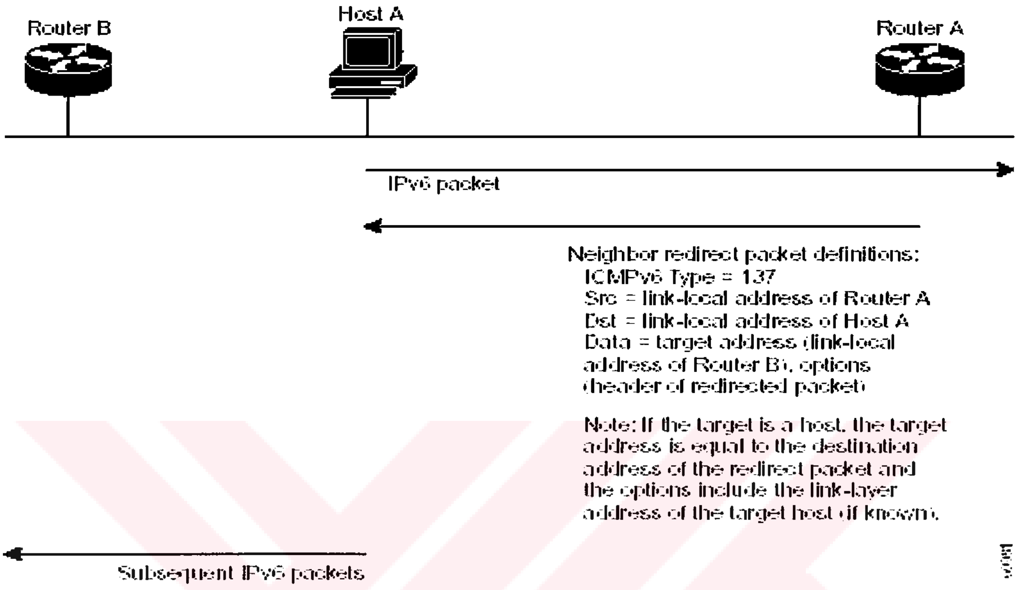

Router B
Host A
Router A
IPv6 packet
Neighbor redirect packet definitions:
ICMPv6 Type = 137
Src = link-local address of Router A
Dst = link-local address of Host A
Data = target address (link-local address of Router B), options (header of redirected packet)
Note: If the target is a host, the target address is equal to the destination address of the redirect packet and the options include the link-layer address of the target host (if known).
Subsequent IPv6 packets

Figure 4. 14 IPv6 Neighbor Discovery—Neighbor Redirect Message

4.3.5. DNS for IPv6

IPv6 introduces new Domain Name System (DNS) record types that are supported in the DNS name-to-address and address-to-name lookup processes. The new DNS record types support IPv6 addresses. Table 4.4 lists the new IPv6 DNS record types.

Table 4. 4 IPv6 DNS Record Types

Record Type	Description	Format
AAAA	Maps a host name to an IPv6 address. (Equivalent to an A record in IPv4.) **Note** Support for AAAA records and A records over an IPv6 transport or IPv4 transport is in the latest release of the Cisco IOS software. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.	www.abc.test AAAA 3FFE:B00:C18:1::2
PTR	Maps an IPv6 address to a host name. (Equivalent to a PTR record in IPv4.) **Note** The Cisco IOS software supports PTR records for the IP6.INT domain.	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8 .1.c.0.0.0.b.0.e.f.f.3.ip6.int PTR www.abc.test

4.3.6. Routing Protocols

IPv6 supports Interior Gateway Protocols (IGPs) and Exterior Gateway Protocols (EGPs). Routing Information Protocol (RIP) is the supported IGP for IPv6. Multiprotocol Border Gateway Protocol (BGP) is the supported EGP for IPv6.

When configuring supported routing protocols in IPv6, we must create the routing process, enable the routing process on interfaces, and customize the routing protocol for our particular network.

RIP in IPv6 functions the same and offers the same benefits as RIP in IPv4. IPv6 enhancements to RIP 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.

IS-IS in IPv6 functions the same and offers many of the same benefits as IS-IS in IPv4. IPv6 enhancements to IS-IS allow IS-IS to advertise IPv6 prefixes in addition to IPv4 and Open System Interconnection (OSI) routes. Extensions to the IS-IS CLI allow configuration of IPv6-specific parameters. IS-IS in IPv6 extends the address families supported by IS-IS to include IPv6, in addition to OSI and IPv4.

Multiprotocol BGP in IPv6 functions the same and offers the same benefits as multiprotocol BGP in IPv4. IPv6 enhancements to multiprotocol BGP include support for an IPv6 address family and network layer reachability information (NLRI) and next hop (the next router in the path to the destination) attributes that use IPv6 addresses and scoped addresses (the next hop attribute uses a global IPv6 address and potentially also a link-local address, when a peer is reachable on the local link). [Lammle,T. & Wallace,K. (2001)]

4.4. Benefits of IPv6

4.4.1. Stateless Autoconfiguration

All interfaces on IPv6 nodes must have a link-local address, which is usually automatically configured from the identifier for an interface and the link-local prefix FE80::0. A link-local address enables a node to communicate with other nodes on the link and can be used to further configure the node.

Nodes can connect to a network and automatically generate site-local and global IPv6 address without the need for manual configuration or help of a server, such as a Dynamic Host Configuration Protocol (DHCP) Server. With IPv6, a router on the link advertises in router advertisement messages any site-local and global prefixes, and its willingness to function as a default router for the link. Router advertisement messages are sent periodically and in response to router solicitation messages, which are sent by hosts at system startup. (Figure 4.15)

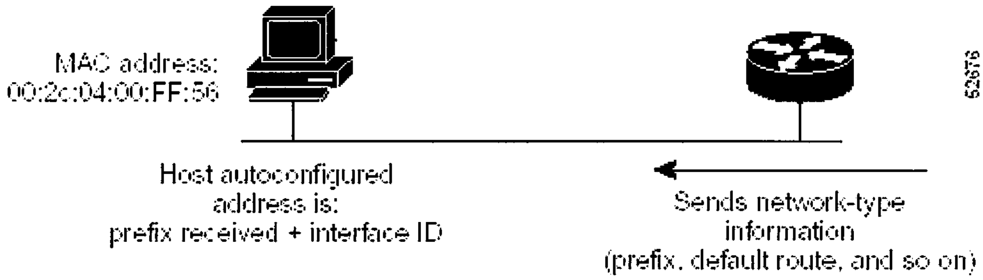

MAC address:
00:2c:04:00:FF:56
Host autoconfigured address is: prefix received + interface ID
Sends network-type information (prefix, default route, and so on)
52676

Figure 4. 15 IPv6 Stateless Autoconfiguration

A node on the link can automatically configure site-local and global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the router advertisement messages. The resulting 128-bit IPv6 addresses configured by the node are then subjected to duplicate address detection to ensure their uniqueness on the link. If the prefixes advertised in the router advertisement messages are globally unique, then the IPv6 addresses configured by the node are also guaranteed to be globally unique. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled router advertisement message. [Gai, S. (1999)]

4.4.2. Simplified Network Renumbering for IPv6 Hosts

The strict aggregation of the global routing table requires that networks be renumbered when the service provider for the network is changed. When the stateless autoconfiguration functionality in IPv6 is used to renumber a network, the prefix from a new service provider is added to router advertisement messages that are sent on the link. (The router advertisement messages contain both the prefix from the old service provider and the prefix from the new service provider.) Nodes on the link automatically configure additional addresses by using the prefix from the new service provider. The nodes can then use the addresses created from the new prefix and the existing addresses created from the old prefix on the link. By configuring the lifetime parameters associated with the old and new prefixes, nodes on the link can make the transition to using only addresses created from the new prefix. During a transition period, the old prefix is removed from router advertisement messages and only addresses that contain the new prefix are used on the link (the renumbering is complete). (Figure 4.16)

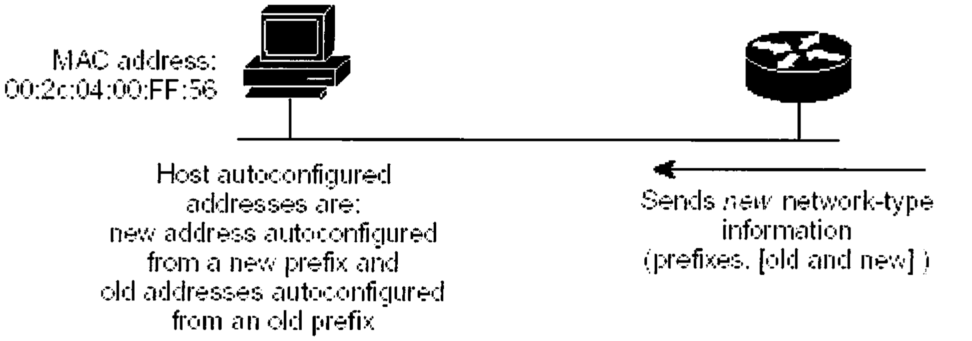


Figure 4. 16 IPv6 Network Renumbering for Hosts Using Stateless Autoconfiguration

4.4.3. Prefix Aggregation

The aggregatable nature of the IPv6 address space enables an IPv6 addressing hierarchy. For example, an enterprise can subdivide a single IPv6 prefix from a service provider into multiple, longer prefixes for use within its internal network. Conversely, a service provider can aggregate all of the prefixes of its customers into a single, shorter prefix that the service provider can then advertise over the IPv6 internet. (Figure 4.17) [Downes, K. (2000)]

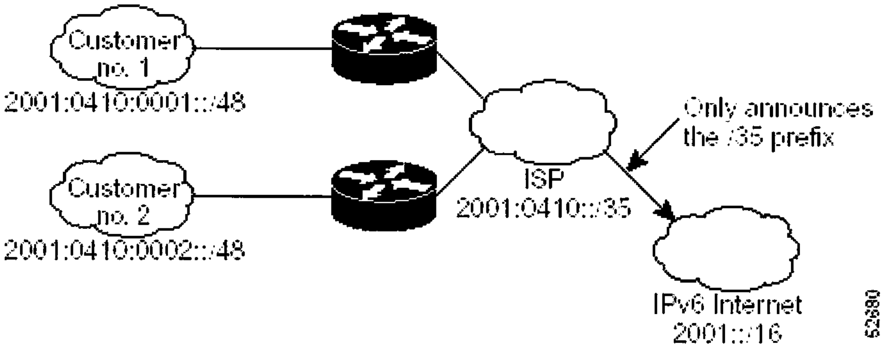


Figure 4. 17. IPv6 Prefix Aggregation

4.4.4. Site Multihoming

Multiple IPv6 prefixes can be assigned to networks and hosts. Having multiple prefixes assigned to a network makes it easy for that network to connect to multiple ISPs without breaking the global routing table. (Figure 4.18)

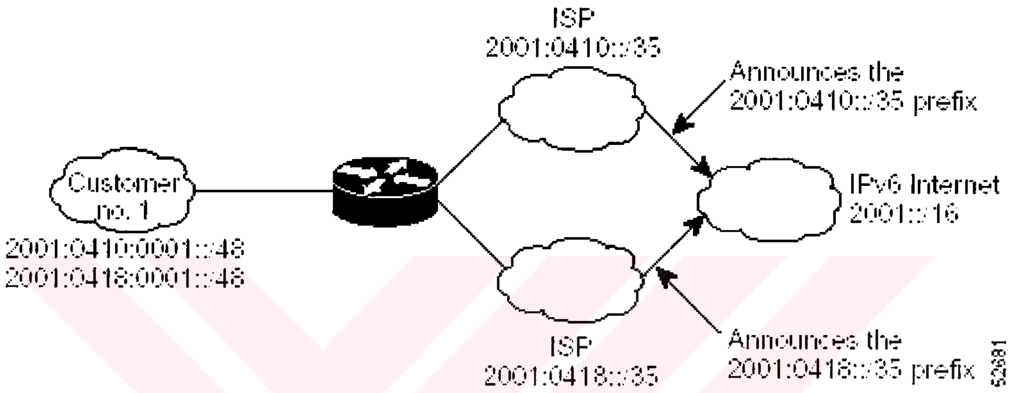


Figure 4. 18 IPv6 Site Multihoming

4.4.5. Mobile IP

Mobile IP provides users the freedom to roam beyond their home subnet while consistently maintaining their home IP address. Mobile IP enables transparent routing of IP datagrams to mobile users during their movement, so that data sessions can be initiated to them while they roam; it also enables sessions to be maintained in spite of physical movement between points of attachment to the Internet or other networks.

In IPv6, Mobile IP is implemented using the routing extension header. The routing extension header enables a mobile node to send IP packets directly to a destination node after the mobile node establishes an initial connection to the home agent. Direct routing in Mobile IP is the ability of a mobile node to bypass the home agent when sending IP packets to a destination node. Optional extensions make direct routing possible in

Mobile IP for IPv4 (the extensions might not be implemented in all deployments of Mobile IP for IPv4), whereas direct routing is built into Mobile IP for IPv6. [Gai, S. (1999), Wegner, J.D. (1999)]

4.4.6. Security

IPSec is a framework of open standards developed by the IETF that provide security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices (peers), such as Cisco routers. IPSec provides the following network security services. These services are optional. In general, local security policy will dictate the use of one or more of these services:

- Data confidentiality—The IPSec sender can encrypt packets before sending them across a network.
- Data integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data origin authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.
- Antireplay—The IPSec receiver can detect and reject replayed packets

With IPSec, data can be sent across a public network without fear of observation, modification, or spoofing. IPSec functionality is essentially identical in both IPv6 and IPv4; however, IPSec in IPv6 can be deployed from end-to-end—data may be encrypted along the entire path between a source node and destination node. (Typically, IPSec in IPv4 is deployed between border routers of separate networks.) In IPv6, IPSec is implemented using the authentication extension header and the ESP extension header. The authentication header provides integrity and authentication of the source. It also provides optional protection against replayed packets. The authentication header protects the integrity of most of the IP header fields and authenticates the source through a signature-based algorithm. The ESP header provides confidentiality, authentication of the source, connectionless integrity of the inner packet, antireplay, and limited traffic flow confidentiality

4.4.7. Transition Richness

Integration and coexistence with IPv4 is a prerequisite to enable the smooth transition of your network to IPv6. Following are some of the techniques available to facilitate the integration of IPv6 networks with IPv4 networks:

- Nodes that support both the IPv4 and IPv6 protocol stacks
- IPv6 tunnels over IPv4 core networks
- Translation gateways (for example, Network Address Translation-Protocol Translation [NAT-PT])
- IPv6 services integration on Multiprotocol Label Switching (MPLS) backbones
- Dedicated IPv6 networks (which support both the IPv6 and IPv4 protocol stacks) over common Layer 2 infrastructures such as Frame Relay, ATM, and optical fiber (for example, Wave-division multiplexing WDM)

Each technique addresses a different set of attributes that are specific to one context or another. For example, supporting both IPv4 and IPv6 protocol stacks enables nodes to send and receive data on both IPv4 and IPv6 networks. Tunneling encapsulates IPv6 packets in IPv4 packets and uses the IPv4 network as a link-layer mechanism. Other transition techniques address having IPv4-only nodes exchange data with IPv6-only nodes. Additionally, techniquess can be combined to achieve a smooth transition of our network to IPv6.

4.4.7.1. Dual IPv4 and IPv6 Protocol Stacks

The preferred technique for a transition to IPv6, the dual IPv4 and IPv6 protocol stack technique enables gradual, one-by-one upgrades to applications running on nodes. Applications running on nodes are upgraded to make use of the IPv6 protocol stack. Applications that are not upgraded—they support only the IPv4 protocol stack—can coexist with upgraded applications on the same node. New and upgraded applications simply make use of both the IPv4 and IPv6 protocol stacks. (Figure 4.19)

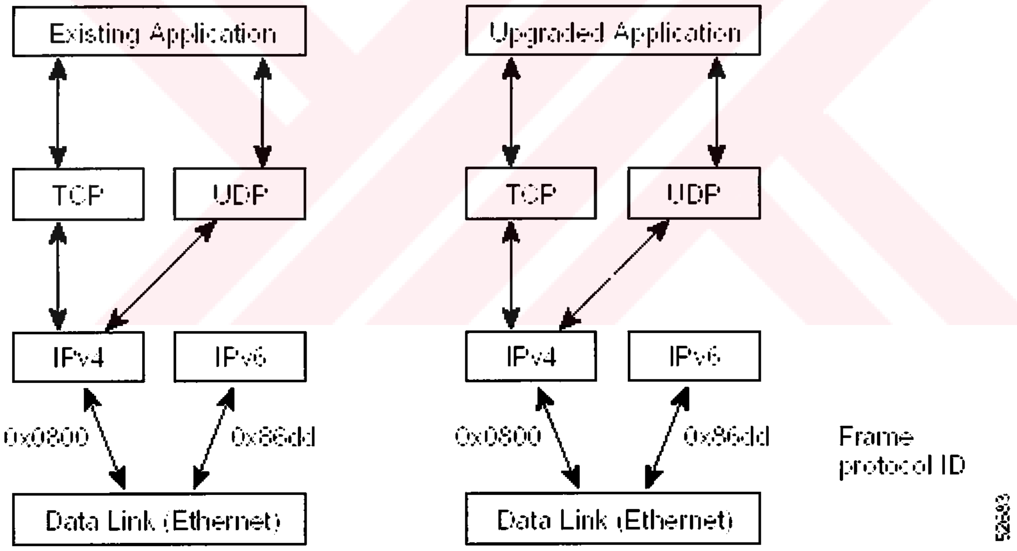


Figure 4. 19 Dual IPv4 and IPv6 Protocol Stack Technique

A new application programming interface (API) has been defined to support both IPv4 and IPv6 addresses and DNS requests. An application can be upgraded to the new API and still use only the IPv4 protocol stack. The Cisco IOS software supports the dual IPv4 and IPv6 protocol stack technique. When an interface is configured with both an IPv4 and an IPv6 address, the interface will forward both IPv4 and IPv6 traffic.

In Figure 4.20, an application that supports dual IPv4 and IPv6 protocol stacks requests all available addresses for the destination host name www.a.com from a DNS server. The DNS server replies with all available addresses (both IPv4 and IPv6 addresses) for www.a.com. The application chooses an address—in most cases, IPv6 addresses are the default choice—and connects the source node to the destination using the IPv6 protocol stack. [Wegner, J.D. (1999)]

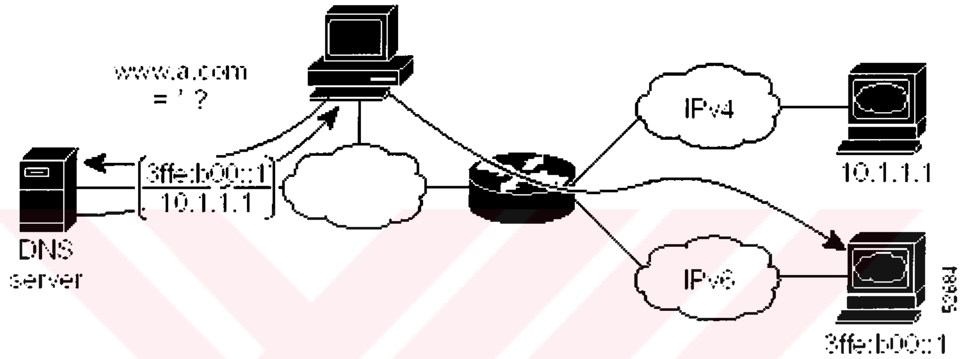


Figure 4. 20 Dual IPv4 and IPv6 Protocol Stack Applications

4.4.7.2. Overlay Tunnels

Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure (a core network or the Internet). (Figure 4.21) By using overlay tunnels, isolated IPv6 networks can communicate without needing to upgrade the IPv4 infrastructure between them. Overlay tunnels can be configured between border routers or between a border router and a host; however, both tunnel endpoints must support both the IPv4 and IPv6 protocol stacks.

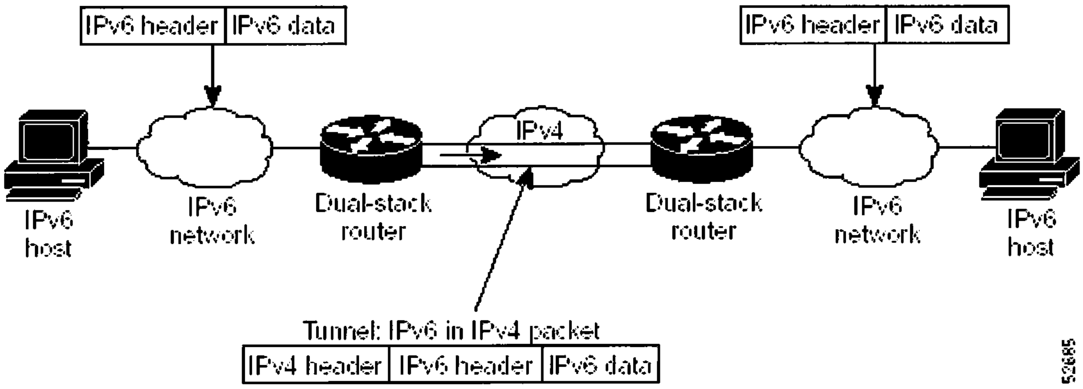


Figure 4. 21 Overlay Tunnels

Following are the different IPv6 overlay tunnel types:

- Manually configured tunnels—A manually configured IPv6 address is configured on a tunnel interface and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks. Manually configured tunnels can be configured between border routers or between a border router and a host.
- Automatic tunnels—The tunnel destination is automatically determined by the IPv4 address in the low-order 32 bits of IPv4-compatible IPv6 addresses. The host or router at each end of an automatic tunnel must support both the IPv4 and IPv6 protocol stacks. Automatic tunnels can be configured between border routers or between a border router and a host.
- 6to4 tunnels—A 6to4 tunnel is an automatic IPv6 tunnel where a border router in an isolated IPv6 network creates a tunnel to a border router in another isolated IPv6 network over an IPv4 infrastructure. The tunnel destination is determined by the IPv4 address of the border router that is concatenated to the prefix 2002::/16, in the format 2002:*IPv4 address of the border router*::/48. The border

router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks. 6to4 tunnels are configured between border routers. [Gai, S. (1999)]

CHAPTER FIVE

IPv6 TRANSITION PLANS

5.1. Planning to Deploy IPv6

As a network manager or operator for an enterprise, we may want to evaluate and assess IPv6 now because of our plans to introduce IPv6 applications within the network in the near future. Although it is not expected that a great number of IPv6-only applications will ship initially, some of the mobile IP offerings being introduced in the market perform and scale better using the direct-path features that will become available in an IPv6 infrastructure, rather than those available with IPv4.

We may also want to assess and evaluate IPv6 because of the end-to-end addressing, integrated autoconfiguration, QoS, and security required by the new environments for mobile phones, or we may want to expand our available address space for some new service such as an IP-based telephone system.

We may want to return to a global environment where the addressing rules of the network are more transparent to the applications, and reintroduce end-to-end security and QoS that are not readily available throughout IPv4 networks that use NAT and other techniques for address conversion, pooling, and temporary allocation.
Two key ways of evaluating and assessing IPv6 products and services are as follows:

- Set up an IPv6 domain and connect to an existing remote IPv6 network such as the 6bone

- Set up two or more IPv6 domains and interconnect these over your existing IPv4 infrastructures

As a network manager or operator for an enterprise, we should begin by choosing the IPv6 applications and services we would like to offer through IPv6, and decide where we want to provide these services. Activities then consist of creating an IPv6 domain and configuring a DNS that supports both IPv4 and IPv6 records, and, if there is a need for intercommunication between IPv6-only and IPv4-only hosts, operating one of the protocol translation mechanisms such as NAT-PT in the router or a TCP-UDP Relay. We should then identify the router or routers in the network that need to be dualstack. They will be part of the IPv6 domain, using IPv6 routing protocols to communicate with the IPv6 applications, and either IPv4 or IPv6 protocols to communicate outside of the domain. The protocol choice will be dependent on whether you are connecting directly to an IPv6 service provider, or using one of the available deployment strategies to carry the IPv6 traffic over the existing IPv4 infrastructure to a remote IPv6 network or domain. In both cases, apply for IPv6 addresses from the relevant service provider. [Downes, K. (2000)]

5.2. Selecting a Deployment Strategy

The key strategies used in deploying IPv6 at the edge of a network involve carrying IPv6 traffic over the IPv4 network, allowing isolated IPv6 domains to communicate with each other before the full transition to a native IPv6 backbone. It is also possible to run IPv4 and IPv6 throughout the network, from all edges through the core, or to translate between IPv4 and IPv6 to allow hosts communicating in one protocol to communicate transparently with hosts running the other protocol. All techniques allow networks to be upgraded and IPv6 deployed incrementally with little to no disruption of IPv4 services.
The four key strategies for deploying IPv6 are as follows:

- Deploying IPv6 over IPv4 tunnels: These tunnels encapsulate the IPv6 traffic within the IPv4 packets, and are primarily for communication between isolated

IPv6 sites or connection to remote IPv6 networks over an IPv4 backbone. The techniques include using manually configured tunnels, generic routing encapsulation (GRE) tunnels, semiautomatic tunnel mechanisms such as tunnel broker services, and fully automatic tunnel mechanisms such as IPv4-compatible and 6to4.

- Deploying IPv6 over dedicated data links: This technique enables isolated IPv6 domains to communicate by using the same Layer 2 infrastructure as for IPv4, but with IPv6 using separate Frame Relay or ATM PVCs, separate optical links, or dense Wave Division Multiplexing (dWDM).

- Deploying IPv6 using dual-stack backbones: This technique allows IPv4 and IPv6 applications to coexist in a dual IP layer routing backbone. All routers in the network need to be upgraded to be dual-stack with IPv4 communication using the IPv4 protocol stack and IPv6 communication using the IPv6 stack.

Table 5.1 summarizes the primary use, benefits, and limitations for each strategy.

Table 5. 1 Deployment Strategies: Primary Uses, Benefits, and Limitations

Deployment Strategy	**Key User/ Primary Use**	**Benefits**	**Limitations**	**Requirements**
IPv6 over IPv4 Tunnels	Service provider wanting to offer initial IPv6 service. Enterprise wanting to interconnect IPv6 domains or link to remote IPv6	Can demonstrate demand for IPv6 for minimal investment. Easy to implement over existing IPv4	Complex management and diagnostics due to the independence of the tunnel and link topologies.	Access to IPv4 through dual-stack router with IPv4 and IPv6 addresses. Access to IPv6 DNS.

	networks.	infrastructures. Low cost, low risk.		
IPv6 over Dedicated Data Links	Service provider WANs or metropolitan area networks (MANs) deploying ATM, Frame Relay, or dWDM.	Can provide end-to-end IPv6 with no impact on the IPv4 traffic and revenue.	Lack of IPv6-specific hardware acceleration and support for IPv6 network management in currently deployed hardware.	Access to the WAN through dual-stack router with IPv4 and IPv6 addresses. Access to IPv6 DNS.
IPv6 Using Dual-Stack Backbones	Small enterprise networks.	Easy to implement for small campus networks with a mixture of IPv4 and IPv6 applications.	Complex dual management of routing protocols. Major upgrade for large networks.	All routers are dual-stack with IPv4 and IPv6 addresses. Access to IPv6 DNS. Enough memory for both IPv4 and IPv6 routing tables.

In addition to the strategies for deploying IPv6 within our IPv4 environment, we also need protocol translation mechanisms (for example, a NAT-PT device to connect IPv6-only web browsers to IPv4-only web servers) or dual-stack servers (for example, an e-mail server that handles IPv4-only and IPv6-only mail clients) to allow communication between applications using IPv4 and applications using IPv6. These mechanisms become increasingly important as IPv6 deployment moves from the testing to the actual usage phase, and more relevant as application developers decide that continuing to support IPv4 is not cost-effective.

Eventually, as IPv6 becomes the protocol of choice, these mechanisms will allow legacy IPv4 systems to be part of the overall IPv6 network. The mechanisms translate between the IPv4 and IPv6 protocols on the end system, or on a dedicated server, or on a router within the IPv6 network, and, together with dual-stack hosts, provide a full set of tools for the incremental deployment of IPv6 with no disruption to the IPv4 traffic. [Gai, S. (1999)]

The following sections provide further information on IPv6 deployment strategies and protocol translation mechanisms:

- Deploying IPv6 over IPv4 Tunnels
- Deploying IPv6 over Dedicated Data Links
- Deploying IPv6 Using Dual-Stack Backbones
- Protocol Translation Mechanisms

5.2.1. Deploying IPv6 over IPv4 Tunnels

Tunneling is the encapsulation of IPv6 traffic within IPv4 packets so that they can be sent over an IPv4 backbone, allowing isolated IPv6 end systems and routers to communicate without the need to upgrade the IPv4 infrastructure that exists between them. Tunneling is one of the key deployment strategies for both service providers and enterprises during the period of IPv4 and IPv6 coexistence. Figure 5.1 shows the use of IPv6 over IPv4 tunnels.

Tunneling allows service providers to offer an end-to-end IPv6 service without major upgrades to the infrastructure and without impacting current IPv4 services. Tunneling allows enterprises to interconnect isolated IPv6 domains over their existing IPv4 infrastructures, or to connect to remote IPv6 networks such as the 6bone.

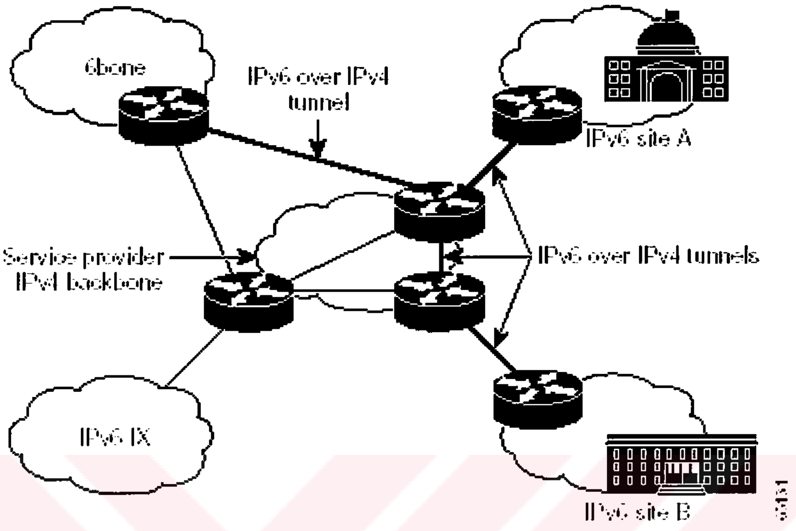


Figure 5. 1 Deploying IPv6 over IPv4 Tunnels

Varieties of tunnel mechanisms are available. These mechanisms include manually created tunnels such as IPv6 manually configured tunnels and IPv6 over IPv4 GRE tunnels, semiautomatic tunnel mechanisms such as that employed by tunnel broker services, and fully automatic tunnel mechanisms such as IPv4-compatible and 6to4. Manual and GRE tunnels are used between two points and require configuration of both the source and destination ends of the tunnel, whereas automatic tunnel mechanisms need only to be enabled and are more transient, they are set up and taken down as required, and last only as long as the communication. [Gai, S. (1999)]

Table 5.2 summarizes the primary use, benefits, and limitations for each tunnelling mechanism.

Table 5. 2 Overlay Tunnel Mechanisms: Primary Uses, Benefits, and Limitations

Tunnel Mechanism	**Primary Use**	**Benefits**	**Limitations**	**Requirements**
IPv6 Manually Configured Tunnel	Stable and secure links for regular communication. Connection to 6bone.	Supported in IPv6 for Cisco IOS software now. DNS with support for IPv6 not required.	Tunnel between two points only. Large management overhead. No independently managed NAT.	ISP-registered IPv6 address. Dual-stack router.
IPv6 over IPv4 GRE Tunnel	Stable and secure links for regular communication.	Well known standard tunnel technique. Supported in IPv6 for Cisco IOS software now.	Tunnel between two points only. Management overhead. No independently managed NAT. Cannot use to connect to 6bone.	ISP-registered IPv6 address. Dual-stack router. Required by i/IS-IS for IPv6.
Tunnel Broker	Standalone isolated IPv6 end systems.	Tunnel set up and managed by ISP.	Potential security implications.	Tunnel broker service must know how to create and send a script for Cisco IOS software.
Automatic IPv4-Compatible Tunnel	Single hosts or small sites. Infrequent communication.	Supported in IPv6 for Cisco IOS software now.	Communication only with other IPv4-compatible sites. Does not scale well. No independently	IPv6 prefix (0::/96). Dual-stack router.

			managed NAT.	
Automatic 6to4 Tunnel	Connection of multiple remote IPv6 domains. Frequent communication.	Easy to set up with no management overhead. Supported in IPv6 for Cisco IOS software now.	No independently managed NAT.	IPv6 prefix (2002::/16). Dual-stack router.
ISATAP Tunnels	Campus sites. Transition of nonrouted sites.	To be supported in the next phase of Cisco IOS software.	Not yet commercially available.	Dual-stack router.
6over4 Tunnels	Campus sites. Transition of nonrouted sites.	—	Not supported by Cisco IOS software.	—

All tunneling mechanisms require that the endpoints of the tunnel run both IPv4 and IPv6, that is, must run in dual-stack mode. The dual-stack routers run both IPv4 and IPv6 protocols simultaneously and thus can interoperate directly with both IPv4 and IPv6 end systems and routers. The design is very similar in concept to running IP and either IPX, DECnet, or AppleTalk on the same router.

Dual-stack end systems allow applications to migrate one at a time from an IPv4 to an IPv6 transport. Applications that are not upgraded (they support only the IPv4 stack) can coexist with upgraded applications on the same end system. Applications choose between using IPv4 or IPv6 based on name lookup; both the IPv4 and IPv6 addresses may be returned from the DNS, with the application (or the system according to the rules defined in the IETF document Default Address Selection for IPv6) selecting the correct address based on the type of IP traffic and particular requirements of the communication.

It may be possible to protect the IPv6 over IPv4 tunnels using IPv4 IPSec by applying a crypto map to both the tunnel interface to encrypt outgoing traffic, and to the physical interface to decrypt the traffic flowing through. Note that it may not be possible to use in all environments due to the limitations of IPSec in IPv4. However, if possible, protecting tunnels in this way may have a substantial impact on performance, and you should balance this loss of performance against the security that can be achieved by careful configuration of your network. [Downes, K. (2000)]

The following sections describe each of the supported tunneling mechanisms in more detail.

- IPv6 Manually Configured Tunnel
- IPv6 over IPv4 GRE Tunnel
- Tunnel Broker
- Automatic IPv4-Compatible Tunnel
- Automatic 6to4 Tunnel

5.2.1.1. IPv6 Manually Configured Tunnel

A manually configured tunnel is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone. The primary use is for stable connections that require regular secure communication between two edge routers or between an end system and an edge router, or for connection to remote IPv6 networks such as the 6bone. The edge routers and end systems, if they are at the end of the tunnel, must be dual-stack implementations.

At each end of the tunnel, you configure the IPv4 and IPv6 addresses of the dual-stack router on the tunnel interface, and identify the entry and exit (or source and destination) points using IPv4 addresses. For enterprises, your ISP provides you with the appropriate IPv6 address prefix for your site. Your ISP also provides you with the required destination IPv4 address for the exit point of the tunnel.

Figure 5.2 shows the configuration of a manually configured tunnel.

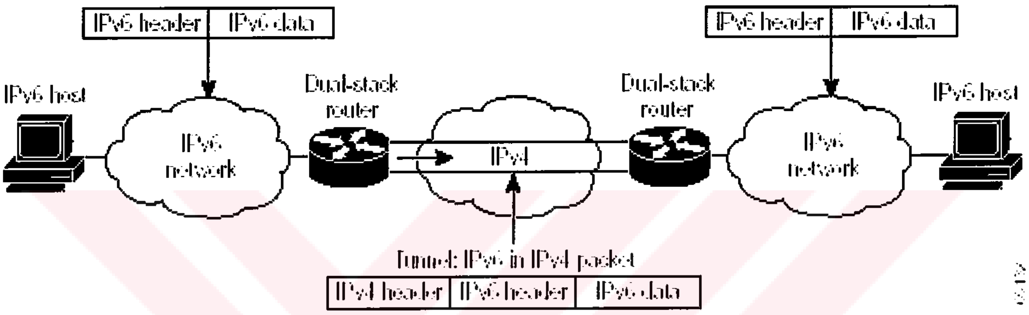


Figure 5. 2 Manually Configured Tunnel

Because each tunnel exists between only two routers, adding routers means adding tunnels to cater for all the paths between the routers. Because each tunnel is independently managed, the more routers you have, the more tunnels you need, and the greater is the management overhead. As with other tunnel mechanisms, NAT, when applied to the outer IPv4 header, is allowed along the path of the tunnel only if the translation map is stable and preestablished.

5.2.1.2. IPv6 over IPv4 GRE Tunnel

The IPv6 over IPv4 GRE tunnel uses the standard GRE tunneling technique that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme. As in IPv6 manually configured tunnels, GRE tunnels are links between two points, with a separate tunnel for each link. The tunnels are not tied to a specific passenger or transport protocol, but in this case carry IPv6 as the passenger protocol over GRE as the carrier protocol.

The primary use is for stable connections that require regular secure communication between two edge routers or between an edge router and an end system. The edge routers and, in the case described, the end systems must be dual-stack implementations.

Figure 5.3 shows the configuration for an IPv6 over IPv4 GRE tunnel.

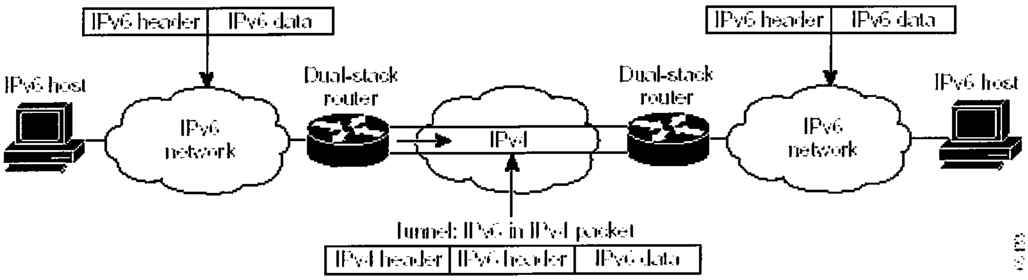


Figure 5. 3 IPv6 over IPv4 GRE Tunnel

As with IPv6 manually configured tunnels, you configure the IPv4 and IPv6 addresses of the dual-stack router on the GRE tunnel interface, and identify the entry and exit (or source and destination) points of the tunnel using IPv4 addresses.

Also, as with manually configured tunnels, each GRE tunnel exists between only two routers, and thus adding routers means adding tunnels to cater for all the paths between the routers. Because each tunnel is independently managed, the more routers you have, the more tunnels you need, and the greater is the management overhead. As with other tunnel mechanisms, NAT, when applied to the outer IPv4 header, is allowed along the path of the tunnel only if the translation map is stable and preestablished.

5.2.1.3. Tunnel Broker

A tunnel broker service allows IPv6 applications on remote dual-stack end systems, or on IPv6 end systems connected to dual-stack routers, access to an IPv6 backbone. The tunnel broker service, using 6-over-4 tunnels to connect the end systems to the IPv6 backbone, automatically manages tunnel requests and configuration for the enterprise, rather than forcing the network administrator to manually configure tunnels.

For instance, an enterprise could register the IPv4 address of the remote end system or router (using IPv4) with the service provider on a dedicated website. The service provider delivers a script that builds a tunnel to the IPv6 network, allocates an IPv6 address to the end system, and allocates a network prefix to the router to allow connectivity for the rest of the site. The tunnel broker manages the creation and deletion of the tunnel to the tunnel server, itself a dual-stack router that is connected to the IPv6 network.

Figure 5.4 shows the steps in the creation of a tunnel.

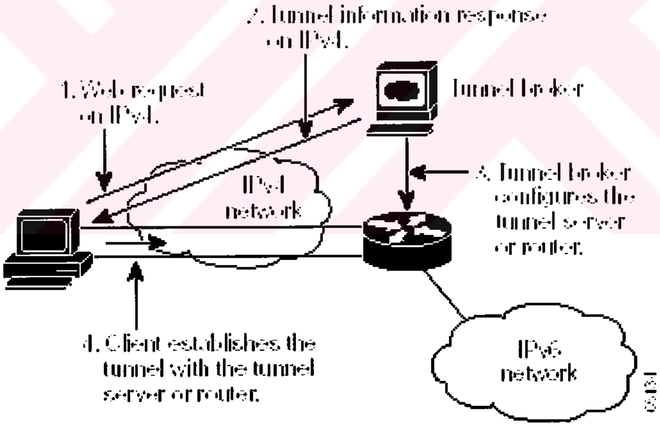

2. Tunnel information response on IPv4.
1. Web request on IPv4.
Tunnel broker
IPv4 network
3. Tunnel broker configures the tunnel server or router.
4. Client establishes the tunnel with the tunnel server or router.
IPv6 network
59191

Figure 5. 4 Tunnel Broker

5.2.1.4. Automatic IPv4-Compatible Tunnel

An automatic IPv4-compatible tunnel can be configured between edge routers or between an edge router and an end system. The edge routers and end systems must be dual-stack implementations.

An IPv4-compatible tunnel is one where the endpoints of the tunnel (the tunnel source and the tunnel destination) are automatically determined by the IPv4 address in the low-order 32 bits of the IPv4-compatible IPv6 address. This IPv4-compatible IPv6 address is a special IPv6 address with 0:0:0:0:0:0 in the high-order 96 bits and the IPv4 address in the low-order 32 bits.

Figure 5.5 shows the configuration of an IPv4-compatible tunnel.

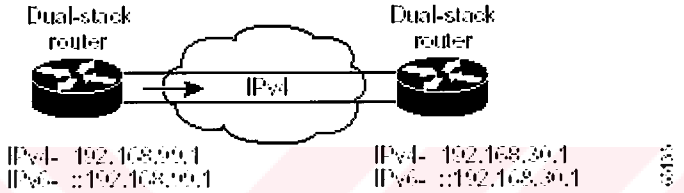


Figure 5. 5 IPv4-Compatible Tunnel

The IPv4-compatible tunnel is a transition mechanism that was defined early in the IPv6 development process, and its use in the future is under discussion in the IETF. Although it is an easy way to create tunnels for IPv6 over IPv4, it is a mechanism that does not scale well for large networks because each host requires an IPv4 address and an IPv6 address to be able to determine the endpoints of the tunnel. A further limitation is that all communication is always only between IPv4-compatible addresses. As with other tunnel mechanisms, NAT, when applied to the outer IPv4 header, is allowed along the path of the tunnel only if the translation map is stable and preestablished.

5.2.1.5. Automatic 6to4 Tunnel

An automatic 6to4 tunnel allows isolated IPv6 domains to be connected over an IPv4 network and allows connections to remote IPv6 networks such as the 6bone. The key difference between this and manually configured tunnels is that the routers are not configured in pairs (and thus do not require manual configuration) because they treat the IPv4 infrastructure as a virtual nonbroadcast link, using an IPv4 address embedded in the IPv6 address to find the other end of the tunnel.

Each IPv6 domain requires a dual-stack router that identifies the IPv4 tunnel by a unique routing prefix in the IPv6 address (the IPv4 address of the tunnel destination is concatenated to the prefix 2002::/16). This unique routing prefix has been assigned permanently by the Internet Assigned Number Authority (IANA) for use in 6to4 schemes. Each site, even if it has just one public IPv4 address, has a unique routing prefix in IPv6. As with the manually configured and IPv4-compatible tunnel mechanisms, management of NAT needs to be linked with the management of the tunnel, and any independently managed NAT is not allowed along the path of the tunnel.

The simplest deployment scenario for 6to4 tunnels is to interconnect multiple IPv6 sites, each of which has at least one connection to a shared IPv4 network. This IPv4 network could be the global Internet or could be your corporate backbone. The key requirement is that each site has a 6to4 IPv6 address. As with other tunnel mechanisms, appropriate entries in a DNS that map between host names and IP addresses for both IPv4 and IPv6 allow the applications to choose the required address.

Figure 5.6 shows the configuration of a 6to4 tunnel for interconnecting 6to4 domains.

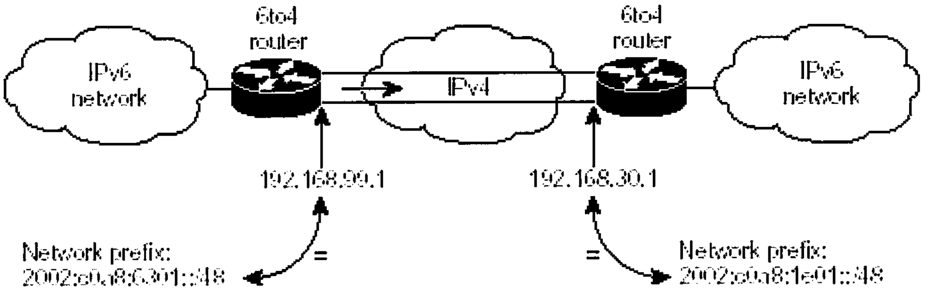


Figure 5. 6 Interconnecting 6to4 Domains

5.2.2. Deploying IPv6 over Dedicated Data Links

Many WANs and MANs have been implemented by deploying Layer 2 technologies such as Frame Relay, ATM, or optical, and are beginning to use dWDM. Figure 5.7 shows a sample configuration for IPv6 over dedicated data links.

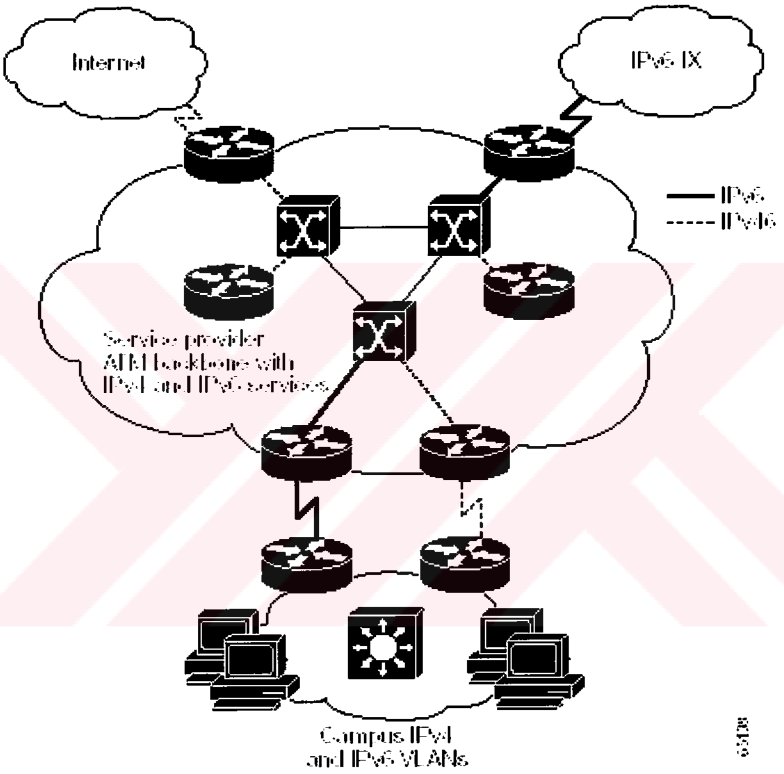


Figure 5. 7 IPv6 over Dedicated Data Links

Routers attached to the ISP WANs or MANs can be configured to use the same Layer 2 infrastructure as for IPv4, but to run IPv6, for example, over separate ATM or Frame Relay PVCs or separate optical lambda. This configuration has the added benefit for the service provider of no loss in service or revenue for the IPv4 traffic.

5.2.3. Deploying IPv6 Using Dual-Stack Backbones

Using dual-stack backbones is a basic strategy for routing both IPv4 and IPv6. All routers in the network need to be upgraded to be dual-stack. IPv4 communication uses the IPv4 protocol stack (with forwarding of IPv4 packets based on routes learned through running IPv4-specific routing protocols), and IPv6 communication uses the IPv6 stack with routes learned through the IPv6-specific routing protocols.

The key requirements are that each site has an IPv6 unicast global prefix and appropriate entries in a DNS that map between host names and IP addresses for both IPv4 and IPv6. Applications choose between using IPv4 or IPv6 based on the response from the DNS resolver library, with the application selecting the correct address based on the type of IP traffic and particular requirements of the communication.

Today, dual-stack routing is a valid deployment strategy for specific network infrastructures with a mixture of IPv4 and IPv6 applications (such as on a campus or an aggregation point of presence), requiring both protocols to be configured. However, apart from the obvious need to upgrade all routers in the network, limitations to this approach are that the routers require a dual addressing scheme to be defined, require dual management of the IPv4 and IPv6 routing protocols, and must be configured with enough memory for both the IPv4 and IPv6 routing tables.

Also, Cisco does not recommend an overall upgrade to a dual-stack network until there is a better parity between features and traffic levels. Although IPv6 for Cisco IOS software fully supports dual-stack, the current implementation of IPv6 requires enhancements to various services (for example, IPv6 multicast and IPv6 QoS) before any network can be upgraded to dual-stack. [Downes, K. (2000)]

5.2.4. Protocol Translation Mechanisms

All of the integration strategies provide IPv6 end to end. Intercommunication between IPv4 and IPv6 requires some level of translation between the IPv4 and IPv6 protocols on the host or router, or dual-stack hosts, with an application-level understanding of which protocol to use.

A variety of protocol translation mechanisms are under consideration by the IETF NGTrans Working Group, as follows:

- Network Address Translation-Protocol Translation (NAT-PT)
- TCP-UDP Relay
- Bump-in-the-Stack (BIS)
- Dual Stack Transition Mechanism (DSTM)
- SOCKS-Based Gateway

These protocol translation mechanisms become more relevant as IPv6 becomes more prevalent, and even as IPv6 becomes the protocol of choice to allow legacy IPv4 systems to be part of the overall IPv6 network.

The mechanisms tend to fall into two categories — those that require no changes to either the IPv4 or IPv6 hosts, and those that do. An example of the former is the TCP-UDP Relay mechanism that runs on a dedicated server and sets up separate connections at the transport level with IPv4 and IPv6 hosts, and then simply transfers information between the two. An example of the latter is the BIS mechanism that requires extra protocol layers to be added to the IPv4 protocol stack.

Table 5.3 provides a summary of the various translation mechanisms, with their primary use, benefits, and limitations.

Table 5. 3 Protocol Translation Mechanisms: Primary Uses, Benefits, and Limitations

Translation Mechanism	Primary Use	Benefits	Limitations	Requirements
NAT-PT	IPv6-only hosts to IPv4-only hosts.	No dual stack. To be supported in IPv6 for Cisco IOS software Phase II.	No end-to-end IPSec. Dedicated server is single point of failure.	Dedicated server. DNS with support for IPv6.
TCP-UDP Relay	Translation between IPv6 and IPv4 on dedicated server.	Freeware. No changes to Cisco IOS software.	No end-to-end IPSec. Dedicated server is single point of failure.	Dedicated server. DNS with support for IPv6.
BIS	IPv4-only hosts communicating with IPv6-only hosts.	End-system implementation.	All stacks must be updated.	Updated IPv4 protocol stack.
DSTM	Dual-stack hosts (but with IPv6 address only).	Temporary IPv4 address allocated from pool.	No current support in Cisco IOS software.	Dedicated server to provide a temporary global IPv4 address.
SOCKS-Based IPv6/IPv4 Gateway	IPv6-only hosts to IPv4-only hosts.	Freeware. No changes to Cisco IOS software.	Additional software in the router.	Client and gateway software in the host and router.

The translation mechanisms that allow communication between IPv6-only and IPv4-only hosts, such as NAT-PT or BIS, use an algorithm called Stateless IP/ICMP Translator (SIIT). This mechanism translates, on a packet-by-packet basis, the headers in the IP packet between IPv4 and IPv6, and translates the addresses in the headers between IPv4 and either IPv4-translated or IPv4-mapped IPv6 addresses. The mechanism assumes that each IPv6 host has a temporary IPv4 address assigned to it.

5.2.4.1. NAT-PT

The NAT-PT translation mechanism translates at the network layer between IPv4 and IPv6. An Application Level Gateway (ALG) translates between the IPv4 and IPv6 DNS requests and responses.

Its greatest use is where new hosts run only native IPv6 or the network has not implemented the dual-stack approach of IPv6 for Cisco IOS software. It has the same benefits as NAT for IPv4, and might be easier to introduce for IPv6 initially due to this familiarity and experience. However, NAT-PT also inherits the same limitations as NAT for IPv4, and makes fast rerouting difficult (ALGs are not as fast as IP routers). Also, the dedicated server is a single point of failure in the network. Although allowing security at an application level, NAT-PT inhibits end-to-end network security, and makes the merging of private-addressed networks extremely difficult.

5.2.4.2. TCP-UDP Relay

The TCP-UDP Relay mechanism is similar to NAT-PT in that it requires a dedicated server and DNS, but it translates at the transport layer rather than the network layer, with the DNS again providing the mapping between IPv4 and IPv6 addresses.

When the TCP relay server receives a request, it establishes separate connections at the transport level with both the source and destination IPv4 and IPv6 hosts, and then simply transfers data from one connection to the other. User Datagram Protocol (UDP) relays work in a similar manner.

The greatest use of this mechanism is for native IPv6 networks that want to access IPv4-only hosts, such as IPv4 web servers, but without the expense of upgrading either the IPv6 or IPv4 sides. The relay mechanism supports bidirectional traffic (multicast is not supported), but, as with NAT-PT, it allows application-level security but inhibits end-to-end network security, and makes the merging of private-addressed networks extremely difficult. Fast rerouting is difficult, and the dedicated server becomes a single point of failure in the network.

5.2.4.3. BIS

The BIS mechanism is for communication between IPv4 applications on an IPv4-only host and IPv6-only hosts.

Three extra layers — name resolver extension, address mapper, and translator — are added to the IPv4 protocol stack between the application and network layers. Whenever an application needs to communicate with an IPv6-only host, the extra layers map an IPv6 address into the IPv4 address of the IPv4 host. The translation mechanism is defined as part of SIIT. This mechanism is for implementation on end systems only.

5.2.4.4. DSTM

The DSTM translation mechanism is for dual-stack hosts in an IPv6 domain that have not yet had an IPv4 address assigned to the IPv4 side, but need to communicate with IPv4 systems or allow IPv4 applications to run on top of their IPv6 protocol stack. The mechanism requires a dedicated server that dynamically provides a temporary global IPv4 address for the duration of the communication (using DHCPv6), and uses dynamic tunnels to carry the IPv4 traffic within an IPv6 packet through the IPv6 domain.

DSTM becomes much more relevant as IPv6 becomes more prevalent and IPv4 addresses become scarce such that they need to be shared between hosts, and where the requirement is to carry IPv4 traffic over IPv6 or communicate between IPv6 hosts in an IPv6 domain and a few remote legacy IPv4 systems.

5.2.4.5. SOCKS-Based IPv6/IPv4 Gateway

The SOCKS-based IPv6/IPv4 gateway mechanism is for communication between IPv4-only and IPv6-only hosts. It consists of additional functionality in both the end system (client) and the dual-stack router (gateway) to permit a communications environment that relays two terminated IPv4 and IPv6 connections at the application layer.

This mechanism is based on the SOCKSv5 protocol, and inherits all the features of that protocol. Existing SOCKSv5 commands are unchanged, and the protocol maintains the end-to-end security between the client and the gateway, and the gateway and the destination.

The mechanism uses a feature called DNS Name Resolving Delegation to determine IPv6 addresses, delegating the name resolving to the gateway, thus requiring no change to existing DNSs. Implementations of the SOCKS-based IPv6/IPv4 gateway are freely available from various locations.

CHAPTER SIX

CONCLUSION

6.1. Conclusion

The intent of this thesis was to investigate new generation Internet Protocol IPv6 and transition plans for migrating from IPv4 to IPv6. Differences between transition plans, current routing and switching technologies for internetworks and changes of these technologies in IPv6 were also investigated.

According to population estimates from the US Census Bureau, the world will be home to about 9 billion people in 2050. Whatever the economic constraints may be, we must clearly plan technically for all of these people to have potential Internet access. It would not be acceptable to produce a technology that simply could not scale to be accessible by the whole human population, under appropriate economic conditions. Furthermore, pervasive use of networked devices will probably mean many devices per person, not just one. Simple arithmetic tells us that the maximum of 4 billion public addresses allowed by the current IP version 4, even if backed up by the inconvenient techniques of private addresses and address translation, will simply be inadequate in the future. If the Internet is truly for everyone, we need more addresses, and IP version 6 is the only way to get them. IPv6 has other benefits, such as provision for "plug and play" automatic configuration, which promises reduced complexity of network deployment and administration. Still, the principal benefit of IPv6 is that of having enough addresses thereby assisting in restoration of the end-to-end model on which the Internet was based.

The continuous growth of the global Internet requires that its overall architecture evolve to accommodate the new technologies that support the growing numbers of users, applications, appliances, and services. IPv6 is designed to meet these requirements and allow a return to a global environment where the addressing rules of the network are again transparent to the applications. The current IP address space is unable to satisfy the potential huge increase in the number of users or the geographical needs of the Internet expansion, let alone the requirements of emerging applications such as Internet-enabled personal digital assistants (PDAs), home area networks (HANs), Internet-connected automobiles, integrated telephony services, and distributed gaming.

We want to return to a global environment where the addressing rules of the network are more transparent to the applications, and reintroduce end-to-end security and QoS that are not readily available throughout IPv4 networks that use NAT and other techniques for address conversion, pooling, and temporary allocation. We may also want to assess and evaluate IPv6 because of the end-to-end addressing, integrated autoconfiguration, QoS, and security required by the new environments for mobile phones, or we may want to expand our available address space for some new service such as an IP-based telephone system.

The purpose of this thesis was to investigate the new protocol IPv6, differences between IPv4 and IPv6 and transition plans for migrating IPv4 to IPv6.Although the success of IPv6 will depend ultimately on the availability of applications that run over IPv6, a key part of the IPv6 design is its ability to integrate into and coexist with existing IPv4 networks. It is expected that IPv4 and IPv6 hosts will need to coexist for a substantial time during the steady migration from IPv4 to IPv6, and the development of transition strategies, tools, and mechanisms has been part of the basic IPv6 design from the start.

REFERENCES

Downes, K. (2000). Internetworking Technologies Handbook (2nd Edition). Cisco Press.

Feibel,W. (1996). The Encyclopedia of Networking(2nd Edition). USA: Network Press.

Gai, S. (1999). Internetworking IPv6 with Cisco Routers. McGraw-Hill Computer Communications Series.

Lammle,T. & Wallace,K. (2001). CCNP Routing Study Guide. USA: Sybex Press.

Lammle,T & Hales,K. (2000). CCNP Switching Study Guide. USA: Sybex Press.

Naugle, M.G. (1998). Illustrated TCP/IP. Wiley Computer Publishing.

Sportack, M. (1999). IP Routing Fundamentals. Cisco Press.

Swartz,J. & Lammle,T.(2001). Cisco Certified Internetwork Expert Study Guide. USA: Sybex Press.

Quinn,T. & Haller,A.K. (1998). Designing Campus Networks. Cisco Press.

Wegner, J.D. (1999). IP Addressing and Subnetting, Including IPv6. USA: Syngress Media Press.