NEAR EAST UNIVERSITY

Faculty of Engineering

Department Of Computer Engineering

Transmission Control Protocol and Internet Protocol

Graduation Project
COM - 400

Student : Selim Aydinoglu

Supervisor: Prof. Dr Fakhreddin Mamedov

Nicosia - 2004

ACKNOWLEDGEMENTS

"First I would like to thank my supervisor Prof. Dr Fakhreddin Mamedov for his great advice and recommendations to finish this work properly.

Although I faced many problem collections data but has guiding me the appropriate references.(Prof. Dr Fakhreddin Mamedov) thanks a lot for your invaluable and continual support.

Second, I thank all the staff of the faculty of engineering for giving me the facilities to practice and solving any problem I was facing during working in this project

Finally thanks for all of my friends for their advices and support.

Abstract

The TCP/IP protocol suite has become the facto standard for computer communications in today's networked world. The ubiquitous implementation of a specific networking standard has led to an incredible dependence on the applications enabled by it. Today, we use the TCP/IP protocols and the Internet not only for entertainment and information, but to conduct our business by performing transactions, buying and selling products, and delivering services to customers. We are continually extending the set of applications that leverage TCP/IP, thereby driving the need for further infrastructural support.

In TCP/IP Tutorial and Technical Overview, we take an in-depth look into the TCP/IP protocol suite. We introduce TCP/IP, providing a basic understanding of the underlying concepts essential to the protocols

TABLE OF CONTENTS

INTRODUCTION

In first chapter, we will see the TCP/IP layered architectures, a history of TCP/IP and the Internet, the structure of the Internet, Internet and IP addresses. Also, We will see How the troubles can be solved in TCP/IP.Using these concepts, we will then move on to look at the TCP/IP family of protocols in more detail.

The next chapter begins with the Internet Protocol (IP), showing how it is used and the format of its header information. The rest of the chapter covers gateway information necessary to piece together the rest of the protocols. We will start an in-depth look at the TCP/IP protocol family with the Internet Protocol. We will cover what IP is and how it does its task of passing datagrams between machines. The construction of the IP datagram and the format of the IP header will be shown in detail. The construction of the IP header is important to many TCP/IP family protocol members. We will also look at the Internet Control Message Protocol (ICMP), an important aspect of the TCP/IP system.

We will also look at the related User Datagram Protocol (UDP). TCP and UDP form the basis for all TCP/IP protocols. Here we will look at TCP in reasonable detail. Combined with the information in the last two chapters, we will now have the theory and background necessary to better understand TCP/IP utilities, such as Telnet and FTP, as well as other protocols that use or closely resemble TCP/IP, such as SMTP and TFTP

Chapter 1 TCP/IP Protocol

1.1 TCP/IP Introduction

In the mid-1970s, the Defense Advanced Research Projects Agency (DARPA) became interested in establishing a packet-switched network to provide communications between research institutions in the United States. DARPA and other government organizations understood the potential of packet-switched technology and were just beginning to face the problem that virtually all companies with networks now have—communication between dissimilar computer systems.

With the goal of heterogeneous connectivity in mind, DARPA funded research by Stanford University and Bolt, Beranek, and Newman (BBN) to create a series of communication protocols. The result of this development effort, completed in the late 1970s, was the Internet Protocol suite, of which the Transmission Control Protocol (TCP) and the Internet Protocol (IP) are the two best-known protocols.

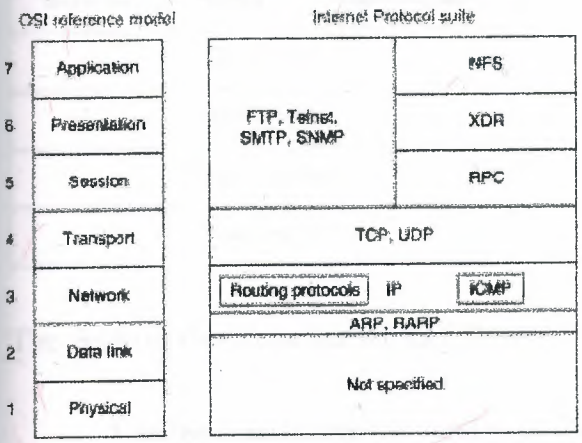
The most widespread implementation of TCP/IP is IPv4 (or IP version 4). In 1995, a new standard, RFC 1883—which addressed some of the problems with IPv4, including address space limitations—was proposed. This new version is called IPv6. Although a lot of work has gone into developing IPv6, no wide-scale deployment has occurred; because of this, IPv6 has been excluded from this text.

1.1.1 Internet Protocols

Internet protocols can be used to communicate across any set of interconnected networks. They are equally well suited for local-area network (LAN) and wide-area network (WAN) communications. The Internet suite includes not only lower-layer specifications (such as TCP and IP), but also specifications for such common applications as e-mail, terminal emulation, and file transfer. Figure 7-1 shows some of the most important Internet protocols and their relationships to the OSI reference model.

As an interesting side note, the seven-layer model actually came about *after* TCP/IP. DARPA used a four-layer model instead, which the OSI later expanded to seven layers. This is why TCP/IP doesn't generally fit all that well into the seven-layer OSI model.

Figure 7-1: The Internet Protocol Suite and the OSI Reference Model

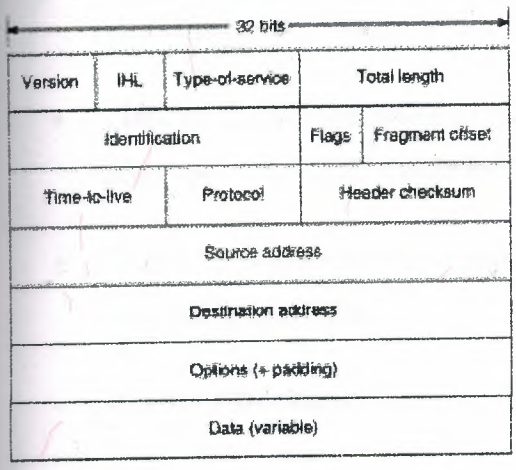


Creation and documentation of the Internet Protocol suite closely resemble an academic research project. The protocols are specified and refined in documents called Requests For Comments (RFCs), which are published, reviewed, and analyzed by the Internet community. Taken together, the RFCs provide a colorful history of the people, companies, and trends that have shaped the development of what is today the world's most popular open-system protocol suite.

1.1.2 The Network Layer

IP is the primary Layer 3 protocol in the TCP/IP suite. IP provides the logical addressing that enables communication across diverse networks. IP also provides fragmentation and reassembly of datagrams and error reporting. Along with TCP, IP represents the heart of the Internet Protocol suite. The IP packet format is shown in Figure 7-2.

Figure 7-2: The IP Packet Format



The fields of the IP packet are as follows:

- **Version**—Indicates the version of this IP datagram.
- **IP Header Length (IHL)**—Indicates the datagram header length in 32-bit words.
- **Type-of-Service**—Specifies how a particular upper-layer protocol would like the current datagram to be handled. Datagrams can be assigned various levels of importance using this field.

  Today this field is used primarily to provide quality of service (QoS) capabilities to TCP/IP for applications requiring predictable bandwidth or delay. RFC 2474 describes a method by which the TOS field is replaced by a DS field that is used to provide differentiated services (DiffServ) on networks. This field is split into two parts. The first 6 bits are used for the DSCP codepoint, which is used to differentiate traffic. The last 2 bits, or CU, are ignored by DiffServ-compliant nodes.

- **Total Length**—Specifies the length of the entire IP packet, including data and header, in bytes.
- **Identification**—Consists of an integer identifying this datagram. This field is used to help piece together datagram fragments.

- **Flags**—Consists of 3 bits, of which the low-order 2 bits control fragmentation. One bit specifies whether the packet can be fragmented; the second bit specifies whether the packet is the last fragment in a series of fragmented packets.
- **Time-to-Live**—Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.
- **Protocol**—Indicates which upper-layer protocol receives incoming packets after IP processing is complete.
- **Header Checksum**—Helps ensure IP header integrity.
- **Source Address**—Specifies the sending node.
- **Destination Address**—Specifies the receiving node.
- **Options**—Allows IP to support various options, such as security.
- **Data**—Contains upper-layer information.

1.1.3 Addressing

As with all network layer protocols, the addressing scheme is integral to the process of routing IP datagrams through an internetwork. An IP address is 32 bits in length, divided into either two or three parts. The first part designates the network address, the second part (if present) designates the subnet address, and the final part designates the host address. Subnet addresses are present only if the network administrator has decided that the network should be divided into subnetworks. The lengths of the network, subnet, and host fields are all variable.

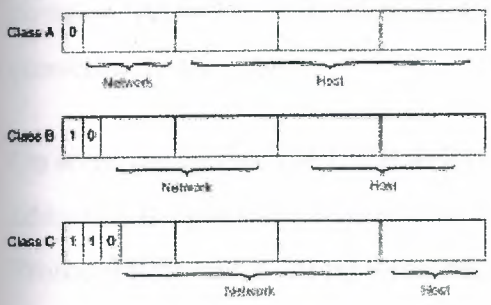
Today's Internet does not segment addresses along classful bounds—it is almost entirely classless. The separation between networks and subnets has been effectively eliminated. The requirement to understand network classes and the difference between a network and a subnet remains solely because of configuration and behavioral issues with network devices.

IP addressing supports five different network classes, and the high-order—far-left—bits indicate the network class:

- Class A networks provide 8 bits for the Network Address field. The high-order bit (at far left) is 0.
- Class B networks allocate 16 bits for the Network Address field and 16 bits for the Host Address field. This address class offers a good compromise between network and host address space. The first 2 high-order bits are 10.
- Class C networks allocate 24 bits for the Network Address field. Class C networks provide only 8 bits for the Host field, however, so the number of hosts per network may be a limiting factor. The first 3 high-order bits are 110.
- Class D addresses are reserved for multicast groups, as described formally in RFC 1112. The first 4 high-order bits are 1110.
- Class E addresses are also defined by IP but are reserved for future use. The first 4 high-order bits are 1111.

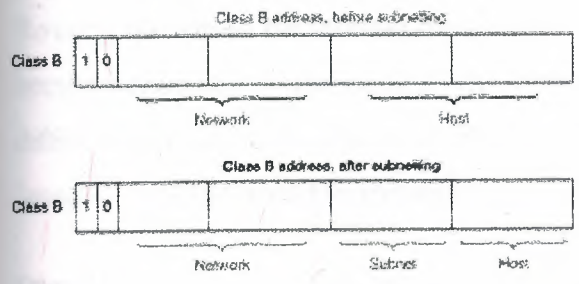
IP addresses are written in dotted decimal format (for example, 34.10.2.1). Figure 7-3 shows the address formats for Class A, B, and C IP networks.

Figure 7-3: Class A, B, and C Address Formats



IP networks can also be divided into smaller units called subnets. Subnets provide extra flexibility for network administrators. For example, assume that a network has been assigned a Class B address, and all the nodes on the network currently conform to a Class B address format. Then assume that the dotted decimal representation of this network's address is 172.16.0.0 (all zeros in the Host field of an address specifies the entire network). Rather than change all the addresses to some other basic network number, the administrator can subdivide the network using subnetting. This is done by borrowing bits from the host portion of the address and using them as a subnet field, as shown in Figure 7-4.

Figure 7-4: Subnet Addresses



If a network administrator has chosen to use 8 bits of subnetting, the third octet of a Class B IP address provides the subnet number. For example, address 172.16.1.0 refers to network 172.16, subnet 1; address 172.16.2.0 refers to network 172.16, subnet 2; and so on. In today's world, the difference between subnet bits and the natural mask has become blurred, and you will often see only a prefix length that specifies the length of the entire mask (natural mask plus subnet bits). It is still important to understand the difference between the natural network mask, which is determined by the network class, and the subnet mask, because routers sometimes make assumptions based on the natural mask of an address. For example, the natural mask of 10.1.1.1/24 is 8 bits because this is a class A network, even though the subnet mask is 24 bits.

Subnet masks can be expressed in two forms: prefix length (as in /24), or dotted-decimal notation (As in 255.255.255.0). Both forms mean exactly the same thing and can easily be converted to the other.

On some media (such as IEEE 802 LANs), the correlation between media addresses and IP addresses is dynamically discovered through the use of two other members of the Internet Protocol suite: the Address Resolution Protocol (ARP) and the Reverse Address Resolution Protocol (RARP). ARP uses broadcast messages to determine the hardware Media Access Control (MAC)-layer address corresponding to a particular IP address. ARP is sufficiently generic to allow use of IP with virtually any type of underlying media-access mechanism. RARP uses broadcast messages to determine the Internet address associated with a particular hardware address. RARP is particularly important to diskless nodes, which may not know their IP address when they boot.

1.1.4 Internet Routing

Routing devices in the Internet have traditionally been called gateways—an unfortunate term because elsewhere in the industry, the term *gateway* applies to a device with somewhat different functionality. Gateways (which we will call *routers* from this point on) within the Internet are organized hierarchically.

Dynamic routing protocols, such as RIP and OSPF, provide a means by which routers can communicate and share information about routes that they have learned or are connected to. This contrasts with static routing, in which routes are established by the network administrator and do not change unless they are manually altered. An IP routing table consists of destination address/next-hop pairs. A sample entry, shown in Figure 7-5, is interpreted as meaning, "To get to network 34.1.0.0 (subnet 1 on network 34), the next stop is the node at address 54.34.23.12."

Figure 7-5: An Example of an IP Routing Table

Destination address	Next hop
34.1.0.0	54.34.23.12
78.2.0.0	54.34.23.12
147.9.5.0	
17.12.0.0	54.32.12.10
	54.32.12.10

IP routing specifies that IP datagrams travel through internetworks one hop at a time; the entire route is not known at the outset of the journey. Instead, at each stop, the next destination is calculated by matching the destination address within the datagram with an entry in the current node's routing table. Each node's involvement in the routing process consists only of forwarding packets based on internal information, regardless of whether the packets get to their final destination. In other words, IP does not provide for error reporting back to the source when routing anomalies occur. This task is left to other Internet protocols, such as the Internet Control Message Protocol (ICMP) and TCP protocol.

1.1.5 ICMP

ICMP performs a number of tasks within an IP internetwork, the principal of which is reporting routing failures back to the source of a datagram. In addition, ICMP provides helpful messages such as the following:

- Echo and reply messages to test node reachability across an internetwork
- Redirect messages to stimulate more efficient routing
- Time exceeded messages to inform sources that a datagram has exceeded its allocated time to exist within the internetwork
- Router advertisement and router solicitation messages to determine the addresses of routers on directly attached subnetworks

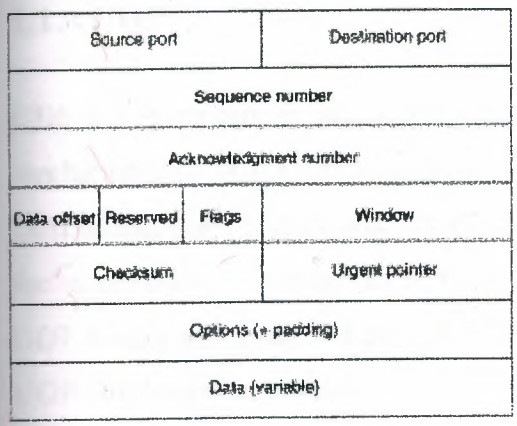
1.1.6 The Transport Layer

The Internet transport layer is implemented by Transport Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP provides connection-oriented data transport, whereas UDP operation is connectionless.

1.1.7 TCP

TCP provides full-duplex, acknowledged, and flow-controlled service to upper-layer protocols. It moves data in a continuous, unstructured byte stream in which bytes are identified by sequence numbers. TCP can support numerous simultaneous upper-layer conversations. The TCP packet format is shown in Figure 7-6.

Figure 7-6: The TCP Packet Format



The fields of the TCP packet are described here:

- **Source port and destination port**—Identify the points (sockets) at which upper-layer source and destination processes receive TCP services.
- **Sequence number**—Usually specifies the number assigned to the first byte of data in the current message. Under certain circumstances, it can also be used to identify an initial sequence number to be used in the upcoming transmission.
- **Acknowledgment number**—Contains the sequence number of the next byte of data that the sender of the packet expects to receive.
- **Data offset**—Indicates the number of 32-bit words in the TCP header.
- **Reserved**—Is reserved for future use.
- **Flags**—Carries a variety of control information.
- **Window**—Specifies the size of the sender's receive window (buffer space available for incoming data).
- **Checksum**—Provides information used to determine whether the header was damaged in transit.
- **Urgent pointer**—Points to the first urgent data byte in the packet.
- **Options**—Specifies various TCP options.

- **Data**—Contains upper-layer information.

1.1.8 UDP

UDP is a much simpler protocol than TCP and is useful in situations in which the reliability mechanisms of TCP are not necessary. The UDP header has only four fields: Source Port, Destination Port, Length, and UDP Checksum. The Source and Destination Port fields serve the same functions as they do in the TCP header. The Length field specifies the length of the UDP header and data, and the UDP Checksum field allows packet integrity checking. The UDP checksum is optional.

1.1.9 Upper-Layer Protocols

The Internet Protocol suite includes many upper-layer protocols representing a wide variety of applications, including network management, file transfer, distributed file services, terminal emulation, and electronic mail. Table 7-1 maps the best-known Internet upper-layer protocols to the applications that they support.

Table 7-1: Internet Protocol/Application Mapping (with Common Port Numbers) Application	**Protocols**
WWW browser	HTTP (TCP port 80)
The Hypertext Transfer Protocol (HTTP) is used by Web browsers and servers to transfer the files that make up web pages.	
File transfer	FTP (TCP ports 20 and 21)
The File Transfer Protocol (FTP) provides a way to move files between computer systems. Telnet allows virtual terminal emulation.	
Terminal emulation	Telnet (TCP port 23)
The Telnet protocol provides terminal emulation services over a reliable TCP stream. The	

Telnet protocol also specifies how a client and server should negotiate the use of certain features and options.	
Electronic mail	SMTP (TCP port 25), POP3 (TCP port 110), IMAP4 (TCP port 143)
The Simple Mail Transfer Protocol (SMTP) is used to transfer electronic mail between mail servers, and is used by mail clients to send mail. Mail clients do not generally use SMTP to receive mail. Instead, they use either the Post Office Protocol version 3 (POP3) or the Internet Message Access Protocol (IMAP); this will be discussed in greater detail later in this chapter.	
Network management	SNMP (UDP port 161)
The Simple Network Management Protocol (SNMP) is a network management protocol used for reporting anomalous network conditions and setting network threshold values.	
Distributed file services	NFS, XDR, RPC (UDP port 111), X Windows (UDP ports 6000-6063)
X Windows is a popular protocol that permits intelligent terminals to communicate with remote computers as if they were directly attached. Network file system (NFS), external data representation (XDR), and remote-procedure call (RPC) combine to allow transparent access to remote network resources.	

These and other network applications use the services of TCP/IP and other lower-layer Internet protocols to provide users with basic network services.

1.1.10 Domain Name System

TCP/IP uses a numeric addressing scheme in which each node is assigned an IP address that is used to route packets to a node on the network. Because it is much easier for people to remember names such as www.somedomain.com instead of 10.1.1.1, a protocol called Domain Name System (DNS) is used to map numbers to names, and vice versa. Most web pages refer to other web pages or links using these names instead of their IP addresses. This provides many advantages; for example, the address can change without breaking any links to a web page if the DNS table is also changed to point to the new address.

1.2 Troubleshooting TCP/IP

The sections in this chapter describe common features of TCP/IP and provide solutions to some of the most common TCP/IP problems. The following items will be covered:

- TCP/IP Introduction
- Tools for Troubleshooting IP Problems
- General IP Troubleshooting Theory and Suggestions
- Troubleshooting Basic IP Connectivity
- Troubleshooting Physical Connectivity Problems
- Troubleshooting Layer 3 Problems
- Troubleshooting Hot Standby Router Protocol (HSRP)

1.3 Tools for Troubleshooting IP Problems

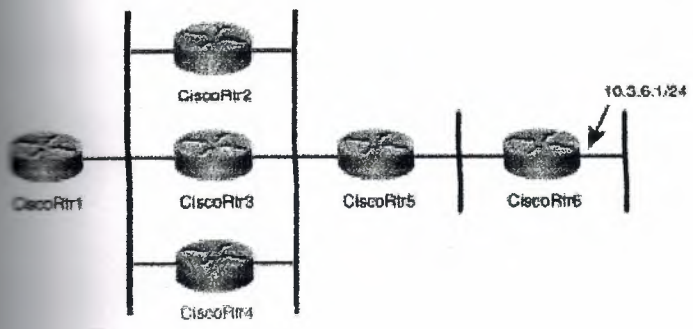
The tools ping and traceroute, both in the TCP/IP protocol suite, will greatly assist in troubleshooting IP connectivity. Most operating systems and IP implementations come with these tools installed by default. On some UNIX platforms, however, you may need to download and install a traceroute package.

Cisco routers provide a basic method of viewing IP traffic switched through the router called *packet debugging*. Packet debugging enables a user to determine whether traffic is travelling along an expected path in the network or whether there are errors in a particular TCP stream. Although in some cases packet debugging can eliminate the need for a packet analyzer, it should not be considered a replacement for this important tool.

Packet debugging can be very intrusive—in some cases, it can cause a router to become inoperable until physically reset. In other instances, packets that are present on the network and switched through the router may not be reported by packet debugging. Thus, a firm conclusion cannot be drawn that a packet was not sent solely from the output of packet debugging; a network analyzer must be used to accurately make this assessment. *Packet debugging should be used with extreme caution by only advanced operators because it can cause the router to lock up and stop routing traffic, if not used carefully.* The risks of using packet debugging may be compounded by the necessity of disabling fast switching for packet debugging to be effective. As a general rule, packet debugging should not be used on a production router unless you have physical access to the router and are willing to risk it going down.

1.3.1 ping

The ping tool uses the IP ICMP echo request and echo reply messages to test reachability to a remote system. In its simplest form, ping simply confirms that an IP packet is capable of getting to and getting back from a destination IP address (Figure 7-7). This tool generally returns two pieces of information: whether the source can reach the destination (and, by inference, vice versa), and the round-trip time (RTT, typically in milliseconds). The RTT returned by ping should be used only as a comparative reference because it can depend greatly on the software implementation and hardware of the system on which ping is run. If ping fails or returns an unusual RTT, traceroute can be used to help narrow down the problem. It is also possible to vary the size of the ICMP echo payload to test problems related to maximum transmission unit (MTU).



Example 7-2 shows ping returning three values separated with the slash "/," the minimum, average, and maximum RTT. Large differences in these values could indicate network congestion or a network problem. In most cases, the average value accurately portrays the network latency to the destination. By default, ping uses small packets for connectivity testing; the packet size will influence the RTT values. The packet size may be changed in some implementations.

Firewalls and routers can be configured to not allow devices to be pinged but to still permit other types of IP traffic. For this reason, a ping failure between two devices should not be misconstrued as a lack of IP connectivity between those devices. Table 7-2 shows a list of some of the codes returned by the Cisco ping utility, along with their meanings and possible cause.

Table 7-2: Cisco ping Return Codes Code	**Meaning**	**Possible Cause(s)**
!	Each exclamation point indicates receipt of an ICMP echo reply.	The ping completed successfully.
.	Each period indicates that the network server timed	This message can indicate many problems:

	out while waiting for a reply.	problems: • ping was blocked by an access list or firewall. • A router along the path did not have a route to the destination and did not send an ICMP destination unreachable message. • A physical connectivity problem occurred somewhere along the path.
U	An ICMP unreachable message was received.	A router along the path did not have a route to the destination address.
C	An ICMP source quench message was received.	A device along the path—possibly the destination—may be receiving to much traffic; check input queues.
&	An ICMP time exceeded message was received.	A routing loop may have occurred.

1.3.2 traceroute

The traceroute utility sends out either ICMP echo request (Windows) or UDP (most implementations) messages with gradually increasing IP TTL values to probe the path by which a packet traverses the network (see Example 7-3). The first packet with the TTL set to 1 will be discarded by the first hop, and the first hop will send back an ICMP TTL exceeded message sourced from its IP address facing the source of the packet. When the machine running the traceroute receives the ICMP TTL exceeded message, it can determine the hop via the source IP address. This continues until the destination is reached. The destination will return either an ICMP echo reply (Windows) or a ICMP port unreachable, indicating that the

destination had been reached. Cisco's implementation of traceroute sends out three packets at each TTL value, allowing traceroute to report routers that have multiple equal-cost paths to the destination.

Although it may also be possible to trace the path between source and destination using ping and the IP record route option, traceroute is preferred because the record route option can alter the way in which packets are forwarded by routers in the network, yielding incorrect path information.

1.4 General IP Troubleshooting Suggestions

This chapter approaches the process of troubleshooting TCP/IP connectivity issues with the assumption that you will have access to the client (or source) and may not have access to the server (or destination). If the problem is determined to be a server issue, you contact the server administrator. If you are the server administrator, you can apply the troubleshooting process in reverse (server to client) to further troubleshoot connectivity issues. This chapter will not address the specifics of troubleshooting server-side IP services; for this, consult the manual or web page for the software or service running on the server.

Because TCP/IP does not store path information in its packets, it is possible for a packet to have a working path from the source to the destination (or vice versa), but not to have a working path in the opposite direction. For this reason, it may be necessary to perform all troubleshooting steps in both directions along an IP path to determine the cause of a connectivity problem.

1.4.1 Narrowing Down the Problem Domain

To efficiently troubleshoot a TCP/IP connectivity problem, it is necessary to identify a single pair of source and destination devices that are exhibiting the connectivity problem. When you've selected the two devices, test to make sure that the problem is actually occurring between these two devices.

Possible problems include these:

- Physical layer issue somewhere along the path
- First-hop Layer 3 connectivity issue, local LAN segment
- Layer 3 IP connectivity issue somewhere along the packet's path
- Name resolution issue

Where to start:

1. Try to ping from the source to destination device by IP address. If the ping fails, verify that you are using the correct address, and try the ping again. If the ping still fails, go to the next section, "Troubleshooting Local Connectivity Problems." Otherwise, proceed to Step 2.

2. Try to ping from the source to the destination device by name. If the ping fails, verify that the name is correctly spelled and that it refers to the destination device, and then try the ping again. If the ping still fails, go to the section "Troubleshooting Domain Name Server Problems," later in this chapter. Otherwise, proceed to Step 3.

3. If you can ping the destination by both name and address, it appears that the problem is an upper-layer problem. Go to the section "Troubleshooting Upper Layer Problems," later in this chapter.

1.5 Troubleshooting Local Connectivity Problems

This section describes how to troubleshoot local connectivity problems on LAN segments such as Ethernet or Token Ring. Going through the methodology in this chapter with help determine and resolve problems moving packets on the local LAN segment or to the next-hop router. If the problem is determined to be past the local LAN segment, then you will be referred to the section "Troubleshooting IP Connectivity and Routing Problems," later in this chapter. If the source device is connected via a modem, then you should consult Chapter 16, "Troubleshooting Dialup Connections."

Possible problems include these:

- Configuration problem
- DHCP or BOOTP issue
- Physical layer issue
- Duplicate IP address

1.5.1 Check for Configuration Problems

To begin troubleshooting, display and examine the IP configuration of the source device. The method to determine this information varies greatly from platform to platform. If you are unsure of how to display this information, consult the manual for the device or operating system. Refer to the following examples:

- On a Cisco router, use **show ip interface** and **show running-config**.
- On Windows 95 or 98, use **winipcfg.exe**.
- On Windows 2000 or NT, use **ipconfig.exe**.
- On a UNIX platform, use **ifconfig**.

Examine the configuration, looking specifically for the IP address and subnet mask. On Windows 9x or Windows 2000 platforms, the default gateway address should also be displayed.

If no IP address is configured, verify that this node receives its IP address from BOOTP or DHCP. Otherwise, an IP address should be statically configured for this interface. Configure an address if one is not present. If the source is configured to receive an IP address via DHCP or BOOTP and is not receiving one, make sure that the bootp (IP) helper address is configured on the router interface facing the source device.

If the incorrect IP address, subnet mask, or default gateway is configured, verify that this node receives its IP address from BOOTP or DHCP, and then contact the DHCP or BOOTP

administrator. Ask the administrator to troubleshoot the DHCP or BOOTP server's configuration. If the address is statically configured, configure the correct address.

1.5.2 Check for Local Connectivity

If the destination is on the same subnet as the source, try pinging the destination by IP address. If the destination is on a different subnet, then try pinging the default gateway or appropriate next hop obtained from the routing table. If the ping fails, double-check the configuration of the next-hop router to see if the subnet and mask match the source's configuration.

If the configuration is correct, check that the source or next-hop router is capable of pinging any other device on the local LAN segment. If you cannot ping the next-hop address, and if the next-hop address is an HSRP virtual address, try pinging one of the next-hop router's actual IP addresses. If the actual address works but the virtual address does not, you may be experiencing an HSRP issue. Failure to communicate with some or all devices on the LAN segment could indicate a physical connectivity problem, a switch or bridge misconfiguration, or a duplicate IP address.

1.5.3 Ruling Out Duplicate IP Addresses

To rule out a duplicate IP address, you can disconnect the suspect device from the LAN or shut down the suspect interface and then try pinging the device from another device on that same LAN segment. If the ping is successful, then there is another device on that LAN segment using the IP address. You will be able to determine the MAC address of the conflicting device by looking at the ARP table on the device that issued the ping.

If at this point you still do not have local connectivity for either the source or the next-hop router, proceed to the next section.

1.6 Troubleshooting Physical Connectivity Problems

This section describes how to troubleshoot Layer 1 and 2 physical connectivity issues on LANs such as Ethernet or Token Ring. For troubleshooting information on dialup links or WAN connections, consult the chapters in Part IV, "Troubleshooting Serial Lines and WAN Connections."

Even though it may seem logical to first troubleshoot at the physical layer, problems can generally be found more quickly by first troubleshooting at Layer 3 and then working backward when a physical problem is found or suspected.

Possible problems include these:

- Configuration is incorrect.
- Cable is faulty or improperly connected.
- Wiring closet cross-connect is faulty or improperly connected.
- Hardware (interface or port) is faulty.
- Interface has too much traffic.

1.6.1 Rule Out a Configuration Problem

Check to make sure that all cables are connected to the appropriate ports. Make sure that all cross-connects are properly patched to the correct location using the appropriate cable and method. Verify that all switch or hub ports are set in the correct VLAN or collision domain and have appropriate options set for spanning tree and other considerations.

1.6.2 Check Cable Connections

Verify that the proper cable is being used. If this is a direct connection between two end systems (for example, a PC and a router) or between two switches, a special crossover cable may be required. Verify that the cable from the source interface is properly connected and is in good condition. If you doubt that the connection is good, reseat the cable and ensure that the connection is secure. Try replacing the cable with a known working cable. If this cable connects to a wall jack, use a cable tester to ensure that the jack is properly wired. Also check any transceiver in use to ensure that it is the correct type, is properly connected, and is properly configured. If replacing the cable does not resolve the problem, try replacing the transceiver if one is being used.

1.6.3 Check the Configuration

Verify that the interface on the device is configured properly and is not shut down. If the device is connected to a hub or switch, verify that the port on the hub or switch is configured properly and is not shut down. Check both speed and duplex.

1.6.4 Check the Network Interface

Most interfaces or NICs will have indicator lights that show whether there is a valid connection; often this light is called the link light. The interface may also have lights to indicate whether traffic is being sent (TX) or received (RX). If the interface has indicator lights that do not show a valid connection, power off the device and reseat the interface card.

1.7 Troubleshooting IP Connectivity and Routing Problems

When troubleshooting IP connectivity problems across large networks, it always helps to have a network diagram handy so that you can understand the path that the traffic should take and compare it to the path that it is actually taking.

When IP packets are routed across a network, there is the potential for problems at every hop between the source and the destination, so test connectivity at each hop to determine where it is broken is the logical troubleshooting methodology.

The following could be wrong:

- A router may not have a route to the source or destination.
- The network might have a routing loop or other routing protocol-related problem.
- A physical connectivity problem might have occurred.
- A resource problem on one router might be prohibiting proper router operation. This could possibly be caused by lack of memory, lack of buffers, or lack of CPU.
- A configuration problem might have occurred on a router.

- A software problem might have occurred on a router.
- A packet filter or firewall might be preventing traffic from passing for an IP address or protocol.
- An MTU mismatch problem might have occurred.

1.7.1 Determining Where to Start

The most detailed method to find a problem would obviously be to start at the next hop away from the source and work your way one hop at a time toward the destination, exploring all possible paths along the way. You would then test basic IP connectivity and possibly protocol connectivity from each router forward. Although in some cases this method is the only one available, the process can generally be shortened by first performing a traceroute from the source to the destination to determine the first problematic hop. If the traceroute method does not provide an answer, you will have to fall back to the longer method.

When you have found a starting point, connect to that router via telnet or console, and verify that it is capable of pinging the source and the destination. When doing this, keep in mind that the router will source the ping packet from the interface closest to the ping target. In some cases, you may want to use an extended ping to specify a source interface because the ping target may not know how to get to the default source address; this is common on serial interfaces configured with private addressing.

1.7.2 Check for Resources

If the router appears sluggish or does not respond (echo) to what you are typing quickly, or if you suspect a resource issue, check the router's resources. Check memory using **show memory**; be sure not to have **terminal length 0** configured when doing this, or it make take a long time. Look at how much memory is available in the **largest free** field. If this number is low (less than 5 percent of total router memory), use **show process memory** to identify which process(es) are "holding" the memory.

Sluggish router response can also be caused by CPU overload. This can be checked using **show process cpu**. You will see two percentages listed (such as 75%/24%). The first number is the total CPU utilization for the router, and the second is interrupt-generated processor

utilization. If the total CPU utilization is greater than 90 percent for an extended period of time (10 to 15 minutes), then you should investigate what is using all the CPU. **Show process cpu** will show which processes are running and how much CPU they are using. If the CPU is too high, it is possible to lose console and Telnet access to the router.

Although I will not cover all the processes that could possibly be running, a few have special meaning. The IP Input process is tied to process-switched traffic. Some traffic that will frequently cause an increase in process-switched traffic includes broadcast traffic, multicast traffic, routing updates, or traffic destined for an IP address on the router. For example, a broadcast storm will cause IP Input to increase and can cause CPU to jump to 99 percent. You will also see processes for the individual routing protocols such as these:

- IP BGP
- IP EIGRP
- IP OSPF

If a routing protocol is converging, it is possible that one of these processes may increase CPU utilization; in most cases, this is normal.

1.7.3 Check for Connectivity

If you cannot ping from this router to either the source or the destination, check the routing table for a route to the ping target. Keep in mind that it may be desirable for the router to use the default route to this destination, and **ip classless** may need to be configured for this to happen. If there is no route to the ping target, you will need to either troubleshoot your routing protocol, if you are running one, or add a static route to the destination network. The router will need to have both a route to the source and to the destination for communication to succeed.

If ping succeeds only a percentage of the time, look to see if there are multiple paths to the destination. If there are multiple paths, it is possible that one path may be failing while the others are working. This can be symptomatic of a routing loop or physical problem somewhere along the path. The only way to test whether a path is failing is to go to all the next hops and test connectivity from there.

Pings with less than 100 percent success rate can also indicate problematic links or links with high utilization. Look at the interface statistics using **show interface** for outgoing interfaces to see if any have problems. When reviewing statistics, keep in mind that the router may have been collecting information for years; always look at the uptime for the router, reported in **show version**, and the last time that the counters were reset, reported at the top of **show interface**. Generally, the counters can be looked at as an accurate percentage of packets received or sent. If the counters have not been reset in a long time, or if a problem is suspected, the counters should be reset using **clear counters** command, and a new reading should be taken after a reasonable period of time has elapsed. If a problem is detected on a WAN or dialup link, refer to Part IV. If a problem is detected on a LAN connection, see the section "Troubleshooting Physical Connectivity Problems," earlier in this chapter.

1.7.4 Check for ACLs

Check this router for any access lists applied to an interface using **ip access-group**, or any other firewall or packet filters configured. Does the packet filtering permit the desired source/destination to communicate using the requested protocol? If you are unsure, see the section "Troubleshooting Upper-Layer Problems."

1.7.5 Check for Network Address Translation

Check to see if this router is configured for network address translation. If it is, is it supposed to translate packets between the source and destination? Has it been configured correctly?

At this point, you will want to move on to one of the next-hop routers. Record routers that you have already visited on a piece of paper. Also record any problems or questions that arose at the router. This record will help you detect routing loops and will provide useful information if you find it necessary to call for support.

1.8 Troubleshooting Upper-Layer Problems

Even though there may be IP connectivity between a source and a destination, problems may still exist for a specific upper-layer protocols such as FTP, HTTP, or Telnet. These protocols ride on top of the basic IP transport but are subject to protocol-specific problems relating to packet filters and firewalls. It is possible that everything except mail will work between a

given source and destination. Before troubleshooting at this level, it is important to first establish whether IP connectivity exists between the source and the destination. If IP connectivity exists, then the issue must be at the application layer.

The following could go wrong:

- A packet filter/firewall issue might have arisen for the specific protocol, data connection, or return traffic.
- The specific service could be down on the server.
- An authentication problem might have occurred on the server for the source or source network.
- There could be a version mismatch or incompatibility with the client and server software.

1.8.1 Generic

To troubleshoot an upper-layer protocol connectivity problem, you must understand how it works. You can generally find this information in the latest RFC for the protocol or on the developer's web page. Questions that you should answer to make certain that you understand the protocol include these:

- What IP protocols does the protocol use (TCP, UDP, ICMP, IGMP, or other)?
- What TCP or UDP port numbers are used by the protocol?
- Does the protocol require any inbound TCP connections or inbound UDP packets?
- Does the protocol embed IP addresses in the data portion of the packet?
- Are you running a client or a server for the protocol?

If the protocol embeds IP addresses in the data portion of the packet and you have NAT configured anywhere along the path of the packet, the NAT gateway will need to know how to deal with that particular protocol, or the connection will fail. NAT gateways do not typically change information in the data portion of a packet unless they have been specifically

coded to do so. Some examples of protocols that embed IP addresses in the data portion of the packet are FTP, SQLNet, and Microsoft WINs.

If there is a question whether a firewall or router is interfering with the flow of data for a particular application or protocol, you can take several steps to see what exactly is happening. These steps may not all be possible in all situations.

- Move the client outside the firewall or address translation device.
- Verify whether the client can talk to a server on the same subnet as the client.
- Capture a network trace at the client's LAN and on the LAN closest to the server (or, preferably, on the server's LAN, if possible).
- If the service is ASCII-based, you can try Telnetting to the service's port from the router closest to the server; then work backward into the network toward the client.

1.8.2 Hypertext Transport Protocol

HTTP is the protocol used to transfer the files that make up web pages. Although the HTTP specification allows for data to be transferred on port 80 using either TCP or UDP, most implementations use TCP. A secure version of the protocol, SHTTP, uses TCP port 443.

You can test HTTP connectivity using any Telnet application that allows a port number to be specified by Telnetting to the IP address of the destination server on port 80. You should see a hello message, which indicates that you have HTTP connectivity to the server.

1.8.3 FTP

FTP uses two or more TCP connections to accomplish data transfers. To start a session, the FTP client opens a TCP connection to port 21 on the FTP server. This connection is called the *control connection* and is used to pass commands and results between the client and the server. No data, such as file transfers or directory listings, is passed over the control connection; instead, data is transferred over a separate TCP connection created specifically to fulfill that request. This *data connection* can be opened in several different ways:

- **Traditional (or active)**—The FTP server opens a TCP connection back to the client's port 20. This method will not work on a multiuser system because many users may make simultaneous FTP requests, and the system will not be capable of matching incoming FTP data connections to the appropriate user.

- **Multiuser traditional (or active)**—The FTP client instructs the FTP server to open a connection on some random port in the range 1024 through 65535. This method creates a rather large security hole because it requires system administrators to permit inbound TCP connections to all ports greater than 1023. Although firewalls that monitor FTP traffic and dynamically allow inbound connections help close this security hole, many corporate networks do not permit this type of traffic. Most command-line FTP clients default to this method of transfer and offer a **passive** command (or something similar) to switch to passive mode.

- **Passive mode**—The FTP client instructs the FTP server that it wants a passive connection, and the server replies with an IP address and port number to which the FTP client can open a TCP data connection. This method is by far the most secure because it requires no inbound TCP connections to the FTP client. Many corporate networks permit only this type of FTP transfer. Although most of the popular web browsers default to this method of FTP transfer, you shouldn't assume that they do.

You can test the FTP control connection using any Telnet application that allows a port number to be specified. Telnet to the IP address of the destination server using port 21, and you should see a hello message indicating that you have FTP connectivity to the server.

Generally, if a client has connectivity via the control connection but cannot retrieve directory listings or transfer files, there is an issue with opening the data connection. Try specifying passive mode because this is permitted by most firewalls.

Another common problem with FTP is being able to transfer small files but not large files, with the transfer generally failing at the same place or time in every file. Remember that the data connection (and the transfer) will be closed if the control connection closes; because the control connection is typically dormant during large file transfers, it is possible for the connection to close in NAT/PAT environments in which there is a timeout on TCP

connections. Increasing the timeout on dormant TCP connections may resolve this problem. If an FTP client is not properly coded, you may also see this problem.

Because FTP file transfers generally create packets of maximum size, an MTU mismatch problem will almost always cause file transfers to fail in a single direction (*gets* may fail, but *puts* may work). This can be caused by a server located on a LAN media that support larger MTUs (such as Token Ring, which can have an MTU of 4096 or larger). Normally this problem is resolved automatically by fragmentation, but misconfigurations or having the IP Don't Fragment option set in the IP datagrams can prevent proper operation.

1.8.4 MAIL (IMAP, POP, and SMTP)

Two types of machines exist in the e-mail universe, and they work in different ways. E-mail servers communicate with each other using the Simple Mail Transport Protocol (SMTP) to send and receive mail. The SMTP protocol transports e-mail messages in ASCII format using TCP; it's possible to connect to an SMTP server by Telnetting to the SMTP port (25). This is a good way to test whether a mail server is reachable.

When a mail server receives a message destined for a local client, it stores that message and waits for the client to collect the mail. There are several ways for mail clients to collect their mail: They can use programs that access the mail server files directly, or they can collect their mail using one of many network protocols. The most popular mail client protocols are POP3 and IMAP4, which both use TCP to transport data. Even though mail clients use these special protocols to collect mail, they almost always use SMTP to send mail. Because two different protocols, and possibly two different servers, are used to send and receive mail, it is possible that mail clients can perform one task and not the other—so you should troubleshoot sending and receiving mail separately.

When verifying the configuration of a mail client, both the mail relay (SMTP) server and mail (POP or IMAP) servers should be verified. The SMTP protocol does not offer much in the way of security and does not require any sort of authentication, so to prevent unauthorized users from bouncing mail messages off their servers, administrators don't often allow hosts that are not part of their network to use their SMTP server to send (or relay) mail.

You can test SMTP, IMAP, and POP connectivity using any Telnet application that allows a port number to be specified. Telnet to the IP address of the destination server using ports 25, 143, and 110 respectively. You should see a hello message, which indicates that you have connectivity to that server.

1.8.5 Telnet

If the Telnet to a particular server fails from one host, try connecting from a router and several other devices. If when Telnetting to a server you do not receive a login prompt, you will want to check the following:

- Are you able to do a reverse DNS lookup on the client's address? Many Telnet servers will not allow connections from IP addresses that have no DNS entry. This is a common problem for DHCP-assigned addresses in which the administrator has not added DNS entries for the DHCP pools.
- It is possible that your Telnet application cannot negotiate the appropriate options and therefore will not connect. On a Cisco router, you can view this negotiation process using **debug telnet**.
- It is possible that Telnet is disabled or has been moved to a port other than 23 on the destination server.

1.9 Troubleshooting Domain Name Server Problems

It is possible for IP connectivity to work but for DNS name resolution to fail. To troubleshoot this situation, use one of the following methods to determine whether DNS is resolving the name of the destination:

- Ping the destination by name, and look for an error message indicating that the name could not be resolved.

  If you are working on a UNIX machine, use **nslookup <fully-qualified domain name>** to perform a DNS lookup on the destination.

If DNS correctly resolves the host's name, go to the section "Narrowing Down the Problem Domain," earlier in this chapter, to start troubleshooting again. Otherwise, continue troubleshooting as follows:

1. Determine which name server you are using; this can be found in different places on each operating system, so if you are unsure of how to find it, consult the device's manual. For examples:

   - On a Cisco router, type **show run** and look for the **name-server**.
   - On Windows 95 or 98, use **winipcfg.exe**.
   - On Windows 2000 or NT, use **ipconfig.exe**.
   - On a UNIX platform, type **cat /etc/resolv.conf** at a command prompt.

2. Verify that you can ping the name server using its IP address. If the ping fails, go to the section "Narrowing Down the Problem Domain," earlier in this chapter, to troubleshoot connectivity between the client and the name server.

3. Verify that you can resolve names within your domain. (For example, if your host is Host1.test.com, you should be able to resolve the names of other hosts in the test.com domain, such as host2.test.com.)

4. Verify that you can resolve one or more domain names outside your domain.

If you cannot resolve names from all domains except that of the destination, there might be a problem with the DNS for the destination host. Contact the administrator of the destination device.

If you cannot resolve names within your domain or a large number of external domains, contact your DNS administrator because there may be a problem with the local DNS (or your host could be using the wrong domain server).

Chapter 2 Internet Protocols

2.1 Background

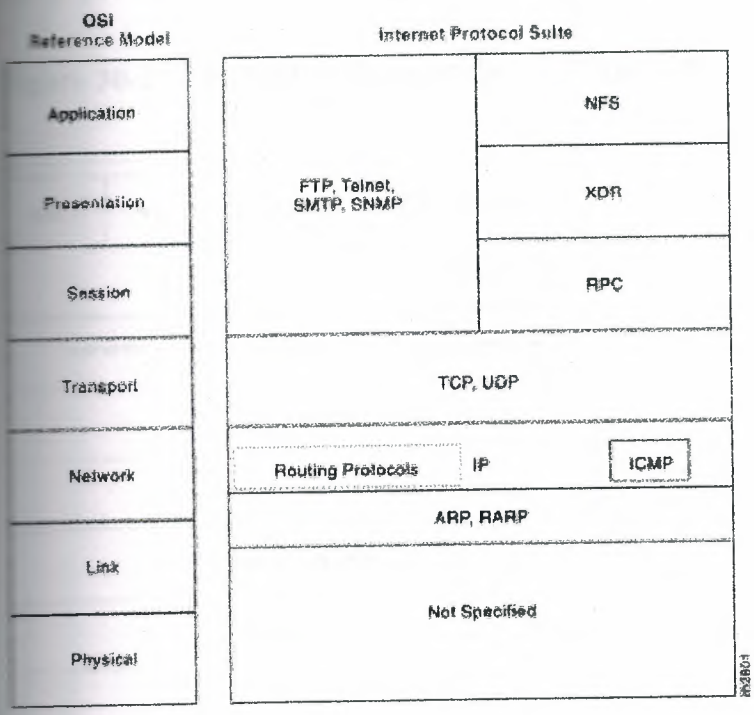
The Internet protocols are the world's most popular open-system (nonproprietary) protocol suite because they can be used to communicate across any set of interconnected networks and are equally well suited for LAN and WAN communications. The Internet protocols consist of a suite of communication protocols, 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. This chapter provides a broad introduction to specifications that comprise the Internet protocols. Discussions include IP addressing and key upper-layer protocols used in the Internet. Specific routing protocols are addressed individually in Part 6, Routing Protocols.

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.

Documentation of the Internet protocols (including new or revised protocols) and policies are specified in technical reports called Request For Comments (RFCs), which are published and then reviewed and analyzed by the Internet community. Protocol refinements are published in the new RFCs. To illustrate the scope of the Internet protocols, Figure 30-1 maps many of the protocols of the Internet protocol suite and their corresponding OSI layers. This chapter addresses the basic elements and operations of these and other key Internet protocols.

Figure 30-1: Internet protocols span the complete range of OSI model layers.



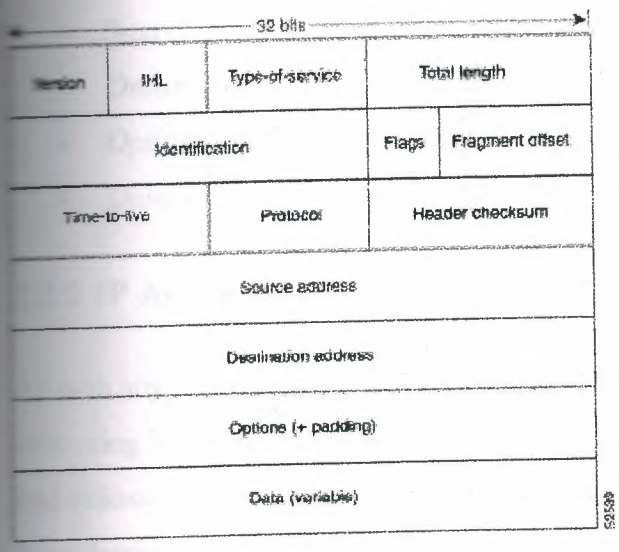
2.2 Internet Protocol (IP)

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.

2.2.1 IP Packet Format

An IP packet contains several types of information, as illustrated in Figure 30-2.

Figure 30-2: Fourteen fields comprise an IP packet.



The following discussion describes the IP packet fields illustrated in Figure 30-2:

- *Version*—Indicates the version of IP currently used.
- *IP Header Length (*IHL)—Indicates the datagram header length in 32-bit words.
- *Type-of-Service*—Specifies how an upper-layer protocol would like a current datagram to be handled, and assigns datagrams various levels of importance.
- *Total Length*—Specifies the length, in bytes, of the entire IP packet, including the data and header.
- *Identification*—Contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.
- *Flags*—Consists of a 3-bit field of which the two low-order (least-significant) bits control fragmentation. The low-order bit specifies whether the packet can be fragmented. The middle bit specifies whether the packet is the last fragment in a series of fragmented packets. The third or high-order bit is not used.
- *Fragment Offset*—Indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.

- *Time-to-Live*—Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.
- *Protocol*—Indicates which upper-layer protocol receives incoming packets after IP processing is complete.
- *Header Checksum*—Helps ensure IP header integrity.
- *Source Address*—Specifies the sending node.
- *Destination Address*—Specifies the receiving node.
- *Options*—Allows IP to support various options, such as security.
- *Data*—Contains upper-layer information.

2.2.2 IP Addressing

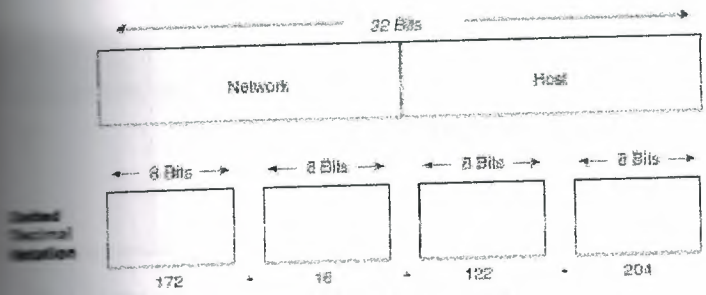
As with any other network-layer protocol, the IP addressing scheme is integral to the process of routing IP datagrams through an internetwork. Each IP address has specific components and follows a basic format. These IP addresses can be subdivided and used to create addresses for subnetworks, as discussed in more detail later in this chapter.

Each host on a TCP/IP network is assigned a unique 32-bit logical address that is divided into two main parts: the network number and the host number. The network number identifies a network and must be assigned by the Internet Network Information Center (InterNIC) if the network is to be part of the Internet. An Internet Service Provider (ISP) can obtain blocks of network addresses from the InterNIC and can itself assign address space as necessary. The host number identifies a host on a network and is assigned by the local network administrator.

2.2.3 IP Address Format

The 32-bit IP address is grouped eight bits at a time, separated by dots, and represented in decimal format (known as *dotted decimal notation*). Each bit in the octet has a binary weight (128, 64, 32, 16, 8, 4, 2, 1). The minimum value for an octet is 0, and the maximum value for an octet is 255. Figure 30-3 illustrates the basic format of an IP address.

Figure 30-3: An IP address consists of 32 bits, grouped into four octets.



2.2.4 IP Address Classes

IP addressing supports five different address classes: A, B,C, D, and E. Only classes A, B, and C are available for commercial use. The left-most (high-order) bits indicate the network class. Table 30-1 provides reference information about the five IP address classes.

Table 30-1: Reference Information About the Five IP Address Classes

IP Address Class	Format	Purpose	High-Order Bit(s)	Address Range	No. Bits Network/Host	Max. Hosts
A	N.H.H.H[1]	Few large organizations	0	1.0.0.0 to 126.0.0.0	7/24	16,777, 214[2] (2^{24} - 2)
B	N.N.H.H	Medium-size organizations	1, 0	128.1.0.0 to 191.254.0.0	14/16	65, 543 (2^{16} - 2)
C	N.N.N.H	Relatively small organizations	1, 1, 0	192.0.1.0 to 223.255.254.0	22/8	245 (2^{8} - 2)

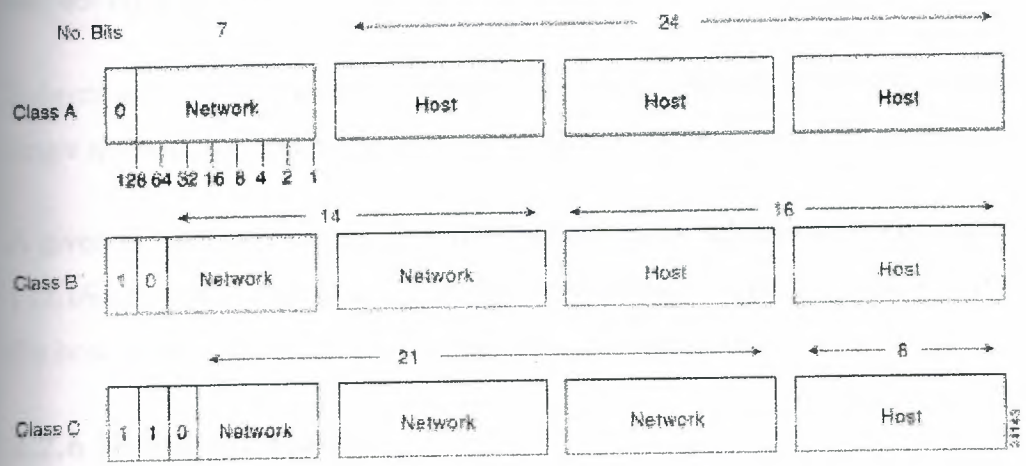
D	N/A	Multicast groups (RFC 1112)	1, 1, 1, 0	224.0.0.0 to 239.255.255.255	N/A (not for commercial use)	N/A
E	N/A	Experimental	1, 1, 1, 1	240.0.0.0 to 254.255.255.255	N/A	N/A

[1]N = Network number, H = Host number.

[2]One address is reserved for the broadcast address, and one address is reserved for the network.

Figure 30-4 illustrates the format of the commercial IP address classes. (Note the high-order bits in each class.)

Figure 30-4: IP address formats A, B, and C are available for commercial use.



The class of address can be determined easily by examining the first octet of the address and mapping that value to a class range in the following table. In an IP address of 172.31.1.2, for example, the first octet is 172. Because 172 falls between 128 and 191, 172.31.1.2 is a Class B address. Figure 30-5 summarizes the range of possible values for the first octet of each address class.

Figure 30-5: A range of possible values exists for the first octet of each address class.

Address Class	First Octet in Decimal	High-Order Bits
Class A	1 Ð 126	0
Class B	128 Ð 191	10
Class C	192 Ð 223	110
Class D	224 Ð 239	1110
Class E	240 Ð 254	1111

2.2.5 IP Subnet Addressing

IP networks can be divided into smaller networks called subnetworks (or subnets). Subnetting provides the network administrator with several benefits, including extra flexibility, more efficient use of network addresses, and the capability to contain broadcast traffic (a broadcast will not cross a router).

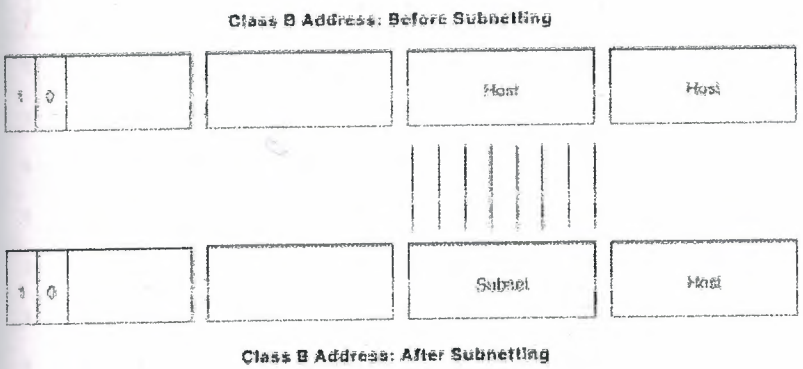
Subnets are under local administration. As such, the outside world sees an organization as a single network and has no detailed knowledge of the organization's internal structure.

A given network address can be broken up into many subnetworks. For example, 172.16.1.0, 172.16.2.0, 172.16.3.0, and 172.16.4.0 are all subnets within network 171.16.0.0. (All 0s in the host portion of an address specifies the entire network.)

2.2.6 IP Subnet Mask

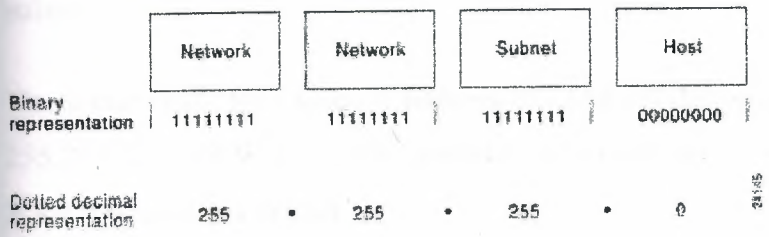
A subnet address is created by "borrowing" bits from the host field and designating them as the subnet field. The number of borrowed bits varies and is specified by the subnet mask. Figure 30-6 shows how bits are borrowed from the host address field to create the subnet address field.

Figure 30-6: Bits are borrowed from the host address field to create the subnet address field.



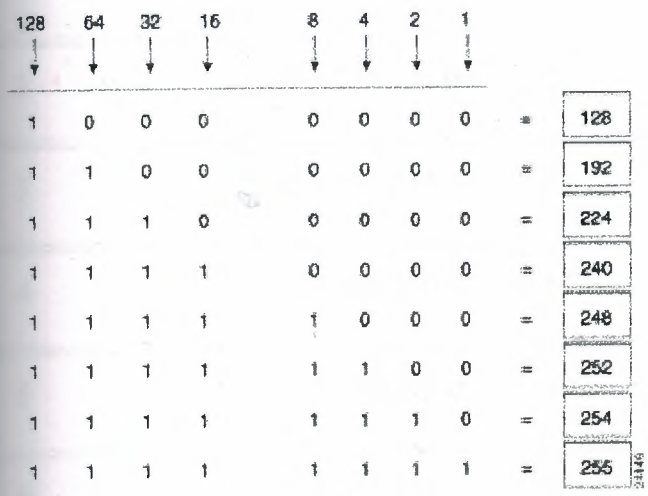
Subnet masks use the same format and representation technique as IP addresses. The subnet mask, however, has binary 1s in all bits specifying the network and subnetwork fields, and binary 0s in all bits specifying the host field. Figure 30-7 illustrates a sample subnet mask.

Figure 30-7: A sample subnet mask consists of all binary 1s and 0s.



Subnet mask bits should come from the high-order (left-most) bits of the host field, as Figure 30-8 illustrates. Details of Class B and C subnet mask types follow. Class A addresses are not discussed in this chapter because they generally are subnetted on an 8-bit boundary.

Figure 30-8: Subnet mask bits come from the high-order bits of the host field.



Various types of subnet masks exist for Class B and C subnets.

The default subnet mask for a Class B address that has no subnetting is 255.255.0.0, while the subnet mask for a Class B address 171.16.0.0 that specifies eight bits of subnetting is 255.255.255.0. The reason for this is that eight bits of subnetting or 2^8 - 2 (1 for the network address and 1 for the broadcast address) = 254 subnets possible, with 2^8 - 2 = 254 hosts per subnet.

The subnet mask for a Class C address 192.168.2.0 that specifies five bits of subnetting is 255.255.255.248. With five bits available for subnetting, 2^5 - 2 = 30 subnets possible, with 2^3 - 2 = 6 hosts per subnet.

The reference charts shown in table 30-2 and table 30-3 can be used when planning Class B and C networks to determine the required number of subnets and hosts, and the appropriate subnet mask.

Table 30-2: Class B Subnetting Reference Chart

Number of Bits	Subnet Mask	Number of Subnets	Number of Hosts
2	255.255.192.0	2	16382
3	255.255.224.0	6	8190
4	255.255.240.0	14	4094
5	255.255.248.0	30	2046
6	255.255.252.0	62	1022
7	255.255.254.0	126	510
8	255.255.255.0	254	254
9	255.255.255.128	510	126
10	255.255.255.192	1022	62
11	255.255.255.224	2046	30
12	255.255.255.240	4094	14
13	255.255.255.248	8190	6
14	255.255.255.252	16382	2

Table 30-3: Class C Subnetting Reference Chart

Number of Bits	Subnet Mask	Number of Subnets	Number of Hosts
2	255.255.255.192	2	62
3	255.255.255.224	6	30
4	255.255.255.240	14	14
5	255.255.255.248	30	6
6	255.255.255.252	62	2

2.2.7 How Subnet Masks are Used to Determine the Network Number

The router performs a set process to determine the network (or more specifically, the subnetwork) address. First, the router extracts the IP destination address from the incoming packet and retrieves the internal subnet mask. It then performs a *logical AND* operation to obtain the network number. This causes the host portion of the IP destination address to be removed, while the destination network number remains. The router then looks up the destination network number and matches it with an outgoing interface. Finally, it forwards the frame to the destination IP address. Specifics regarding the logical AND operation are discussed in the following section.

Logical AND Operation

Three basic rules govern logically "ANDing" two binary numbers. First, 1 "ANDed" with 1 yields 1. Second, 1 "ANDed" with 0 yields 0. Finally, 0 "ANDed" with 0 yields 0. The truth table provided in table 30-4 illustrates the rules for logical AND operations.

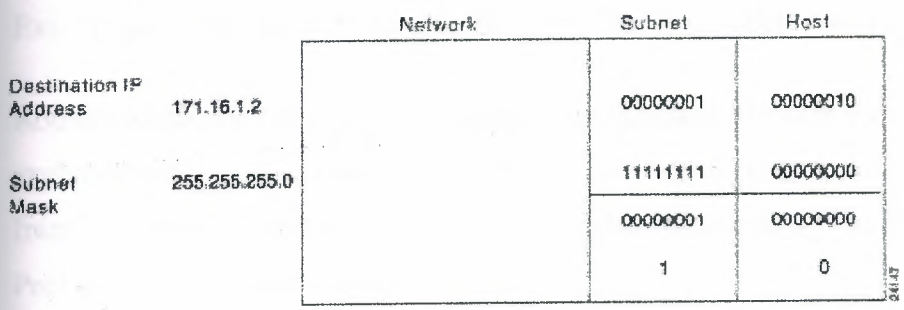
Table 30-4: Rules for Logical AND Operations

Input	Input	Output
1	1	1
1	0	0
0	1	0
0	0	0

Two simple guidelines exist for remembering logical AND operations: Logically "ANDing" a 1 with a 1 yields the original value, and logically "ANDing" a 0 with any number yields 0.

Figure 30-9 illustrates that when a logical AND of the destination IP address and the subnet mask is performed, the subnetwork number remains, which the router uses to forward the packet.

Figure 30-9: Applying a logical AND the destination IP address and the subnet mask produces the subnetwork number.



2.2.5 Address Resolution Protocol (ARP) Overview

For two machines on a given network to communicate, they must know the other machine's physical (or MAC) addresses. By broadcasting Address Resolution Protocols (ARPs), a host can dynamically discover the MAC-layer address corresponding to a particular IP network-layer address.

After receiving a MAC-layer address, IP devices create an ARP cache to store the recently acquired IP-to-MAC address mapping, thus avoiding having to broadcast ARPS when they want to recontact a device. If the device does not respond within a specified time frame, the cache entry is flushed.

In addition to the Reverse Address Resolution Protocol (RARP) is used to map MAC-layer addresses to IP addresses. RARP, which is the logical inverse of ARP, might be used by diskless workstations that do not know their IP addresses when they boot. RARP relies on the presence of a RARP server with table entries of MAC-layer-to-IP address mappings.

2.3 Internet Routing

Internet routing devices traditionally have been called gateways. In today's terminology, however, the term gateway refers specifically to a device that performs application-layer protocol translation between devices. Interior gateways refer to devices that perform these protocol functions between machines or networks under the same administrative control or authority, such as a corporation's internal network. These are known as autonomous systems. Exterior gateways perform protocol functions between independent networks.

Routers within the Internet are organized hierarchically. Routers used for information exchange within autonomous systems are called interior routers, which use a variety of Interior Gateway Protocols (IGPs) to accomplish this purpose. The Routing Information Protocol (RIP) is an example of an IGP.

Routers that move information between autonomous systems are called exterior routers. These routers use an exterior gateway protocol to exchange information between autonomous systems. The Border Gateway Protocol (BGP) is an example of an exterior gateway protocol.

2.3.1 IP Routing

IP routing protocols are dynamic. Dynamic routing calls for routes to be calculated automatically at regular intervals by software in routing devices. This contrasts with static routing, where routers are established by the network administrator and do not change until the network administrator changes them.

An IP routing table, which consists of destination address/next hop pairs, is used to enable dynamic routing. An entry in this table, for example, would be interpreted as follows: to get to network 172.31.0.0, send the packet out Ethernet interface 0 (E0).

IP routing specifies that IP datagrams travel through internetworks one hop at a time. The entire route is not known at the onset of the journey, however. Instead, at each stop, the next destination is calculated by matching the destination address within the datagram with an entry in the current node's routing table.

Each node's involvement in the routing process is limited to forwarding packets based on internal information. The nodes do not monitor whether the packets get to their final destination, nor does IP provide for error reporting back to the source when routing anomalies occur. This task is left to another Internet protocol, the Internet Control-Message Protocol (ICMP), which is discussed in the following section.

2.4 Internet Control Message Protocol (ICMP)

The *Internet Control Message Protocol (ICMP)* is a network-layer Internet protocol that provides message packets to report errors and other information regarding IP packet processing back to the source. ICMP is documented in RFC 792.

2.4.1 ICMP Messages

ICMPs generate several kinds of useful messages, including Destination Unreachable, Echo Request and Reply, Redirect, Time Exceeded, and Router Advertisement and Router Solicitation. If an ICMP message cannot be delivered, no second one is generated. This is to avoid an endless flood of ICMP messages.

2.4.2 ICMP Router-Discovery Protocol (IDRP)

IDRP uses Router-Advertisement and Router-Solicitation messages to discover the addresses of routers on directly attached subnets. Each router periodically multicasts Router-Advertisement messages from each of its interfaces. Hosts then discover addresses of routers on directly attached subnets by listening for these messages. Hosts can use Router-Solicitation messages to request immediate advertisements rather than waiting for unsolicited messages.

IRDP offers several advantages over other methods of discovering addresses of neighboring routers. Primarily, it does not require hosts to recognize routing protocols, nor does it require manual configuration by an administrator.

Router-Advertisement messages enable hosts to discover the existence of neighboring routers, but not which router is best to reach a particular destination. If a host uses a poor first-hop router to reach a particular destination, it receives a Redirect message identifying a better choice.

2.5 Transmission Control Protocol (TCP)

The TCP provides reliable transmission of data in an IP environment. TCP corresponds to the transport layer (Layer 4) of the OSI reference model. Among the services TCP provides are stream data transfer, reliability, efficient flow control, full-duplex operation, and multiplexing.

With stream data transfer, TCP delivers an unstructured stream of bytes identified by sequence numbers. This service benefits applications because they do not have to chop data into blocks before handing it off to TCP. Instead, TCP groups bytes into segments and passes them to IP for delivery.

TCP offers reliability by providing connection-oriented, end-to-end reliable packet delivery through an internetwork. It does this by sequencing bytes with a forwarding acknowledgment number that indicates to the destination the next byte the source expects to receive. Bytes not acknowledged within a specified time period are retransmitted. The reliability mechanism of TCP allows devices to deal with lost, delayed, duplicate, or misread packets. A time-out mechanism allows devices to detect lost packets and request retransmission.

TCP offers efficient flow control, which means that, when sending acknowledgments back to the source, the receiving TCP process indicates the highest sequence number it can receive without overflowing its internal buffers.

Full-duplex operation means that TCP processes can both send and receive at the same time.

Finally, TCP's multiplexing means that numerous simultaneous upper-layer conversations can be multiplexed over a single connection.

2.5.1 TCP Connection Establishment

To use reliable transport services, TCP hosts must establish a connection-oriented session with one another. Connection establishment is performed by using a "three-way handshake" mechanism.

A three-way handshake synchronizes both ends of a connection by allowing both sides to agree upon initial sequence numbers. This mechanism also guarantees that both sides are ready to transmit data and know that the other side is ready to transmit as well. This is necessary so that packets are not transmitted or retransmitted during session establishment or after session termination.

Each host randomly chooses a sequence number used to track bytes within the stream it is sending and receiving. Then, the three-way handshake proceeds in the following manner:

The first host (Host A) initiates a connection by sending a packet with the initial sequence number (X) and SYN bit set to indicate a connection request. The second host (Host B) receives the SYN, records the sequence number X, and replies by acknowledging the SYN (with an ACK = X + 1). Host B includes its own initial sequence number (SEQ = Y). An ACK = 20 means the host has received bytes 0 through 19 and expects byte 20 next. This technique is called *forward acknowledgment*. Host A then acknowledges all bytes Host B sent with a forward acknowledgment indicating the next byte Host A expects to receive (ACK = Y + 1). Data transfer then can begin.

2.5.2 Positive Acknowledgment and Retransmission (PAR)

A simple transport protocol might implement a reliability-and-flow-control technique where the source sends one packet, starts a timer, and waits for an acknowledgment before sending a new packet. If the acknowledgment is not received before the timer expires, the source retransmits the packet. Such a technique is called *positive acknowledgment and retransmission* (PAR).

By assigning each packet a sequence number, PAR enables hosts to track lost or duplicate packets caused by network delays that result in premature retransmission. The sequence numbers are sent back in the acknowledgments so that the acknowledgments can be tracked.

PAR is an inefficient use of bandwidth, however, because a host must wait for an acknowledgment before sending a new packet, and only one packet can be sent at a time.

2.5.3 TCP Sliding Window

A *TCP sliding window* provides more efficient use of network bandwidth than PAR because it enables hosts to send multiple bytes or packets before waiting for an acknowledgment.

In TCP, the receiver specifies the current window size in every packet. Because TCP provides a byte-stream connection, window sizes are expressed in bytes. This means that a window is the number of data bytes that the sender is allowed to send before waiting for an acknowledgment. Initial window sizes are indicated at connection setup, but might vary throughout the data transfer to provide flow control. A window size of zero, for instance, means "Send no data."

In a TCP sliding-window operation, for example, the sender might have a sequence of bytes to send (numbered 1 to 10) to a receiver who has a window size of five. The sender then would place a window around the first five bytes and transmit them together. It would then wait for an acknowledgment.

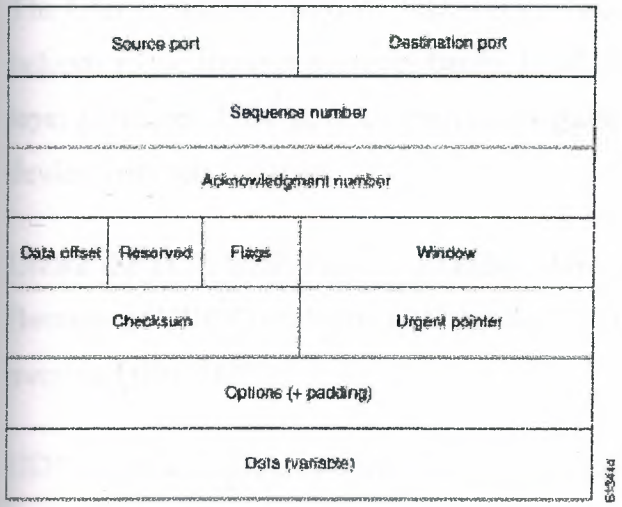
The receiver would respond with an ACK = 6, indicating that it has received bytes 1 to 5 and is expecting byte 6 next. In the same packet, the receiver would indicate that its window size is 5. The sender then would move the sliding window five bytes to the right and transmit bytes 6 to 10. The receiver would respond with an ACK = 11, indicating that it is expecting

sequenced byte 11 next. In this packet, the receiver might indicate that its window size is 0 (because, for example, its internal buffers are full). At this point, the sender cannot send any more bytes until the receiver sends another packet with a window size greater than 0.

2.5.4 TCP Packet Format

Figure 30-10 illustrates the fields and overall format of a TCP packet.

Figure 30-10: Twelve fields comprise a TCP packet.



2.5.5 TCP Packet Field Descriptions

The following descriptions summarize the TCP packet fields illustrated in Figure 30-10:

- *Source Port* and *Destination Port*—Identifies points at which upper-layer source and destination processes receive TCP services.
- *Sequence Number*—Usually specifies the number assigned to the first byte of data in the current message. In the connection-establishment phase, this field also can be used to identify an initial sequence number to be used in an upcoming transmission.
- *Acknowledgment Number*—Contains the sequence number of the next byte of data the sender of the packet expects to receive.
- *Data Offset*—Indicates the number of 32-bit words in the TCP header.
- *Reserved*—Remains reserved for future use.

- *Flags*—Carries a variety of control information, including the SYN and ACK bits used for connection establishment, and the FIN bit used for connection termination.
- *Window*—Specifies the size of the sender's receive window (that is, the buffer space available for incoming data).
- *Checksum*—Indicates whether the header was damaged in transit.
- *Urgent Pointer*—Points to the first urgent data byte in the packet.
- *Options*—Specifies various TCP options.
- *Data*—Contains upper-layer information.

2.5.6 User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) is a connectionless transport-layer protocol (Layer 4) that belongs to the Internet protocol family. UDP is basically an interface between IP and upper-layer processes. UDP protocol ports distinguish multiple applications running on a single device from one another.

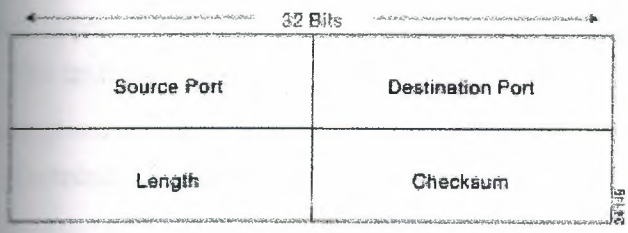
Unlike the TCP, UDP adds no reliability, flow-control, or error-recovery functions to IP. Because of UDP's simplicity, UDP headers contain fewer bytes and consume less network overhead than TCP.

UDP is useful in situations where the reliability mechanisms of TCP are not necessary, such as in cases where a higher-layer protocol might provide error and flow control.

UDP is the transport protocol for several well-known application-layer protocols, including Network File System (NFS), Simple Network Management Protocol (SNMP), Domain Name System (DNS), and Trivial File Transfer Protocol (TFTP).

The UDP packet format contains four fields, as shown in Figure 30-11. These include source and destination ports, length, and checksum fields.

Figure 30-11: A UDP packet consists of four fields.



Source and destination ports contain the 16-bit UDP protocol port numbers used to demultiplex datagrams for receiving application-layer processes. A length field specifies the length of the UDP header and data. Checksum provides an (optional) integrity check on the UDP header and data.

2.6 Internet Protocols Application-Layer Protocols

The Internet protocol suite includes many application-layer protocols that represent a wide variety of applications, including the following:

- *File Transfer Protocol (FTP)*—Moves files between devices
- *Simple Network-Management Protocol (SNMP)*—Primarily reports anomalous network conditions and sets network threshold values
- *Telnet*—Serves as a terminal emulation protocol
- *X Windows*—Serves as a distributed windowing and graphics system used for communication between X terminals and UNIX workstations
- *Network File System (NFS), External Data Representation (XDR), and Remote Procedure Call (RPC)*—Work together to enable transparent access to remote network resources
- *Simple Mail Transfer Protocol (SMTP)*—Provides electronic mail services
- *Domain Name System (DNS)*—Translates the names of network nodes into network addresses

Table 30-5 lists these higher-layer protocols and the applications that they support.

Table 30-5: Higher-Layer Protocols and Their Applications

Application	Protocols
File transfer	FTP
Terminal emulation	Telnet
Electronic mail	SMTP
Network management	SNMP
Distributed file services	NFS, XDR, RPC, X Windows

Chapter 3 Subnetting an IP Address Space

This appendix provides a partial listing of a Class B area intended to be divided into approximately 500 Open Shortest Path First (OSPF) areas. For the purposes of this example, the network is assumed to be a Class B network with the address 150.100.0.0.

Only the address space for two of 512 areas is shown in Table A-1. These areas are defined with the base address 150.100.2.0. Illustrating the entire address space for 150.100.0.0 would require hundreds of additional pages of addressing information. Each area would require the equivalent number of entries for each of the example areas illustrated here.

Table A-1 illustrates the assignment of 255 IP addresses that have been split between two OSPF areas. Table A-1 also illustrates the boundaries of the subnets and of the two OSPF areas shown (area 8 and area 17).

For the purposes of this discussion, consider a network that requires point-to-point serial links in each area to be assigned a subnet mask that allows two hosts per subnet. All other subnets are to be allowed 14 hosts per subnet. The use of bit-wise subnetting and variable-length subnet masks (VLSMs) permit you to customize your address space by facilitating the division of address spaces into smaller groupings than is allowed when subnetting along octet boundaries. The address layout shown in Table A-1 illustrates a structured approach to assigning addresses that uses VLSM. Table A-1 presents two subnet masks: 255.255.255.240 and of 255.255.255.252. The first mask creates subnet address spaces that are four bits wide; the second mask creates subnet address spaces that are two bits wide.

Because of the careful assignment of addresses, each area can be summarized with a single area router configuration command (used to define address range). The first set of addresses starting with 150.100.2.0xxxxxxx (last octet represented here in binary) can be summarized into the backbone.

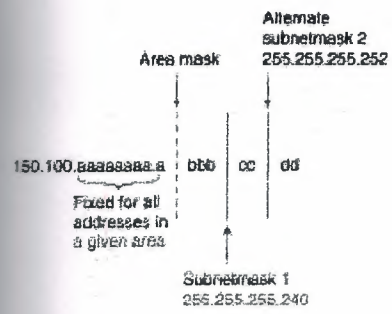
Allocation of subnets allows you to decide where to draw the line between the subnet and host (using a subnet mask) within each area. Note that in this example there are only seven bits remaining to use because of the creation of the artificial area mask. The nine bits to the left of the area mask are actually part of the subnet portion of the address. By keeping these nine bits

the same for all addresses in a given area, route summarization is easily achieved at area border routers, as illustrated by the scheme used in Table A-1.

Table A-1 lists individual subnets, valid IP addresses, subnet identifiers, and broadcast addresses. This method of assigning addresses for the VLSM portion of the address space guarantees that there is no address overlap. If the requirement had been different, any number of the larger subnets might be chosen and divided into smaller ranges with fewer hosts, or combined into several ranges to create subnets with more hosts.

The design approach used in this appendix allows the area mask boundary and subnet masks to be assigned to any point in the address space, which provides significant design flexibility. A change in the specification of the area mask boundary or subnet masks may be required if a network outgrows its initial address space design. In Table A-1, the area mask boundary is to the right of the most significant bit of the last octet of the address, as shown by Figure A-1.

Figure A-1: Breakdown of the addresses assigned by the example.



With a subnet mask of 255.255.255.240, the a and b bits together represent the subnet portion of the address, whereas the c and d bits together provide four-bit host identifiers. When a subnet mask of 255.255.255.252 (a typical subnet mask for point-to-point serial lines), the a, b, and c bits together represent the subnet portion of the address, and the d bits provide two-bit host identifiers. As mentioned earlier, the purpose of the area mask is to keep all of the a bits constant in a given OSPF area (independent of the subnet mask) so that route summarization is easy to apply.

The following steps outline the process used to allocate addresses:

Step 1 Determine the number of areas required for your OSPF network. A value of 500 is used for this example.

Step 2 Create an artificial area mask boundary in your address space. This example uses nine bits of subnet addressing space to identify the areas uniquely. Because $2^9 = 512$, nine bits of subnet meet our requirement of 500 areas.

Step 3 Determine the number of subnets required in each area and the maximum number of hosts required per subnet. This allows you to determine the placement of the subnet mask(s). In Table A-1, the requirement is for seven subnets with 14 hosts each and four subnets with two hosts each.

Chapter 4 Designing Large-Scale IP Internetworks

This chapter focuses on the following design implications of the Enhanced Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF) protocols, and the Border Gateway Protocol (BGP):

- Network Topology
- Addressing and Route Summarization
- Route Selection
- Convergence
- Network Scalability
- Security

Enhanced IGRP, OSPF, and BGP are routing protocols for the Internet Protocol (IP). An introductory discussion outlines general routing protocol issues; subsequent discussions focus on design guidelines for the specific IP protocols.

4.1Implementing Routing Protocols

The following discussion provides an overview of the key decisions you must make when selecting and deploying routing protocols. This discussion lays the foundation for subsequent discussions regarding specific routing protocols.

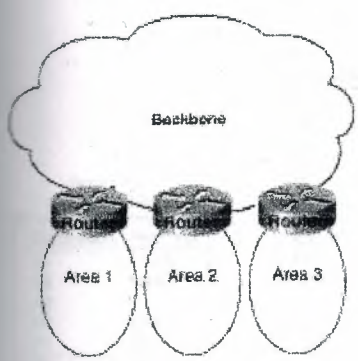
4.1.1 Network Topology

The physical topology of an internetwork is described by the complete set of routers and the networks that connect them. Networks also have a logical topology. Different routing protocols establish the logical topology in different ways.

Some routing protocols do not use a logical hierarchy. Such protocols use addressing to segregate specific areas or domains within a given internetworking environment and to establish a logical topology. For such nonhierarchical, or *flat*, protocols, no manual topology creation is required.

Other protocols require the creation of an explicit hierarchical topology through establishment of a backbone and logical areas. The OSPF and Intermediate System-to-Intermediate System (IS-IS) protocols are examples of routing protocols that use a hierarchical structure. A general hierarchical network scheme is illustrated in Figure 3-1. The explicit topology in a hierarchical scheme takes precedence over the topology created through addressing.

Figure 3-1: Hierarchical network.



If a hierarchical routing protocol is used, the addressing topology should be assigned to reflect the hierarchy. If a flat routing protocol is used, the addressing implicitly creates the topology. There are two recommended ways to assign addresses in a hierarchical network. The simplest way is to give each area (including the backbone) a unique network address. An alternative is to assign address ranges to each area.

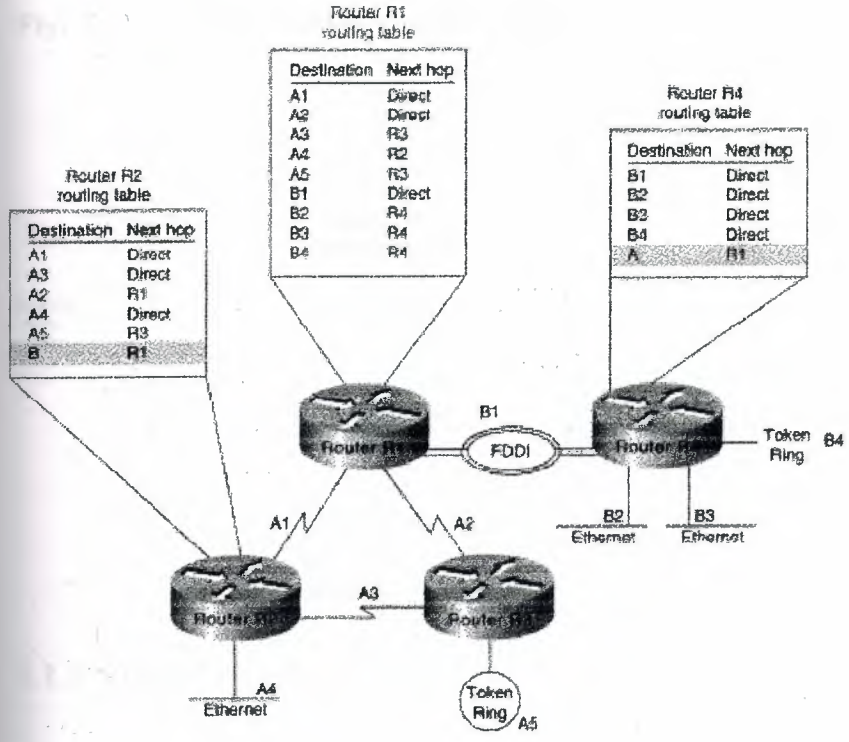
Areas are logical collections of contiguous networks and hosts. Areas also include all the routers having interfaces on any one of the included networks. Each area runs a separate copy of the basic routing algorithm. Therefore, each area has its own topological database.

4.1.2 Addressing and Route Summarization

Route summarization procedures condense routing information. Without summarization, each router in a network must retain a route to every subnet in the network. With summarization, routers can reduce some sets of routes to a single advertisement, reducing both the load on the router and the perceived complexity of the network. The importance of route summarization increases with network size.

Figure 3-2 illustrates an example of route summarization. In this environment, Router R2 maintains one route for all destination networks beginning with B, and Router R4 maintains one route for all destination networks beginning with A. This is the essence of route summarization. Router R1 tracks all routes because it exists on the boundary between A and B.

Figure 3-2: Route summarization example.

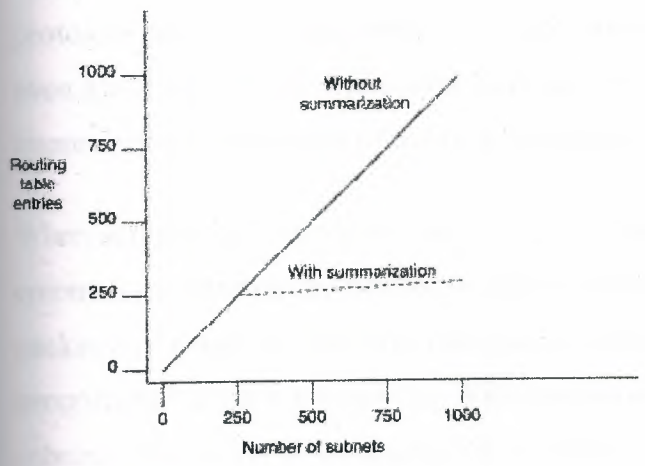


The reduction in route propagation and routing information overhead can be significant. Figure 3-3 illustrates the potential savings. The vertical axis of Figure 3-3 shows the number of routing table entries. The horizontal axis measures the number of subnets. Without summarization, each router in a network with 1,000 subnets must contain 1,000 routes. With summarization, the picture changes considerably. If you assume a Class B network with eight bits of subnet address space, each router needs to know all of the routes for each subnet in its network number (250 routes, assuming that 1,000 subnets fall into four major networks of 250 routers each) plus one route for each of the other networks (three) for a total of 253 routes. This represents a nearly 75-percent reduction in the size of the routing table.

The preceding example shows the simplest type of route summarization: collapsing all the subnet routes into a single network route. Some routing protocols also support route summarization at any bit boundary (rather than just at major network number boundaries) in a network address. A routing protocol can summarize on a bit boundary only if it supports *variable-length subnet masks* (VLSMs).

Some routing protocols summarize automatically. Other routing protocols require manual configuration to support route summarization, as shown in Figure 3-3.

Figure 3-3: Route summarization benefits.

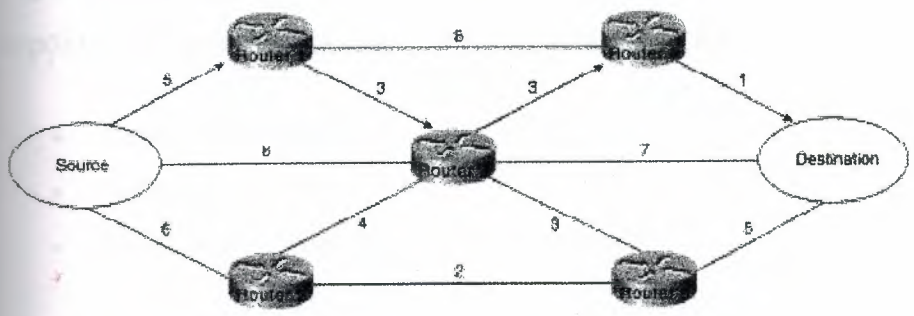


4.1.3 Route Selection

Route selection is trivial when only a single path to the destination exists. However, if any part of that path should fail, there is no way to recover. Therefore, most networks are designed with multiple paths so there are alternatives in case a failure occurs.

Routing protocols compare route metrics to select the best route from a group of possible routes. Route metrics are computed by assigning a characteristic or set of characteristics to each physical network. The metric for the route is an aggregation of the characteristics of each physical network in the route. Figure 3-4 shows a typical meshed network with metrics assigned to each link and the best route from source to destination identified.

Figure 3-4: Routing metrics and route selection.



Routing protocols use different techniques for assigning metrics to individual networks. Further, each routing protocol forms a metric aggregation in a different way. Most routing protocols can use multiple paths if the paths have an equal cost. Some routing protocols can even use multiple paths when paths have an unequal cost. In either case, load balancing can improve overall allocation of network bandwidth.

When multiple paths are used, there are several ways to distribute the packets. The two most common mechanisms are *per-packet load balancing* and *per-destination load balancing*. Per-packet load balancing distributes the packets across the possible routes in a manner proportional to the route metrics. With equal-cost routes, this is equivalent to a round-robin scheme. One packet or destination (depending on switching mode) is distributed to each possible path. Per-destination load balancing distributes packets across the possible routes based on destination. Each new destination is assigned the next available route. This technique tends to preserve packet order.

Note Most TCP implementations can accommodate out-of-order packets. However, out-of-order packets may cause performance degradation.

When fast switching is enabled on a router (default condition), route selection is done on a per- destination basis. When fast switching is disabled, route selection is done on a per-packet basis. For line speeds of 56 Kbps and faster, fast switching is recommended.

4.1.4 Convergence

When *network* topology changes, network traffic must reroute quickly. The phrase "convergence time" describes the time it takes a router to start using a new route after a topology changes. Routers must do three things after a topology changes:

- Detect the change
- Select a new route
- Propagate the changed route information

Some changes are immediately detectable. For example, serial line failures that involve carrier loss are immediately detectable by a router. Other failures are harder to detect. For example, if a serial line becomes unreliable but the carrier is not lost, the unreliable link is not immediately detectable. In addition, some media (Ethernet, for example) do not provide physical indications such as carrier loss. When a router is reset, other routers do not detect this immediately. In general, failure detection is dependent on the media involved and the routing protocol used.

Once a failure has been detected, the routing protocol must select a new route. The mechanisms used to do this are protocol-dependent. All routing protocols must propagate the changed route. The mechanisms used to do this are also protocol-dependent.

4.1.5 Network Scalability

The capability to extend your internetwork is determined, in part, by the scaling characteristics of the routing protocols used and the quality of the network design.

Network scalability is limited by two factors: operational issues and technical issues. Typically, operational issues are more significant than technical issues. Operational scaling concerns encourage the use of large areas or protocols that do not require hierarchical structures. When hierarchical protocols are required, technical scaling concerns promote the use of small areas. Finding the right balance is the art of network design.

From a technical standpoint, routing protocols scale well if their resource use grows less than linearly with the growth of the network. Three critical resources are used by routing protocols: memory, central processing unit (CPU), and bandwidth.

4.1.5.1 Memory

Routing protocols use memory to store routing tables and topology information. Route summarization cuts memory consumption for all routing protocols. Keeping areas small reduces the memory consumption for hierarchical routing protocols.

4.1.5.2 CPU

CPU usage is protocol-dependent. Some protocols use CPU cycles to compare new routes to existing routes. Other protocols use CPU cycles to regenerate routing tables after a topology change. In most cases, the latter technique will use more CPU cycles than the former. For link-state protocols, keeping areas small and using summarization reduces CPU requirements by reducing the effect of a topology change and by decreasing the number of routes that must be recomputed after a topology change.

4.1.5.3 Bandwidth

Bandwidth usage is also protocol-dependent. Three key issues determine the amount of bandwidth a routing protocol consumes:

- *When routing information is sent*—Periodic updates are sent at regular intervals. Flash updates are sent only when a change occurs.
- *What routing information is sent*—Complete updates contain all routing information. Partial updates contain only changed information.
- *Where routing information is sent*—Flooded updates are sent to all routers. Bounded updates are sent only to routers that are affected by a change.

Note These three issues also affect CPU usage.

Distance vector protocols such as Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), Internetwork Packet Exchange (IPX) RIP, IPX Service Advertisement Protocol (SAP), and Routing Table Maintenance Protocol (RTMP), broadcast their complete routing table periodically, regardless of whether the routing table has changed. This periodic advertisement varies from every 10 seconds for RTMP to every 90 seconds for

IGRP. When the network is stable, distance vector protocols behave well but waste bandwidth because of the periodic sending of routing table updates, even when no change has occurred. When a failure occurs in the network, distance vector protocols do not add excessive load to the network, but they take a long time to reconverge to an alternative path or to flush a bad path from the network.

Link-state routing protocols, such as Open Shortest Path First (OSPF), Intermediate System-to-Intermediate System (IS-IS), and NetWare Link Services Protocol (NLSP), were designed to address the limitations of distance vector routing protocols (slow convergence and unnecessary bandwidth usage). Link-state protocols are more complex than distance vector protocols, and running them adds to the router's overhead. The additional overhead (in the form of memory utilization and bandwidth consumption when link-state protocols first start up) constrains the number of neighbors that a router can support and the number of neighbors that can be in an area.

When the network is stable, link-state protocols minimize bandwidth usage by sending updates only when a change occurs. A hello mechanism ascertains reachability of neighbors. When a failure occurs in the network, link-state protocols flood link-state advertisements (LSAs) throughout an area. LSAs cause every router within the failed area to recalculate routes. The fact that LSAs need to be flooded throughout the area in failure mode and the fact that all routers recalculate routing tables constrain the number of neighbors that can be in an area.

Enhanced IGRP is an advanced distance vector protocol that has some of the properties of link-state protocols. Enhanced IGRP addresses the limitations of conventional distance vector routing protocols (slow convergence and high bandwidth consumption in a steady state network). When the network is stable, Enhanced IGRP sends updates only when a change in the network occurs. Like link-state protocols, Enhanced IGRP uses a hello mechanism to determine the reachability of neighbors. When a failure occurs in the network, Enhanced IGRP looks for feasible successors by sending messages to its neighbors. The search for feasible successors can be aggressive in terms of the traffic it generates (updates, queries, and replies) to achieve convergence. This behavior constrains the number of neighbors that is possible.

In WANs, consideration of bandwidth is especially critical. For example, Frame Relay, which statistically multiplexes many logical data connections (virtual circuits) over a single physical link, allows the creation of networks that share bandwidth. Public Frame Relay networks use bandwidth sharing at all levels within the network. That is, bandwidth sharing may occur within the Frame Relay network of Corporation X, as well as between the networks of Corporation X and Corporation Y.

Two factors have a substantial effect on the design of public Frame Relay networks:

- Users are charged for each permanent virtual circuit (PVC), which encourages network designers to minimize the number of PVCs.
- Public carrier networks sometimes provide incentives to avoid the use of committed information rate (CIR) circuits. Although service providers try to ensure sufficient bandwidth, packets can be dropped.

Overall, WANs can lose packets because of lack of bandwidth. For Frame Relay networks, this possibility is compounded because Frame Relay does not have a broadcast replication facility, so for every broadcast packet that is sent out a Frame Relay interface, the router must replicate it for each PVC on the interface. This requirement limits the number of PVCs that a router can handle effectively.

In addition to bandwidth, network designers must consider the size of routing tables that need to be propagated. Clearly, the design considerations for an interface with 50 neighbors and 100 routes to propagate are very different from the considerations for an interface with 50 neighbors and 10,000 routes to propagate. Table 3-1 gives a rough estimate of the number of WAN neighbors that a routing protocol can handle effectively.

Table 3-1: Routing Protocols and Number of WAN Neighbors

Routing Protocol	Number of Neighbors per Router
Distance vector	50
Link state	30
Advanced distance vector	30

4.1.6 Security

Controlling access to network resources is a primary concern. Some routing protocols provide techniques that can be used as part of a security strategy. With some routing protocols, you can insert a filter on the routes being advertised so that certain routes are not advertised in some parts of the network.

Some routing protocols can authenticate routers that run the same protocol. Authentication mechanisms are protocol specific and generally weak. In spite of this, it is worthwhile to take advantage of the techniques that exist. Authentication can increase network stability by preventing unauthorized routers or hosts from participating in the routing protocol, whether those devices are attempting to participate accidentally or deliberately.

4.2 Enhanced IGRP Internetwork Design Guidelines

The Enhanced Interior Gateway Routing Protocol (Enhanced IGRP) is a routing protocol developed by Cisco Systems and introduced with Software Release 9.21 and Cisco Internetworking Operating System (Cisco IOS) Software Release 10.0. Enhanced IGRP combines the advantages of distance vector protocols, such as IGRP, with the advantages of link-state protocols, such as Open Shortest Path First (OSPF). Enhanced IGRP uses the Diffusing Update ALgorithm (DUAL) to achieve convergence quickly.

Enhanced IGRP includes support for IP, Novell NetWare, and AppleTalk. The discussion on Enhanced IGRP covers the following topics:

- Enhanced IGRP Network Topology
- Enhanced IGRP Addressing
- Enhanced IGRP Route Summarization
- Enhanced IGRP Route Selection
- Enhanced IGRP Convergence
- Enhanced IGRP Network Scalability
- Enhanced IGRP Security

If you are using candidate default route in IP Enhanced IGRP and have installed multiple releases of Cisco router software within your internetwork that include any versions prior to September 1994, contact your Cisco technical support representative for version compatibility and software upgrade information. Refer to your software release notes for details.

4.2.1 Enhanced IGRP Network Topology

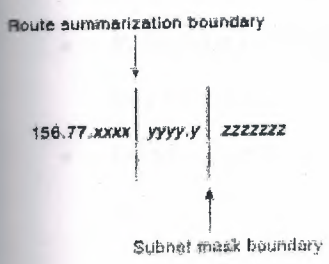
Enhanced IGRP uses a nonhierarchical (or flat) topology by default. Enhanced IGRP automatically summarizes subnet routes of directly connected networks at a network number boundary. This automatic summarization is sufficient for most IP networks. See the section "Enhanced IGRP Route Summarization" later in this chapter for more details.

4.2.2 Enhanced IGRP Addressing

The first step in designing an Enhanced IGRP network is to decide on how to address the network. In many cases, a company is assigned a single NIC address (such as a Class B network address) to be allocated in a corporate internetwork. Bit-wise subnetting and variable-length subnetwork masks (VLSMs) can be used in combination to save address space. Enhanced IGRP for IP supports the use of VLSMs.

Consider a hypothetical network where a Class B address is divided into subnetworks, and contiguous groups of these subnetworks are summarized by Enhanced IGRP. The Class B network 156.77.0.0 might be subdivided as illustrated in Figure 3-5.

Figure 3-5: Variable-length subnet masks (VLSMs) and route summarization boundaries.



In Figure 3-5, the letters x, y, and z represent bits of the last two octets of the Class B network as follows:

- The four *x* bits represent the route summarization boundary.
- The five *y* bits represent up to 32 subnets per summary route.
- The seven *z* bits allow for 126 (128-2) hosts per subnet.

4.2.3 Enhanced IGRP Route Summarization

With Enhanced IGRP, subnet routes of directly connected networks are automatically summarized at network number boundaries. In addition, a network administrator can configure route summarization at any interface with any bit boundary, allowing ranges of networks to be summarized arbitrarily.

4.2.4 Enhanced IGRP Route Selection

Routing protocols compare route metrics to select the best route from a group of possible routes. The following factors are important to understand when designing an Enhanced IGRP internetwork. Enhanced IGRP uses the same vector of metrics as IGRP. Separate metric values are assigned for bandwidth, delay, reliability, and load. By default, Enhanced IGRP computes the metric for a route by using the minimum bandwidth of each hop in the path and adding a media-specific delay for each hop. The metrics used by Enhanced IGRP are as follows:

- *Bandwidth*—Bandwidth is deduced from the interface type. Bandwidth can be modified with the **bandwidth** command.
- *Delay*—Each media type has a propagation delay associated with it. Modifying delay is very useful to optimize routing in network with satellite links. Delay can be modified with the **delay** command.
- *Reliability*—Reliability is dynamically computed as a rolling weighted average over five seconds.
- *Load*—Load is dynamically computed as a rolling weighted average over five seconds.

When Enhanced IGRP summarizes a group of routes, it uses the metric of the best route in the summary as the metric for the summary.

4.2.5 Enhanced IGRP Convergence

Enhanced IGRP implements a new convergence algorithm known as DUAL (Diffusing Update ALgorithm). DUAL uses two techniques that allow Enhanced IGRP to converge very quickly. First, each Enhanced IGRP router stores its neighbors' routing tables. This allows the router to use a new route to a destination instantly if another *feasible* route is known. If no feasible route is known based upon the routing information previously learned from its neighbors, a router running Enhanced IGRP becomes *active* for that destination and sends a query to each of its neighbors, asking for an alternative route to the destination. These queries propagate until an alternative route is found. Routers that are not affected by a topology change remain *passive* and do not need to be involved in the query and response.

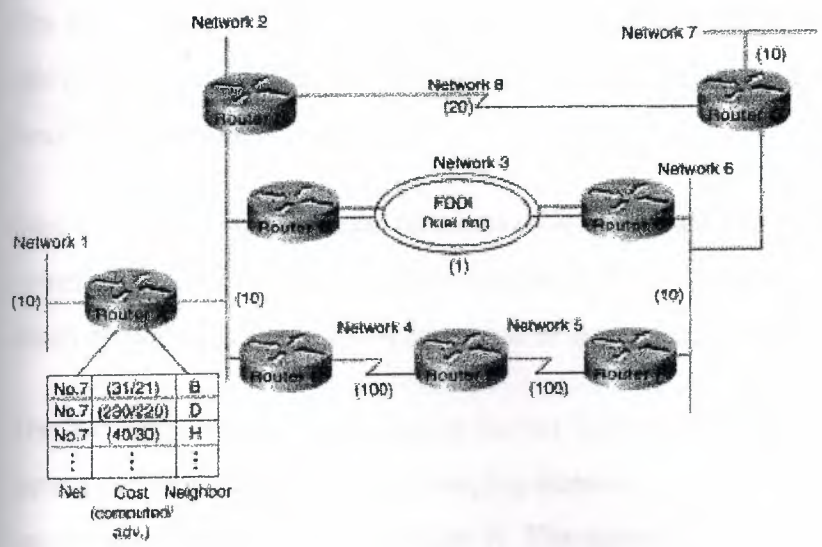
A router using Enhanced IGRP receives full routing tables from its neighbors when it first communicates with the neighbors. Thereafter, only *changes* to the routing tables are sent and only to routers that are affected by the change. A successor is a neighboring router that is currently being used for packet forwarding, provides the least cost route to the destination, and is not part of a routing loop. Information in the routing table is based on feasible successors. Feasible successor routes can be used in case the existing route fails. Feasible successors provide the next least-cost path without introducing routing loops.

The routing table keeps a list of the computed costs of reaching networks. The topology table keeps a list of all routes advertised by neighbors. For each network, the router keeps the real cost of getting to that network and also keeps the advertised cost from its neighbor. In the event of a failure, convergence is instant if a feasible successor can be found. A neighbor is a feasible successor if it meets the feasibility condition set by DUAL. DUAL finds feasible successors by the performing the following computations:

- Determines membership of V_1. V_1 is the set of all neighbors whose advertised distance to network *x* is less than FD. (FD is the feasible distance and is defined as the best metric during an active-to-passive transition.)
- Calculates D_{min}. D_{min} is the minimum computed cost to network *x*.
- Determines membership of V_2. V_2 is the set of neighbors that are in V1 whose computed cost to network *x* equals D_{min}.

The feasibility condition is met when V_2 has one or more members. The concept of feasible successors is illustrated in Figure 3-6. Consider Router A's topology table entries for Network 7. Router B is the *successor* with a computed cost of 31 to reach Network 7, compared to the computed costs of Router D (230) and Router H (40).

Figure 3-6: DUAL feasible successor.



If Router B becomes unavailable, Router A will go through the following three-step process to find a feasible successor for Network 7:

Step 1 Determining which neighbors have an advertised distance to Network 7 that is less than Router A's feasible distance (FD) to Network 7. The FD is 31 and Router H meets this condition. Therefore, Router H is a member of V_1.

Step 2 Calculating the minimum computed cost to Network 7. Router H provides a cost of 40, and Router D provides a cost of 230. D_{min} is, therefore, 40.

Step 3 Determining the set of neighbors that are in V_1 whose computed cost to Network 7 equals D_{min} (40). Router H meets this condition.

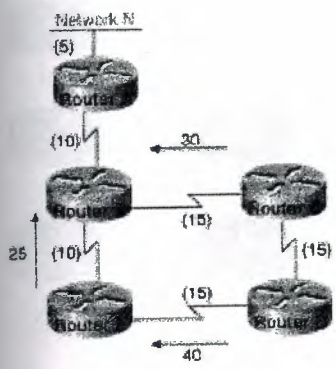
The feasible successor is Router H which provides a least cost route of 40 from Router A to Network 7. If Router H now also becomes unavailable, Router A performs the following computations:

Step 1 Determines which neighbors have an advertised distance to Network 7 that is less than the FD for Network 7. Because both Router B and H have become unavail- able, only Router D remains. However, the advertised cost of Router D to Network 7 is 220, which is greater than Router A's FD (31) to Network 7. Router D, therefore, cannot be a member of V_1. The FD remains at 31—the FD can only change during an active-to-passive transition, and this did not occur. There was no transition to active state for Network 7; this is known as a *local computation.*

Step 2 Because there are no members of V1, there can be no feasible successors. Router A, therefore, transitions from passive to active state for Network 7 and queries its neighbors about Network 7. There was a transition to active; this is known as a *diffusing computation.*

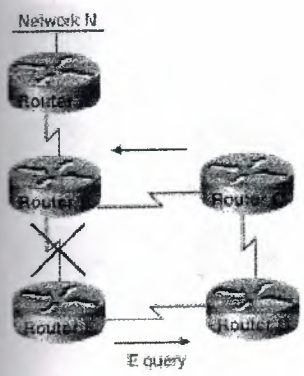
The following example and graphics further illustrate how Enhanced IGRP supports virtually instantaneous convergence in a changing internetwork environment. In Figure 3-7, all routers can access one another and Network N. The computed cost to reach other routers and Network N is shown. For example, the cost from Router E to Router B is 10. The cost from Router E to Network N is 25 (cumulative of 10 + 10 + 5 = 25).

Figure 3-7: DUAL example (part 1): initial network connectivity.



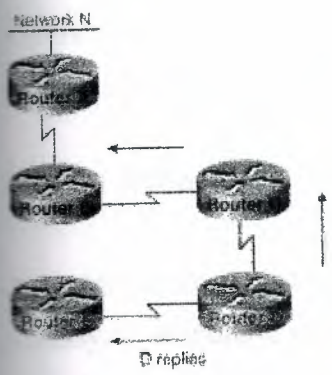
In Figure 3-8, the connection between Router B and Router E fails. Router E sends a multicast query to all of its neighbors and puts Network N into an active state.

Figure 3-8: DUAL example (part 2): sending queries.



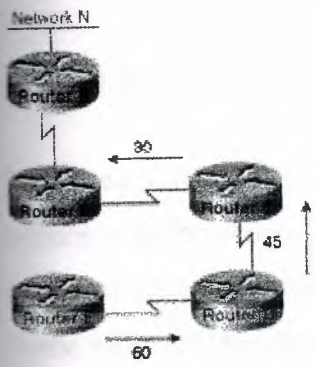
Next, as illustrated in Figure 3-9, Router D determines that it has a feasible successor. It changes its successor from Router E to Router C and sends a reply to Router E.

Figure 3-9: UAL example (part 3): switching to a feasible successor.



In Figure 3-10, Router E has received replies from all neighbors and therefore brings Network N out of active state. Router E puts Network N into its routing table at a distance of 60.

Figure 3-10: Flow of intersubnet traffic with layer 3 switches.



Note Router A, Router B, and Router C were not involved in route recomputation. Router D recomputed its path to Network N without first needing to learn new routing information from its downstream neighbors.

4.2.6 Enhanced IGRP Network Scalability

Network scalability is limited by two factors: operational issues and technical issues. Operationally, Enhanced IGRP provides easy configuration and growth. Technically, Enhanced IGRP uses resources at less than a linear rate with the growth of a network.

4.2.6.1 Memory

A router running Enhanced IGRP stores all routes advertised by neighbors so that it can adapt quickly to alternative routes. The more neighbors a router has, the more memory a router uses. Enhanced IGRP automatic route aggregation bounds the routing table growth naturally. Additional bounding is possible with manual route aggregation.

4.2.6.2 CPU

Enhanced IGRP uses the DUAL algorithm to provide fast convergence. DUAL recomputes only routes which are affected by a topology change. DUAL is not computationally complex, so it does not require a lot of CPU.

4.2.6.3 Bandwidth

Enhanced IGRP uses partial updates. Partial updates are generated only when a change occurs; only the changed information is sent, and this changed information is sent only to the routers affected. Because of this, Enhanced IGRP is very efficient in its usage of bandwidth. Some additional bandwidth is used by Enhanced IGRP's HELLO protocol to maintain adjacencies between neighboring routers.

4.2.7 Enhanced IGRP Security

Enhanced IGRP is available only on Cisco routers. This prevents accidental or malicious routing disruption caused by hosts in a network. In addition, route filters can be set up on any interface to prevent learning or propagating routing information inappropriately.

4.3 OSPF Internetwork Design Guidelines

OSPF is an Interior Gateway Protocol (IGP) developed for use in Internet Protocol (IP)-based internetworks. As an IGP, OSPF distributes routing information between routers belonging to a single autonomous system (AS). An AS is a group of routers exchanging routing information via a common routing protocol. The OSPF protocol is based on shortest-path-first, or link-state, technology.

The OSPF protocol was developed by the OSPF working group of the Internet Engineering Task Force (IETF). It was designed expressly for the Internet Protocol (IP) environment, including explicit support for IP subnetting and the tagging of externally derived routing information. OSPF Version 2 is documented in Request for Comments (RFC) 1247.

Whether you are building an OSPF internetwork from the ground up or converting your internetwork to OSPF, the following design guidelines provide a foundation from which you can construct a reliable, scalable OSPF-based environment.

Two design activities are critically important to a successful OSPF implementation:

- Definition of area boundaries
- Address assignment

Ensuring that these activities are properly planned and executed will make all the difference in your OSPF implementation. Each is addressed in more detail with the discussions that follow. These discussions are divided into nine sections:

- OSPF Network Topology
- OSPF Addressing and Route Summarization
- OSPF Route Selection
- OSPF Convergence
- OSPF Network Scalability
- OSPF Security
- OSPF NSSA (Not-So-Stubby Area) Capabilities
- OSPF On Demand Circuit Protocol Issues
- OSPF over Non-Broadcast Networks

4.3.1 OSPF Network Topology

OSPF works best in a hierarchical routing environment. The first and most important decision when designing an OSPF network is to determine which routers and links are to be included in the backbone and which are to be included in each area. There are several important guidelines to consider when designing an OSPF topology:

- *The number of routers in an area*—OSPF uses a CPU-intensive algorithm. The number of calculations that must be performed given *n* link-state packets is proportional to *n log n*. As a result, the larger and more unstable the area, the greater the likelihood for performance problems associated with routing protocol recalculation. Generally, an area should have no more than 50 routers. Areas with unstable links should be smaller.
- *The number of neighbors for any one router*—OSPF floods all link-state changes to all routers in an area. Routers with many neighbors have the most work to do when link-state changes occur. In general, any one router should have no more than 60 neighbors.
- *The number of areas supported by any one router*—A router must run the link-state algorithm for each link-state change that occurs for every area in which the router resides. Every area border router is in at least two areas (the backbone and one area). In general, to maximize stability, one router should not be in more than three areas.
- *Designated router selection*—In general, the designated router and backup designated router on a local-area network (LAN) have the most OSPF work to do. It is a good idea to select routers that are not already heavily loaded with CPU-intensive activities to be the designated router and backup designated router. In addition, it is generally not a good idea to select the same router to be designated router on many LANs simultaneously.

The discussions that follow address topology issues that are specifically related to the backbone and the areas.

4.3.1.1 Backbone Considerations

Stability and *redundancy* are the most important criteria for the backbone. Stability is increased by keeping the size of the backbone reasonable. This is caused by the fact that every router in the backbone needs to recompute its routes after every link-state change. Keeping the backbone small reduces the likelihood of a change and reduces the amount of CPU cycles required to recompute routes. As a general rule, each area (including the backbone) should contain no more than 50 routers. If link quality is high and the number of routes is small, the number of routers can be increased. Redundancy is important in the backbone to prevent partition when a link fails. Good backbones are designed so that no single link failure can cause a partition.

OSPF backbones must be contiguous. All routers in the backbone should be directly connected to other backbone routers. OSPF includes the concept of virtual links. A virtual link creates a path between two area border routers (an area border router is a router connects an area to the backbone) that are not directly connected. A virtual link can be used to heal a partitioned backbone. However, it is not a good idea to design an OSPF network to require the use of virtual links. The stability of a virtual link is determined by the stability of the underlying area. This dependency can make troubleshooting more difficult. In addition, virtual links cannot run across stub areas. See the section "Backbone-to-Area Route Advertisement" later in this chapter for a detailed discussion of stub areas.

Avoid placing hosts (such as workstations, file servers, or other shared resources) in the backbone area. Keeping hosts out of the backbone area simplifies internetwork expansion and creates a more stable environment.

4.3.1.2 Area Considerations

Individual areas must be contiguous. In this context, a contiguous area is one in which a continuous path can be traced from any router in an area to any other router in the same area. This does not mean that all routers must share common network media. It is not possible to

use virtual links to connect a partitioned area. Ideally, areas should be richly connected internally to prevent partitioning. The two most critical aspects of area design follow:

- Determining how the area is addressed
- Determining how the area is connected to the backbone

Areas should have a contiguous set of network and/or subnet addresses. Without a contiguous address space, it is not possible to implement route summarization. The routers that connect an area to the backbone are called *area border routers.* Areas can have a single area border router or they can have multiple area border routers. In general, it is desirable to have more than one area border router per area to minimize the chance of the area becoming disconnected from the backbone.

When creating large-scale OSPF internetworks, the definition of areas and assignment of resources within areas must be done with a pragmatic view of your internetwork. The following are general rules that help ensure that your internetwork remains flexible and provides the kind of performance needed to deliver reliable resource access:

- *Consider physical proximity when defining areas*—If a particular location is densely connected, create an area specifically for nodes at that location.
- *Reduce the maximum size of areas if links are unstable*—If your internetwork includes unstable links, consider implementing smaller areas to reduce the effects of route flapping. Whenever a route is lost or comes online, each affected area must converge on a new topology. The Dykstra algorithm will run on all the affected routers. By segmenting your internetwork into smaller areas, you can isolate unstable links and deliver more reliable overall service.

4.3.2 OSPF Addressing and Route Summarization

Address assignment and route summarization are inextricably linked when designing OSPF internetworks. To create a scalable OSPF internetwork, you should implement route summarization. To create an environment capable of supporting route summarization, you must implement an effective hierarchical addressing scheme. The addressing structure that you implement can have a profound impact on the performance and scalability of your OSPF

internetwork. The following sections discuss OSPF route summarization and three addressing options:

Separate network numbers for each area

- Network Information Center (NIC)-authorized address areas created using bit-wise subnetting and VLSM
- Private addressing, with a *demilitarized zone* (DMZ) buffer to the official Internet world

---

Note You should keep your addressing scheme as simple as possible, but be wary of oversimplifying your address assignment scheme. Although simplicity in addressing saves time later when operating and troubleshooting your network, taking shortcuts can have certain severe consequences. In building a scalable addressing environment, use a structured approach. If necessary, use bit-wise subnetting— but make sure that route summarization can be accomplished at the area border routers.

4.3.2.1 OSPF Route Summarization

Route summarization is extremely desirable for a reliable and scalable OSPF internetwork. The effectiveness of route summarization, and your OSPF implementation in general, hinges on the addressing scheme that you adopt. Summarization in an OSPF internetwork occurs between each area and the backbone area. Summarization must be configured manually in OSPF. When planning your OSPF internetwork, consider the following issues:

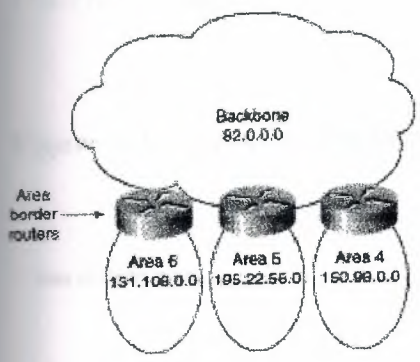
- Be sure that your network addressing scheme is configured so that the range of subnets assigned within an area is contiguous.
- Create an address space that will permit you to split areas easily as your network grows. If possible, assign subnets according to simple octet boundaries. If you cannot assign addresses in an easy-to-remember and easy-to-divide manner, be sure to have a thoroughly defined addressing structure. If you know how your entire address space is assigned (or will be assigned), you can plan for changes more effectively.
- Plan ahead for the addition of new routers to your OSPF environment. Be sure that new routers are inserted appropriately as area, backbone, or border routers. Because

the addition of new routers creates a new topology, inserting new routers can cause unexpected routing changes (and possibly performance changes) when your OSPF topology is recomputed.

4.3.2.2 Separate Address Structures for Each Area

One of the simplest ways to allocate addresses in OSPF is to assign a separate network number for each area. With this scheme, you create a backbone and multiple areas, and assign a separate IP network number to each area. Figure 3-11 illustrates this kind of area allocation.

Figure 3-11: Assignment of NIC addresses example.



The following are the basic steps for creating such a network:

Step 1 Define your structure (identify areas and allocate nodes to areas).

Step 2 Assign addresses to networks, subnets, and end stations.

In the network illustrated in Figure 3-11, each area has its own unique NIC-assigned address. These can be Class A (the backbone in Figure 3-11), Class B (areas 4 and 6), or Class C (Area 5). The following are some clear benefits of assigning separate address structures to each area:

- Address assignment is relatively easy to remember.
- Configuration of routers is relatively easy and mistakes are less likely.
- Network operations are streamlined because each area has a simple, unique network number.

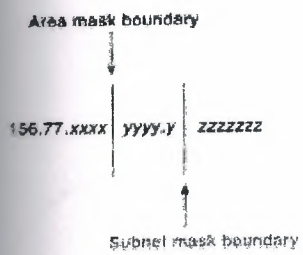
In the example illustrated in Figure 3-11, the route summarization configuration at the area border routers is greatly simplified. Routes from Area 4 injecting into the backbone can be summarized as follows: *All routes starting with 150.98 are found in Area 4.*

The main drawback of this approach to address assignment is that it wastes address space. If you decide to adopt this approach, be sure that area border routers are configured to do route summarization. Summarization must be explicitly set; it is disabled by default in OSPF.

4.3.2.3 Bit-Wise Subnetting and VLSM

Bit-wise subnetting and variable-length subnetwork masks (VLSMs) can be used in combination to save address space. Consider a hypothetical network where a Class B address is subdivided using an area mask and distributed among 16 areas. The Class B network, 156.77.0.0, might be sub- divided as illustrated in Figure 3-12.

Figure 3-12: Areas and subnet masking.



In Figure 3-12, the letters *x*, *y*, and *z* represent bits of the last two octets of the Class B network as follows:

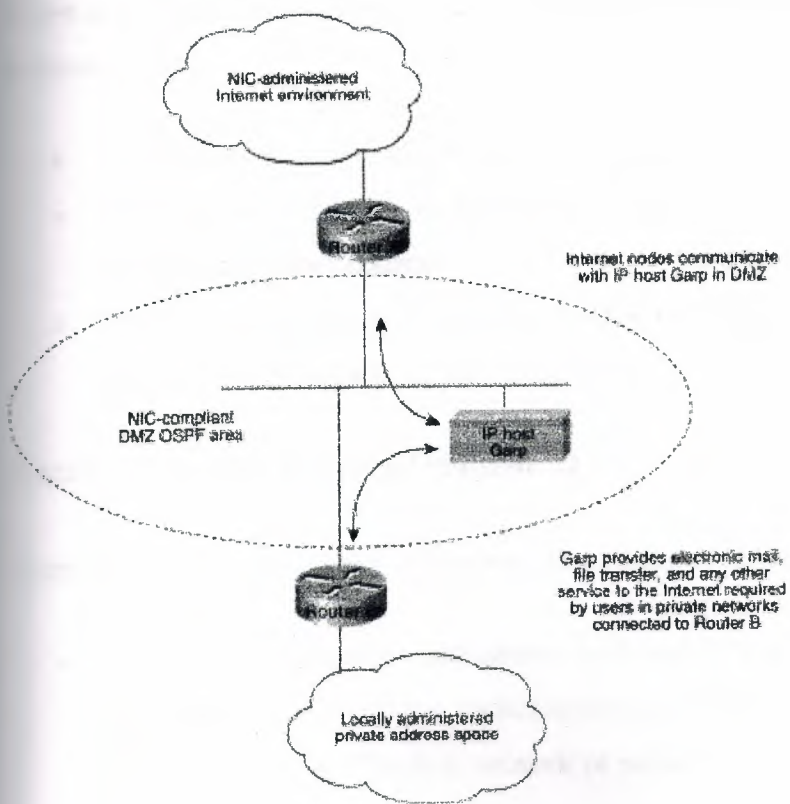
- The four *x* bits are used to identify 16 areas.
- The five *y* bits represent up to 32 subnets per area.
- The seven *z* bits allow for 126 (128-2) hosts per subnet.Private Addressing

Private addressing is another option often cited as simpler than developing an area scheme using bit-wise subnetting. Although private address schemes provide an excellent level of flexibility and do not limit the growth of your OSPF internetwork, they have certain disadvantages. For instance, developing a large-scale internetwork of privately addressed IP

nodes limits total access to the Internet, and mandates the implementation of what is referred to as a *demilitarized zone* (DMZ). If you need to connect to the Internet, Figure 3-13 illustrates the way in which a DMZ provides a buffer of valid NIC nodes between a privately addressed network and the Internet.

All nodes (end systems and routers) on the network in the DMZ must have NIC-assigned IP addresses. The NIC might, for example, assign a single Class C network number to you. The DMZ shown in Figure 3-13 has two routers and a single application gateway host (Garp). Router A provides the interface between the DMZ and the Internet, and Router B provides the firewall between the DMZ and the private address environment. All applications that need to run over the Internet must access the Internet through the application gateway.

Figure 3-13: Connecting to the Internet from a privately addressed network.



4.3.2.4 Route Summarization Techniques

Route summarization is particularly important in an OSPF environment because it increases the stability of the network. If route summarization is being used, routes within an area that change do not need to be changed in the backbone or in other areas. Route summarization addresses two important questions of route information distribution:

- What information does the backbone need to know about each area? The answer to this question focuses attention on area-to-backbone routing information.
- What information does each area need to know about the backbone and other areas? The answer to this question focuses attention on backbone-to-area routing information.

Area-to-Backbone Route Advertisement

There are several key considerations when setting up your OSPF areas for proper summarization:

- OSPF route summarization occurs in the area border routers.
- OSPF supports VLSM, so it is possible to summarize on any bit boundary in a network or subnet address.
- OSPF requires manual summarization. As you design the areas, you need to determine summarization at each area border router.

Backbone-to-Area Route Advertisement

There are four potential types of routing information in an area:

- *Default*—If an explicit route cannot be found for a given IP network or subnetwork, the router will forward the packet to the destination specified in the default route.
- *Intra-area routes*—Explicit network or subnet routes must be carried for all networks or subnets inside an area.
- *Interarea routes*—Areas may carry explicit network or subnet routes for networks or subnets that are in this AS but not in this area.

- *External routes*—When different ASs exchange routing information, the routes they exchange are referred to as external routes.

In general, it is desirable to restrict routing information in any area to the minimal set that the area needs. There are three types of areas, and they are defined in accordance with the routing information that is used in them:

- *Nonstub areas*—Nonstub areas carry a default route, static routes, intra-area routes, interarea routes, and external routes. An area must be a nonstub area when it contains a router that uses both OSPF and any other protocol, such as the Routing Information Protocol (RIP). Such a router is known as an autonomous system border router (ASBR). An area must also be a nonstub area when a virtual link is configured across the area. Nonstub areas are the most resource-intensive type of area.
- *Stub areas*—Stub areas carry a default route, intra-area routes and interarea routes, but they do not carry external routes. Stub areas are recommended for areas that have only one area border router and they are often useful in areas with multiple area border routers. See "Controlling Interarea Traffic" later in this chapter for a detailed discussion of the design trade-offs in areas with multiple area border routers.There are *two restrictions on the use of stub areas: Virtual links cannot be configured across* them and they cannot contain an ASBR.
- *Stub areas without summaries*—Software releases 9.1(11), 9.21(2), and 10.0(1) and later support stub areas without summaries, allowing you to create areas that carry only a default route and intra-area routes. Stub areas without summaries do not carry interarea routes or external routes. This type of area is recommended for simple configurations in which a single router connects an area to the backbone.

Table 3-2 shows the different types of areas according to the routing information that they use.

Routing Information Used in OSPF Areas

Area Type	Default Route	Intra-area Routes	Interarea Routes	External Routes
Nonstub	Yes	Yes	Yes	Yes
Stub	Yes	Yes	Yes	No
Stub without summaries	Yes	Yes	No	No

Stub areas are configured using the **area** *area-id* **stub** router configuration command. Routes are summarized using the **area** *area-id* **range** *address mask* router configuration command. Refer to your *Router Products Configuration Guide* and *Router Products Command Reference* publications for more information regarding the use of these commands.

4.3.3 OSPF Route Selection

When designing an OSPF internetwork for efficient route selection, consider three important topics:

- Tuning OSPF Metrics
- Controlling Interarea Traffic
- Load Balancing in OSPF Internetworks

4.3.3.1 Tuning OSPF Metrics

The default value for OSPF metrics is based on bandwidth. The following characteristics show how OSPF metrics are generated:

- Each link is given a metric value based on its bandwidth. The metric for a specific link is the inverse of the bandwidth for that link. Link metrics are normalized to give FDDI

a metric of 1. The metric for a route is the sum of the metrics for all the links in the route.

Note In some cases, your network might implement a media type that is faster than the fastest default media configurable for OSPF (FDDI). An example of a faster media is ATM. By default, a faster media will be assigned a cost equal to the cost of an FDDI link—a link-state metric cost of 1. Given an environment with both FDDI and a faster media type, you must manually configure link costs to configure the faster link with a lower metric. Configure any FDDI link with a cost greater than 1, and the faster link with a cost less than the assigned FDDI link cost. Use the **ip ospf cost** interface configuration command to modify link-state cost.

- When route summarization is enabled, OSPF uses the metric of the best route in the summary.
- There are two forms of external metrics: type 1 and type 2. Using an external type 1 metric results in routes adding the internal OSPF metric to the external route metric. External type 2 metrics do not add the internal metric to external routes. The external type 1 metric is generally preferred. If you have more than one external connection, either metric can affect how multiple paths are used.

4.3.3.2 Controlling Interarea Traffic

When an area has only a single area border router, all traffic that does not belong in the area will be sent to the area border router. In areas that have multiple area border routers, two choices are available for traffic that needs to leave the area:

- Use the area border router closest to the originator of the traffic. (Traffic leaves the area as soon as possible.)
- Use the area border router closest to the destination of the traffic. (Traffic leaves the area as late as possible.)

If the area border routers inject only the default route, the traffic goes to the area border router that is closest to the source of the traffic. Generally, this behavior is desirable because the

backbone typically has higher bandwidth lines available. However, if you want the traffic to use the area border router that is nearest the destination (so that traffic leaves the area as late as possible), the area border routers should inject summaries into the area instead of just injecting the default route.

Most network designers prefer to avoid asymmetric routing (that is, using a different path for packets that are going from A to B than for those packets that are going from B to A). It is important to understand how routing occurs between areas to avoid asymmetric routing.

4.3.3.3 Load Balancing in OSPF Internetworks

Internetwork topologies are typically designed to provide redundant routes in order to prevent a partitioned network. Redundancy is also useful to provide additional bandwidth for high traffic areas. If equal-cost paths between nodes exist, Cisco routers automatically load balance in an OSPF environment.

Cisco routers can use up to four equal-cost paths for a given destination. Packets might be distributed either on a per-destination (when fast switching) or a per-packet basis. Per-destination load balancing is the default behavior. Per-packet load balancing can be enabled by turning off fast switching using the **no ip route-cache** interface configuration command. For line speeds of 56 Kbps and faster, it is recommended that you enable fast switching.

4.3.4 OSPF Convergence

One of the most attractive features about OSPF is the capability to quickly adapt to topology changes. There are two components to routing convergence:

- *Detection of topology changes*—OSPF uses two mechanisms to detect topology changes. Interface status changes (such as carrier failure on a serial link) is the first mechanism. The second mechanism is failure of OSPF to receive a hello packet from its neighbor within a timing window called a *dead timer*. After this timer expires, the router assumes the neighbor is down. The dead timer is configured using the **ip ospf dead-interval** interface configuration command. The default value of the dead timer is four times the value of the Hello interval. That results in a dead timer default of 40 seconds for broadcast networks and two minutes for nonbroadcast networks.

- *Recalculation of routes*—After a failure has been detected, the router that detected the failure sends a link-state packet with the change information to all routers in the area. All the routers recalculate all of their routes using the Dykstra (or SPF) algorithm. The time required to run the algorithm depends on a combination of the size of the area and the number of routes in the database.

4.3.5 OSPF Network Scalability

Your ability to scale an OSPF internetwork depends on your overall network structure and addressing scheme. As outlined in the preceding discussions concerning network topology and route summarization, adopting a hierarchical addressing environment and a structured address assignment will be the most important factors in determining the scalability of your internetwork. Network scalability is affected by operational and technical considerations:

- Operationally, OSPF networks should be designed so that areas do not need to be split to accommodate growth. Address space should be reserved to permit the addition of new areas.
- Technically, scaling is determined by the utilization of three resources: memory, CPU, and bandwidth, all discussed in the following sections.

4.3.5.1 Memory

An OSPF router stores all of the link states for all of the areas that it is in. In addition, it can store summaries and externals. Careful use of summarization and stub areas can reduce memory use substantially.

4.3.5.2 CPU

An OSPF router uses CPU cycles whenever a link-state change occurs. Keeping areas small and using summarization dramatically reduces CPU use and creates a more stable environment for OSPF.

4.3.5.3 Bandwidth

OSPF sends partial updates when a link-state change occurs. The updates are flooded to all routers in the area. In a quiet network, OSPF is a quiet protocol. In a network with substantial topology changes, OSPF minimizes the amount of bandwidth used.

4.3.6 OSPF Security

Two kinds of security are applicable to routing protocols:

- *Controlling the routers that participate in an OSPF network*

  OSPF contains an optional authentication field. All routers within an area must agree on the value of the authentication field. Because OSPF is a standard protocol available on many platforms, including some hosts, using the authentication field prevents the inadvertent startup of OSPF in an uncontrolled platform on your network and reduces the potential for instability.

- *Controlling the routing information that routers exchange*

  All routers must have the same data within an OSPF area. As a result, it is not possible to use route filters in an OSPF network to provide security.

4.3.7 OSPF NSSA (Not-So-Stubby Area) Overview

Prior to NSSA, to disable an area from receiving external (Type 5) link-state advertisements (LSAs), the area needed to be defined as a stub area. Area Border Routers (ABRs) that connect stub areas do not flood any external routes they receive into the stub areas. To return packets to destinations outside of the stub area, a default route through the ABR is used.

RFC 1587 defines a hybrid area called the Not-So-Stubby Area (NSSA). An OSPF NSSA is similar to an OSPF stub area but allows for the following capabilities:

- Importing (redistribution) of external routes as Type 7 LSAs into NSSAs by NSSA Autonomous System Boundary Routers (ASBRs).
- Translation of specific Type 7 LSAs routes into Type 5 LSAs by NSSA ABRs.

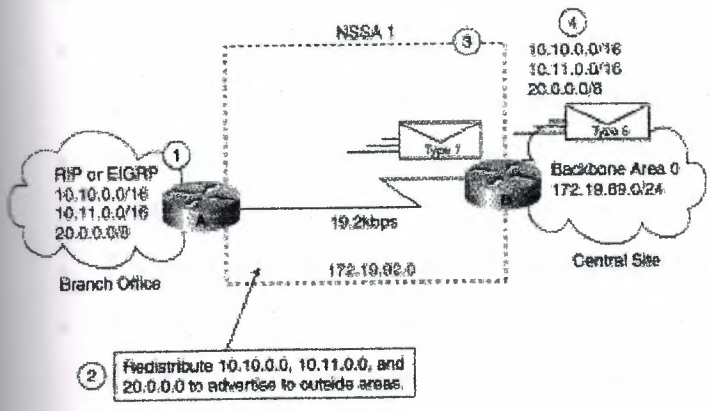
4.3.7.1 Using OSPF NSSA

Use OSPF NSSA in the following scenarios:

- When you want to summarize or filter Type 5 LSAs before they are forwarded into an OSPF area. The OSPF Specification (RFC 1583) prohibits the summarizing or filtering of Type 5 LSAs. It is an OSPF requirement that Type 5 LSAs always be flooding throughout a routing domain. When you define an NSSA, you can import specific external routes as Type 7 LSAs into the NSSA. In addition, when translating Type 7 LSAs to be imported into nonstub areas, you can summarize or filter the LSAs before importing them as Type 5 LSAs.
- If you are an Internet service provider (ISP) or a network administrator that has to connect a central site using OSPF to a remote site that is using a different protocol, such as RIP or EIGRP, you can use NSSA to simplify the administration of this kind of topology. Prior to NSSA, the connection between the corporate site ABR and the remote router used RIP or EIGRP. This meant maintaining two routing protocols. Now, with NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA, as shown in Figure 3-14. You cannot expand the normal OSPF area to the remote site because the Type 5 external will overwhelm both the slow link and the remote router.

In Figure 3-14, the central site and branch office are interconnected through a slow WAN link. The branch office is not using OSPF, but the central site is. Rather than define an RIP domain to connect the sites, you can define an NSSA.

Figure 3-14: OSPF NSSA operation.



In this scenario, Router A is defined as an ASBR (autonomous system border router). It is configured to redistribute any routes within the RIP/EIGRP domain to the NSSA. The following lists what happens when the area between the connecting routers is defined as an NSSA:

1. Router A receives RIP or EGRP routes for networks 10.10.0.0/16, 10.11.0.0/16, and 20.0.0.0/8.

2. Because Router A is also connected to an NSSA, it redistributes the RIP or EIGRP routers as Type 7 LSAs into the NSSA.

3. Router B, an ABR between the NSSA and the backbone Area 0, receives the Type 7 LSAs.

4. After the SPF calculation on the forwarding database, Router B translates the Type 7 LSAs into Type 5 LSAs and then floods them throughout Backbone Area 0. It is at this point that router B could have summarized routes 10.10.0.0/16 and 10.11.0.0/16 as 10.0.0.0/8, or could have filtered one or more of the routes.

4.3.7.2 Type 7 LSA Characteristics

Type 7 LSAs have the following characteristics:

- They are originated only by ASBRs that are connected between the NSSA and autonomous system domain.

- They include a forwarding address field. This field is retained when a Type 7 LSA is translated as a Type 5 LSA.
- They are advertised only within an NSSA.
- They are not flooded beyond an NSSA. The ABR that connects to another nonstub area reconverts the Type 7 LSA into a Type 5 LSA before flooding it.
- NSSA ABRs can be configured to summarize or filter Type 7 LSAs into Type 5 LSAs.
- NSSA ABRs can advertise a Type 7 default route into the NSSA.
- Type 7 LSAs have a lower priority than Type 5 LSAs, so when a route is learned with a Type 5 LSA and Type 7 LSA, the route defined in the Type 5 LSA will be selected first.

4.3.7.3 Configuring OSPF NSSA

The steps used to configure OSPF NSSA are as follows:

Step 1 Configure standard OSPF operation on one or more interfaces that will be attached to NSSAs.

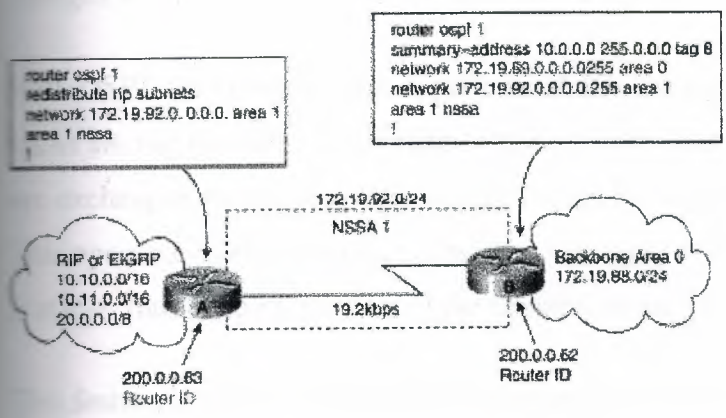
Step 2 Configure an area as NSSA using the following commands:

```
router(config)#area area-id nssa
```

Step 3 (Optional) Control the summarization or filtering during the translation. Figure 3-15 shows how Router will summarize routes using the following command:

```
router(config)#summary-address prefix mask [not-advertise] [tag tag]
```

Figure 3-15: Configuring OSPF NSSA.



4.3.7.4 NSSA Implementation Considerations

Be sure to evaluate these considerations before implementing NSSA. As shown in Figure 3-15, you can set a Type 7 default route that can be used to reach external destinations. The command to issue a Type 7 default route is as follows:

```
router(config)#area area-id nssa [default-information-originate]
```

When configured, the router generates a Type 7 default into the NSSA by the NSSA ABR. Every router within the same area must agree that the area is NSSA; otherwise, the routers will not be able to communicate with one another.

If possible, avoid doing explicit redistribution on NSSA ABR because you could get confused about which packets are being translated by which router.

4.3.8 OSPF On Demand Circuit

OSPF On Demand Circuit is an enhancement to the OSPF protocol that allows efficient operation over on-demand circuits such as ISDN, X.25 SVCs, and dial-up lines. This feature supports RFC 1793, OSPF Over On Demand Circuits. This RFC is useful in understanding the operation of this feature. It has good examples and explains the operation of OSPF in this type of environment.

Prior to this feature, OSPF periodic Hello and link-state advertisement (LSA) updates would be exchanged between routers that connected the on-demand link even when there were no changes in the Hello or LSA information.

With OSPF On Demand Circuit, periodic Hellos are suppressed and periodic refreshes of LSAs are not flooded over demand circuits. These packets bring up the links only when they are exchanged for the first time, or when there is a change in the information they contain. This operation allows the underlying data link layer to be closed when the network topology is stable, thus keeping the cost of the demand circuit to a minimum.

This feature is a standards-based mechanism that is similar to the Cisco Snapshot feature used for distance vector protocols such as RIP.

4.3.8.1 Why Use OSPF On Demand Circuit?

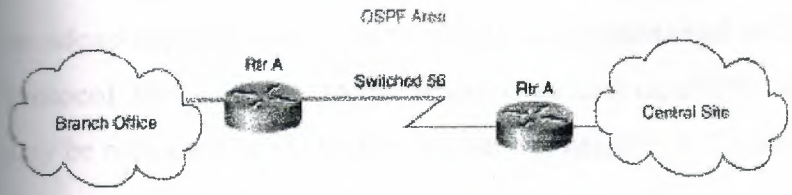
This feature is useful when you want to have an OSPF backbone at the central site and you want to connect telecommuters or branch offices to the central site. In this case, OSPF On Demand Circuit allows the benefits of OSPF over the entire domain without excessive connection costs. Periodic refreshes of Hello updates and LSA updates and other protocol overhead are prevented from enabling the on-demand circuit when there is no "real" data to transmit.

Overhead protocols such as Hellos and LSAs are transferred over the on-demand circuit only upon initial setup and when they reflect a change in the topology. This means that topology-critical changes that require new shortest path first (SPF) calculations are transmitted in order to maintain network topology integrity, but periodic refreshes that do not include changes are not transmitted across the link.

4.3.8.2 OSPF On Demand Circuit Operation

Figure 3-16 illustrates general OSPF operation over on-demand circuits.

Figure 3-16: OSPF area.



The following steps describe the procedure shown in Figure 3-16:

1. Upon initialization, Router A brings up the on demand circuit to exchange Hellos and synchronize LSA databases with Router B. Because both routers are configured for OSPF On Demand Circuit, each router's Hello packets and database description packets have the demand circuit (DC) bit set. As a result, both routers know to suppress periodic Hello packet updates. When each router floods LSAs over the network, the LSAs will have the DoNotAge (DNA) bit set. This means that the LSAs will not age. They can be updated if a new LSA is received with changed information, but no periodic LSA refreshes will be issued over the demand circuit.

2. When Router A receives refreshed LSAs for existing entries in its database, it will determine whether the LSAs include changed information. If not, Router A will update the existing LSA entries, but it will not flood the information to Router B. Therefore, both routers will have the same entries, but the entry sequence numbers may not be identical.

3. When Router A does receive an LSA for a new route or an LSA that includes changed information, it will update its LSA database, bring up the on-demand circuit, and flood the information to Router B. At this point, both routers will have identical sequence numbers for this LSA entry.

4. If there is no data to transfer while the link is up for the updates, the link is terminated.

5. When a host on either side needs to transfer data to another host at the remote site, the link will be brought up.

4.3.9 OSPF Over Non-Broadcast Networks

NBMA networks are those networks that support many (more than two) routers, but have no broadcast capability. Neighboring routers are maintained on these nets using OSPF's Hello Protocol. However, due to the lack of broadcast capability, some configuration information may be necessary to aid in the discovery of neighbors. On non-broadcast networks, OSPF protocol packets that are normally multicast need to be sent to each neighboring router, in turn. An X.25 Public Data Network (PDN) is an example of a non-broadcast network. Note the following:

- *OSPF runs in one of two modes over non-broadcast networks.* The first mode, called non-broadcast multiaccess or NBMA, simulates the operation of OSPF on a broadcast network. The second mode, called point-to-multipoint, treats the non-broadcast network as a collection of point-to-point links. Non-broadcast networks are referred to as NBMA networks or point-to-multipoint networks, depending on OSPF's mode of operation over the network.
- *In NBMA mode, OSPF emulates operation over a broadcast network.* A Designated Router is elected for the NBMA network, and the Designated Router originates an LSA for the network. The graph representation for broadcast networks and NBMA networks is identical.

4.3.9.1 NBMA Mode

NBMA mode is the most efficient way to run OSPF over non-broadcast networks, both in terms of link-state database size and in terms of the amount of routing protocol traffic. However, it has one significant restriction: It requires all routers attached to the NBMA network to be able to communicate directly. This restriction may be met on some non-broadcast networks, such as an ATM subnet utilizing SVCs. But it is often not met on other non-broadcast networks, such as PVC-only Frame Relay networks.

On non-broadcast networks in which not all routers can communicate directly, you can break the non-broadcast network into logical subnets, with the routers on each subnet being able to communicate directly. Then each separate subnet can be run as an NBMA network or a point-to-point network if each virtual circuit is defined as a separate logical subnet. This setup,

however, requires quite a bit of administrative overhead, and is prone to misconfiguration. It is probably better to run such a non-broadcast network in Point-to-MultiPoint mode.

4.3.9.2 Point-to-MultiPoint Mode

Point-to-MultiPoint networks have been designed to work simply and naturally when faced with partial mesh connectivity. In Point-to-MultiPoint mode, OSPF treats all router-to-router connections over the non-broadcast network as if they were point-to-point links. No Designated Router is elected for the network, nor is there an LSA generated for the network. It may be necessary to configure the set of neighbors that are directly reachable over the Point-to-MultiPoint network. Each neighbor is identified by its IP address on the Point-to-MultiPoint network. Because no Designated Routers are elected on Point-to-MultiPoint networks, the Designated Router eligibility of configured neighbors is undefined.

Alternatively, neighbors on Point-to-MultiPoint networks may be dynamically discovered by lower-level protocols such as Inverse ARP. In contrast to NBMA networks, Point-to-MultiPoint networks have the following properties:

1. Adjacencies are established between all neighboring routers. There is no Designated Router or Backup Designated Router for a Point-to-MultiPoint network. No network-LSA is originated for Point-to-MultiPoint networks. Router Priority is not configured for Point-to-MultiPoint interfaces, nor for neighbors on Point-to-MultiPoint networks.

2. When originating a router-LSA, Point-to-MultiPoint interface is reported as a collection of "point-to-point links" to all of the interface's adjacent neighbors, together with a single stub link advertising the interface's IP address with a cost of 0.

3. When flooding out a non-broadcast interface (when either in NBMA or Point-to-MultiPoint mode) the Link State Update or Link State Acknowledgment packet must be replicated in order to be sent to each of the interface's neighbors.

4.4 BGP Internetwork Design Guidelines

The Border Gateway Protocol (BGP) is an interautonomous system routing protocol. The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems. This network reachability information includes information on the list of Autonomous Systems (ASs) that reachability information traverses. BGP-4 provides a new set of mechanisms for supporting classless interdomain routing. These mechanisms include support for advertising an IP prefix and eliminate the concept of network *class* within BGP. BGP-4 also introduces mechanisms that allow aggregation of routes, including aggregation of AS paths. These changes provide support for the proposed supernetting scheme. This section describes how BGP works and it can be used to participate in routing with other networks that run BGP. The following topics are covered:

- BGP operation
- BGP attributes
- BGP path selection criteria
- Understanding and defining BGP routing policies

4.4.1 BGP Operation

This section presents fundamental information about BGP, including the following topics:

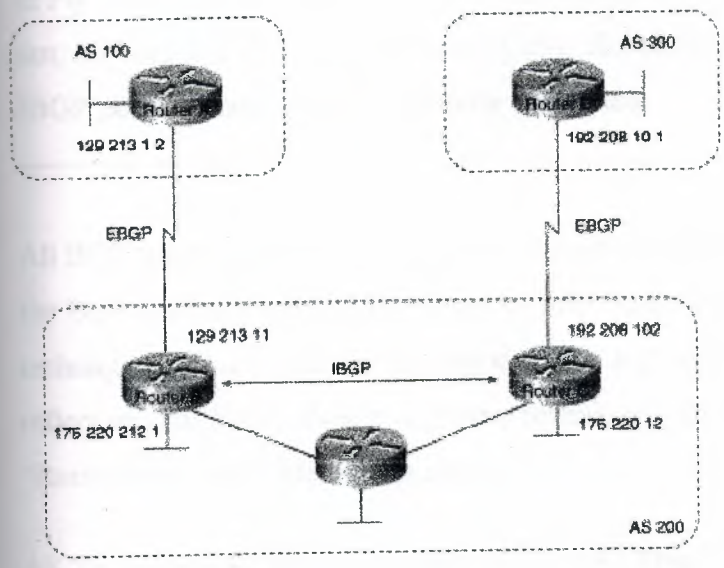
- Internal BGP
- External BGP
- BGP and Route Maps
- Advertising Networks

Routers that belong to the same AS and exchange BGP updates are said to be running internal BGP (IBGP). Routers that belong to different ASs and exchange BGP updates are said to be running external BGP (EBGP).

With the exception of the neighbor **ebgp-multihop** router configuration command (described in the section "External BGP (EBGP)" later in this chapter), the commands for configuring EBGP and IBGP are the same. This chapter uses the terms EBGP and IBGP as a reminder that, for any particular context, routing updates are being exchanged between ASs (EBGP) or

within an AS (IBGP). Figure 3-17 shows a network that demonstrates the difference between EBGP and IBGP.

Figure 3-17: EBGP, IBGP, and multiple ASs.



Before it exchanges information with an external AS, BGP ensures that networks within the AS are reachable. This is done by a combination of internal BGP peering among routers within the AS and by redistributing BGP routing information to Interior Gateway Protocols (IGPs) that run within the AS, such as Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), Routing Information Protocol (RIP), and Open Shortest Path First (OSPF).

BGP uses the Transmission Control Protocol (TCP) as its transport protocol (specifically, port 179). Any two routers that have opened a TCP connection to each other for the purpose of exchanging routing information are known as peers or neighbors. In Figure 3-17, Routers A and B are BGP peers, as are Routers B and C, and Routers C and D. The routing information consists of a series of AS numbers that describe the full path to the destination network. BGP uses this information to construct a loop-free map of ASs. Note that within an AS, BGP peers do not have to be directly connected.

BGP peers initially exchange their full BGP routing tables. Thereafter, BGP peers send incremental updates only. BGP peers also exchange keepalive messages (to ensure that the connection is up) and notification messages (in response to errors or special conditions).

Note Routers A and B are running EBGP, and Routers B and C are running IBGP, as shown in Figure 3-17. Note that the EBGP peers are directly connected and that the IBGP peers are not. As long as there is an IGP running that allows the two neighbors to reach each other, IBGP peers do not have to be directly connected.

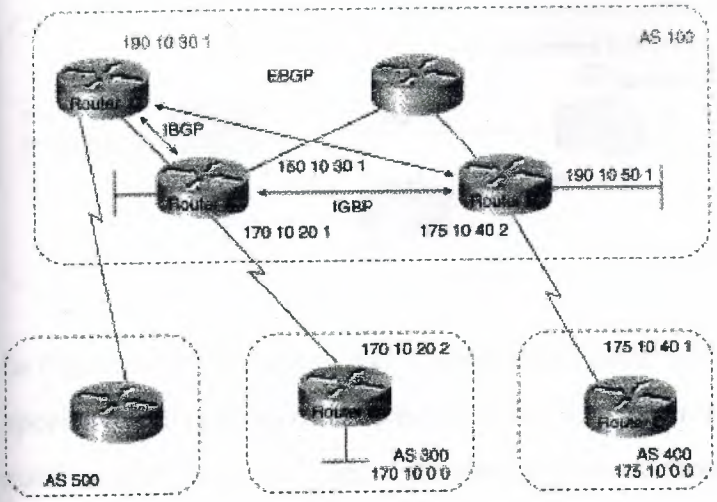
All BGP speakers within an AS must establish a peer relationship with one another. That is, the BGP speakers within an AS must be fully meshed logically. BGP-4 provides two techniques that alleviate the requirement for a logical full mesh: confederations and route reflectors. For information about these techniques, see the sections "Confederations" and "Route Reflectors" later in this chapter.

AS 200 is a transit AS for AS 100 and AS 300. That is, AS 200 is used to transfer packets between AS 100 and AS 300.

4.4.1.1 Internal BGP

Internal BGP (IBGP) is the form of BGP that exchanges BGP updates within an AS. Instead of IBGP, the routes learned via EBGP could be redistributed into IGP within the AS and then redistributed again into another AS. However, IBGP is more flexible, more scalable, and provides more efficient ways of controlling the exchange of information within the AS. It also presents a consistent view of the AS to external neighbors. For example, IBGP provides ways to control the exit point from an AS. Figure 3-18 shows a topology that demonstrates IBGP.

Figure 3-18: Internal BGP example.

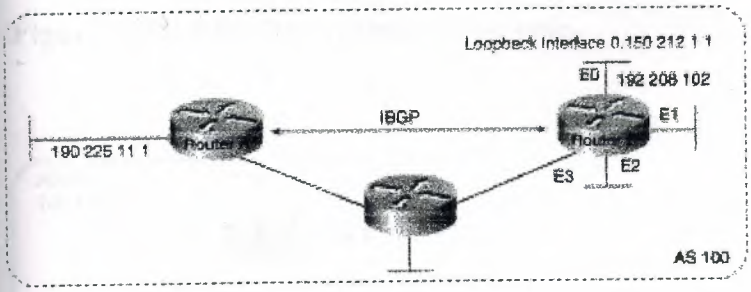


When a BGP speaker receives an update from other BGP speakers in its own AS (that is, via IBGP), the receiving BGP speaker uses EBGP to forward the update to external BGP speakers only. This behavior of IBGP is why it is necessary for BGP speakers within an AS to be fully meshed.

For example, in Figure 3-18, if there were no IBGP session between Routers B and D, Router A would send updates from Router B to Router E but not to Router D. If you want Router D to receive updates from Router B, Router B must be configured so that Router D is a BGP peer.

Loopback Interfaces. Loopback interfaces are often used by IBGP peers. The advantage of using loopback interfaces is that they eliminate a dependency that would otherwise occur when you use the IP address of a physical interface to configure BGP. Figure 3-19 shows a network in which using the loopback interface is advantageous.

Figure 3-19: Use of loopback interfaces.



In Figure 3-19, Routers A and B are running IBGP within AS 100. If Router A were to specify the IP address of Ethernet interface 0, 1, 2, or 3 in the **neighbor remote-as** router configuration command, and if the specified interface were to become unavailable, Router A would not be able to establish a TCP connection with Router B. Instead, Router A specifies the IP address of the loopback interface that Router B defines. When the loopback interface is used, BGP does not have to rely on the availability of a particular interface for making TCP connections.

Note Loopback interfaces are rarely used between EBGP peers because EBGP peers are usually directly connected and, therefore, depend on a particular physical interface for connectivity.

4.4.1.2 External BGP (EBGP)

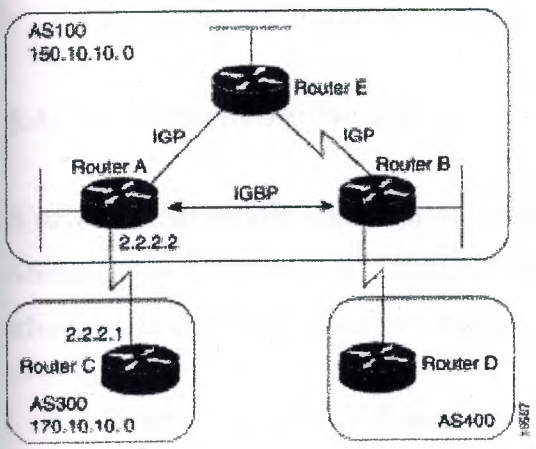
When two BGP speakers that are not in the same AS run BGP to exchange routing information, they are said to be running EBGP.

Synchronization

When an AS provides transit service to other ASs when there are non-BGP routers in the AS, transit traffic might be dropped if the intermediate non-BGP routers have not learned routes for that traffic via an IGP. The BGP synchronization rule states that if an AS provides transit service to another AS, BGP should not advertise a route until all of the routers within the AS

have learned about the route via an IGP. The topology shown in Figure 3-20 demonstrates this synchronization rule.

Figure 3-20: EBGP synchronization rule.



In Figure 3-20, Router C sends updates about network 170.10.0.0 to Router A. Routers A and B are running IBGP, so Router B receives updates about network 170.10.0.0 via IBGP. If Router B wants to reach network 170.10.0.0, it sends traffic to Router E. If Router A does not redistribute network 170.10.0.0 into an IGP, Router E has no way of knowing that network 170.10.0.0 exists and will drop the packets.

If Router B advertises to AS 400 that it can reach 170.10.0.0 before Router E learns about the network via IGP, traffic coming from Router D to Router B with a destination of 170.10.0.0 will flow to Router E and be dropped.

This situation is handled by the synchronization rule of BGP. It states that if an AS (such as AS 100 in Figure 3-20) passes traffic from one AS to another AS, BGP does not advertise a route before all routers within the AS (in this case, AS 100) have learned about the route via an IGP. In this case, Router B waits to hear about network 170.10.0.0 via an IGP before it sends an update to Router D.

Disabling Synchronization

In some cases, you might want to disable synchronization. Disabling synchronization allows BGP to converge more quickly, but it might result in dropped transit packets. You can disable synchronization if one of the following conditions is true:

- Your AS does not pass traffic from one AS to another AS.
- All the transit routers in your AS run BGP.

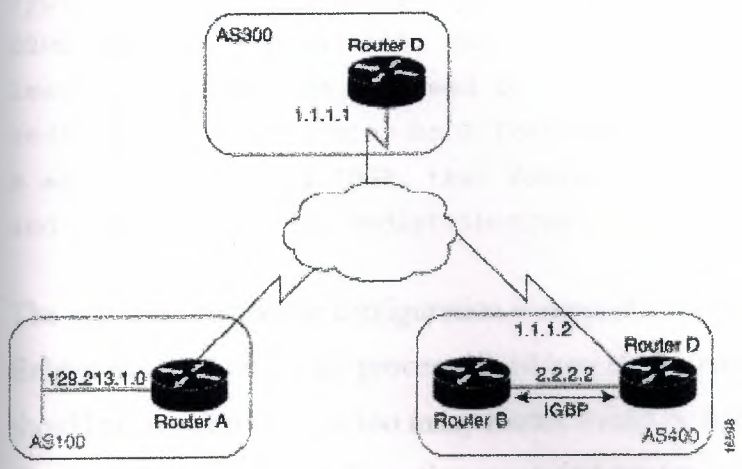
4.4.1.4 Advertising Networks

A network that resides within an AS is said to originate from that network. To inform other ASs about its networks, the AS advertises them. BGP provides three ways for an AS to advertise the networks that it originates:

- Redistributing Static Routes
- Redistributing Dynamic Routes
- Using the **network** Command

This section uses the topology shown in Figure 3-22 to demonstrate how networks that originate from an AS can be advertised.

Figure 3-22: Network advertisement example 1.



4.4.1.5 Redistributing Static Routes

One way to advertise that a network or a subnet originates from an AS is to redistribute static routes into BGP. The only difference between advertising a static route and advertising a dynamic route is that when you redistribute a static route, BGP sets the origin attribute of updates for the route to Incomplete. (For a discussion of other values that can be assigned to the origin attribute, see the section "Origin Attribute" later in this chapter.) To configure Router C in Figure 3-22 to originate network 175.220.0.0 into BGP,

The **redistribute router** configuration command and the **static** keyword cause all static routes to be redistributed into BGP. The **ip route** global configuration command establishes a static route for network 175.220.0.0. In theory, the specification of the null 0 interface would cause a packet destined for network 175.220.0.0 to be discarded. In practice, there will be a more specific match for the packet than 175.220.0.0, and the router will send it out the appropriate interface. Redistributing a static route is the best way to advertise a supernet because it prevents the route from flapping.

Note Regardless of route type (static or dynamic), the **redistribute router** configuration command is the only way to inject BGP routes into an IGP.

Redistributing Dynamic Routes

Another way to advertise networks is to redistribute dynamic routes. Typically, you redistribute IGP routes (such as Enhanced IGRP, IGRP, IS-IS, OSPF, and RIP routes) into BGP. Some of your IGP routes might have been learned from BGP, so you need to use access lists to prevent the redistribution of routes back into BGP. Assume that in Figure 3-22, Routers B and C are running IBGP, that Router C is learning 129.213.1.0 via BGP, and that Router B is redistributing 129.213.1.0 back into Enhanced IGRP.

The **redistribute router** configuration command with the **eigrp** keyword redistributes Enhanced IGRP routes for process ID 10 into BGP. (Normally, distributing BGP into IGP should be avoided because too many routes would be injected into the AS.) The **neighbor distribute-list router** configuration command applies access list 1 to outgoing advertisements to the neighbor whose IP address is 1.1.1.1 (that is, Router D). Access list 1 specifies that

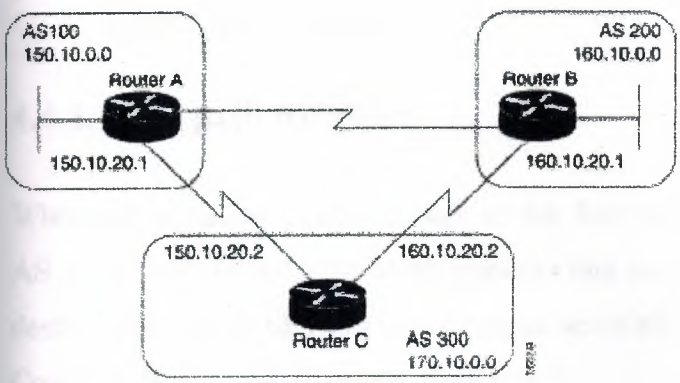
network 175.220.0.0 is to be advertised. All other networks, such as network 129.213.1.0, are implicitly prevented from being advertised. The access list prevents network 129.213.1.0 from being injected back into BGP as if it originated from AS 200, and allows BGP to advertise network 175.220.0.0 as originating from AS 200.

Using the network Command

Another way to advertise networks is to use the **network router** configuration command. When used with BGP, the **network** command specifies the networks that the AS originates. (By way of contrast, when used with an IGP such as RIP, the **network** command identifies the interfaces on which the IGP is to run.) The **network** command works for networks that the router learns dynamically or that are configured as static routes. The origin attribute of routes that are injected into BGP by means of the **network** command is set to IGP. The following commands configure Router C to advertise network 175.220.0.0:

The **network router** configuration command causes Router C to generate an entry in the BGP routing table for network 175.220.0.0. Figure 3-23 shows another topology that demonstrates the effects of the **network** command.

Figure 3-23: Network advertisement example 2.



To ensure a loop-free interdomain topology, BGP does not accept updates that originated from its own AS. For example, in Figure 3-23, if Router A generates an update for network 150.10.0.0 with the origin set to AS 100 and sends it to Router C, Router C will pass the update to Router B with the origin still set to AS 100. Router B will send the update (with the origin still set to AS 100) to Router A, which will recognize that the update originated from its own AS and will ignore it.

4.4.2 BGP Attributes

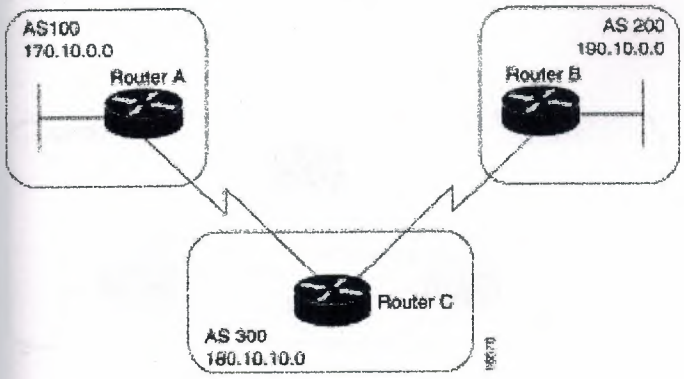
When a BGP speaker receives updates from multiple ASs that describe different paths to the same destination, it must choose the single best path for reaching that destination. Once chosen, BGP propagates the best path to its neighbors. The decision is based on the value of attributes (such as next hop, administrative weights, local preference, the origin of the route, and path length) that the update contains and other BGP-configurable factors. This section describes the following attributes and factors that BGP uses in the decision-making process:

- AS_path Attribute
- Origin Attribute
- Next Hop Attribute
- Weight Attribute
- Local Preference Attribute
- Multi-Exit Discriminator Attribute
- Community Attribute

4.4.2.1 AS_path Attribute

Whenever an update passes through an AS, BGP prepends its AS number to the update. The AS_path attribute is the list of AS numbers that an update has traversed in order to reach a destination. An AS-SET is a mathematical set of all the ASs that have been traversed. Consider the network shown in Figure 3-24.

Figure 3-24: AS_path attribute.



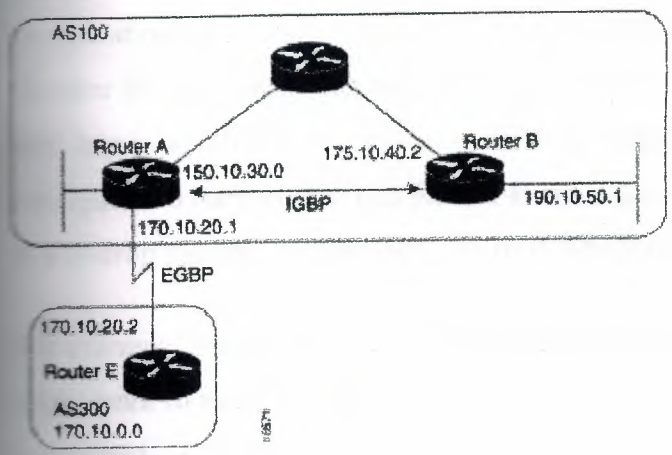
4.4.2.2 Origin Attribute

The origin attribute provides information about the origin of the route. The origin of a route can be one of three values:

- **IGP**—The route is interior to the originating AS. This value is set when the **network router** configuration command is used to inject the route into BGP. The IGP origin type is represented by the letter *i* in the output of the **show ip bgp** EXEC command.
- **EGP**—The route is learned via the Exterior Gateway Protocol (EGP). The EGP origin type is represented by the letter *e* in the output of the **show ip bgp** EXEC command.
- **Incomplete**—The origin of the route is unknown or learned in some other way. An origin of Incomplete occurs when a route is redistributed into BGP. The Incomplete origin type is represented by the *?* symbol in the output of the **show ip bgp** EXEC command.

Figure 3-25 shows a network that demonstrates the value of the origin attribute.

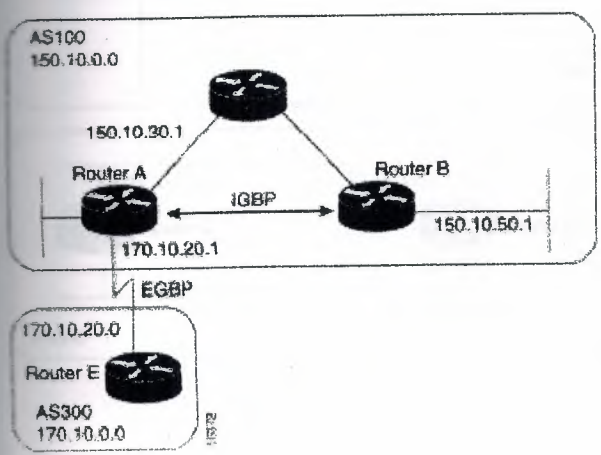
Figure 3-25: Origin attribute.



4.4.2.3 Next Hop Attribute

The BGP next hop attribute is the IP address of the next hop that is going to be used to reach a certain destination. For EBGP, the next hop is usually the IP address of the neighbor specified by the **neighbor remote-as router** configuration command. (The exception is when the next hop is on a multiaccess media, in which case, the next hop could be the IP address of the router in the same subnet.) Consider the network shown in Figure 3-26.

Figure 3-26: Next hop attribute.



In Figure 3-26, Router C advertises network 170.10.0.0 to Router A with a next hop attribute of 170.10.20.2, and Router A advertises network 150.10.0.0 to Router C with a next hop attribute of 170.10.20.1.

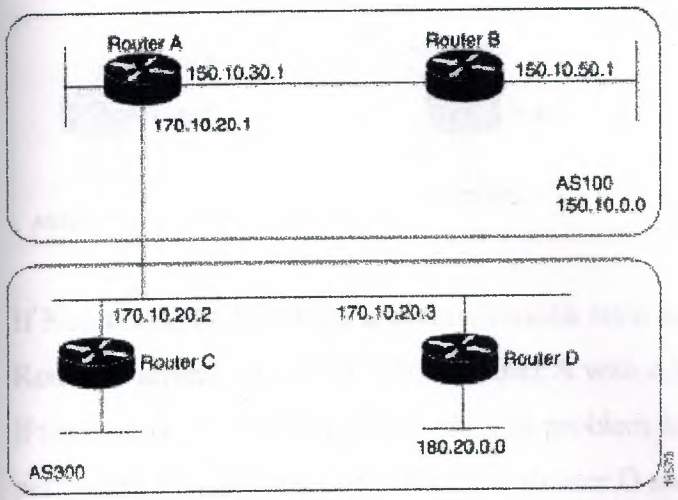
BGP specifies that the next hop of EBGP-learned routes should be carried without modification into IBGP. Because of that rule, Router A advertises 170.10.0.0 to its IBGP peer (Router B) with a next hop attribute of 170.10.20.2. As a result, according to Router B, the next hop to reach 170.10.0.0 is 170.10.20.2, instead of 150.10.30.1. For that reason, the configuration must ensure that Router B can reach 170.10.20.2 via an IGP. Otherwise, Router B will drop packets destined for 170.10.0.0 because the next hop address is inaccessible.

For example, if Router B runs IGRP, Router A should run IGRP on network 170.10.0.0. You might want to make IGRP passive on the link to Router C so that only BGP updates are exchanged.

4.4.2.4 Next Hop Attribute and Multiaccess Media

BGP might set the value of the next hop attribute differently on multiaccess media, such as Ethernet. Consider the network shown in Figure 3-27.

Figure 3-27: Multiaccess media network.

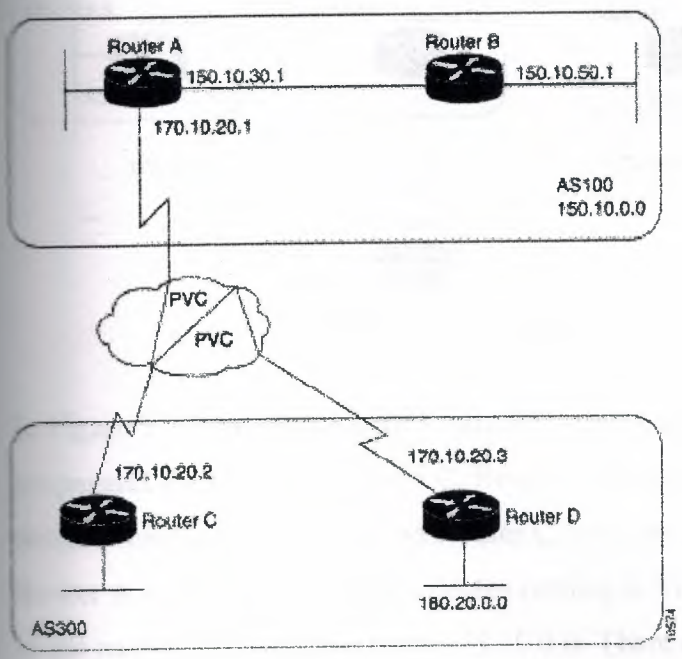


In Figure 3-27, Routers C and D in AS 300 are running OSPF. Router C is running BGP with Router A. Router C can reach network 180.20.0.0 via 170.10.20.3. When Router C sends a BGP update to Router A regarding 180.20.0.0, it sets the next hop attribute to 170.10.20.3, instead of its own IP address (170.10.20.2). This is because Routers A, B, and C are in the same subnet, and it makes more sense for Router A to use Router D as the next hop rather than taking an extra hop via Router C.

4.4.2.5 Next Hop Attribute and Nonbroadcast Media Access

In Figure 3-28, three networks are connected by a nonbroadcast media access (NBMA) cloud, such as Frame Relay.

Figure 3-28: Next Hop attritbute and nonbroadcast media access.

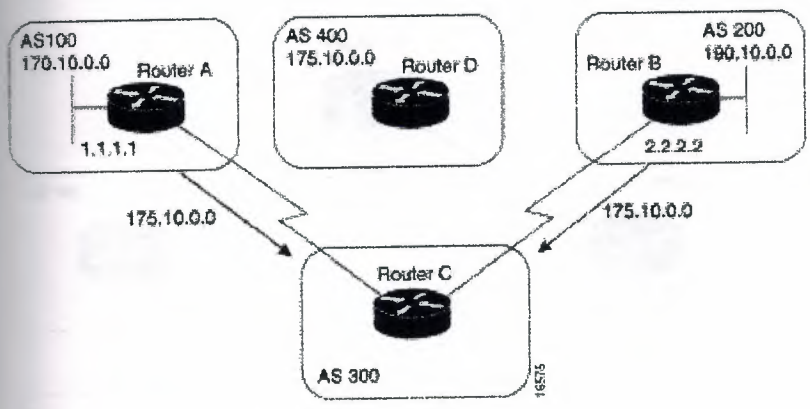


If Routers A, C, and D use a common media such as Frame Relay (or any NBMA cloud), Router C advertises 180.20.0.0 to Router A with a next hop of 170.10.20.3, just as it would do if the common media were Ethernet. The problem is that Router A does not have a direct permanent virtual connection (PVC) to Router D and cannot reach the next hop, so routing will fail.

4.4.2.6 Weight Attribute

The weight attribute is a special Cisco attribute that is used in the path selection process when there is more than one route to the same destination. The weight attribute is local to the router on which it is assigned, and it is not propagated in routing updates. By default, the weight attribute is 32768 for paths that the router originates and zero for other paths. Routes with a higher weight are preferred when there are multiple routes to the same destination. Consider the network shown in Figure 3-29.

Figure 3-29: Weight attribute example.



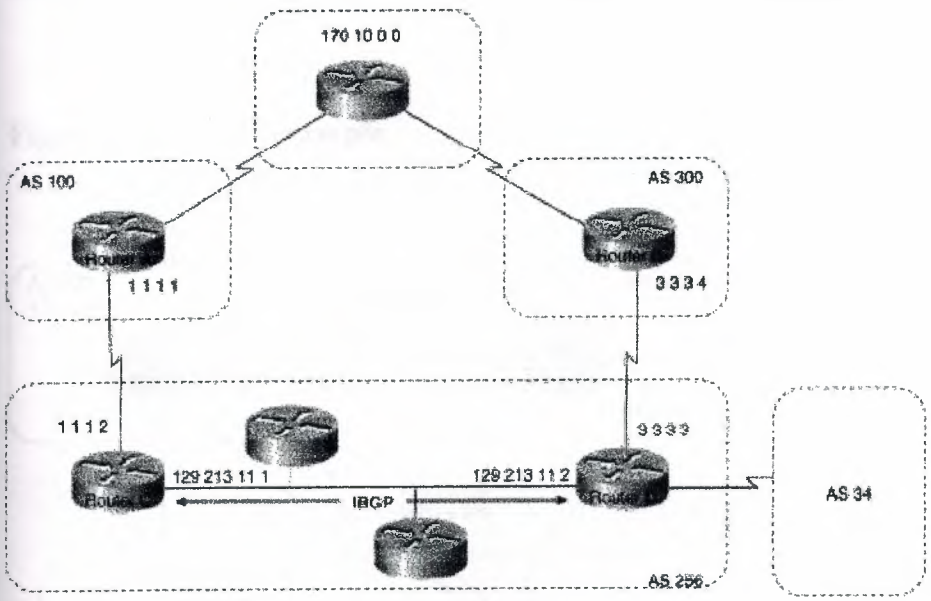
In Figure 3-29, Routers A and B learn about network 175.10.0.0 from AS 400, and each propagates the update to Router C. Router C has two routes for reaching 175.10.0.0 and has to decide which route to use. If, on Router C, you set the weight of the updates coming in from Router A to be higher than the updates coming in from Router B, Router C will use Router A as the next hop to reach network 175.10.0.0. There are three ways to set the weight for updates coming in from Router A:

- Using an Access List to Set the Weight Attribute
- Using a Route Map to Set the Weight Attribute
- Using the **neighbor weight** Command to Set the Weight Attribute

4.4.2.7 Local Preference Attribute

When there are multiple paths to the same destination, the local preference attribute indicates the preferred path. The path with the higher preference is preferred (the default value of the local preference attribute is 100). Unlike the weight attribute, which is relevant only to the local router, the local preference attribute is part of the routing update and is exchanged among routers in the same AS. The network shown in Figure 3-30 demonstrates the local preference attribute.

Figure 3-30: Local preference.



In Figure 3-30, AS 256 receives route updates for network 170.10.0.0 from AS 100 and AS 300. There are two ways to set local preference:

- Using the **bgp default local-preference** Command
- Using a Route Map to Set Local Preference

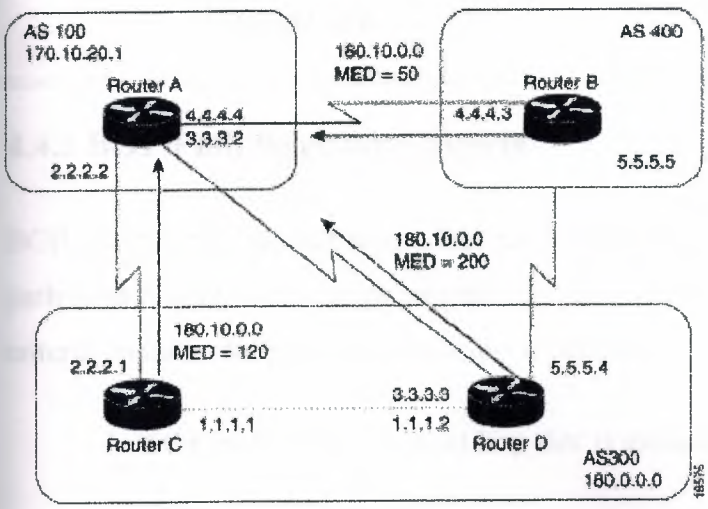
4.4.2.8 Multi-Exit Discriminator Attribute

The multi-exit discriminator (MED) attribute is a hint to external neighbors about the preferred path into an AS when there are multiple entry points into the AS. A lower MED value is preferred over a higher MED value. The default value of the MED attribute is 0.

Unlike local preference, the MED attribute is exchanged between ASs, but a MED attribute that comes into an AS does not leave the AS. When an update enters the AS with a certain MED value, that value is used for decision making within the AS. When BGP sends that update to another AS, the MED is reset to 0.

Unless otherwise specified, the router compares MED attributes for paths from external neighbors that are in the same AS. If you want MED attributes from neighbors in other ASs to be compared, you must configure the **bgp always-compare-med** command. The network shown in Figure 3-31 demonstrates the use of the MED attribute.

Figure 3-31: MED example.



In Figure 3-31, AS 100 receives updates regarding network 180.10.0.0 from Routers B, C, and D. Routers C and D are in AS 300, and Router B is in AS 400.

4.4.2.9 Community Attribute

The community attribute provides a way of grouping destinations (called communities) to which routing decisions (such as acceptance, preference, and redistribution) can be applied. Route maps are used to set the community attribute. A few predefined communities are listed in Table 3-3.

Table 3-2: Predefined Communities

Community	Meaning
no-export	Do not advertise this route to EBGP peers.
no-advertised	Do not advertise this route to any peer.
internet	Advertise this route to the Internet community; all routers in the network belong to it.

4.4.3 BGP Path Selection Criteria

BGP selects only one path as the best path. When the path is selected, BGP puts the selected path in its routing table and propagates the path to its neighbors. BGP uses the following criteria, in the order presented, to select a path for a destination:

1. If the path specifies a next hop that is inaccessible, drop the update.

2. Prefer the path with the largest weight.

3. If the weights are the same, prefer the path with the largest local preference.

4. If the local preferences are the same, prefer the path that was originated by BGP running on this router.

5. If no route was originated, prefer the route that has the shortest AS_path.

6. If all paths have the same AS_path length, prefer the path with the lowest origin type (where IGP is lower than EGP, and EGP is lower than Incomplete).

7. If the origin codes are the same, prefer the path with the lowest MED attribute.

8. If the paths have the same MED, prefer the external path over the internal path.

9. If the paths are still the same, prefer the path through the closest IGP neighbor.

10. Prefer the path with the lowest IP address, as specified by the BGP router ID.

4.4.4 Understanding and Defining BGP Routing Policies

This section describes how to understand and define BGP Policies to control the flow of BGP updates. The techniques include the following:

- Administrative Distance
- BGP Filtering
- BGP Peer Groups
- CIDR and Aggregate Addresses
- Confederations
- Route Reflectors
- Route Flap Dampening

4.4.4.1 Administrative Distance

Normally, a route could be learned via more than one protocol. Administrative distance is used to discriminate between routes learned from more than one protocol. The route with the lowest administrative distance is installed in the IP routing table. By default, BGP uses the administrative distances shown in Table 3-3.

Table 3-3: BGP Administrative Distances

Distance	Default Value	Function
External	20	Applied to routes learned from EBGP
Internal	200	Applied to routes learned from IBGP
Local	200	Applied to routes originated by the router

4.4.4.2 BGP Filtering

You can control the sending and receiving of updates by using the following filtering methods:

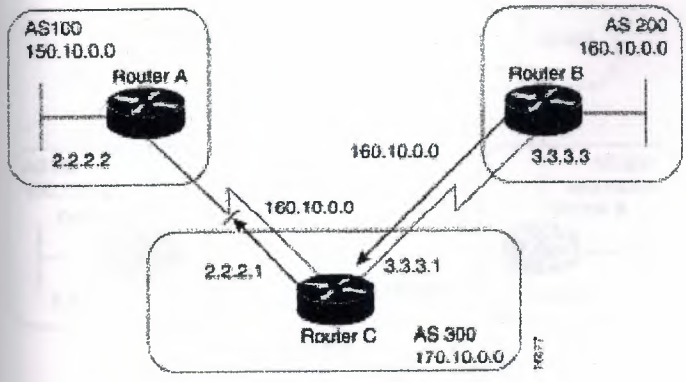
- Prefix Filtering
- AS_path Filtering
- Route Map Filtering
- Community Filtering

Each method can be used to achieve the same result—the choice of method depends on the specific network configuration.

Prefix Filtering

To restrict the routing information that the router learns or advertises, you can filter based on routing updates to or from a particular neighbor. The filter consists of an access list that is applied to updates to or from a neighbor. The network shown in Figure 3-32 demonstrates the usefulness of prefix filtering.

Figure 3-32: Prefix route filtering.



In Figure 3-32, Router B is originating network 160.10.0.0 and sending it to Router C. If you want to prevent Router C from propagating updates for network 160.10.0.0 to AS 100, you can apply an access list to filter those updates when Router C exchanges updates with Router

Figure 3-33: AS_path filtering.

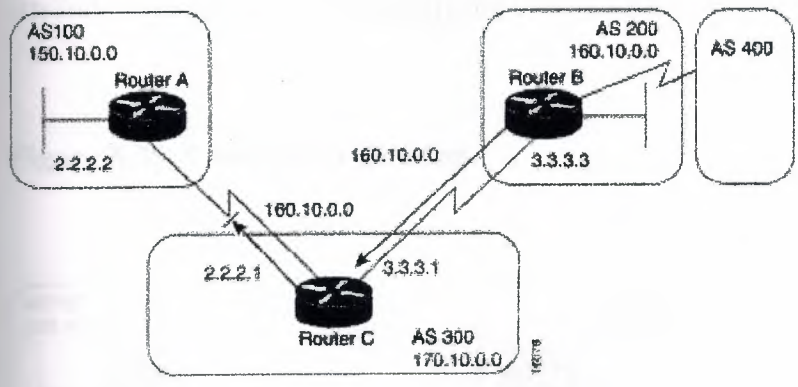
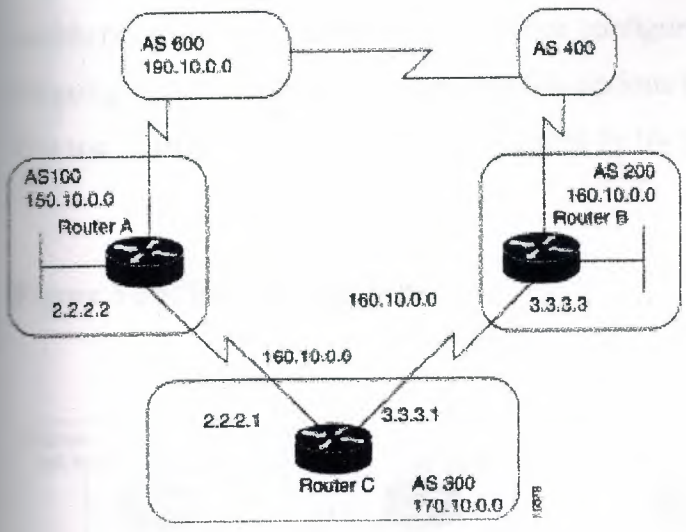
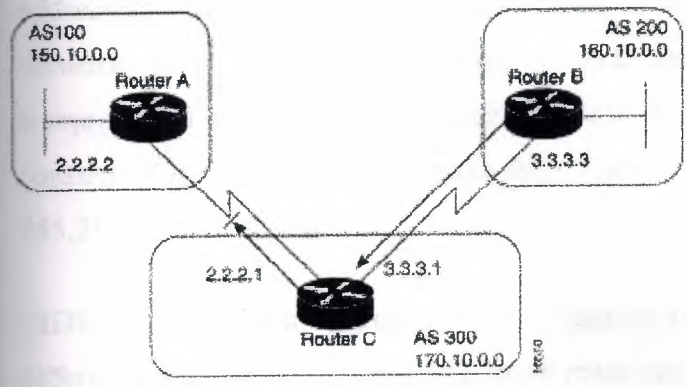


Figure 3-34: BGP route map filtering.



Assume that in Figure 3-34, you want Router C to learn about networks that are local to AS 200 only. (That is, you do not want Router C to learn about AS 100, AS 400, or AS 600 from AS 200.) Also, on those routes that Router C accepts from AS 200, you want the weight attribute to be set to 20. The following configuration for Router C accomplishes this goal:

Figure 3-35: Community filtering.



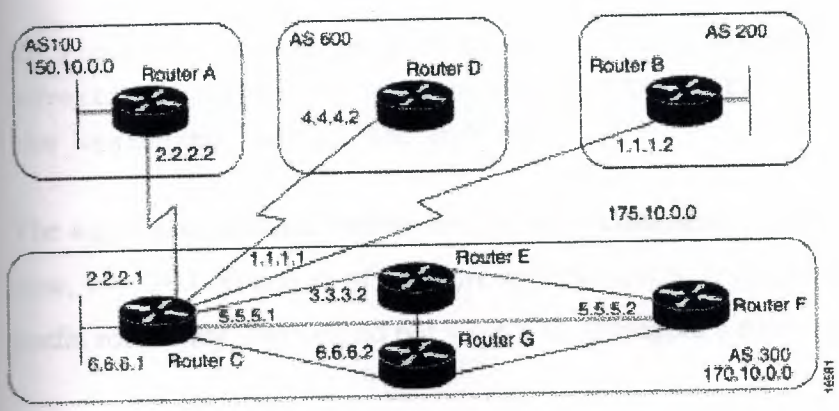
4.4.4.3 BGP Peer Groups

A *BGP peer group* is a group of BGP neighbors that share the same update policies. Update policies are usually set by route maps, distribution lists, and filter lists. Instead of defining the

same policies for each individual neighbor, you define a peer group name and assign policies to the peer group.

Members of a peer group inherit all of the configuration options of the peer group. Peer group members can also be configured to override configuration options if the options do not affect outgoing updates. That is, you can override options that are set only for incoming updates. The use of BGP peer groups is demonstrated by the network shown in Figure 3-36

Figure 3-36: BGP peer groups.

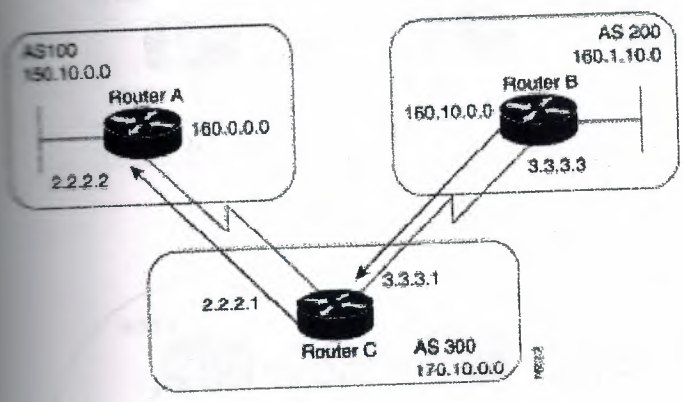


4.4.4.4 CIDR and Aggregate Addresses

BGP4 supports classless interdomain routing (CIDR). CIDR is a new way of looking at IP addresses that eliminates the concept of classes (Class A, Class B, and so on). For example, network 192.213.0.0, which is an illegal Class C network number, is a legal supernet when it is represented in CIDR notation as 192.213.0.0/16. The /16 indicates that the subnet mask consists of 16 bits (counting from the left). Therefore, 192.213.0.0/16 is similar to 192.213.0.0 255.255.0.0.

CIDR makes it easy to aggregate routes. Aggregation is the process of combining several different routes in such a way that a single route can be advertised, which minimizes the size of routing tables. Consider the network shown in Figure 3-37.

Figure 3-37: Aggregation example.



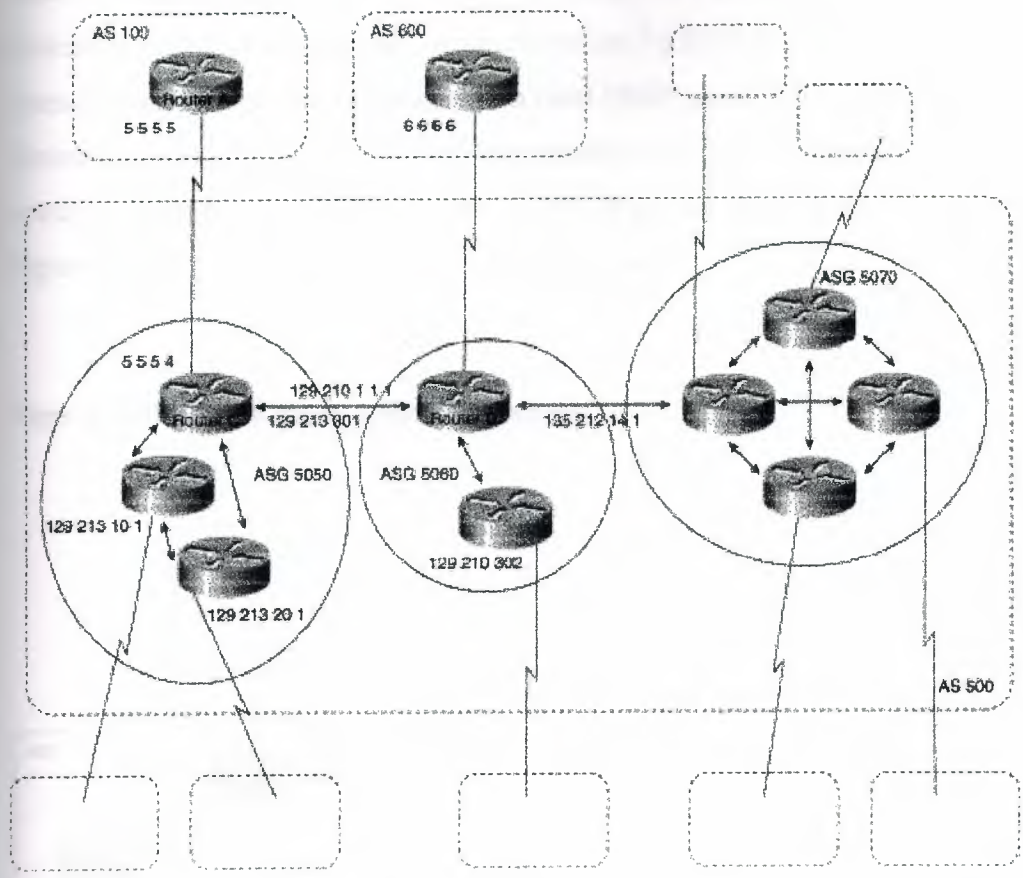
In Figure 3-37, Router B in AS 200 is originating network 160.11.0.0 and advertising it to Router C in AS 300. To configure Router C to propagate the aggregate address 160.0.0.0 to

The **aggregate-address router** configuration command advertises the prefix route (in this case, 160.0.0.0/8) and all of the more specific routes. If you want Router C to propagate the prefix route only, and you do not want it to propagate a more specific route,

4.4.4.5 Confederations

A *confederation* is a technique for reducing the IBGP mesh inside the AS. Consider the network shown in Figure 3-38.

Figure 3-38: Example of confederations.



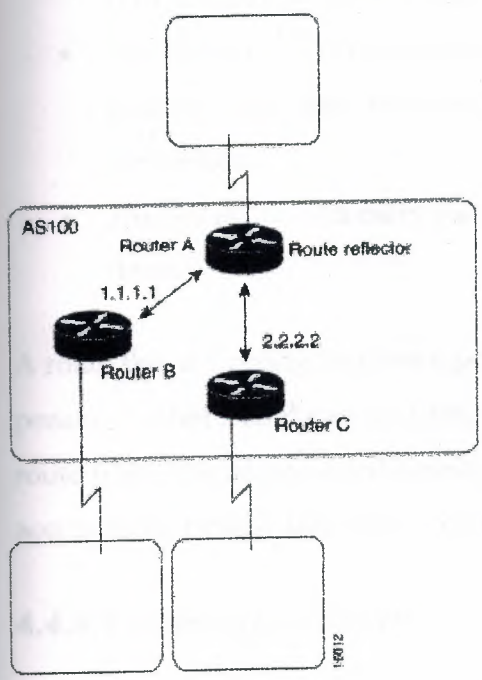
In Figure 3-38, AS 500 consists of nine BGP speakers (although there might be other routers that are not configured for BGP). Without confederations, BGP would require that the routers in AS 500 be fully meshed. That is, each router would need to run IBGP with each of the other eight routers, and each router would need to connect to an external AS and run EBGP, for a total of nine peers for each router.

Confederations reduce the number of peers within the AS, as shown in Figure 3-38. You use confederations to divide the AS into multiple mini-ASs and assign the mini-ASs to a confederation. Each mini-AS is fully meshed, and IBGP is run among its members. Each mini-AS has a connection to the other mini-ASs within the confederation. Even though the mini-ASs have EBGP peers to ASs within the confederation, they exchange routing updates as if they were using IBGP. That is, the next hop, MED, and local preference information is preserved. To the outside world, the confederation looks like a single AS.

4.4.4.6 Route Reflectors

Route reflectors are another solution for the explosion of IBGP peering within an AS. As described earlier in the section "Synchronization," a BGP speaker does not advertise a route learned from another IBGP speaker to a third IBGP speaker. Route reflectors ease this limitation and allow a router to advertise (reflect) IBGP-learned routes to other IBGP speakers, thereby reducing the number of IBGP peers within an AS. The network shown in Figure 3-39 demonstrates how route reflectors work.

Figure 3-39: imple route reflector example.



Without a route reflector, the network shown in Figure 3-39 would require a full IBGP mesh (that is, Router A would have to be a peer of Router B). If Router C is configured as a route reflector, IBGP peering between Routers A and B is not required because Router C will reflect updates from Router A to Router B and from Router B to Router A. To configure

4.4.4.7 Route Flap Dampening

Route flap dampening (introduced in Cisco IOS Release 11.0) is a mechanism for minimizing the instability caused by route flapping. The following terms are used to describe route flap dampening:

- *Penalty*—A numeric value that is assigned to a route when it flaps.
- *Half-life time*—A configurable numeric value that describes the time required to reduce the penalty by one half.
- *Suppress limit*—A numeric value that is compared with the penalty. If the penalty is greater than the suppress limit, the route is suppressed.
- *Suppressed*—A route that is not advertised even though it is up. A route is suppressed if the penalty is more than the suppressed limit.
- *Reuse limit*—A configurable numeric value that is compared with the penalty. If the penalty is less than the reuse limit, a suppressed route that is up will no longer be suppressed.
- *History entry*—An entry that is used to store flap information about a route that is down.

A route that is flapping receives a penalty of 1000 for each flap. When the accumulated penalty reaches a configurable limit, BGP suppresses advertisement of the route even if the route is up. The accumulated penalty is decremented by the half-life time. When the accumulated penalty is less than the reuse limit, the route is advertised again (if it is still up).

4.4.4.8 Summary of BGP

The primary function of a BGP system is to exchange network reachability information with other BGP systems. This information is used to construct a graph of AS connectivity from which routing loops are pruned and with which AS-level policy decisions are enforced. BGP provides a number of techniques for controlling the flow of BGP updates, such as route, path, and community filtering. It also provides techniques for consolidating routing information, such as CIDR aggregation, confederations, and route reflectors. BGP is a powerful tool for providing loop-free interdomain routing within and between ASs.

4.5 Summary

Recall the following design implications of the Enhanced Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF) protocols, and the BGP protocol:

- Network topology
- Addressing and route summarization
- Route selection
- Convergence
- Network scalability
- Security

This chapter outlined these general routing protocol issues and focused on design guidelines for the specific IP protocols.

Chapter 5 Internet Protocol Multicast

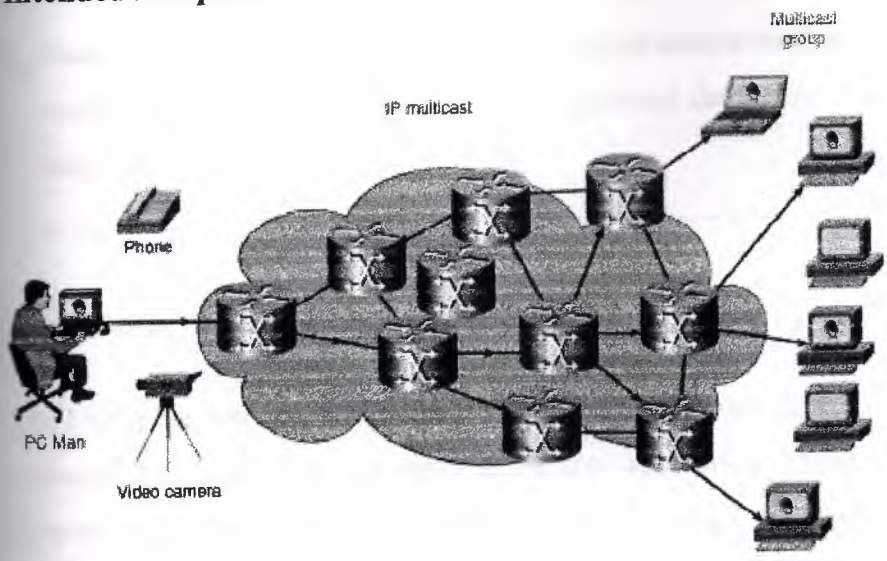
5.1 Background

Internet Protocol (IP) multicast is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to thousands of corporate recipients and homes. Applications that take advantage of multicast include videoconferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news.

IP Multicast delivers source traffic to multiple receivers without adding any additional burden on the source or the receivers while using the least network bandwidth of any competing technology. Multicast packets are replicated in the network by Cisco routers enabled with Protocol Independent Multicast (PIM) and other supporting multicast protocols resulting in the most efficient delivery of data to multiple receivers possible. All alternatives require the source to send more than one copy of the data. Some even require the source to send an individual copy to each receiver. If there are thousands of receivers, even low-bandwidth applications benefit from using Cisco IP Multicast. High-bandwidth applications, such as MPEG video, may require a large portion of the available network bandwidth for a single stream. In these applications, the only way to send to more than one receiver simultaneously is by using IP Multicast. Figure 43-1 demonstrates how data from one source is delivered to several interested recipients using IP multicast.

Figure 43-1: Multicast Transmission Sends a Single Multicast Packet Addressed to All

Intended Recipients



5.2 Multicast Group Concept

Multicast is based on the concept of a group. An arbitrary group of receivers expresses an interest in receiving a particular data stream. This group does not have any physical or geographical boundaries—the hosts can be located anywhere on the Internet. Hosts that are interested in receiving data flowing to a particular group must join the group using IGMP. Hosts must be a member of the group to receive the data stream.

5.3 IP Multicast Addresses

Multicast addresses specify an arbitrary group of IP hosts that have joined the group and want to receive traffic sent to this group.

5.3.1 IP Class D Addresses

The *Internet Assigned Numbers Authority (IANA)* controls the assignment of IP multicast addresses. It has assigned the old Class D address space to be used for IP multicast. This means that all IP multicast group addresses will fall in the range of 224.0.0.0 to 239.255.255.255.

Note This address range is only for the group address or destination address of IP multicast traffic. The source address for multicast datagrams is always the unicast source address.

5.3.2 Reserved Link Local Addresses

The IANA has reserved addresses in the 224.0.0.0 through 224.0.0.255 to be used by network protocols on a local network segment. Packets with these addresses should never be forwarded by a router; they remain local on a particular LAN segment. They are always transmitted with a time-to-live (TTL) of 1.

Network protocols use these addresses for automatic router discovery and to communicate important routing information. For example, OSPF uses 224.0.0.5 and 224.0.0.6 to exchange link state information. Table 43-1 lists some of the well-known addresses.

Table 43-1: Link Local Addresses Address	**Usage**
224.0.0.1	All systems on this subnet
224.0.0.2	All routers on this subnet
224.0.0.5	OSPF routers
224.0.0.6	OSPF designated routers
224.0.0.12	DHCP server/relay agent

5.3.3 Globally Scoped Address

The range of addresses from 224.0.1.0 through 238.255.255.255 are called globally scoped addresses. They can be used to multicast data between organizations and across the Internet.

Some of these addresses have been reserved for use by multicast applications through IANA. For example, 224.0.1.1 has been reserved for Network Time Protocol (NTP).

More information about reserved multicast addresses can be found at http://www.isi.edu/in-notes/iana/assignments/multicast-addresses.

5.3.4 Limited Scope Addresses

The range of addresses from 239.0.0.0 through 239.255.255.255 contains limited scope addresses or administratively scoped addresses. These are defined by RFC 2365 to be constrained to a local group or organization. Routers are typically configured with filters to prevent multicast traffic in this address range from flowing outside an autonomous system (AS) or any user-defined domain. Within an autonomous system or domain, the limited scope address range can be further subdivided so those local multicast boundaries can be defined. This also allows for address reuse among these smaller domains.

5.3.5 Glop Addressing

RFC 2770 proposes that the 233.0.0.0/8 address range be reserved for statically defined addresses by organizations that already have an AS number reserved. The AS number of the domain is embedded into the second and third octets of the 233.0.0.0/8 range.

For example, the AS 62010 is written in hex as F23A. Separating out the two octets F2 and 3A, we get 242 and 58 in decimal. This would give us a subnet of 233.242.58.0 that would be globally reserved for AS 62010 to use.

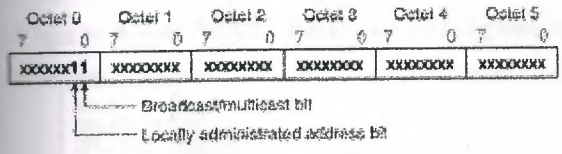
5.3.6 Layer 2 Multicast Addresses

Normally, network interface cards (NICs) on a LAN segment will receive only packets destined for their burned-in MAC address or the broadcast MAC address. Some means had to be devised so that multiple hosts could receive the same packet and still be capable of differentiating among multicast groups.

Fortunately, the IEEE LAN specifications made provisions for the transmission of broadcast and/or multicast packets. In the 802.3 standard, bit 0 of the first octet is used to indicate a

broadcast and/or multicast frame. Figure 43-2 shows the location of the broadcast/multicast bit in an Ethernet frame.

Figure 43-2: IEEE 802.3 MAC Address Format



This bit indicates that the frame is destined for an arbitrary group of hosts or all hosts on the network (in the case of the broadcast address, 0xFFFF.FFFF.FFFF).

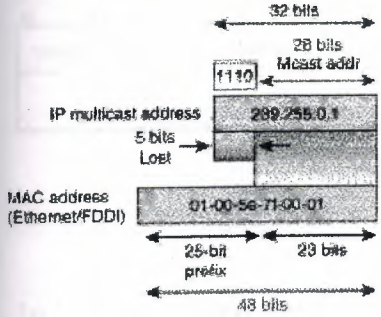
IP multicast makes use of this capability to transmit IP packets to a group of hosts on a LAN segment.

5.3.7 Ethernet MAC Address Mapping

The IANA owns a block of Ethernet MAC addresses that start with 01:00:5E in hexadecimal. Half of this block is allocated for multicast addresses. This creates the range of available Ethernet MAC addresses to be 0100.5e00.0000 through 0100.5e7f.ffff.

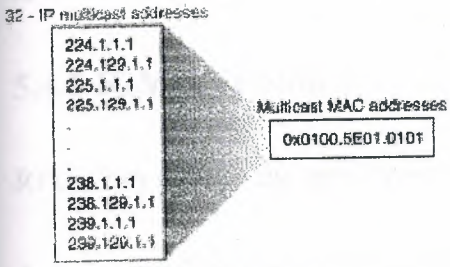
This allocation allows for 23 bits in the Ethernet address to correspond to the IP multicast group address. The mapping places the lower 23 bits of the IP multicast group address into these available 23 bits in the Ethernet address (shown in Figure 43-3).

Figure 43-3: Mapping of IP Multicast to Ethernet/FDDI MAC Address



Because the upper 5 bits of the IP multicast address are dropped in this mapping, the resulting address is not unique. In fact, 32 different multicast group IDs all map to the same Ethernet address (see Figure 43-4).

Figure 43-4: MAC Address Ambiguities



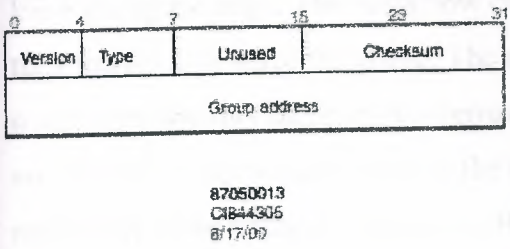
5.4 Internet Group Management Protocol

IGMP is used to dynamically register individual hosts in a multicast group on a particular LAN. Hosts identify group memberships by sending IGMP messages to their local multicast router. Under IGMP, routers listen to IGMP messages and periodically send out queries to discover which groups are active or inactive on a particular subnet.

5.4.1 IGMP Version 1

RFC 1112 defines the specification for IGMP Version 1. A diagram of the packet format is found in Figure 43-5.

Figure 43-5: IGMP Version 1 Packet Format



In Version 1, there are just two different types of IGMP messages:

- Membership query
- Membership report

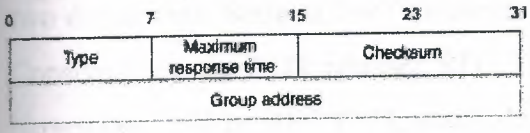
Hosts send out IGMP membership reports corresponding to a particular multicast group to indicate that they are interested in joining that group. The router periodically sends out an IGMP membership query to verify that at least one host on the subnet is still interested in receiving traffic directed to that group. When there is no reply to three consecutive IGMP membership queries, the router times out the group and stops forwarding traffic directed toward that group.

5.4.2 IGMP Version 2

RFC 2236 defines the specification for IGMP Version 2.

A diagram of the packet format follows in Figure 43-6.

Figure 43-6: IGMPv2 Message Format



In Version 2, there are four types of IGMP messages:

- Membership query
- Version 1 membership report
- Version 2 membership report
- Leave group

IGMP Version 2 works basically the same as Version 1. The main difference is that there is a leave group message. The hosts now can actively communicate to the local multicast router their intention to leave the group. The router then sends out a group-specific query and determines whether there are any remaining hosts interested in receiving the traffic. If there are no replies, the router times out the group and stops forwarding the traffic. This can greatly reduce the leave latency compared to IGMP Version 1. Unwanted and unnecessary traffic can be stopped much soo

Chapter 6 Routing Information Protocol

6.1 Background

The Routing Information Protocol, or RIP, as it is more commonly called, is one of the most enduring of all routing protocols. RIP is also one of the more easily confused protocols because a variety of RIP-like routing protocols proliferated, some of which even used the same name! RIP and the myriad RIP-like protocols were based on the same set of algorithms that use distance vectors to mathematically compare routes to identify the best path to any given destination address. These algorithms emerged from academic research that dates back to 1957.

Today's open standard version of RIP, sometimes referred to as IP RIP, is formally defined in two documents: Request For Comments (RFC) 1058 and Internet Standard (STD) 56. Consequently, the IETF released RFC 1388 in January 1993, which was then superceded in November 1994 by RFC 1723, which describes RIP 2 (the second version of RIP). These RFCs described an extension of RIP's capabilities but did not attempt to obsolete the previous version of RIP. RIP 2 enabled RIP messages to carry more information, which permitted the use of a simple authentication mechanism to secure table updates. More importantly, RIP 2 supported subnet masks, a critical feature that was not available in RIP.

This chapter summarizes the basic capabilities and features associated with RIP. Topics include the routing update process, RIP routing metrics, routing stability, and routing timers.

6.2 Routing Updates

RIP sends routing-update messages at regular intervals and when the network topology changes. When a router receives a routing update that includes changes to an entry, it updates its routing table to reflect the new route. RIP routers maintain only the best route (the route with the lowest metric value) to a destination. After updating its routing table, the router immediately begins transmitting routing updates to inform other network routers of the change. These updates are sent independently of the regularly scheduled updates that RIP routers send.

6.3 RIP Routing Metric

RIP uses a single routing metric (hop count) to measure the distance between the source and a destination network. Each hop in a path from source to destination is assigned a hop count value, which is typically 1. When a router receives a routing update that contains a new or changed destination network entry, the router adds 1 to the metric value indicated in the update and enters the network in the routing table. The IP address of the sender is used as the next hop.

6.4 RIP Stability Features

RIP prevents routing loops from continuing indefinitely by implementing a limit on the number of hops allowed in a path from the source to a destination. The maximum number of hops in a path is 15. If a router receives a routing update that contains a new or changed entry, and if increasing the metric value by 1 causes the metric to be infinity (that is, 16), the network destination is considered unreachable. The downside of this stability feature is that it limits the maximum diameter of a RIP network to less than 16 hops.

RIP includes a number of other stability features that are common to many routing protocols. These features are designed to provide stability despite potentially rapid changes in a network's topology. For example, RIP implements the split horizon and holddown mechanisms to prevent incorrect routing information from being propagated.

6.5 RIP Timers

RIP uses numerous timers to regulate its performance. These include a routing-update timer, a route-timeout timer, and a route-flush timer. The routing-update timer clocks the interval between periodic routing updates. Generally, it is set to 30 seconds, with a small random amount of time added whenever the timer is reset. This is done to help prevent congestion, which could result from all routers simultaneously attempting to update their neighbors. Each routing table entry has a route-timeout timer associated with it. When the route-timeout timer expires, the route is marked invalid but is retained in the table until the route-flush timer expires.

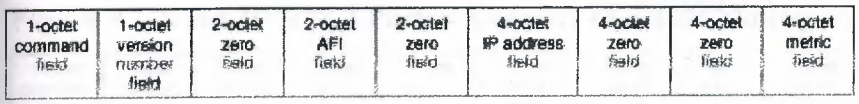
6.6 Packet Formats

The following section focuses on the IP RIP and IP RIP 2 packet formats illustrated in Figures 44-1 and 44-2. Each illustration is followed by descriptions of the fields illustrated.

6.6.1 RIP Packet Format

Figure 47-1 illustrates the IP RIP packet format.

Figure 47-1: An IP RIP Packet Consists of Nine Fields



The following descriptions summarize the IP RIP packet format fields illustrated in Figure 47-1:

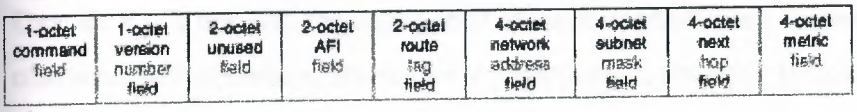
- **Command**—Indicates whether the packet is a request or a response. The request asks that a router send all or part of its routing table. The response can be an unsolicited regular routing update or a reply to a request. Responses contain routing table entries. Multiple RIP packets are used to convey information from large routing tables.
- **Version number**—Specifies the RIP version used. This field can signal different potentially incompatible versions.
- **Zero**—This field is not actually used by RFC 1058 RIP; it was added solely to provide backward compatibility with prestandard varieties of RIP. Its name comes from its defaulted value: zero.
- **Address-family identifier (AFI)**—Specifies the address family used. RIP is designed to carry routing information for several different protocols. Each entry has an address-family identifier to indicate the type of address being specified. The AFI for IP is 2.
- **Address**—Specifies the IP address for the entry.

- **Metric**—Indicates how many internetwork hops (routers) have been traversed in the trip to the destination. This value is between 1 and 15 for a valid route, or 16 for an unreachable route.

6.6.2 RIP 2 Packet Format

The RIP 2 specification (described in RFC 1723) allows more information to be included in RIP packets and provides a simple authentication mechanism that is not supported by RIP. Figure 47-2 shows the IP RIP 2 packet format.

Figure 47-2: An IP RIP 2 Packet Consists of Fields Similar to Those of an IP RIP Packet



The following descriptions summarize the IP RIP 2 packet format fields illustrated in Figure 47-2:

- **Command**—Indicates whether the packet is a request or a response. The request asks that a router send all or a part of its routing table. The response can be an unsolicited regular routing update or a reply to a request. Responses contain routing table entries. Multiple RIP packets are used to convey information from large routing tables.

- **Version**—Specifies the RIP version used. In a RIP packet implementing any of the RIP 2 fields or using authentication, this value is set to 2.

- **Unused**—Has a value set to zero.

- **Address-family identifier (AFI)**—Specifies the address family used. RIPv2's AFI field functions identically to RFC 1058 RIP's AFI field, with one exception: If the AFI for the first entry in the message is 0xFFFF, the remainder of the entry contains authentication information. Currently, the only authentication type is simple password.

- **Route tag**—Provides a method for distinguishing between internal routes (learned by RIP) and external routes (learned from other protocols).

- **IP address**—Specifies the IP address for the entry.

- **Subnet mask**—Contains the subnet mask for the entry. If this field is zero, no subnet mask has been specified for the entry.
- **Next hop**—Indicates the IP address of the next hop to which packets for the entry should be forwarded.
- **Metric**—Indicates how many internetwork hops (routers) have been traversed in the trip to the destination. This value is between 1 and 15 for a valid route, or 16 for an unreachable route.

6.7 Summary

Despite RIP's age and the emergence of more sophisticated routing protocols, it is far from obsolete. RIP is mature, stable, widely supported, and easy to configure. Its simplicity is well suited for use in stub networks and in small autonomous systems that do not have enough redundant paths to warrant the overheads of a more sophisticated protocol.

6.8 Review Questions

Q—*Name RIP's various stability features.*

A—RIP has numerous stability features, the most obvious of which is RIP's maximum hop count. By placing a finite limit on the number of hops that a route can take, routing loops are discouraged, if not completely eliminated. Other stability features include its various timing mechanisms that help ensure that the routing table contains only valid routes, as well as split horizon and holddown mechanisms that prevent incorrect routing information from being disseminated throughout the network.

Q—*What is the purpose of the timeout timer?*

A—The timeout timer is used to help purge invalid routes from a RIP node. Routes that aren't refreshed for a given period of time are likely invalid because of some change in the network. Thus, RIP maintains a timeout timer for each known route. When a route's timeout timer expires, the route is marked invalid but is retained in the table until the route-flush timer expires.

Q—*What two capabilities are supported by RIP 2 but not RIP?*

A—RIP 2 enables the use of a simple authentication mechanism to secure table updates. More importantly, RIP 2 supports subnet masks, a critical feature that is not available in RIP.

Q—*What is the maximum network diameter of a RIP network?*

A—A RIP network's maximum diameter is 15 hops. RIP can count to 16, but that value is considered an error condition rather than a valid hop count.

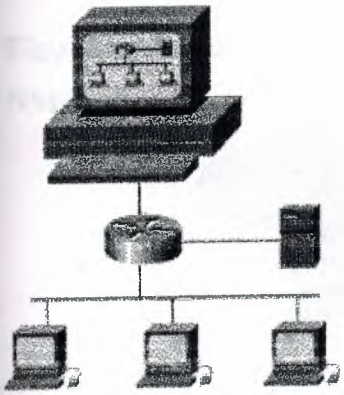
Chapter 7 Simple Network Management Protocol

7.1 Background

The *Simple Network Management Protocol (SNMP)* is an application layer protocol that facilitates the exchange of management information between network devices. It is part of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth.

Two versions of SNMP exist: SNMP version 1 (SNMPv1) and SNMP version 2 (SNMPv2). Both versions have a number of features in common, but SNMPv2 offers enhancements, such as additional protocol operations. Standardization of yet another version of SNMP—SNMP Version 3 (SNMPv3)—is pending. This chapter provides descriptions of the SNMPv1 and SNMPv2 protocol operations. Figure 56-1 illustrates a basic network managed by SNMP.

Figure 56-1: SNMP Facilitates the Exchange of Network Information Between Devices



7.2 SNMP Basic Components

An SNMP-managed network consists of three key components: managed devices, agents, and network-management systems (NMSs).

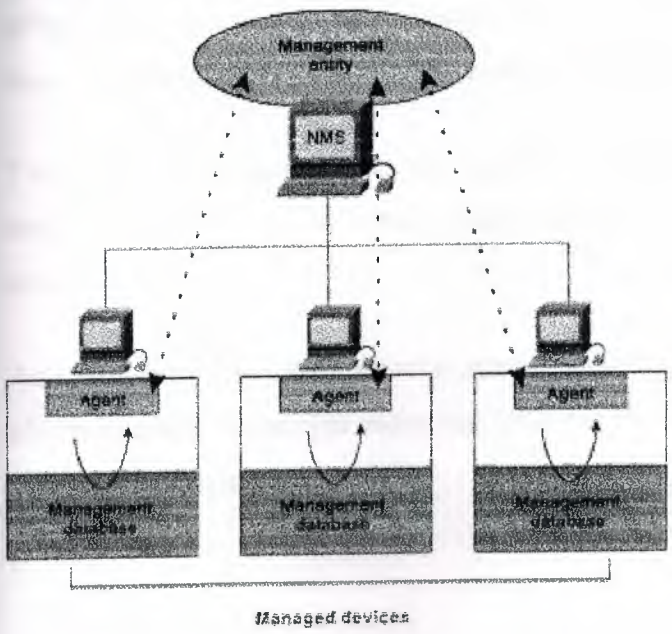
A *managed device* is a network node that contains an SNMP agent and that resides on a managed network. Managed devices collect and store management information and make this information available to NMSs using SNMP. Managed devices, sometimes called network elements, can be routers and access servers, switches and bridges, hubs, computer hosts, or printers.

An *agent* is a network-management software module that resides in a managed device. An agent has local knowledge of management information and translates that information into a form compatible with SNMP.

An *NMS* executes applications that monitor and control managed devices. NMSs provide the bulk of the processing and memory resources required for network management. One or more NMSs must exist on any managed network.

Figure 56-2 illustrates the relationships of these three components.

Figure 56-2: An SNMP-Managed Network Consists of Managed Devices, Agents, and NMSs



7.3 SNMP Basic Commands

Managed devices are monitored and controlled using four basic SNMP commands: **read**, **write**, **trap**, and traversal operations.

The **read** command is used by an NMS to monitor managed devices. The NMS examines different variables that are maintained by managed devices.

The **write** command is used by an NMS to control managed devices. The NMS changes the values of variables stored within managed devices.

The **trap** command is used by managed devices to asynchronously report events to the NMS. When certain types of events occur, a managed device sends a trap to the NMS.

Traversal operations are used by the NMS to determine which variables a managed device supports and to sequentially gather information in variable tables, such as a routing table.

7.4 SNMP Management Information Base

A *Management Information Base (MIB)* is a collection of information that is organized hierarchically. MIBs are accessed using a network-management protocol such as SNMP. They are comprised of managed objects and are identified by object identifiers.

A managed object (sometimes called a MIB object, an object, or a MIB) is one of any number of specific characteristics of a managed device. Managed objects are comprised of one or more object instances, which are essentially variables.

Two types of managed objects exist: scalar and tabular. *Scalar objects* define a single object instance. *Tabular objects* define multiple related object instances that are grouped in MIB tables.

An example of a managed object is atInput, which is a scalar object that contains a single object instance, the integer value that indicates the total number of input AppleTalk packets on a router interface.

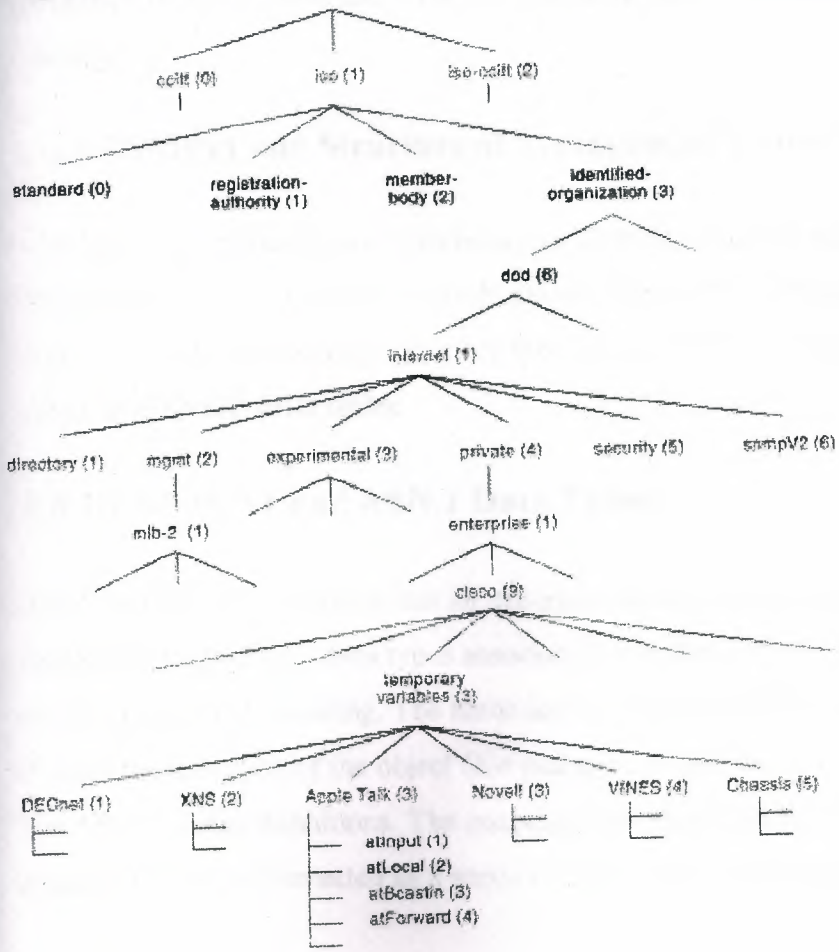
An object identifier (or object ID) uniquely identifies a managed object in the MIB hierarchy. The MIB hierarchy can be depicted as a tree with a nameless root, the levels of which are assigned by different organizations. Figure 56-3 illustrates the MIB tree.

The top-level MIB object IDs belong to different standards organizations, while lower-level object IDs are allocated by associated organizations.

Vendors can define private branches that include managed objects for their own products. MIBs that have not been standardized typically are positioned in the experimental branch.

The managed object atInput can be uniquely identified either by the object name—iso.identified-organization.dod.internet.private.enterprise.cisco.temporary variables.AppleTalk.atInput—or by the equivalent object descriptor, 1.3.6.1.4.1.9.3.3.1.

Figure 56-3: The MIB Tree Illustrates the Various Hierarchies Assigned by Different Organizations



7.5 SNMP and Data Representation

SNMP must account for and adjust to incompatibilities between managed devices. Different computers use different data representation techniques, which can compromise the capability of SNMP to exchange information between managed devices. SNMP uses a subset of Abstract Syntax Notation One (ASN.1) to accommodate communication between diverse systems.

7.6 SNMP Version 1

SNMP version 1 (SNMPv1) is the initial implementation of the SNMP protocol. It is described in Request For Comments (RFC) 1157 and functions within the specifications of the Structure of Management Information (SMI). SNMPv1 operates over protocols such as User Datagram Protocol (UDP), Internet Protocol (IP), OSI Connectionless Network Service (CLNS), AppleTalk Datagram-Delivery Protocol (DDP), and Novell Internet Packet Exchange (IPX). SNMPv1 is widely used and is the *de facto* network-management protocol in the Internet community.

7.6.1 SNMPv1 and Structure of Management Information

The *Structure of Management Information (SMI)* defines the rules for describing management information, using Abstract Syntax Notation One (ASN.1). The SNMPv1 SMI is defined in RFC 1155. The SMI makes three key specifications: ASN.1 data types, SMI-specific data types, and SNMP MIB tables.

7.6.1.1 SNMPv1 and ASN.1 Data Types

The SNMPv1 SMI specifies that all managed objects have a certain subset of Abstract Syntax Notation One (ASN.1) data types associated with them. Three ASN.1 data types are required: name, syntax, and encoding. The name serves as the object identifier (object ID). The syntax defines the data type of the object (for example, integer or string). The SMI uses a subset of the ASN.1 syntax definitions. The encoding data describes how information associated with a managed object is formatted as a series of data items for transmission over the network.

SNMPv1 and SMI-Specific Data Types

The *SNMPv1 SMI* specifies the use of a number of SMI-specific data types, which are divided into two categories: simple data types and application-wide data types.

Three simple data types are defined in the SNMPv1 SMI, all of which are unique values: integers, octet strings, and object IDs. The integer data type is a signed integer in the range of -2,147,483,648 to 2,147,483,647. Octet strings are ordered sequences of 0 to 65,535 octets. Object IDs come from the set of all object identifiers allocated according to the rules specified in ASN.1.

Seven application-wide data types exist in the SNMPv1 SMI: network addresses, counters, gauges, time ticks, opaques, integers, and unsigned integers. Network addresses represent an address from a particular protocol family. SNMPv1 supports only 32-bit IP addresses. Counters are non-negative integers that increase until they reach a maximum value and then return to zero. In SNMPv1, a 32-bit counter size is specified. Gauges are non-negative integers that can increase or decrease but that retain the maximum value reached. A time tick represents a hundredth of a second since some event. An opaque represents an arbitrary encoding that is used to pass arbitrary information strings that do not conform to the strict data typing used by the SMI. An integer represents signed integer-valued information. This data type redefines the integer data type, which has arbitrary precision in ASN.1 but bounded precision in the SMI. An unsigned integer represents unsigned integer-valued information and is useful when values are always non-negative. This data type redefines the integer data type, which has arbitrary precision in ASN.1 but bounded precision in the SMI.

7.6.1.2 SNMP MIB Tables

The SNMPv1 SMI defines highly structured tables that are used to group the instances of a tabular object (that is, an object that contains multiple variables). Tables are composed of zero or more rows, which are indexed in a way that allows SNMP to retrieve or alter an entire row with a single **Get**, **GetNext**, or **Set** command.

7.6.2 SNMPv1 Protocol Operations

SNMP is a simple request/response protocol. The network-management system issues a request, and managed devices return responses. This behavior is implemented by using one of four protocol operations: Get, GetNext, Set, and Trap. The Get operation is used by the NMS to retrieve the value of one or more object instances from an agent. If the agent responding to the Get operation cannot provide values for all the object instances in a list, it does not provide any values. The GetNext operation is used by the NMS to retrieve the value of the next object instance in a table or a list within an agent. The Set operation is used by the NMS to set the values of object instances within an agent. The Trap operation is used by agents to asynchronously inform the NMS of a significant event.

7.7 SNMP Version 2

SNMP version 2 (SNMPv2) is an evolution of the initial version, SNMPv1. Originally, SNMPv2 was published as a set of proposed Internet standards in 1993; currently, it is a draft standard. As with SNMPv1, SNMPv2 functions within the specifications of the Structure of Management Information (SMI). In theory, SNMPv2 offers a number of improvements to SNMPv1, including additional protocol operations.

7.7.1 SNMPv2 and Structure of Management Information

The Structure of Management Information (SMI) defines the rules for describing management information, using ASN.1.

The SNMPv2 SMI is described in RFC 1902. It makes certain additions and enhancements to the SNMPv1 SMI-specific data types, such as including bit strings, network addresses, and counters. Bit strings are defined only in SNMPv2 and comprise zero or more named bits that specify a value. Network addresses represent an address from a particular protocol family. SNMPv1 supports only 32-bit IP addresses, but SNMPv2 can support other types of addresses as well. Counters are non-negative integers that increase until they reach a maximum value and then return to zero. In SNMPv1, a 32-bit counter size is specified. In SNMPv2, 32-bit and 64-bit counters are defined.

7.7.2 SMI Information Modules

The SNMPv2 SMI also specifies information modules, which specify a group of related definitions. Three types of SMI information modules exist: MIB modules, compliance statements, and capability statements. MIB modules contain definitions of interrelated managed objects. Compliance statements provide a systematic way to describe a group of managed objects that must be implemented for conformance to a standard. Capability statements are used to indicate the precise level of support that an agent claims with respect to a MIB group. An NMS can adjust its behavior toward agents according to the capabilities statements associated with each agent.

7.7.3 SNMPv2 Protocol Operations

The Get, GetNext, and Set operations used in SNMPv1 are exactly the same as those used in SNMPv2. However, SNMPv2 adds and enhances some protocol operations. The SNMPv2 Trap operation, for example, serves the same function as that used in SNMPv1, but it uses a different message format and is designed to replace the SNMPv1 Trap.

SNMPv2 also defines two new protocol operations: GetBulk and Inform. The GetBulk operation is used by the NMS to efficiently retrieve large blocks of data, such as multiple rows in a table. GetBulk fills a response message with as much of the requested data as will fit. The Inform operation allows one NMS to send trap information to another NMS and to then receive a response. In SNMPv2, if the agent responding to GetBulk operations cannot provide values for all the variables in a list, it provides partial results.

7.8 SNMP Management

SNMP is a distributed-management protocol. A system can operate exclusively as either an NMS or an agent, or it can perform the functions of both. When a system operates as both an NMS and an agent, another NMS might require that the system query manage devices and provide a summary of the information learned, or that it report locally stored management information.

7.9 SNMP Security

SNMP lacks any authentication capabilities, which results in vulnerability to a variety of security threats. These include masquerading occurrences, modification of information, message sequence and timing modifications, and disclosure. Masquerading consists of an unauthorized entity attempting to perform management operations by assuming the identity of an authorized management entity. Modification of information involves an unauthorized entity attempting to alter a message generated by an authorized entity so that the message results in unauthorized accounting management or configuration management operations. Message sequence and timing modifications occur when an unauthorized entity reorders, delays, or copies and later replays a message generated by an authorized entity. Disclosure results when an unauthorized entity extracts values stored in managed objects, or learns of notifiable events by monitoring exchanges between managers and agents. Because SNMP does not implement authentication, many vendors do not implement Set operations, thereby reducing SNMP to a monitoring facility.

7.10 SNMP Interoperability

As presently specified, SNMPv2 is incompatible with SNMPv1 in two key areas: message formats and protocol operations. SNMPv2 messages use different header and protocol data unit (PDU) formats than SNMPv1 messages. SNMPv2 also uses two protocol operations that are not specified in SNMPv1. Furthermore, RFC 1908 defines two possible SNMPv1/v2 coexistence strategies: proxy agents and bilingual network-management systems.

7.10.1 Proxy Agents

An SNMPv2 agent can act as a proxy agent on behalf of SNMPv1 managed devices, as follows:

- An SNMPv2 NMS issues a command intended for an SNMPv1 agent.
- The NMS sends the SNMP message to the SNMPv2 proxy agent.
- The proxy agent forwards Get, GetNext, and Set messages to the SNMPv1 agent unchanged.

- GetBulk messages are converted by the proxy agent to GetNext messages and then are forwarded to the SNMPv1 agent.

The proxy agent maps SNMPv1 trap messages to SNMPv2 trap messages and then forwards them to the NMS.

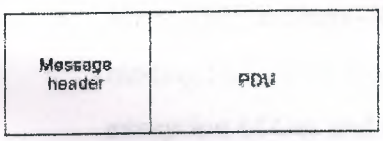
7.10.2 Bilingual Network-Management System

Bilingual SNMPv2 network-management systems support both SNMPv1 and SNMPv2. To support this dual-management environment, a management application in the bilingual NMS must contact an agent. The NMS then examines information stored in a local database to determine whether the agent supports SNMPv1 or SNMPv2. Based on the information in the database, the NMS communicates with the agent using the appropriate version of SNMP.

7.11 SNMP Reference: SNMPv1 Message Formats

SNMPv1 messages contain two parts: a message header and a protocol data unit (PDU). Figure 56-4 illustrates the basic format of an SNMPv1 message.

Figure 56-4: An SNVPv1 Message Consists of a Header and a PDU



7.11.1 SNMPv1 Message Header

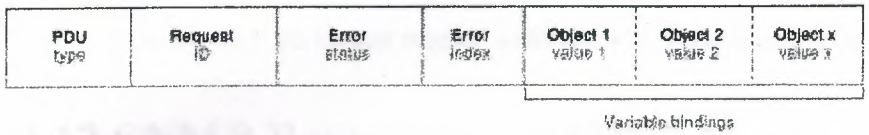
SNMPv1 message headers contain two fields: Version Number and Community Name. The following descriptions summarize these fields:

- **Version number**—Specifies the version of SNMP used.
- **Community name**—Defines an access environment for a group of NMSs. NMSs within the community are said to exist within the same administrative domain. Community names serve as a weak form of authentication because devices that do not know the proper community name are precluded from SNMP operations.

7.11.2 SNMPv1 Protocol Data Unit

SNMPv1 PDUs contain a specific command (Get, Set, and so on) and operands that indicate the object instances involved in the transaction. SNMPv1 PDU fields are variable in length, as prescribed by ASN.1. Figure 56-5 illustrates the fields of the SNMPv1 Get, GetNext, Response, and Set PDUs transactions.

Figure 56-5: SNMPv1 Get, GetNext, Response, and Set PDUs Contain the Same Fields



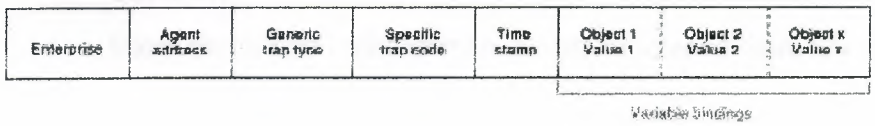
The following descriptions summarize the fields illustrated in Figure 56-5:

- **PDU type**—Specifies the type of PDU transmitted.
- **Request ID**—Associates SNMP requests with responses.
- **Error status**—Indicates one of a number of errors and error types. Only the response operation sets this field. Other operations set this field to zero.
- **Error index**—Associates an error with a particular object instance. Only the response operation sets this field. Other operations set this field to zero.
- **Variable bindings**—Serves as the data field of the SNMPv1 PDU. Each variable binding associates a particular object instance with its current value (with the exception of Get and GetNext requests, for which the value is ignored).

7.11.3 Trap PDU Format

Figure 56-6 illustrates the fields of the SNMPv1 Trap PDU.

Figure 56-6: The SNMPv1 Trap PDU Consists of Eight Fields



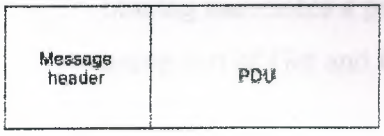
The following descriptions summarize the fields illustrated in Figure 56-6:

- **Enterprise**—Identifies the type of managed object generating the trap.
- **Agent address**—Provides the address of the managed object generating the trap.
- **Generic trap type**—Indicates one of a number of generic trap types.
- **Specific trap code**—Indicates one of a number of specific trap codes.
- **Time stamp**—Provides the amount of time that has elapsed between the last network reinitialization and generation of the trap.
- **Variable bindings**—The data field of the SNMPv1 Trap PDU. Each variable binding associates a particular object instance with its current value.

7.12 SNMP Reference: SNMPv2 Message Format

SNMPv2 messages consist of a header and a PDU. Figure 56-7 illustrates the basic format of an SNMPv2 message.

Figure 56-7: SNMPv2 Messages Also Consist of a Header and a PDU



7.12.1 SNMPv2 Message Header

SNMPv2 message headers contain two fields: Version Number and Community Name. The following descriptions summarize these fields:

- **Version number**—Specifies the version of SNMP that is being used.
- **Community name**—Defines an access environment for a group of NMSs. NMSs within the community are said to exist within the same administrative domain. Community names serve as a weak form of authentication because devices that do not know the proper community name are precluded from SNMP operations.

7.12.2 SNMPv2 Protocol Data Unit

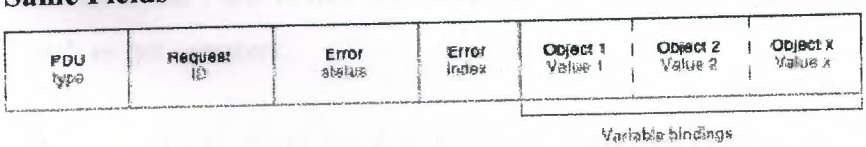
SNMPv2 specifies two PDU formats, depending on the SNMP protocol operation. SNMPv2 PDU fields are variable in length, as prescribed by Abstract Syntax Notation One (ASN.1).

Figure 56-8 illustrates the fields of the SNMPv2 Get, GetNext, Inform, Response, Set, and Trap PDUs.

The following descriptions summarize the fields illustrated in Figure 56-8:

- **PDU type**—Identifies the type of PDU transmitted (Get, GetNext, Inform, Response, Set, or Trap).
- **Request ID**—Associates SNMP requests with responses.
- **Error status**—Indicates one of a number of errors and error types. Only the response operation sets this field. Other operations set this field to zero.
- **Error index**—Associates an error with a particular object instance. Only the response operation sets this field. Other operations set this field to zero.
- **Variable bindings**—Serves as the data field of the SNMPv2 PDU. Each variable binding associates a particular object instance with its current value (with the exception of Get and GetNext requests, for which the value is ignored).

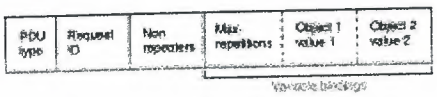
Figure 56-8: SNMPv2 Get, GetNext, Inform, Response, Set, and Trap PDUs Contain the Same Fields



7.12.2.1 GetBulk PDU Format

Figure 56-9 illustrates the fields of the SNMPv2 GetBulk PDU.

Figure 56-9: The SNMPv2 GetBulk PDU Consists of Seven Fields



The following descriptions summarize the fields illustrated in Figure 56-9:

- **PDU type**—Identifies the PDU as a GetBulk operation.
- **Request ID**—Associates SNMP requests with responses.
- **Non repeaters**—Specifies the number of object instances in the variable bindings field that should be retrieved no more than once from the beginning of the request. This field is used when some of the instances are scalar objects with only one variable.
- **Max repetitions**—Defines the maximum number of times that other variables beyond those specified by the Non repeaters field should be retrieved.
- **Variable bindings**—Serves as the data field of the SNMPv2 PDU. Each variable binding associates a particular object instance with its current value (with the exception of Get and GetNext requests, for which the value is ignored).

7.13 Review Questions

Q—*What are MIBs, and how are they accessed?*

A—A Management Information Base (MIB) is a collection of information that is organized hierarchically. MIBs are accessed using a network-management protocol such as SNMP. They are comprised of managed objects and are identified by object identifiers.

Q—*SNMP uses a series of _____ and _____to manage the network.*

A—Gets and Puts. SNMP uses a Get object and a Put object to manage devices on a network such as get counters.

Q—*Name three of the seven fields of the SNMP v2 GETBULK.*

A—PDU Type, Request ID, Nonrepeaters, Max Repetitions, Variable Bindings (the variable bindings consists of variable object fields that make up the three remaining fields).

Chapter 8 UDP Broadcast Flooding

A *broadcast* is a data packet that is destined for multiple hosts. Broadcasts can occur at the data link layer and the network layer. Data-link broadcasts are sent to all hosts attached to a particular physical network. Network layer broadcasts are sent to all hosts attached to a particular logical network. The Transmission Control Protocol/Internet Protocol (TCP/IP) supports the following types of broadcast packets:

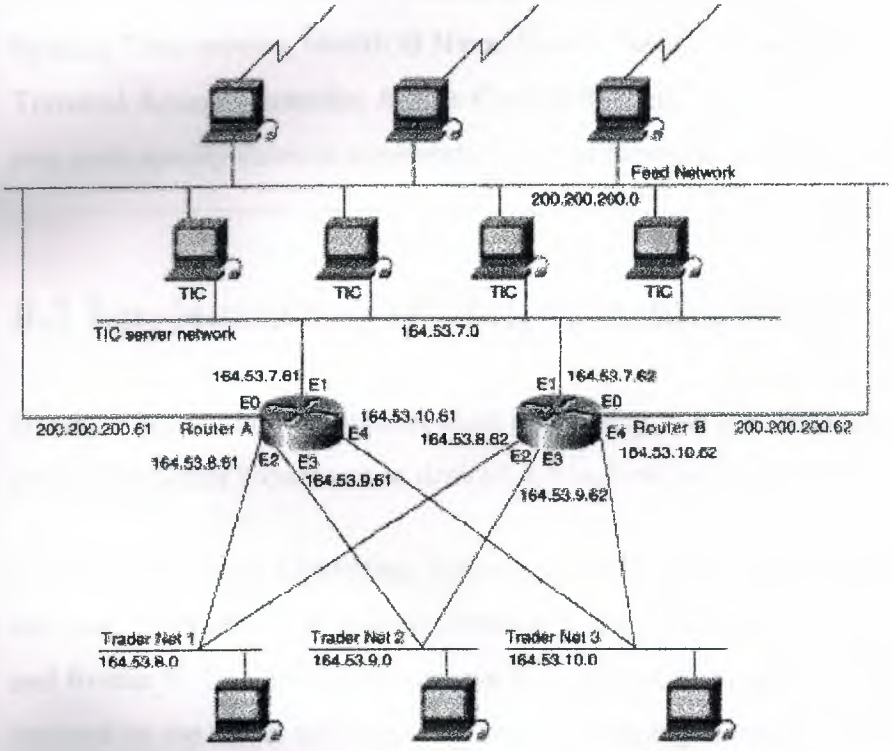
- *All ones*—By setting the broadcast address to all ones (255.255.255.255), all hosts on the network receive the broadcast.
- *Network*—By setting the broadcast address to a specific network number in the network portion of the IP address and setting all ones in the host portion of the broadcast address, all hosts on the specified network receive the broadcast. For example, when a broadcast packet is sent with the broadcast address of 131.108.255.255, all hosts on network number 131.108 receive the broadcast.
- *Subnet*—By setting the broadcast address to a specific network number and a specific subnet number, all hosts on the specified subnet receive the broadcast. For example, when a broadcast packet is set with the broadcast address of 131.108.4.255, all hosts on subnet 4 of network 131.108 receive the broadcast.

Because broadcasts are recognized by all hosts, a significant goal of router configuration is to control unnecessary proliferation of broadcast packets. Cisco routers support two kinds of broadcasts: *directed* and *flooded*. A directed broadcast is a packet sent to a specific network or series of networks, whereas a flooded broadcast is a packet sent to every network. In IP internetworks, most broadcasts take the form of User Datagram Protocol (UDP) broadcasts.

Although current IP implementations use a broadcast address of all ones, the first IP implementations used a broadcast address of all zeros. Many of the early implementations do not recognize broadcast addresses of all ones and fail to respond to the broadcast correctly. Other early implementations forward broadcasts of all ones, which causes a serious network overload known as a *broadcast storm*. Implementations that exhibit these problems include systems based on versions of BSD UNIX prior to Version 4.3.

In the brokerage community, applications use UDP broadcasts to transport market data to the desktops of traders on the trading floor. This case study gives examples of how brokerages have implemented both directed and flooding broadcast schemes in an environment that consists of Cisco routers and Sun workstations. Figure 19-1 illustrates a typical topology. Note that the addresses in this network use a 10-bit netmask of 255.255.255.192.

Figure 19-1: Topology that requires UDP broadcast forwarding.



In Figure 19-1, UDP broadcasts must be forwarded from a source segment (Feed network) to many destination segments that are connected redundantly. Financial market data, provided, for example, by Reuters, enters the network through the Sun workstations connected to the Feed network and is disseminated to the TIC servers. The TIC servers are Sun workstations running Teknekron Information Cluster software. The Sun workstations on the trader networks subscribe to the TIC servers for the delivery of certain market data, which the TIC servers deliver by means of UDP broadcasts. The two routers in this network provide redundancy so that if one router becomes unavailable, the other router can assume the load of the failed router without intervention from an operator. The connection between each router and the Feed network is for network administration purposes only and does not carry user traffic.

Two different approaches can be used to configure Cisco routers for forwarding UDP broadcast traffic: IP helper addressing and UDP flooding. This case study analyzes the advantages and disadvantages of each approach.

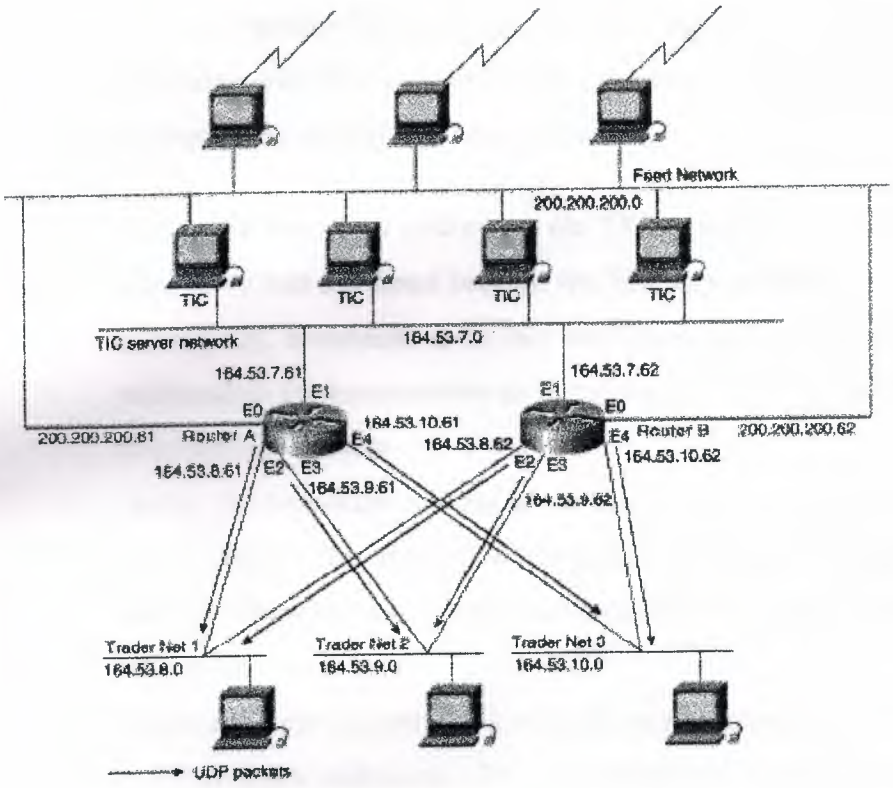
Note Regardless of whether you implement IP helper addressing or UDP flooding, you must use the **ip forward-protocol udp** global configuration command to enable the UDP forwarding. By default, the **ip forward-protocol udp** command enables forwarding for ports associated with the following protocols: Trivial File Transfer Protocol, Domain Name System, Time service, NetBIOS Name Server, NetBIOS Datagram Server, Boot Protocol, and Terminal Access Controller Access Control System. To enable forwarding for other ports, you must specify them as arguments to the **ip forward-protocol udp** command.

8.1 Implementing IP Helper Addressing

IP helper addressing is a form of static addressing that uses directed broadcasts to forward local and all-nets broadcasts to desired destinations within the internetwork.

To configure helper addressing, you must specify the **ip helper-address** command on every interface on every router that receives a broadcast that needs to be forwarded. On Router A and Router B, IP helper addresses can be configured to move data from the TIC server network to the trader networks. IP helper addressing in not the optimal solution for this type of topology because each router receives unnecessary broadcasts from the other router, as shown in Figure 19-2.

Figure 19-2: Flow of UDP packets from routers to trader networks using IP helper addressing.



In this case, Router A receives each broadcast sent by Router B *three times*, one for each segment, and Router B receives each broadcast sent by Router A three times, one for each segment. When each broadcast is received, the router must analyze it and determine that the broadcast does not need to be forwarded. As more segments are added to the network, the routers become overloaded with unnecessary traffic, which must be analyzed and discarded.

When IP helper addressing is used in this type of topology, no more than one router can be configured to forward UDP broadcasts (unless the receiving applications can handle duplicate broadcasts). This is because duplicate packets arrive on the trader network. This restriction limits redundancy in the design and can be undesirable in some implementations.

To send UDP broadcasts bidirectionally in this type of topology, a second **ip helper address** command must be applied to every router interface that receives UDP broadcasts. As more segments and devices are added to the network, more **ip helper address** commands are required to reach them, so the administration of these routers becomes more complex over

time. Note, too, that bidirectional traffic in this topology significantly impacts router performance.

Although IP helper addressing is well-suited to nonredundant, nonparallel topologies that do not require a mechanism for controlling broadcast loops, in view of these drawbacks, IP helper addressing does not work well in this topology. To improve performance, network designers considered several other alternatives:

- *Setting the broadcast address on the TIC servers to all ones (255.255.255.255)*—This alternative was dismissed because the TIC servers have more than one interface, causing TIC broadcasts to be sent back onto the Feed network. In addition, some workstation implementations do not allow all ones broadcasts when multiple interfaces are present.
- *Setting the broadcast address of the TIC servers to the major net broadcast (164.53.0.0)*—This alternative was dismissed because the Sun TCP/IP implementation does not allow the use of major net broadcast addresses when the network is subnetted.
- *Eliminating the subnets and letting the workstations use Address Resolution Protocol (ARP) to learn addresses*—This alternative was dismissed because the TIC servers cannot quickly learn an alternative route in the event of a primary router failure.

With alternatives eliminated, the network designers turned to a simpler implementation that supports redundancy without duplicating packets and that ensures fast convergence and minimal loss of data when a router fails: UDP flooding.

8.2 Implementing UDP Flooding

UDP flooding uses the spanning tree algorithm to forward packets in a controlled manner. Bridging is enabled on each router interface for the sole purpose of building the spanning tree. The spanning tree prevents loops by stopping a broadcast from being forwarded out an interface on which the broadcast was received. The spanning tree also prevents packet duplication by placing certain interfaces in the blocked state (so that no packets are forwarded) and other interfaces in the forwarding state (so that packets that need to be forwarded are forwarded).

To enable UDP flooding, the router must be running software that supports transparent bridging and bridging must be configured on each interface that is to participate in the flooding. If bridging is not configured for an interface, the interface will receive broadcasts, but the router will not forward those broadcasts and will not use that interface as a destination for sending broadcasts received on a different interface.

When configured for UPD flooding, the router uses the destination address specified by the **ip broadcast-address** command on the output interface to assign a destination address to a flooded UDP datagram. Thus, the destination address might change as the datagram propagates through the network. The source address, however, does not change.

With UDP flooding, both routers shown in Figure 19-1 use a spanning tree to control the network topology for the purpose of forwarding broadcasts.

The **bridge protocol** command can specify either the **dec** keyword (for the DEC spanning-tree protocol) or the **ieee** keyword (for the IEEE Ethernet protocol). All routers in the network must enable the same spanning tree protocol. The **ip forward-protocol spanning tree** command uses the database created by the **bridge protocol** command. Only one broadcast packet arrives at each segment, and UDP broadcasts can traverse the network in both directions.

Note Because bridging is enabled only to build the spanning tree database, use access lists to prevent the spanning tree from forwarding non-UDP traffic. The configuration examples later in this chapter configure an access list that blocks all bridged packets.

To determine which interface forwards or blocks packets, the router configuration specifies a path cost for each interface. The default path cost for Ethernet is 100. Setting the path cost for each interface on Router B to 50 causes the spanning tree algorithm to place the interfaces in Router B in forwarding state. Given the higher path cost (100) for the interfaces in Router A, the interfaces in Router A are in the blocked state and do not forward the broadcasts. With these interface states, broadcast traffic flows through Router B. If Router B fails, the spanning tree algorithm will place the interfaces in Router A in the forwarding state, and Router A will forward broadcast traffic.

With one router forwarding broadcast traffic from the TIC server network to the trader networks, it is desirable to have the other forward unicast traffic. For that reason, each router enables the ICMP Router Discovery Protocol (IRDP), and each workstation on the trader networks runs the **irdp** daemon. On Router A, the **preference** keyword sets a higher IRDP preference than does the configuration for Router B, which causes each **irdp** daemon to use Router A as its preferred default gateway for unicast traffic forwarding. Users of those workstations can use **netstat -rn** to see how the routers are being used.

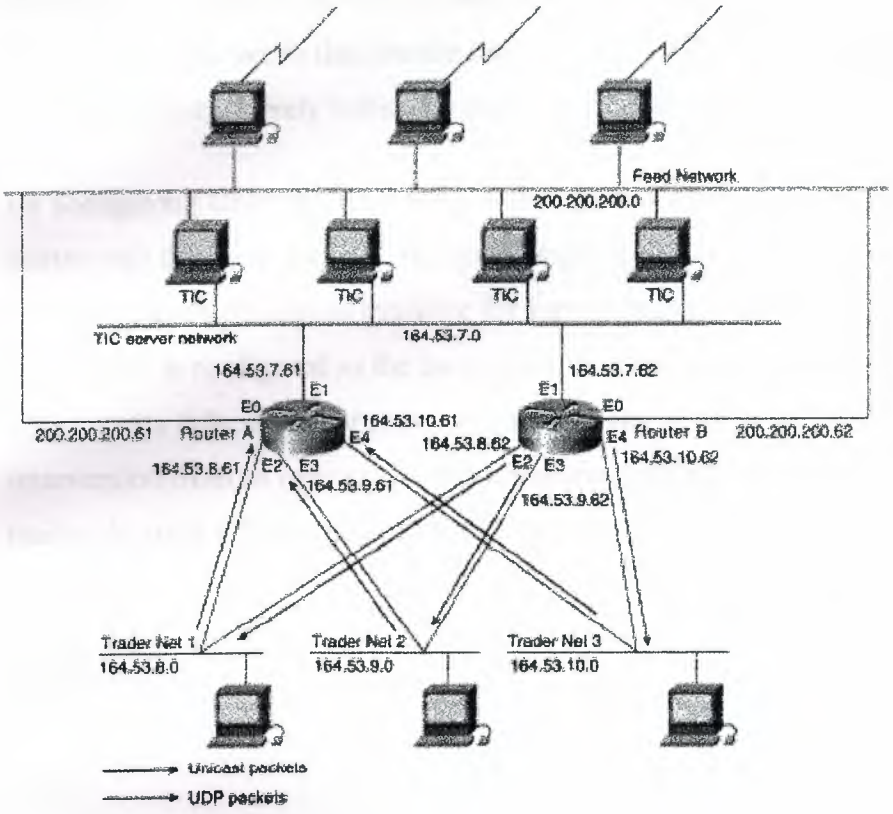
On the routers, the **holdtime**, **maxadvertinterval**, and **minadvertinterval** keywords reduce the advertising interval from the default so that the **irdp** daemons running on the hosts expect to see advertisements more frequently. With the advertising interval reduced, the workstations will adopt Router B more quickly if Router A becomes unavailable. With this configuration, when a router becomes unavailable, IRDP offers a convergence time of less than one minute.

IRDP is preferred over the Routing Information Protocol (RIP) and default gateways for the following reasons:

- RIP takes longer to converge, typically from one to two minutes.
- Configuration of Router A as the default gateway on each Sun workstation on the trader networks would allow those Sun workstations to send unicast traffic to Router A, but would not provide an alternative route if Router A becomes unavailable.

Figure 19-3 shows how data flows when the network is configured for UDP flooding.

Figure 19-3: Data flow with UDP flooding and IRDP.



Note This topology is broadcast intensive—broadcasts sometimes consume 20 percent of the Ethernet bandwidth. However, this is a favorable percentage when compared to the configuration of IP helper addressing, which, in the same network, causes broadcasts to consume up to 50 percent of the Ethernet bandwidth.

If the hosts on the trader networks do not support IRDP, the Hot Standby Routing Protocol (HSRP) can be used to select which router will handle unicast traffic. HSRP allows the standby router to take over quickly if the primary router becomes unavailable. For information about configuring HSRP, see "Using HSRP for Fault-Tolerant IP Routing."

Summary

Although IP helper addressing is useful in networks that do not require redundancy, when configured in networks that feature redundancy, IP helper addressing results in packet duplication that severely reduces router and network performance.

By configuring UDP flooding, one router forwards UDP traffic without burdening the second router with duplicate packets. By dedicating one router to the task of forwarding UDP traffic, the second router becomes available for forwarding unicast traffic. At the same time, because each router is configured as the backup for the other router, redundancy is maintained; if either router fails, the other router can assume the work of the failed router without intervention from an operator. When compared with IP helper addressing, UDP flooding makes the most efficient use of router resources.

Conclusion

I believe that TCP/IP is important because it set the "standard" for network communication. We have learnt that without standards some aspects of computing would not work, without the TCP/IP standard file sharing and networking communication would be almost impossible across the different networks that make up the Internet today. At the time of TCP/IP development several other commercial companies had been developing their own protocols. If TCP/IP had not set the standard it is probable that we would now have several independent networks. Each would be running under their own rules and connecting them together would be virtually impossible.

I also believe that if TCP/IP had been taken out of the equation then we would possibly have been faced with a very closed network. It wouldn't be feasible for a home PC user to try to connect to one of these networks; indeed the resources provided would probably be of little use to us. Therefore I conclude that without TCP/IP the growth of the Internet that we have witnessed over the past few years would have been greatly diminished, if it had happened at all.

REFERENCES

[1] Adolfo Rodriguez ,John Gatrell ,John Karas ,Roland Peschke
TCP/IP Tutorial and Technical Overview

[2] Cisco Systems, Inc
http://www.cisco.com/

[3] Yale University Library
http://www.yale.edu

[4] Microsoft Corporation, Inc
http://www.microsoft.com