NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

NETWORKS AND ROUTING PROBLEM

Graduation Project
COM- 400

Student: AYMAN AL-SALLAJ

Supervisor: Assoc.Prof.Dr.RAHIB ABIYEV

Nicosia - 2004

Acknowledgement

It is my honor to thank my supervisor for this project assoc. Prof .Dr. Rahib Abeiv for showing his help and for being so co-operative throughout his supervision at this work and other courses that I have taken with him during my academic study.

I would like to express my tight thanks to my Parents; my father Dr. Al-Sallaj; and great mum, then I'm in dept to my brother Dr. Naman Sallaj, also my brothers Yazan and Zaid, who helped me many times in my whole life, they boosted me up of my study life. Their love has motivated me to make the gradation day come true.

With my thanks to the Near East University stuff professors, Assoc., profs., Drs., and Mrs., especially prof, Dr. Fakhredden Maedove, Mr.Tyseer Al-Shanablah, Mr. Umit Ilhan.

I would also like to thank my friends, who helped me so much in doing my project .they encouraged me a lot in completing my project.

I have to save the thank to my aunts and my grand mum; for being the special person that you are; you make every day special for everyone around you; for your kindness, warmth and company; and for the fantastic meals and gifts.

ABSTRACT

A local area network (LAN) is a group of computers and associated devices that share a common communications line or wireless link and typically share the resources of a single processor or server within a small geographic area (for example, within an office building). Usually, the server has applications and data storage that are shared in common by multiple computer users. A local area network may serve as few as two or three users (for example, in a home network) or many as thousands of users (for example, in an FDDI network).

Routed protocols are transported by routing protocols across an internetwork. In general, routed protocols in this context also are referred to as *network* protocols. These network protocols perform a variety of functions required for communication between user applications in source and destination devices, and these functions can differ widely among protocol suites. Network protocols occur at the upper four layers of the OSI reference model: the transport layer, the session layer, the presentation layer, and the application layer.LAN-TO-LAN routing the network layer must understand and be able to interface with various lower layers. Routers must be capable of seamlessly handling packets encapsulated into various lower-level frames without changing the packets' Layer 3 addressing.

LAN-TO WAN routing the network layer must relate to, and interface with, various lower layers for LAN-to-WAN traffic. As an internetwork grows, the path taken by a packet may encounter several relay points and a variety of data link types beyond the LANs

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

CONTENTS

CHAPTER ONE

LOCAL AREA NETWORK AND ROUTING

1. LOCAL AREA NETWORK
1.1 Basic OF LAN

A local area network (LAN) is a group of computers and associated devices that share a common communications line or wireless link and typically share the resources of a single processor or server within a small geographic area (for example, within an office building). Usually, the server has applications and data storage that are shared in common by multiple computer users. A local area network may serve as few as two or three users (for example, in a home network) or many as thousands of users (for example, in an FDDI network).

The main local area network technologies are:

- Ethernet
- Token Ring
- ARCNET
- FDDI (Fiber Distributed Data Interface)

1. Ethernet

is the most widely-installed local area network (LAN) technology. Specified in a standard, IEEE 802.3, Ethernet was originally developed by Xerox and then developed further by Xerox, DEC, and Intel. An Ethernet LAN typically uses coaxial cable or special grades of twisted pair wires. Ethernet is also used in wireless LANs. The most commonly installed Ethernet systems are called 10BASE-T and provide transmission speeds up to 10 Mbps. Devices are connected to the cable and compete for access using a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol.
Fast Ethernet or 100BASE-T provides transmission speeds up to 100 megabits per second and is typically used for LAN backbone systems, supporting workstations with 10BASE-T cards. Gigabit Ethernet provides an even higher level of backbone support at 1000 megabits per second (1 gigabit or 1 billion bits per second). 10-Gigabit Ethernet provides up to 10 billion bits per second.

1.1.2 Token Ring

Token Ring network is a local area network (LAN) in which all computers are connected in a ring or star topology and a bit- or token-passing scheme is used in order to prevent the collision of data between two computers that want to send messages at the same time. The Token Ring protocol is the second most widely-used protocol on local area networks after Ethernet. The IBM Token Ring protocol led to a standard version, specified as IEEE 802.5. Both protocols are used and are very

similar. The IEEE 802.5 Token Ring technology provides for data transfer rates of either 4 or 16 megabits per second. Very briefly, here is how it works:
Empty information frames are continuously circulated on the ring.
When a computer has a message to send, it inserts a token in an empty frame (this may consist of simply changing a 0 to a 1 in the token bit part of the frame) and inserts a message and a destination identifier in the frame.
The frame is then examined by each successive workstation. If the workstation sees that it is the destination for the message, it copies the message from the frame and changes the token back to 0.
When the frame gets back to the originator, it sees that the token has been changed to 0 and that the message has been copied and received. It removes the message from the frame.
The frame continues to circulate as an "empty" frame, ready to be taken by a workstation when it has a message to send.
The token scheme can also be used with bus topology LANs.
The standard for the Token Ring protocol is Institute of Electrical and Electronics Engineers (IEEE) 802.5. The Fiber Distributed-Data Interface (FDDI) also uses a Token Ring protocol.

1.1.3 ARCNET

ARCNET is a widely-installed local area network (LAN) technology that uses a *token-bus* scheme for managing line sharing among the workstations and other devices connected on the LAN. The LAN server continuously circulates empty message frames on a bus (a line in which every message goes through every device on the line and a device uses only those with its address). When a device wants to send a message, it inserts a "token" (this can be as simple as setting a token bit to 1) in an empty frame in which it also inserts the message. When the destination device or LAN server reads the message, it resets the token to 0 so that the frame can be reused by any other device. The scheme is very efficient when traffic increases since all devices are afforded the same opportunity to use the shared network.
ARCNET can use coaxial cable or fiber optic lines. ARCNET is one of four major LAN technologies, which also include Ethernet, Token Ring, and FDDI.

1.1.4 FDDT (Fiber Distributed data Interface)

FDDI (Fiber Distributed Data Interface) is a set of ANSI and ISO standards for data transmission on fiber optic lines in a local area network (LAN) that can extend in range up to 200 km (124 miles). The FDDI protocol is based on the Token Ring protocol. In addition to being large geographically, an FDDI local area network can support thousands of users. FDDI is frequently used on the backbone for a wide area network (WAN).
An FDDI network contains two token rings, one for possible backup in case the primary ring fails. The primary ring offers up to 100 Mbps capacity. If the secondary ring is not needed for backup, it can also carry data, extending capacity to 200 Mbps. The single ring can extend the maximum distance; a dual ring can extend 100 km (62 miles).
FDDI is a product of American National Standards Committee X3-T9 and conforms to the Open Systems Interconnection (OSI) model of functional layering. It can be used to interconnect LANs using other protocols. FDDI-II is a version of FDDI that

adds the capability to add circuit-switched service to the network so that voice signals can also be handled. Work is underway to connect FDDI networks to the developing Synchronous Optical Network (SONET).
Typically, a suite of application programs can be kept on the LAN server. Users who need an application frequently can download it once and then run it from their local hard disk. Users can order printing and other services as needed through applications run on the LAN server. A user can share files with others at the LAN server; read and write access is maintained by a LAN administrator. A LAN server may also be used as a Web server if safeguards are taken to secure internal applications and data from outside access.
In some situations, a **wireless LAN** may be preferable to a wired LAN because it is cheaper to install and maintain.
6.2 Networking Basics
Here are some of the fundamental parts of a network:

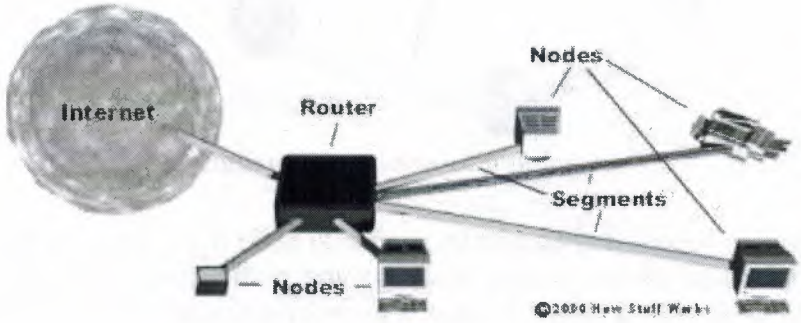


figure : fundamental parts of a network

Network - A network is a group of computers connected together in a way that allows information to be exchanged between the computers.
Node - A node is anything that is connected to the network. While a node is typically a computer, it can also be something like a printer or CD-ROM tower.
Segment - A segment is any portion of a network that is separated, by a switch, bridge or router, from other parts of the network.
Backbone - The backbone is the main cabling of a network that all of the segments connect to. Typically, the backbone is capable of carrying more information than the individual segments. For example, each segment may have a transfer rate of 10 Mbps (megabits per second), while the backbone may operate at 100 Mbps.
Topology - Topology is the way that each node is physically connected to the network. Common topologies include:
Bus - Each node is daisy-chained (connected one right after the other) along the same backbone, similar to Christmas lights. Information sent from a node travels along the backbone until it reaches its destination node. Each end of a bus network must be **terminated** with a resistor to keep the signal that is sent by a node across the network from bouncing back when it reaches the end of the cable.
Ring - Like a bus network, rings have the nodes daisy-chained. The difference is that the end of the network comes back around to the first node, creating a complete circuit. In a ring network, each node takes a turn sending and receiving information through the use of a token. The token, along with any data, is sent from the first node to the second node, which extracts the data addressed to it and adds any data it wishes to send. Then, the second node passes the token and data to the third node, and so on

until it comes back around to the first node again. Only the node with the token is allowed to send data. All other nodes must wait for the token to come to them.

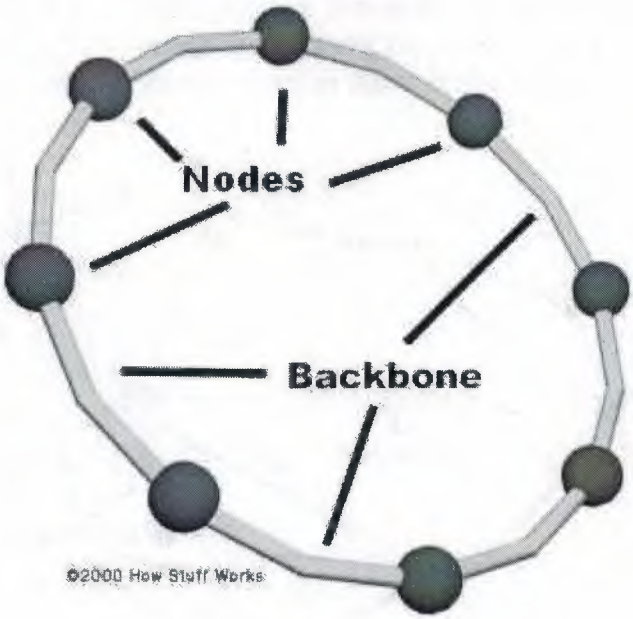


figure : Ring network topology

Star - In a star network, each node is connected to a central device called a **hub**. The hub takes a signal that comes from any node and passes it along to all the other nodes in the network. A hub does not perform any type of filtering or routing of the data. It is simply a junction that joins all the different nodes together.

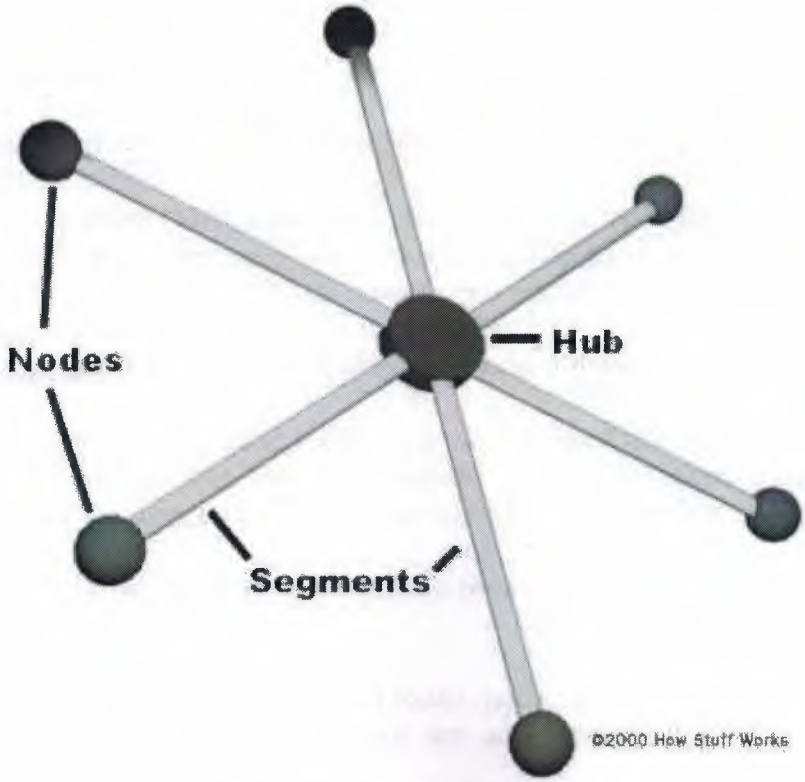


figure : Star network topology

Star bus - Probably the most common network topology in use today, star bus combines elements of the star and bus topologies to create a versatile network environment. Nodes in particular areas are connected to hubs (creating stars), and the hubs are connected together along the network backbone (like a bus network). Quite often, stars are nested within stars, as seen in the example below:

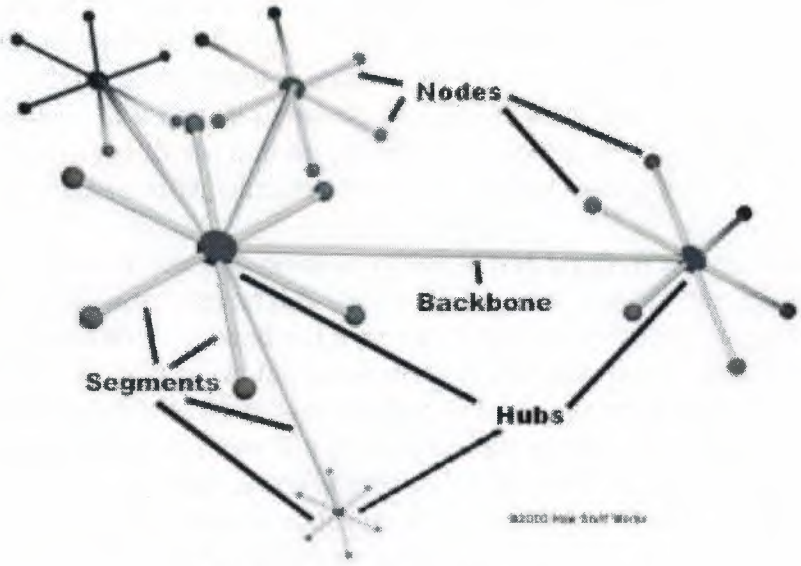


figure :A typical star bus network

Local Area Network (LAN) - A LAN is a network of computers that are in the same general physical location, usually within a building or a campus. If the computers are far apart (such as across town or in different cities), then a **Wide Area Network** (WAN) is typically used.

Network Interface Card (NIC) - Every computer (and most other devices) is connected to a network through an NIC. In most desktop computers, this is an Ethernet card (normally 10 or 100 Mbps) that is plugged into a slot on the computer's motherboard.

Media Access Control (MAC) address - This is the *physical* address of any device -- such as the NIC in a computer -- on the network. The MAC address has two parts, each 3 bytes long. The first 3 bytes identify the company that made the NIC. The second 3 bytes are the serial number of the NIC itself.

Unicast - A unicast is a transmission from one node addressed specifically to another node.

Multicast - In a multicast, a node sends a packet addressed to a special group address. Devices that are interested in this group register to receive packets addressed to the group. An example might be a Cisco router sending out an update to all of the other Cisco routers.

Broadcast - In a broadcast, a node sends out a packet that is intended for transmission to all other nodes on the network.

Adding Switches

In the most basic type of network found today, nodes are simply connected together using hubs. As a network grows, there are some potential problems with this configuration:

Scalability - In a hub network, limited shared bandwidth makes it difficult to accommodate significant growth without sacrificing performance. Applications today

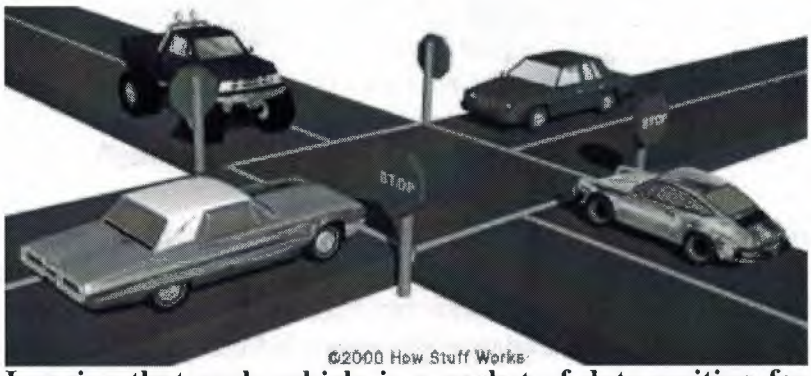
need more bandwidth than ever before. Quite often, the entire network must be redesigned periodically to accommodate growth.

Latency - This is the amount of time that it takes a packet to get to its destination. Since each node in a hub-based network has to wait for an opportunity to transmit in order to avoid collisions, the latency can increase significantly as you add more nodes. Or, if someone is transmitting a large file across the network, then all of the other nodes have to wait for an opportunity to send their own packets. You have probably seen this before at work -- you try to access a server or the Internet and suddenly everything slows down to a crawl.

Network failure - In a typical network, one device on a hub can cause problems for other devices attached to the hub due to incorrect speed settings (100 Mbps on a 10-Mbps hub) or excessive broadcasts. Switches can be configured to limit broadcast levels.

Collisions - Ethernet uses a process called **CSMA/CD** (Carrier Sense Multiple Access with Collision Detection) to communicate across the network. Under CSMA/CD, a node will not send out a packet unless the network is clear of traffic. If two nodes send out packets at the same time, a collision occurs and the packets are lost. Then both nodes wait a random amount of time and retransmit the packets. Any part of the network where there is a possibility that packets from two or more nodes will interfere with each other is considered to be part of the same collision domain. A network with a large number of nodes on the same segment will often have a lot of collisions and therefore a large collision domain.

While hubs provide an easy way to scale up and shorten the distance that the packets must travel to get from one node to another, they do not break up the actual network into discrete segments. That is where switches come in.



Imagine that each vehicle is a packet of data waiting for an opportunity to continue on its trip.

Think of a hub as a four-way intersection where everyone has to stop. If more than one car reaches the intersection at the same time, they have to wait for their turn to proceed. Now imagine what this would be like with a dozen or even a hundred roads intersecting at a single point. The amount of waiting and the potential for a collision increases significantly. But wouldn't it be amazing if you could take an exit ramp from any one of those roads to the road of your choosing? That is exactly what a switch does for network traffic. A switch is like a cloverleaf intersection -- each car can take an exit ramp to get to its destination without having to stop and wait for other traffic to go by.

A vital difference between a hub and a switch is that all the nodes connected to a hub share the bandwidth among themselves, while a device connected to a switch port has

the **full bandwidth** all to itself. For example, if 10 nodes are communicating using a hub on a 10-Mbps network, then each node may only get a portion of the 10 Mbps if other nodes on the hub want to communicate as well. But with a switch, each node could possibly communicate at the full 10 Mbps. Think about our road analogy. If all of the traffic is coming to a common intersection, then each car it has to share that intersection with every other car. But a cloverleaf allows all of the traffic to continue at full speed from one road to the next.

In a fully switched network, switches replace all the hubs of an Ethernet network with a dedicated segment for every node. These segments connect to a switch, which supports multiple dedicated segments (sometimes in the hundreds). Since the only devices on each segment are the switch and the node, the switch picks up every transmission before it reaches another node. The switch then forwards the frame over the appropriate segment. Since any segment contains only a single node, the frame only reaches the intended recipient. This allows many conversations to occur simultaneously on a switched network.

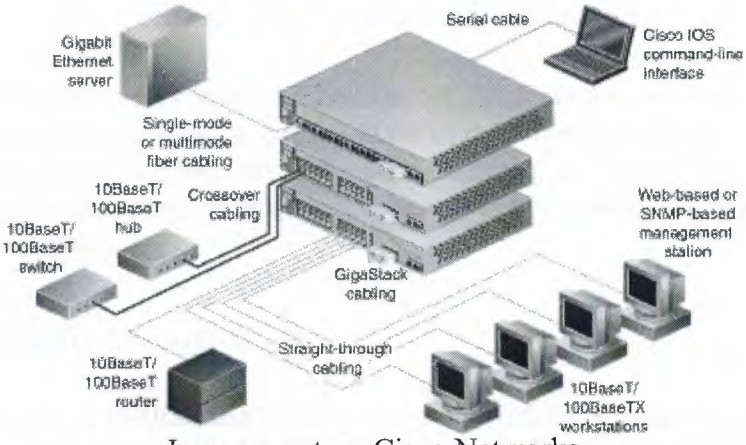


Image courtesy Cisco Networks

figure: An example of a network using a switch

Switching allows a network to maintain **full-duplex** Ethernet. Before switching, Ethernet was half-duplex, which means that data could be transmitted in only one direction at a time. In a fully switched network, each node communicates only with the switch, not directly with other nodes. Information can travel from node to switch and from switch to node simultaneously.

Fully switched networks employ either twisted-pair or fiber-optic cabling, both of which use separate conductors for sending and receiving data. In this type of environment, Ethernet nodes can forgo the collision detection process and transmit at will, since they are the only potential devices that can access the medium. In other words, traffic flowing in each direction has a lane to itself. This allows nodes to transmit to the switch as the switch transmits to them -- it's a collision-free environment.Transmitting in both directions can effectively double the apparent speed of the network when two nodes are exchanging information. If the speed of the network is 10 Mbps, then each node can transmit simultaneously at 10 Mbps.

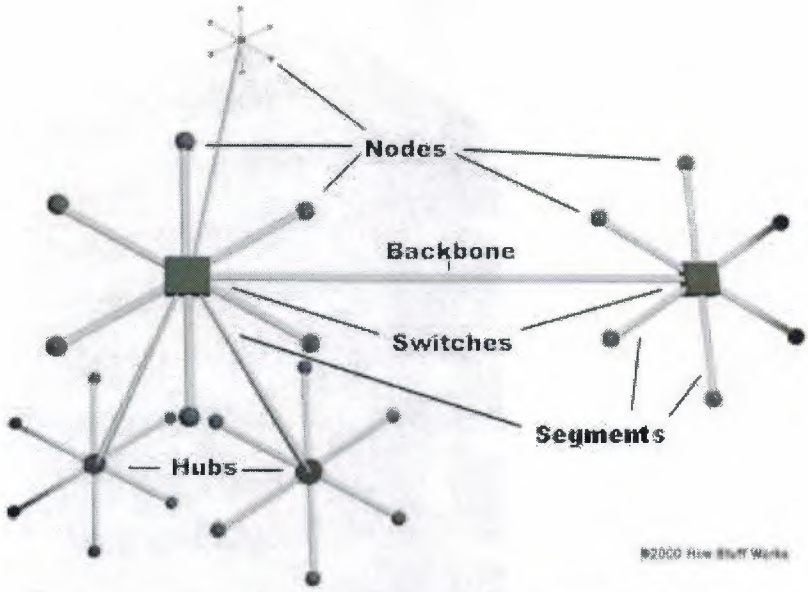


figure: mixed network with two switches and three hubs

Most networks are not fully switched because of the costs incurred in replacing all of the hubs with switches. Instead, a combination of switches and hubs are used to create an efficient yet cost-effective network. For example, a company may have hubs connecting the computers in each department and then a switch connecting all of the department-level hubs.

1.1.5SwitchingTechnologies

You can see that a switch has the potential to radically change the way nodes communicate with each other. But you may be wondering what makes it different from a router. Switches usually work at **Layer 2** (Data or Data link) of the OSI Reference Model, using MAC addresses, while routers work at Layer 3 (Network) with Layer 3 addresses (IP, IPX or AppleTalk, depending on which Layer 3 protocols are being used).The algorithm that switches use to decide how to forward packets is different from the algorithms used by routers to forward packets.

One of these differences in the algorithms between switches and routers is how **broadcasts** are handled. On any network, the concept of a broadcast packet is vital to the operability of a network. Whenever a device needs to send out information but doesn't know who it should send it to, it sends out a broadcast. For example, every time a new computer or other device comes on to the network, it sends out a broadcast packet to announce its presence. The other nodes (such as a domain server) can add the computer to their **browser list** (kind of like an address directory) and communicate directly with that computer from that point on. Broadcasts are used any time a device needs to make an announcement to the rest of the network or is unsure of who the recipient of the information should be.

figure : The OSI Reference Model consists of seven layers that build from the wire (Physical) to the software (Application).

A hub or a switch will pass along any broadcast packets they receive to all the other segments in the broadcast domain, but a router will not. Think about our four-way intersection again: All of the traffic passed through the intersection no matter where it was going. Now imagine that this intersection is at an international border. To pass through the intersection, you must provide the border guard with the specific address that you are going to. If you don't have a specific destination, then the guard will not let you pass. A router works like this. Without the specific address of another device, it will not let the data packet through. This is a good thing for keeping networks separate from each other, but not so good when you want to talk between different parts of the same network. This is where switches come in.

LAN switches rely on packet-switching. The switch establishes a connection between two segments just long enough to send the current packet. Incoming packets (part of an Ethernet frame) are saved to a temporary memory area (buffer); the MAC address contained in the frame's header is read and then compared to a list of addresses maintained in the switch's lookup table. In an Ethernet-based LAN, an Ethernet frame contains a normal packet as the payload of the frame, with a special header that includes the MAC address information for the source and destination of the packet.

Packet-based switches use one of three methods for routing traffic:

- **Cut-through**

- **Store-and-forward**
- **Fragment-free**

Cut-through switches read the MAC address as soon as a packet is detected by the switch. After storing the 6 bytes that make up the address information, they immediately begin sending the packet to the destination node, even as the rest of the packet is coming into the switch.

A switch using **store-and-forward** will save the entire packet to the buffer and check it for CRC errors or other problems before sending. If the packet has an error, it is discarded. Otherwise, the switch looks up the MAC address and sends the packet on to the destination node. Many switches combine the two methods, using cut-through until a certain error level is reached and then changing over to store-and-forward. Very few switches are strictly cut-through, since this provides no error correction.

A less common method is fragment-free. It works like cut-through except that it stores the first 64 bytes of the packet before sending it on. The reason for this is that most errors, and all collisions, occur during the initial 64 bytes of a packet.

LAN switches vary in their physical design. Currently, there are three popular configurations in use:

Shared memory - This type of switch stores all incoming packets in a common memory buffer shared by all the switch ports (input/output connections), then sends them out via the correct port for the destination node.

Matrix - This type of switch has an internal grid with the input ports and the output ports crossing each other. When a packet is detected on an input port, the MAC address is compared to the lookup table to find the appropriate output port. The switch then makes a connection on the grid where these two ports intersect.

Bus architecture - Instead of a grid, an internal transmission path (common bus) is shared by all of the ports using TDMA. A switch based on this configuration has a dedicated memory buffer for each port, as well as an ASIC to control the internal bus access.

1.1.6TheLayers

Think of the seven layers as the assembly line in the computer. At each layer, certain things happen to the data that prepare it for the next layer. The seven layers, which separate into two sets, are:

Application Set

Layer 7: Application - This is the layer that actually interacts with the operating system or application whenever the user chooses to transfer files, read messages or performs other network-related activities.

Layer 6: Presentation - Layer 6 takes the data provided by the Application layer and converts it into a standard format that the other layers can understand.

Layer 5: Session - Layer 5 establishes, maintains and ends communication with the receiving device.

Transport Set

Layer 4: Transport - This layer maintains flow control of data and provides for error checking and recovery of data between the devices. Flow control means that the Transport layer looks to see if data is coming from more than one application and integrates each application's data into a single stream for the physical network.

Layer 3: Network - The way that the data will be sent to the recipient device is determined in this layer. Logical protocols, routing and addressing are handled here.

Layer 2: Data - In this layer, the appropriate physical protocol is assigned to the data. Also, the type of network and the packet sequencing is defined.
Layer 1: Physical - This is the level of the actual hardware. It defines the physical characteristics of the network such as connections, voltage levels and timing.

figure :The seven layers of the OSI Reference Model

The OSI Reference Model is really just a guideline. Actual **protocol stacks** often combine one or more of the OSI layers into a single layer.

1.1.7 How Web Servers Work

Have you ever wondered about the mechanisms that delivered this page to you? Chances are you are sitting at a computer right now, viewing this page in a browser -- so when you clicked on the link for this page, or typed in its URL (uniform resource locator), what happened behind the scenes to bring this page onto your screen?
If you've ever been curious about the process, or have ever wanted to know some of the specific mechanisms that allow you to surf the Internet, then read on. In this edition of How Stuff Works, you will learn how Web servers bring pages into your home, school or office. Let's get started!
The Basic Process
let's say that you are sitting at your computer, surfing the Web, and you get a call from a friend who says, "I just read a great article! Type in this URL and check it out! It's at http://computer.howstuffworks.com/web-server.htm." So you type that URL into your browser and press return. And magically, no matter where in the world that URL lives, the page pops up on your screen!
At the most basic level possible, the following diagram shows the steps that brought that page to your screen:

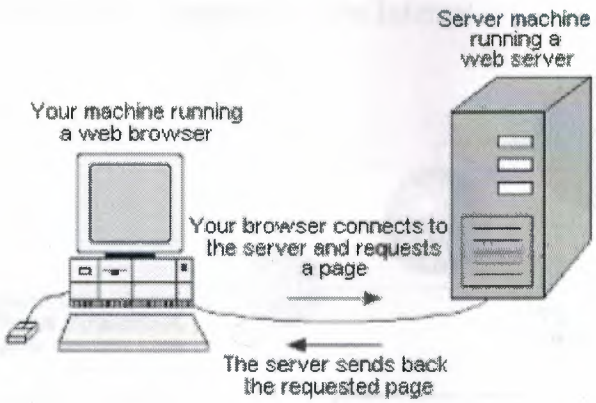


Figure: shows the steps that brought that page to your screen

Your browser formed a connection to a Web server, requested a page and received it. If you want to get into a bit more detail, here are the basic steps that occurred behind the scenes:

The browser broke the URL into three parts:

The protocol ("http")

The server name ("www.howstuffworks.com")

The file name ("web-server.htm")

The browser communicated with a name server to translate the server name "www.howstuffworks.com" into an IP Address, which it uses to connect to the server machine.

The browser then formed a connection to the server at that IP address on port 80. (We'll discuss ports later in this article.)

Following the HTTP protocol, the browser sent a GET request to the server, asking for the file "http://computer.howstuffworks.com/web-server.htm." (Note that **cookies** may be sent from browser to server with the GET request -- see How Internet Cookies Work for details.)

The server then sent the HTML text for the Web page to the browser. (Cookies may also be sent from server to browser in the header for the page.)

The browser read the HTML tags and formatted the page onto your screen.

If you've never explored this process before, that's a lot of new vocabulary. To understand this whole process in detail, you need to learn about IP addresses, ports, protocols... The following sections will lead you through a complete explanation.

The Internet

So what is "the Internet"? The Internet is a gigantic collection of millions of computers, all linked together on a **computer network**. The network allows all of the computers to communicate with one another. A home computer may be linked to the Internet using a phone-line modem, DSL or cable modem that talks to an Internet service provider (ISP). A computer in a business or university will usually have a network interface card (NIC) that directly connects it to a local area network (LAN) inside the business. The business can then connect its LAN to an ISP using a high-speed phone line like a T1 line. A T1 line can handle approximately 1.5 million bits per second, while a normal phone line using a modem can typically handle 30,000 to 50,000 bits per second.

ISPs then connect to larger ISPs, and the largest ISPs maintain fiber-optic "backbones" for an entire nation or region. Backbones around the world are connected through fiber-optic lines, undersea cables or satellite links (see this page for a nice

backbone and connection diagram). In this way, every computer on the Internet is connected to every other computer on the Internet.

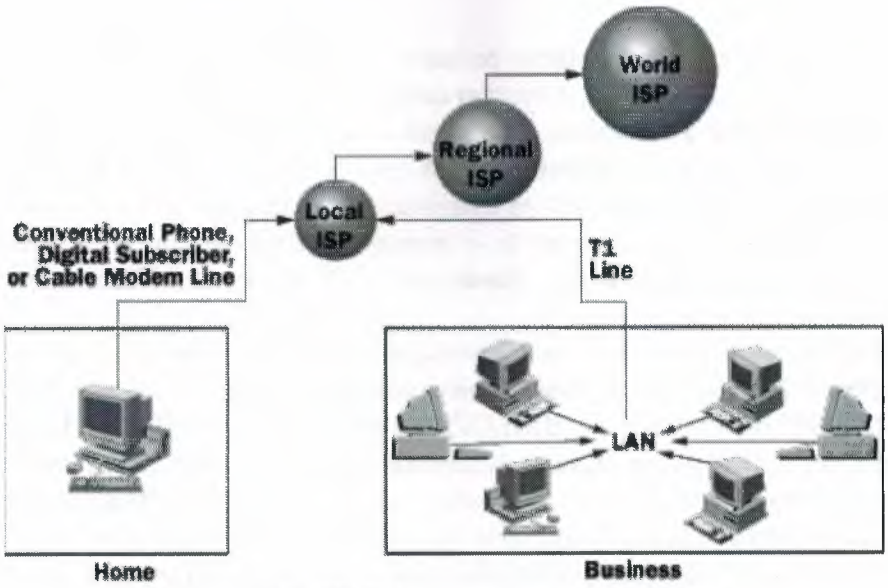


Figure: SHOW THE CONECTION OF LAN

1.1.8 Clients and Servers

In general, all of the machines on the Internet can be categorized as two types: servers and clients. Those machines that provide services (like Web servers or FTP servers) to other machines are **servers**. And the machines that are used to connect to those services are **clients**. When you connect to Yahoo! at www.yahoo.com to read a page, Yahoo! is providing a machine (probably a cluster of very large machines), for use on the Internet, to service your request. Yahoo! is providing a server. Your machine, on the other hand, is probably providing no services to anyone else on the Internet. Therefore, it is a user machine, also known as a client. It is possible and common for a machine to be both a server and a client, but for our purposes here you can think of most machines as one or the other.

A server machine may provide one or more services on the Internet. For example, a server machine might have software running on it that allows it to act as a Web server, an e-mail server and an FTP server. Clients that come to a server machine do so with a specific intent, so clients direct their requests to a specific software server running on the overall server machine. For example, if you are running a Web browser on your machine, it will most likely want to talk to the Web server on the server machine. Your Telnet application will want to talk to the Telnet server, your e-mail application will talk to the e-mail server, and so on...

1.1.9IP Addresses

To keep all of these machines straight, each machine on the Internet is assigned a unique address called an **IP address**. IP stands for **Internet protocol**, and these

addresses are 32-bit numbers, normally expressed as four "octets" in a "dotted decimal number." A typical IP address looks like this:

216.27.61.137

The four numbers in an IP address are called **octets** because they can have values between 0 and 255, which is 2^8 possibilities per octet.

Every machine on the Internet has a unique IP address. A server has a static IP address that does not change very often. A home machine that is dialing up through a modem often has an IP address that is assigned by the ISP when the machine dials in. That IP address is unique for that session -- it may be different the next time the machine dials in. This way, an ISP only needs one IP address for each modem it supports, rather than for each customer.

If you are working on a Windows machine, you can view a lot of the Internet information for your machine, including your current IP address and hostname, with the command WINIPCFG.EXE (IPCONFIG.EXE for Windows 2000/XP). On a UNIX machine, type nslookup at the command prompt, along with a machine name, like www.howstuffworks.com -- e.g. "nslookup www.howstuffworks.com" -- to display the IP address of the machine, and you can use the command hostname to learn the name of your machine. (For more information on IP addresses, see IANA.)

As far as the Internet's machines are concerned, an IP address is all you need to talk to a server. For example, in your browser, you can type the URL http://209.116.69.66 and arrive at the machine that contains the Web server for HowStuffWorks. On some servers, the IP address alone is not sufficient, but on most large servers it is -- keep reading for details.

Name Servers

Because most people have trouble remembering the strings of numbers that make up IP addresses, and because IP addresses sometimes need to change, all servers on the Internet also have human-readable names, called **domain names**. For example, www.howstuffworks.com is a permanent, human-readable name. It is easier for most of us to remember www.howstuffworks.com than it is to remember 209.116.69.66.

The name www.howstuffworks.com actually has three parts:

The host name ("www")

The domain name ("howstuffworks")

The top-level domain name ("com")

Domain names are managed by a company called VeriSign. VeriSign creates the top-level domain names and guarantees that all names within a top-level domain are unique. VeriSign also maintains contact information for each site and runs the "whois" database. The host name is created by the company hosting the domain. "www" is a very common host name, but many places now either omit it or replace it with a different host name that indicates a specific area of the site. For example, in encarta.msn.com, the domain name for Microsoft's Encarta encyclopedia, "encarta" is designated as the host name instead of "www."

A set of servers called domain name servers (DNS) maps the human-readable names to the IP addresses. These servers are simple databases that map names to IP addresses, and they are distributed all over the Internet. Most individual companies, ISPs and universities maintain small name servers to map host names to IP addresses. There are also central name servers that use data supplied by VeriSign to map domain names to IP addresses.

If you type the URL "http://computer.howstuffworks.com/web-server.htm" into your browser, your browser extracts the name "www.howstuffworks.com," passes it to a

domain name server, and the domain name server returns the correct IP address for www.howstuffworks.com. A number of name servers may be involved to get the right IP address. For example, in the case of www.howstuffworks.com, the name server for the "com" top-level domain will know the IP address for the name server that knows host names, and a separate query to that name server, operated by the HowStuffWorks ISP, may deliver the actual IP address for the HowStuffWorks server machine.
On a UNIX machine, you can access the same service using the nslookup command. Simply type a name like "www.howstuffworks.com" into the command line, and the command will query the name servers and deliver the corresponding IP address to you.
So here it is: The Internet is made up of millions of machines, each with a unique IP address. Many of these machines are server machines, meaning that they provide services to other machines on the Internet. You have heard of many of these servers: e-mail servers, Web servers, FTP servers, Gopher servers and Telnet servers, to name a few. All of these are provided by server machines.
Ports
Any server machine makes its services available to the Internet using numbered **ports**, one for each service that is available on the server. For example, if a server machine is running a Web server and an FTP server, the Web server would typically be available on port 80, and the FTP server would be available on port 21. Clients connect to a service at a specific IP address and on a specific port.
Each of the most well-known services is available at a well-known port number. Here are some common port numbers:
echo 7
Daytime 13
qotd 17 (Quote of the Day)
Ftp 21
Telnet 23
SMTP 25 (Simple Mail Transfer, meaning e-mail)
Time 37
Name server 42
Nickname 43 (Who Is)
Gopher 70
Finger 79
WWW 80
If the server machine accepts connections on a port from the outside world, and if a firewall is not protecting the port, you can connect to the port from anywhere on the Internet and use the service. Note that there is nothing that forces, for example, a Web server to be on port 80. If you were to set up your own machine and load Web server software on it, you could put the Web server on port 918, or any other unused port, if you wanted to. Then, if your machine were known as xxx.yyy.com, someone on the Internet could connect to your server with the URL **http://xxx.yyy.com:918**. The ":918" explicitly specifies the port number, and would have to be included for someone to reach your server. When no port is specified, the browser simply assumes that the server is using the well-known port 80.

Protocols

Once a client has connected to a service on a particular port, it accesses the service using a specific protocol. The protocol is the pre-defined way that someone who wants to use a service talks with that service. The "someone" could be a person, but

more often it is a computer program like a Web browser. Protocols are often text, and simply describe how the client and server will have their conversation.
Perhaps the simplest protocol is the daytime protocol. If you connect to port 13 on a machine that supports a daytime server, the server will send you its impression of the current date and time and then close the connection. The protocol is, "If you connect to me, I will send you the date and time and then disconnect." Most UNIX machines support this server. If you would like to try it out, you can connect to one with the Telnet application. In UNIX, the session would look like this:

```
%telnet web67.ntx.net 13
Trying 216.27.61.137...
Connected to web67.ntx.net.
Escape character is '^]'.
Sun Oct 25 08:34:06 1998
Connection closed by foreign host.
```

On a Windows machine, you can access this server by typing "telnet web67.ntx.net 13" at the MSDOS prompt.
In this example, web67.ntx.net is the server's UNIX machine, and 13 is the port number for the daytime service. The Telnet application connects to port 13 (telnet naturally connects to port 23, but you can direct it to connect to any port), then the server sends the date and time and disconnects. Most versions of Telnet allow you to specify a port number, so you can try this using whatever version of Telnet you have available on your machine.
Most protocols are more involved than daytime and are specified in Request for Comment (RFC) documents that are publicly available. Every Web server on the Internet conforms to the HTTP protocol, summarized nicely in this article. The most basic form of the protocol understood by an HTTP server involves just one command: GET. If you connect to a server that understands the HTTP protocol and tell it to "GET filename," the server will respond by sending you the contents of the named file and then disconnecting. Here's a typical session:

```
%telnet www.howstuffworks.com 80
Trying 216.27.61.137...
Connected to howstuffworks.com.
Escape character is '^]'.
GET http://computer.howstuffworks.com/
<html>
<head>
<title>Welcome to How Stuff Works</title>
 ...
</body>
</html>
Connection closed by foreign host.
```

In the original HTTP protocol, all you would have sent was the actual filename, such as "/" or "/web-server.htm." The protocol was later modified to handle the sending of the complete URL. This has allowed companies that host virtual domains, where many domains live on a single machine, to use one IP address for all of the domains they host. It turns out that hundreds of domains are hosted on 209.116.69.66 – the

Putting It Altogether

Now you know a tremendous amount about the Internet. You know that when you type a URL into a browser, the following steps occur:
The browser breaks the URL into three parts:
The protocol ("http")
The server name ("www.howstuffworks.com")
The file name ("web-server.htm")
The browser communicates with a name server to translate the server name, "www.howstuffworks.com," into an IP address, which it uses to connect to that server machine.
The browser then forms a connection to the Web server at that IP address on port 80. Following the HTTP protocol, the browser sends a GET request to the server, asking for the file "http://computer.howstuffworks.com/web-server.htm." (Note that cookies may be sent from browser to server with the GET request)The server sends the HTML text for the Web page to the browser. (Cookies may also be sent from server to browser in the header for the page.) The browser reads the HTML tags and formats the page onto your screen

1.2 **INTRODUCTION TO ROUTING**

1.2.1 What is Routing?

Routing is the act of moving information across an internetwork from a source to a destination. Along the way, at least one intermediate node typically is encountered. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer. The primary difference between the two is that bridging occurs at Layer 2 (the link layer) of the OSI reference model, whereas routing occurs at Layer 3 (the network layer). This distinction provides routing and bridging with different information to use in the process of moving information from source to destination, so the two functions accomplish their tasks in different ways.
The topic of routing has been covered in computer science literature for more than two decades, but routing achieved commercial popularity as late as the mid-1980s. The primary reason for this time lag is that networks in the 1970s were fairly simple, homogeneous environments. Only relatively recently has large-scale internetworking become popular.

1.2.2 Routing Components

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

1.3 Path Determination

Path determination, for traffic going through a network cloud, occurs at the network layer (Layer 3). The path determination function enables a router to evaluate the available paths to a destination and to establish the preferred handling of a packet. Routing services use network topology information when evaluating network paths. This information can be configured by the network administrator or collected through dynamic processes running in the network.

The network layer provides best-effort end-to-end packet delivery across interconnected networks. The network layer uses the IP routing table to send packets from the source network to the destination network. After the router determines which path to use, it proceeds with forwarding the packet. It takes the packet that it accepted on one interface and forwards it to another interface or port that reflects the best path to the packet's destination.

A metric is a standard of measurement, such as path length, that is used by routing algorithms to determine the optimal path to a destination. To aid the process of path determination, routing algorithms initialize and maintain routing tables, which contain route information. Route information varies depending on the routing algorithm used. Routing algorithms fill routing tables with a variety of information. Destination/next hop associations tell a router that a particular destination can be gained optimally by sending the packet to a particular router representing the "next hop" on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with a next hop. Figure 1-1 depicts a sample destination/next hop routing table.

To reach network:	Send to:
27	Node A
57	Node B
17	Node C
24	Node A
52	Node A
16	Node B
26	Node A
.	.
.	.
.	.

Figure 1-1: Destination/next hop associations determine the data's optimal path.

Routing tables also can contain other information, such as data about the desirability of a path. Routers compare metrics to determine optimal routes, and these metrics differ depending on the design of the routing algorithm used. A variety of common metrics will be introduced and described later in this chapter.
Routers communicate with one another and maintain their routing tables through the transmission of a variety of messages. The routing update message is one such message that generally consists of all or a portion of a routing table. By analyzing routing updates from all other routers, a router can build a detailed picture of network topology. A link-state advertisement, another example of a message sent between routers, informs other routers of the state of the sender's links. Link information also can be used to build a complete picture of topology to enable routers to determine optimal routes to network destinations.

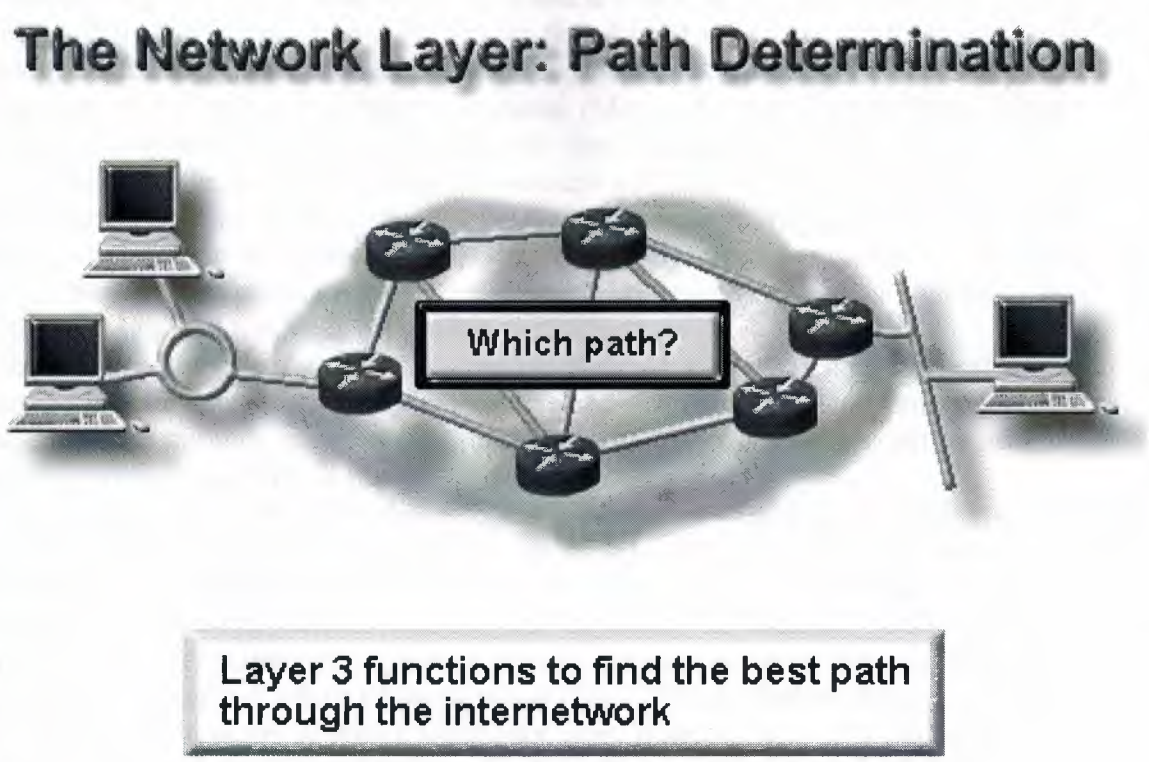


Figure 1-2: Path determination

1.4 How routers route packets from source to destination

To be truly practical, a network must consistently represent the paths available between routers. As Figure 1-2 shows, each line between the routers has a number that the routers use as a network address. These addresses must convey information that can be used by a routing process to pass packets from a source toward a destination. Using these addresses, the network layer can provide a relay connection that interconnects independent networks.

The consistency of Layer 3 addresses across the entire internetwork also improves the use of bandwidth by preventing unnecessary broadcasts. Broadcasts invoke unnecessary process overhead and waste capacity on any devices or links that do not need to receive the broadcasts. By using consistent end-to-end addressing to represent the path of media connections, the network layer can find a path to the destination without unnecessarily burdening the devices or links on the internetwork with broadcasts.

1.5 Switching

Switching algorithms are relatively simple and are basically the same for most routing protocols. In most cases, a host determines that it must send a packet to another host. Having acquired a router's address by some means, the source host sends a packet addressed specifically to a router's physical (Media Access Control [MAC]-layer) address, this time with the protocol (network- layer) address of the destination host.
As it examines the packet's destination protocol address, the router determines that it either knows or does not know how to forward the packet to the next hop. If the router does not know how to forward the packet, it typically drops the packet. If the router knows how to forward the packet, it changes the destination physical address to that of the next hop and transmits the packet.
The next hop may, in fact, be the ultimate destination host. If not, the next hop is usually another router, which executes the same switching decision process. As the packet moves through the internetwork, its physical address changes, but its protocol address remains constant, as illustrated in Figure 5-3.
The preceding discussion describes switching between a source and a destination end system. The International Organization for Standardization (ISO) has developed a hierarchical terminology that is useful in describing this process. Using this terminology, network devices without the capability to forward packets between subnetworks are called end systems (ESs), whereas network devices with these capabilities are called intermediate systems (ISs). ISs are further divided into those that can communicate within routing domains (intradomain ISs) and those that communicate both within and between routing domains (interdomain ISs). A routing domain generally is considered to be a portion of an internetwork under common administrative authority that is regulated by a particular set of administrative guidelines. Routing domains are also called autonomous systems. With certain protocols, routing domains can be divided into routing areas, but intradomain routing protocols are still used for switching both within and between areas.
A router generally relays a packet from one data link to another, using two basic functions:

- a path determination function
- a switching function.

The router uses addressing for these routing and switching functions. The router uses the network portion of the address to make path selections to pass the packet to the next router along the path.
The switching function allows a router to accept a packet on one interface and forward it through a second interface. The path determination function enables the router to select the most appropriate interface for forwarding a packet. The node portion of the address is used by the final router (the router connected to the destination network) to deliver the packet to the correct host.

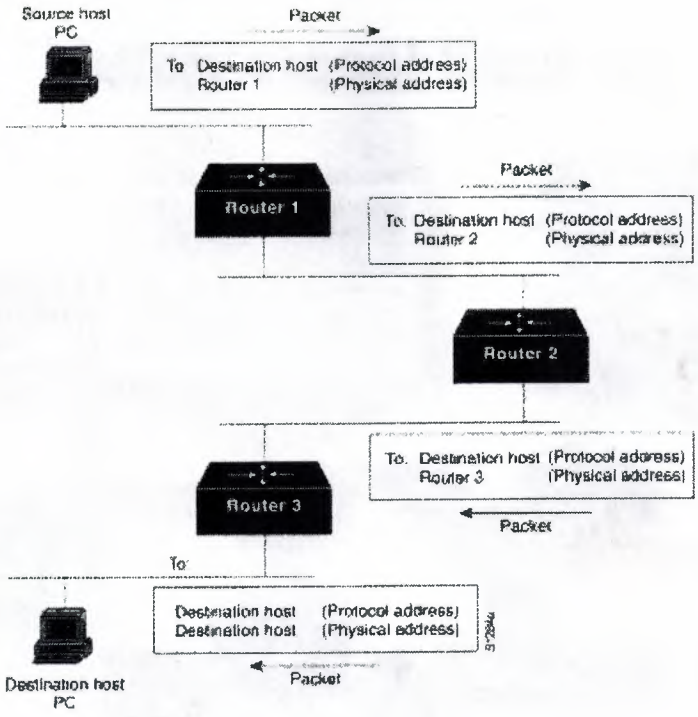


Figure 5-3: Numerous routers may come into play during the switching process.

1.6 Routed versus routing protocol

Because of the similarity of the two terms, confusion often exists with routed protocol and routing protocol.

Routed protocol is any network protocol that provides enough information in its network layer address to allow a packet to be forwarded from one host to another host based on the addressing scheme. Routed protocols define the field formats within a packet. Packets are generally conveyed from end system to end system. The Internet Protocol (IP) is an example of a routed protocol.

Routing protocols support a routed protocol by providing mechanisms for sharing routing information. Routing protocol messages move between the routers. A routing protocol allows the routers to communicate with other routers to update and maintain tables. TCP/IP examples of routing protocols are:

- RIP (Routing Information Protocol)
- IGRP (Interior Gateway Routing Protocol)
- EIGRP (Enhanced Interior Gateway Routing Protocol)
- OSPF (Open Shortest Path First)

1.7 Multiprotocol routing

Routers are capable of supporting multiple independent routing protocols and maintaining routing tables for several routed protocols. This capability allows a router to deliver packets from several routed protocols over the same data links.

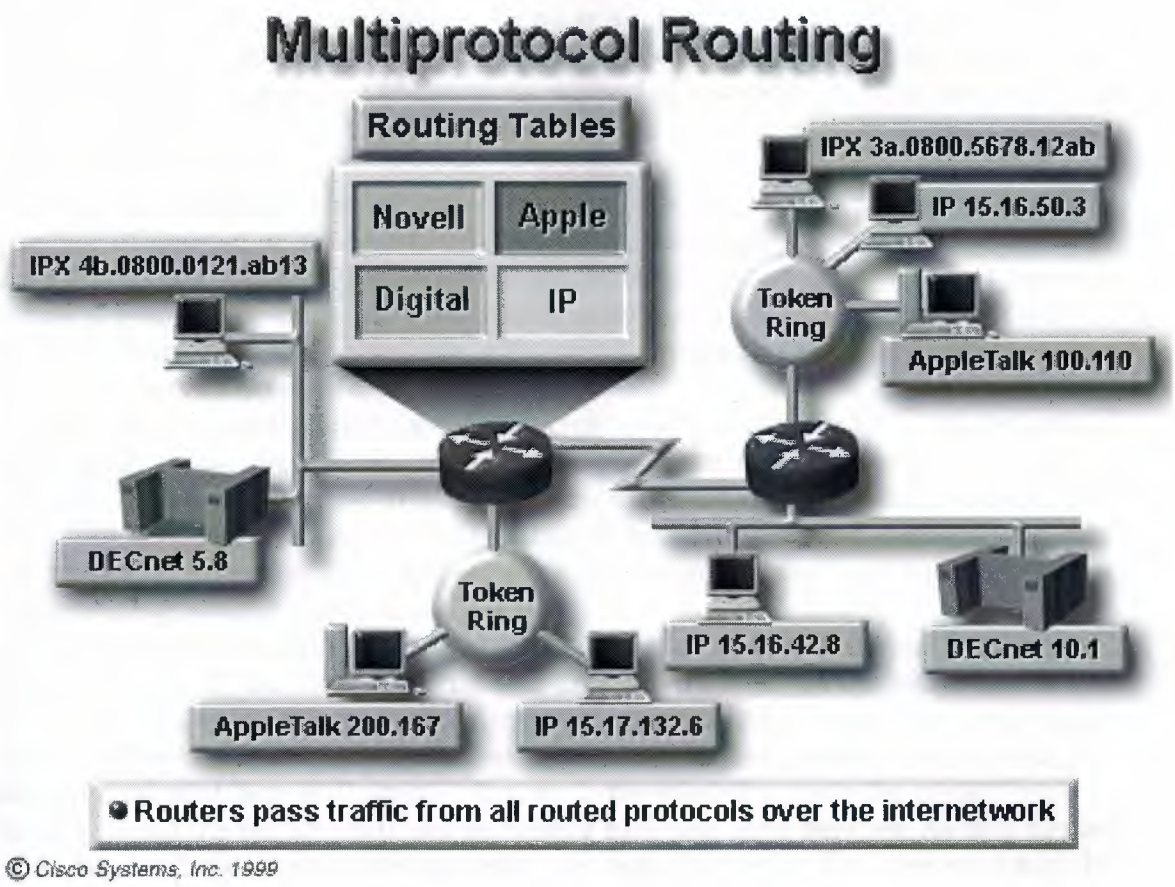


Figure: multi protocol routing

1.8 Routing Algorithms

Routing algorithms can be differentiated based on several key characteristics. First, the particular goals of the algorithm designer affect the operation of the resulting routing protocol. Second, various types of routing algorithms exist, and each algorithm has a different impact on network and router resources. Finally, routing algorithms use a variety of metrics that affect calculation of optimal routes. The following sections analyze these routing algorithm attributes.

1.9 Design Goals

Routing algorithms often have one or more of the following design goals:

- Optimality
- Simplicity and low overhead
- Robustness and stability
- Rapid convergence
- Flexibility

Optimality refers to the capability of the routing algorithm to select the best route, which depends on the metrics and metric weightings used to make the calculation. One routing algorithm, for example, may use a number of hops and delays, but may weight delay more heavily in the calculation. Naturally, routing protocols must define their metric calculation algorithms strictly.

Routing algorithms also are designed to be as simple as possible. In other words, the routing algorithm must offer its functionality efficiently, with a minimum of software and utilization overhead. Efficiency is particularly important when the software implementing the routing algorithm must run on a computer with limited physical resources.

Routing algorithms must be robust, which means that they should perform correctly in the face of unusual or unforeseen circumstances, such as hardware failures, high load conditions, and incorrect implementations. Because routers are located at network junction points, they can cause considerable problems when they fail. The best routing algorithms are often those that have withstood the test of time and have proven stable under a variety of network conditions.

In addition, routing algorithms must converge rapidly. Convergence is the process of agreement, by all routers, on optimal routes. When a network event causes routes either to go down or become available, routers distribute routing update messages that permeate networks, stimulating recalculation of optimal routes and eventually causing all routers to agree on these routes. Routing algorithms that converge slowly can cause routing loops or network outages.

In the routing loop displayed in Figure 5-4, a packet arrives at Router 1 at time t1. Router 1 already has been updated and thus knows that the optimal route to the destination calls for Router 2 to be the next stop. Router 1 therefore forwards the packet to Router 2, but because this router has not yet been updated, it believes that the optimal next hop is Router 1. Router 2 therefore forwards the packet back to Router 1, and the packet continues to bounce back and forth between the two routers until Router 2 receives its routing update or until the packet has been switched the maximum number of times allowed.

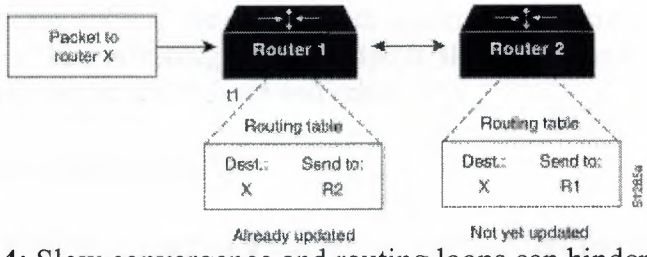


Figure 5-4: Slow convergence and routing loops can hinder progress.

Routing algorithms should also be flexible, which means that they should quickly and accurately adapt to a variety of network circumstances. Assume, for example, that a network segment has gone down. As they become aware of the problem, many routing algorithms will quickly select the next-best path for all routes normally using that segment. Routing algorithms can be programmed to adapt to changes in network bandwidth, router queue size, and network delay, among other variables.

1.10 Algorithm Types

Routing algorithms can be classified by type. Key differentiators include:

- Static versus dynamic
- Single-path versus multi-path
- Flat versus hierarchical
- Host-intelligent versus router-intelligent

- Intradomain versus interdomain
- Link state versus distance vector

1.10.1 Static Versus Dynamic

Static routing algorithms are hardly algorithms at all, but are table mappings established by the network administrator prior to the beginning of routing. These mappings do not change unless the network administrator alters them. Algorithms that use static routes are simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple.
Because static routing systems cannot react to network changes, they generally are considered unsuitable for today's large, changing networks. Most of the dominant routing algorithms in the 1990s are dynamic routing algorithms, which adjust to changing network circumstances by analyzing incoming routing update messages. If the message indicates that a network change has occurred, the routing software recalculates routes and sends out new routing update messages. These messages permeate the network, stimulating routers to rerun their algorithms and change their routing tables accordingly.
Dynamic routing algorithms can be supplemented with static routes where appropriate. A router of last resort (a router to which all unroutable packets are sent), for example, can be designated to act as a repository for all unroutable packets, ensuring that all messages are at least handled in some way.

1.10.2 Single-Path Versus Multipath

Some sophisticated routing protocols support multiple paths to the same destination. Unlike single-path algorithms, these multipath algorithms permit traffic multiplexing over multiple lines. The advantages of multipath algorithms are obvious: They can provide substantially better throughput and reliability.

1.10.3 Flat Versus Hierarchical

Some routing algorithms operate in a flat space, while others use routing hierarchies. In a flat routing system, the routers are peers of all others. In a hierarchical routing system, some routers form what amounts to a routing backbone. Packets from non-backbone routers travel to the backbone routers, where they are sent through the backbone until they reach the general area of the destination. At this point, they travel from the last backbone router through one or more non-backbone routers to the final destination.
Routing systems often designate logical groups of nodes, called domains, autonomous systems, or areas. In hierarchical systems, some routers in a domain can communicate with routers in other domains, while others can communicate only with routers within their domain. In very large networks, additional hierarchical levels may exist, with routers at the highest hierarchical level forming the routing backbone.
The primary advantage of hierarchical routing is that it mimics the organization of most companies and therefore supports their traffic patterns well. Most network communication occurs within small company groups (domains). Because intradomain routers need to know only about other routers within their domain, their routing

algorithms can be simplified, and, depending on the routing algorithm being used, routing update traffic can be reduced accordingly.

1.10.4 Host-Intelligent Versus Router-Intelligent

Some routing algorithms assume that the source end-node will determine the entire route. This is usually referred to as source routing. In source-routing systems, routers merely act as store-and-forward devices, mindlessly sending the packet to the next stop.
Other algorithms assume that hosts know nothing about routes. In these algorithms, routers determine the path through the internetwork based on their own calculations. In the first system, the hosts have the routing intelligence. In the latter system, routers have the routing intelligence.
The trade-off between host-intelligent and router-intelligent routing is one of path optimality versus traffic overhead. Host-intelligent systems choose the better routes more often, because they typically discover all possible routes to the destination before the packet is actually sent. They then choose the best path based on that particular system's definition of "optimal." The act of determining all routes, however, often requires substantial discovery traffic and a significant amount of time.

1.10.5 Intradomain versus Interdomain

Some routing algorithms work only within domains; others work within and between domains. The nature of these two algorithm types is different. It stands to reason, therefore, that an optimal intradomain- routing algorithm would not necessarily be an optimal interdomain- routing algorithm.

1.10.6 Link State Versus Distance Vector

Link- state algorithms (also known as shortest path first algorithms) flood routing information to all nodes in the internetwork. Each router, however, sends only the portion of the routing table that describes the state of its own links. Distance- vector algorithms (also known as Bellman-Ford algorithms) call for each router to send all or some portion of its routing table, but only to its neighbors. In essence, link- state algorithms send small updates everywhere, while distance- vector algorithms send larger updates only to neighboring routers.
Because they converge more quickly, link- state algorithms are somewhat less prone to routing loops than distance- vector algorithms. On the other hand, link- state algorithms require more CPU power and memory than distance- vector algorithms. Link-state algorithms, therefore, can be more expensive to implement and support. Despite their differences, both algorithm types perform well in most circumstances.

1.11 Routing Metrics

Routing tables contain information used by switching software to select the best route. But how, specifically, are routing tables built? What is the specific nature of the information they contain? How do routing algorithms determine that one route is preferable to others?

Routing algorithms have used many different metrics to determine the best route. Sophisticated routing algorithms can base route selection on multiple metrics, combining them in a single (hybrid) metric. All the following metrics have been used:

- Path Length
- Reliability
- Delay
- Bandwidth
- Load
- Communication Cost

Path length is the most common routing metric. Some routing protocols allow network administrators to assign arbitrary costs to each network link. In this case, path length is the sum of the costs associated with each link traversed. Other routing protocols define hop count, a metric that specifies the number of passes through internetworking products, such as routers, that a packet must take en route from a source to a destination.

Reliability, in the context of routing algorithms, refers to the dependability (usually described in terms of the bit-error rate) of each network link. Some network links might go down more often than others. After a network fails, certain network links might be repaired more easily or more quickly than other links. Any reliability factors can be taken into account in the assignment of the reliability ratings, which are arbitrary numeric values usually assigned to network links by network administrators.

Routing delay refers to the length of time required to move a packet from source to destination through the internetwork. Delay depends on many factors, including the bandwidth of intermediate network links, the port queues at each router along the way, network congestion on all intermediate network links, and the physical distance to be travelled. Because delay is a conglomeration of several important variables, it is a common and useful metric.

Bandwidth refers to the available traffic capacity of a link. All other things being equal, a 10-Mbps Ethernet link would be preferable to a 64-kbps leased line. Although bandwidth is a rating of the maximum attainable throughput on a link, routes through links with greater bandwidth do not necessarily provide better routes than routes through slower links. If, for example, a faster link is busier, the actual time required to send a packet to the destination could be greater.

Load refers to the degree to which a network resource, such as a router, is busy. Load can be calculated in a variety of ways, including CPU utilization and packets processed per second. Monitoring these parameters on a continual basis can be resource-intensive itself.

Communication cost is another important metric, especially because some companies may not care about performance as much as they care about operating expenditures. Even though line delay may be longer, they will send packets over their own lines rather than through the public lines that cost money for usage time.+

1.12 Network Protocols

Routed protocols are transported by routing protocols across an internetwork. In general, routed protocols in this context also are referred to as network protocols.
These network protocols perform a variety of functions required for communication between user applications in source and destination devices, and these functions can differ widely among protocol suites. Network protocols occur at the upper four layers of the OSI reference model: the transport layer, the session layer, the presentation layer, and the application layer.
Confusion about the terms routed protocol and routing protocol is common. Routed protocols are protocols that are routed over an internetwork. Examples of such protocols are the Internet Protocol (IP), DECnet, AppleTalk, Novell NetWare, OSI, Banyan VINES, and Xerox Network System (XNS). Routing protocols, on the other hand, are protocols that implement routing algorithms. Put simply, routing protocols direct protocols through an internetwork. Examples of these protocols include Interior Gateway Routing Protocol (IGRP), Enhanced Interior Gateway Routing Protocol (Enhanced IGRP), Open Shortest Path First (OSPF), Exterior Gateway Protocol (EGP), Border Gateway Protocol (BGP), Intermediate System to Intermediate System (IS-IS), and Routing Information Protocol (RIP). Routed and routing protocols are discussed in detail later.

Routed protocol is any network protocol that provides enough information in its network layer address to allow a packet to be forwarded from one host to another host based on the addressing scheme user information.
Routed protocols define the field formats and use within a packet. Packets are generally conveyed from end system to end system. The Internet Protocol (IP)
Routing protocol supports a routed protocol by providing mechanisms for sharing routing information. Routing protocol messages move between the routers. A routing protocol allows the routers to communicate with other routers to update and maintain tables. TCP/IP examples of routing protocols are:

- RIP (Routing Information Protocol)
- IGRP (Interior Gateway Routing Protocol)
- EIGRP (Enhanced Interior Gateway Routing Protocol)
- OSPF (Open Shortest Path First)

1.13 INITIAL ROUTER CONFIGURATION

1.13.1 Setup mode

After testing the hardware and loading the Cisco IOS system image, the router finds and applies the configuration statements. These entries provide the router with details about router-specific attributes, protocol functions, and interface addresses. However, if the router is unable to locate a valid startup-config file, it enters an initial router configuration mode called setup mode.
With the setup mode command facility, you can answer questions in the system configuration dialog. This facility prompts you for basic configuration information. The answers you enter allow the router to use a sufficient, but minimal-feature, router configuration that includes the following:

- an inventory of interfaces

- an opportunity to enter global parameters
- an opportunity to enter interface parameters
- a setup script review
- an opportunity to indicate whether you want the router to use this configuration

After you approve setup mode entries, the router uses the entries as a running configuration. The router also stores the configuration in NVRAM as a new startup-config, and you can start using the router. For additional protocol and interface changes, you can use the enable mode and enter the command configure.

1.13.2 Initial IP routing table

Initially, a router must refer to entries about networks or subnets that are directly connected to it. Each interface must be configured with an IP address and a mask. The Cisco IOS software learns about this IP address and mask information from a configuration that has been input from some source. The initial source of addressing is a user who types it into a configuration file.

In the lab that follows, you will start up your router in a just-received condition, a state that lacks another source for the startup configuration. This condition on the router will permit you to use the setup-mode command facility and answer prompts for basic configuration information. The answers you enter will include address-to-port commands to set up router interfaces for IP.

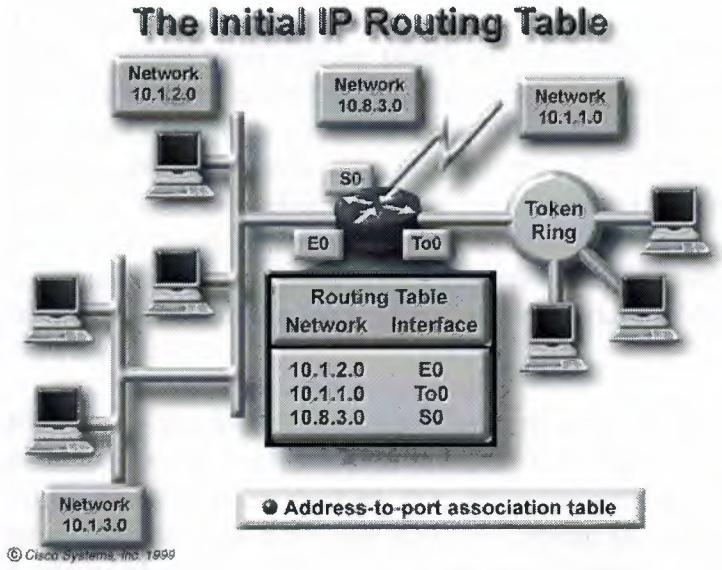


Figure: initial IP routing table

So how a router learns about destinations? By default, routers learn paths to destinations three different ways

- static routes-manually defined by the system administrator as the next hop to a destination; useful for security and traffic reduction
- default routes-manually defined by the system administrator as the path to take when there is no known route to the destination
- dynamic routing-the router learns of paths to destinations by receiving periodic updates from other routers.

1.13.3 The IP route command

The Ip route command sets up a static route.
The administrative distance is a rating of the trustworthiness of a routing information source, expressed as a numeric value from 0 to 255. The higher the number, the lower the trustworthiness rating.
A static route allows manual configuration of the routing table. No dynamic changes to this table entry will occur as long as the path is active. A static route may reflect some special knowledge of the networking situation known to the network administrator. Manually-entered administrative distance values for static routes are usually low numbers (1 is the default). Routing updates are not sent on a link if they are only defined by a static route, therefore, they conserve bandwidth.

Using the ip route command , The assignment of a static route to reach the stub network 172.16.1.0 is proper for Cisco A because there is only one way to reach that network. The assignment of a static route from Cisco B to the cloud networks is also possible. However, a static route assignment is required for each destination network, in which case a default route may be more appropriate.

1.13.4 IP default-network command

The ip default-network command establishes a default route in networks using dynamic routing protocols..
Default routes keep routing tables shorter. When an entry for a destination network does not exist in a routing table, the packet is sent to the default network. Because a router does not have complete knowledge about all destination networks, it can use a default network number to indicate the direction to take for unknown network numbers. Use the default network number when you need to locate a route but have only partial information about the destination network. The ip default-network command must be added to all routers in the network or used with the additional command redistribute static so all networks have knowledge of the candidate default network

How using the ip default-network command? In the example, the global command ip default-network 192.168.17.0 defines the Class C network 192.168.17.0 as the destination path for packets that have no routing table entries. The Company X administrator does not want updates coming in from the public network. Router A could need a firewall for routing updates. Router A may need a mechanism to group those networks that will share Company X's routing strategy. One such mechanism is an autonomous system number.

CHAPTER TWO

ROUTING PROTOCOLS AND CONTEXT

2.1 Why **ROUTING PROTOCOLS are necessary?**

2.1.1 Static versus dynamic routes

Static route knowledge is administered manually by a network administrator who enters it into a router's configuration. The administrator must manually update this static route entry whenever an internetwork topology change requires an update.
Dynamic route knowledge works differently. After a network administrator enters configuration commands to start dynamic routing, the route knowledge is automatically updated by a routing process whenever new information is received from the internetwork. Changes in dynamic knowledge are exchanged between routers as part of the update process.

2.1.2 Why use a static route

Static routing has several useful applications. Dynamic routing tends to reveal everything known about an internetwork, for security reasons, you may want to hide parts of an internetwork. Static routing enables you to specify the information you want to reveal about restricted networks.
When a network is accessible by only one path, a static route to the network can be sufficient. This type of network is called a stub network. Configuring static routing to a stub network avoids the overhead of dynamic routing.

2.1.3 How a default route is used

The Figure shows a use for a default route - a routing table entry that directs packets to the next hop when that hop is not explicitly listed in the routing table. You can set default routes as part of the static configuration.
In this example, the company X routers possess specific knowledge of the topology of the company X network, but not of other networks. Maintaining knowledge of every other network accessible by way of the Internet cloud is unnecessary and unreasonable, if not impossible. Instead of maintaining specific network knowledge, each router in company X is informed of the default route that it can use to reach any unknown destination by directing the packet to the Internet.

2.1.4 Why dynamic routing is necessary

The network shown in the Figure adapts differently to topology changes depending on whether it uses statically or dynamically configured routing information.
Static routing allows routers to properly route a packet from network to network based on configured information. The router refers to its routing table and follows the static knowledge residing there to relay the packet to Router D. Router D does the

same, and relays the packet to Router C. Router C delivers the packet to the destination host.
If the path between Router A and Router D fails, Router A will not be able to relay the packet to Router D using that static route. Until Router A is manually reconfigured to relay packets by way of Router B, communication with the destination network is impossible.
Dynamic routing offers more flexibility. According to the routing table generated by Router A, a packet can reach its destination over the preferred route through Router D. However, a second path to the destination is available by way of Router B. When Router A recognizes that the link to Router D is down, it adjusts its routing table, making the path through Router B the preferred path to the destination. The routers continue sending packets over this link.
When the path between Routers A and D is restored to service, Router A can once again change its routing table to indicate a preference for the counterclockwise path through Routers D and C to the destination network. Dynamic routing protocols can also direct traffic from the same session over different paths in a network for better performance. This is known as loadsharing.

2.1.5 Dynamic routing operations

The success of dynamic routing depends on two basic router functions:

- maintenance of a routing table
- timely distribution of knowledge, in the form of routing updates, to other routers

Dynamic routing relies on a routing protocol to share knowledge among routers. A routing protocol defines the set of rules used by a router when it communicates with neighboring routers. For example, a routing protocol describes:

- how to send updates
- what knowledge is contained in these updates
- when to send this knowledge
- how to locate recipients of the updates

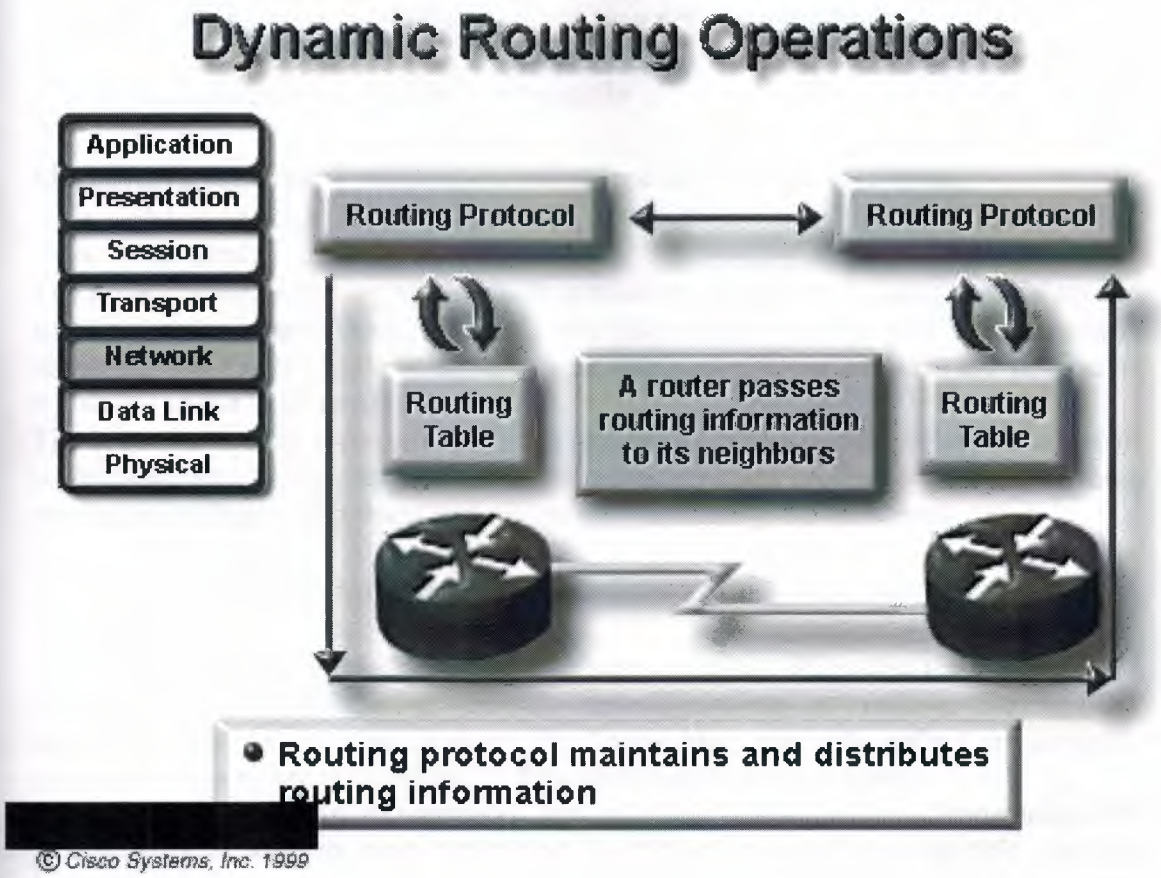


Figure: dynamic routing operation

2.1.6 How distances on network paths are determined by various metrics

When a routing algorithm updates a routing table, its primary objective is to determine the best information to include in the table. Each routing algorithm interprets what is best in its own way. The algorithm generates a number, called the metric value, for each path through the network. Typically, the smaller the metric number, the better the path.

You can calculate metrics based on a single characteristic of a path; you can calculate more complex metrics by combining several characteristics. The metrics most commonly used by routers are as follows:

- bandwidth-the data capacity of a link; (normally, a 10 Mbps Ethernet link is preferable to a 64 kbps leased line)
- delay-the length of time required to move a packet along each link from source to destination
- load-the amount of activity on a network resource such as a router or link
- reliability-usually refers to the error rate of each network link
- hop count-the number of routers a packet must travel through before reaching its destination
- ticks-the delay on a data link using IBM PC clock ticks (approximately 55 milliseconds).

- cost-an arbitrary value, usually based on bandwidth, monetary expense, or other measurement, that is assigned by a network administrator

2.1.7 Three classes of routing protocols

Most routing algorithms can be classified as one of two basic algorithms:

- distance vector; or
- link state.

The distance-vector routing approach determines the direction (vector) and distance to any link in the internetwork. The link-state (also called shortest path first) approach re-creates the exact topology of the entire internetwork (or at least the portion in which the router is situated).
The balanced hybrid approach combines aspects of the link-state and distance-vector algorithms. The next several pages cover procedures and problems for each of these routing algorithms and present techniques for minimizing the problems.

2.1.8 Time to convergence

The routing algorithm is fundamental to dynamic routing. Whenever the topology of a network changes because of growth, reconfiguration, or failure, the network knowledge base must also change. The knowledge needs to reflect an accurate, consistent view of the new topology. This view is called convergence.
When all routers in an internetwork are operating with the same knowledge, the internetwork is said to have converged. Fast convergence is a desirable network feature because it reduces the period of time in which routers would continue to make incorrect/wasteful routing decisions.

2.2 Distance-vector versus link-state routing protocols

You can compare distance-vector routing to link-state routing in several key areas:

- Distance-vector routing gets topological data from the routing table information of its neighbors. Link-state routing obtains a wide view of the entire internetwork topology by accumulating all necessary LSAs.
- Distance-vector routing determines the best path by adding to the metric value that it receives as routing information is passed from router to router. For link-state routing, each router works separately to calculate its own shortest path to destination networks.
- With most distance-vector routing protocols, updates for topology changes come in periodic table updates. The information passes from router to router, usually resulting in slower convergence. With link-state routing protocols, updates are usually triggered by topology changes. Relatively small LSAs passed to all other routers usually result in faster time to converge on any internetwork topology change.

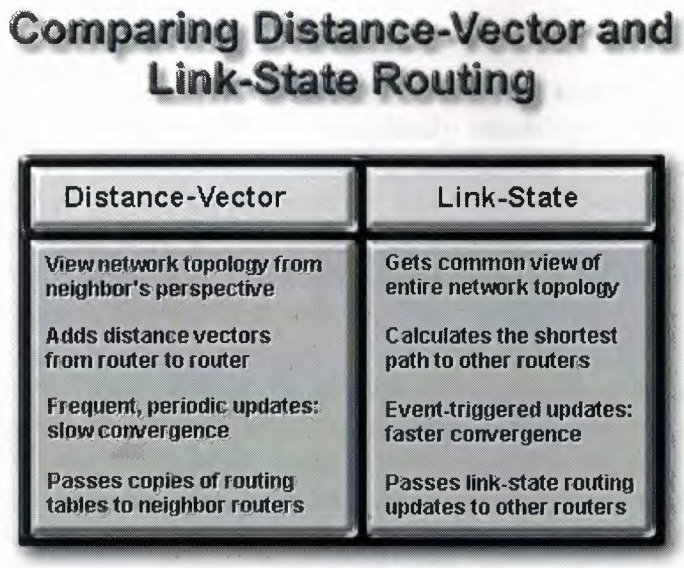


Figure: show and compare between link-state and distance-vector

2.3 Hybrid routing protocols

An emerging third type of routing protocol combines aspects of both distance-vector and link-state routing. This third type is called balanced-hybrid routing. Balanced-hybrid routing protocols use distance vectors with more accurate metrics to determine the best paths to destination networks. However, they differ from most distance-vector protocols by using topology changes to trigger routing database updates.
The balanced-hybrid routing protocol converges rapidly, like the link-state protocols. However, it differs from distance-vector and link-state protocols by using fewer resources such as bandwidth, memory, and processor overhead. Examples of hybrid protocols are OSI's IS-IS (Intermediate System-to-Intermediate System), and Cisco's EIGRP (Enhanced Interior Gateway Routing Protocol).

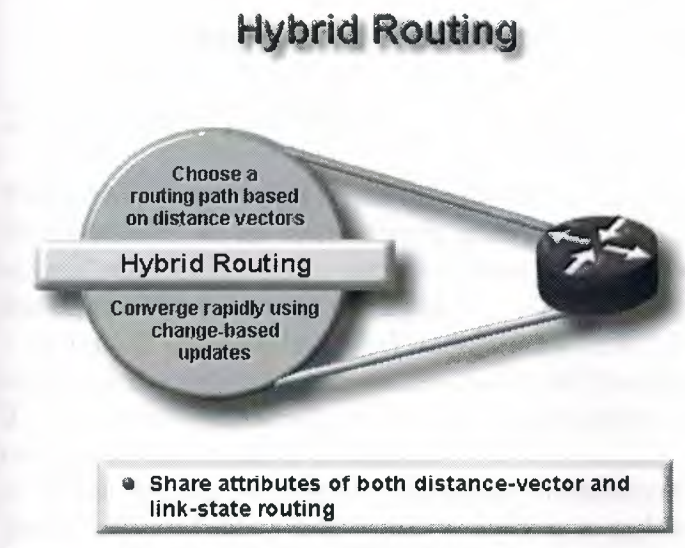


Figure: hybrid routing

2.4 LAN-to-LAN routing

The network layer must understand and be able to interface with various lower layers. Routers must be capable of seamlessly handling packets encapsulated into various lower-level frames without changing the packets' Layer 3 addressing.
The Figure shows an example of this with LAN-to-LAN routing. In this example, packet traffic from source Host 4 on Ethernet Network 1 needs a path to destination Host 5 on Network 2. The LAN hosts depend on the router and its consistent network addressing to find the best path.

When the router checks its routing table entries, it discovers that the best path to destination Network 2 uses outgoing port To0, the interface to a token-ring LAN. Although the lower-layer framing must change as the router passes packet traffic from Ethernet on Network 1 to token-ring on Network 2, the Layer 3 addressing for source and destination remains the same. In the Figure, the destination address remains Network 2, Host 5, regardless of the different lower-layer encapsulations.

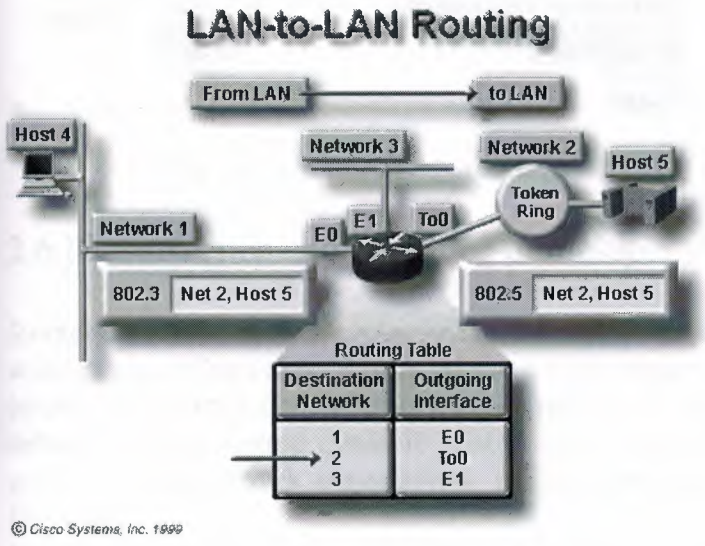


Figure: LAN-to-LAN routing

2.5 LAN-to-WAN routing

The network layer must relate to, and interface with, various lower layers for LAN-to-WAN traffic. As an internetwork grows, the path taken by a packet may encounter several relay points and a variety of data link types beyond the LANs. For example, in the Figure, the following takes place:
1. A packet from the top workstation at address 1.3 must traverse three data links to reach the file server at address 2.4, shown on the bottom.
2. The workstation sends a packet to the file server by first encapsulating it in a token-ring frame addressed to Router A.
3. When Router A receives the frame, it removes the packet from the token-ring frame, encapsulates it in a Frame Relay frame, and forwards the frame to Router B.
4. Router B removes the packet from the Frame Relay frame and forwards it to the file server in a newly created Ethernet frame.

5. When the file server at 2.4 receives the Ethernet frame, it extracts and passes the packet to the appropriate upper-layer process.
Routers enable LAN-to-WAN packet flow by keeping the end-to-end source and destination addresses constant while encapsulating the packet in data link frames, as appropriate, for the next hop along the path.

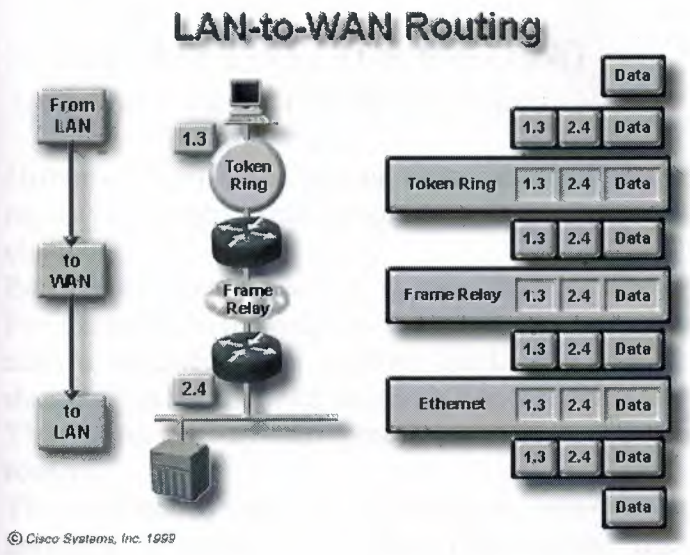


Figure: LAN-to-WAN routing

2.6 Path selection and switching of multiple protocols and media

Routers are devices that implement the network service. They provide interfaces for a wide range of links and subnetworks at a wide range of speeds. Routers are active and intelligent network nodes that can participate in managing a network. Routers manage networks by providing dynamic control over resources and supporting the tasks and goals for internetwork connectivity, reliable performance, management control, and flexibility.
In addition to the basic switching and routing functions, routers have a variety of additional features that help to improve the cost-effectiveness of the internetwork. These features include sequencing traffic based on priority and traffic filtering.
Typically, routers are required to support multiple protocol stacks, each with its own routing protocols, and to allow these different environments to operate in parallel. In practice, routers also incorporate bridging functions and sometimes serve as a limited form of hub.

CHAPTER THREE

DISTANCE-VECTOR ROUTING

3.1 DISTANCE-VECTOR ROUTING

3.1.1 Distance-vector routing basics

Distance-vector-based routing algorithms pass periodic copies of a routing table from router to router. These regular updates between routers communicate topology changes.

Each router receives a routing table from its directly connected neighboring routers. For example, in the graphic, Router B receives information from Router A. Router B adds a distance-vector number (such as a number of hops), which increases the distance vector and then passes this new routing table to its other neighbor, Router C. This same step-by-step process occurs in all directions between direct-neighbor routers.

The algorithm eventually accumulates network distances so that it can maintain a database of network topology information. Distance-vector algorithms do not, however, allow a router to know the exact topology of an internetwork.

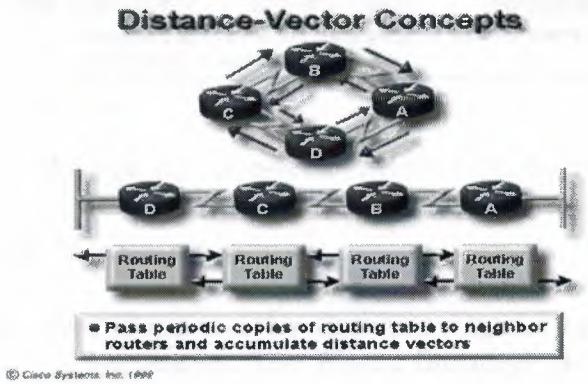


Figure: distance-vector concepts

3.1.2 How distance-vector protocols exchange routing tables

Each router that uses distance-vector routing begins by identifying its own neighbors. In the Figure, the interface that leads to each directly-connected network is shown as having a distance of 0. As the distance-vector network discovery process proceeds, routers discover the best path to destination networks based on the information they receive from each neighbor. For example, Router A learns about other networks based on the information that it receives from Router B. Each of the other network entries in the routing table has an accumulated distance vector to show how far away that network is in a given direction.

3.1.3 How topology changes propagate through the network of routers

When the topology in a distance-vector protocol network changes, routing table updates must occur. As with the network discovery process, topology change updates proceed step-by-step from router to router. Distance-vector algorithms call for each router to send its entire routing table to each of its adjacent neighbors. The routing tables include information about the total path cost (defined by its metric) and the logical address of the first router on the path to each network contained in the table.

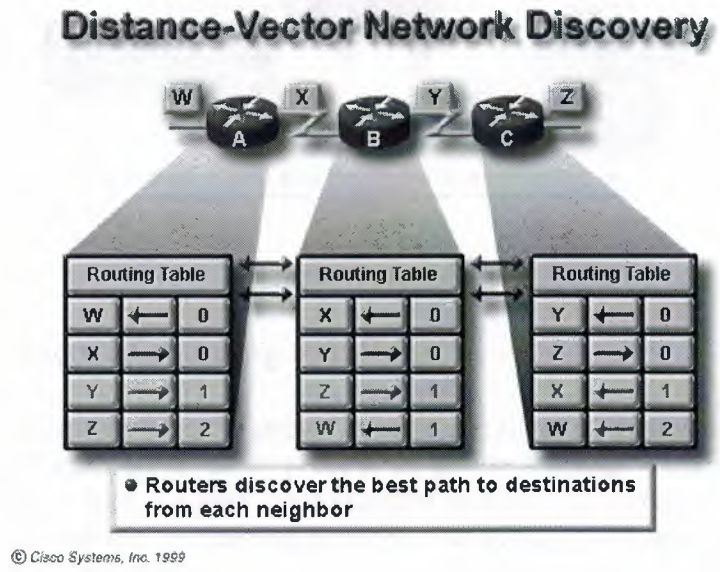


Figure: distance-vector network discovery

3.1.4 The problem of routing loops

Routing loops can occur if a network's slow convergence on a new configuration causes inconsistent routing entries. The Figure illustrates how a routing loop can occur:

1. Just before the failure of Network 1, all routers have consistent knowledge and correct routing tables. The network is said to have converged. Assume for the remainder of this example that Router C's preferred path to Network 1 is by way of Router B, and the distance from Router C to Network 1 is 3.

2. When Network 1 fails, Router E sends an update to Router A. Router A stops routing packets to Network 1, but Routers B, C, and D continue to do so because they have not yet been informed of the failure. When Router A sends out its update, Routers B and D stop routing to Network 1; however, Router C has not received an update. To Router C, Network 1 is still reachable via Router B.

3. Now Router C sends a periodic update to Router D, indicating a path to Network 1 by way of Router B. Router D changes its routing table to reflect this good, but incorrect, information, and propagates the information to Router A. Router A

propagates the information to Routers B and E, and so on. Any packet destined for Network 1 will now loop from Router C to B to A to D and back to again to C.

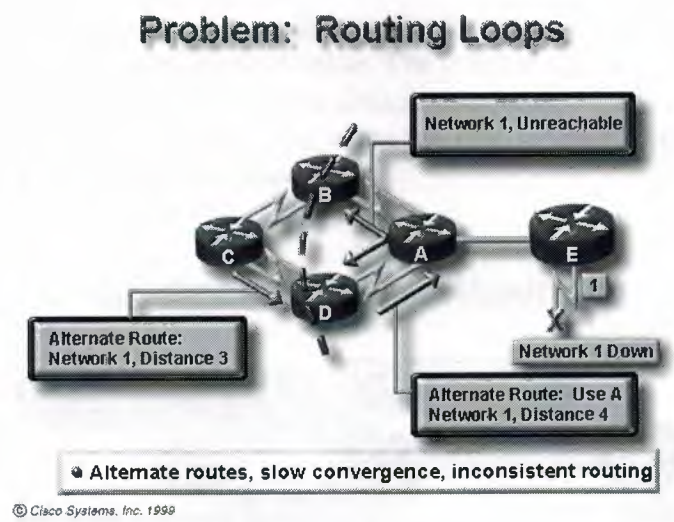


Figure: routing loops

3.1.5 The problem of counting to infinity

Continuing the example from the previous page, the invalid updates of Network 1 will continue to loop until some other process stops the looping. This condition, called count to infinity, loops packets continuously around the network in spite of the fundamental fact that the destination network, Network 1, is down. While the routers are counting to infinity, the invalid information allows a routing loop to exist.
Without countermeasures to stop the process, the distance vector (metric) of hop count increments each time the packet passes through another router. These packets loop through the network because of wrong information in the routing tables.

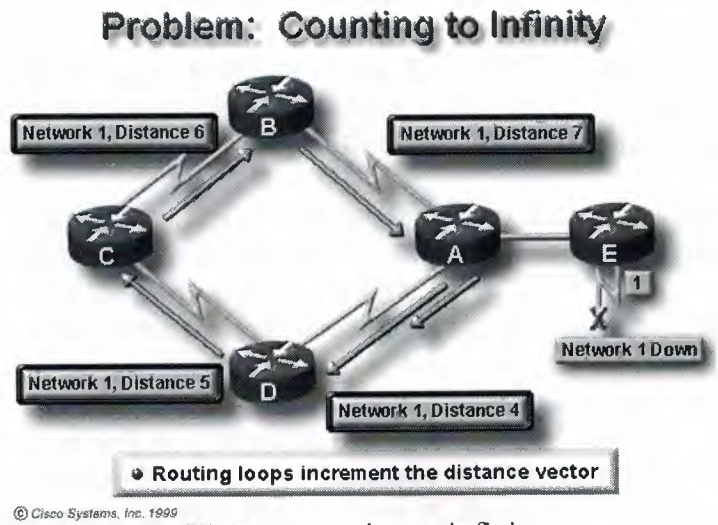


Figure: counting to infinity

3.1.6 The solution of defining a maximum

Distance-vector routing algorithms are self-correcting, but a routing loop problem can require a count to infinity first. To avoid this prolonged problem, distance-vector protocols define infinity as a specific maximum number. This number refers to a routing metric (e.g. a simple hop count).
With this approach, the routing protocol permits the routing loop to continue until the metric exceeds its maximum allowed value. The graphic shows the metric value as 16 hops, which exceeds the distance-vector default maximum of 15 hops, and the packet is discarded by the router. In any case, when the metric value exceeds the maximum value, Network 1 is considered unreachable.

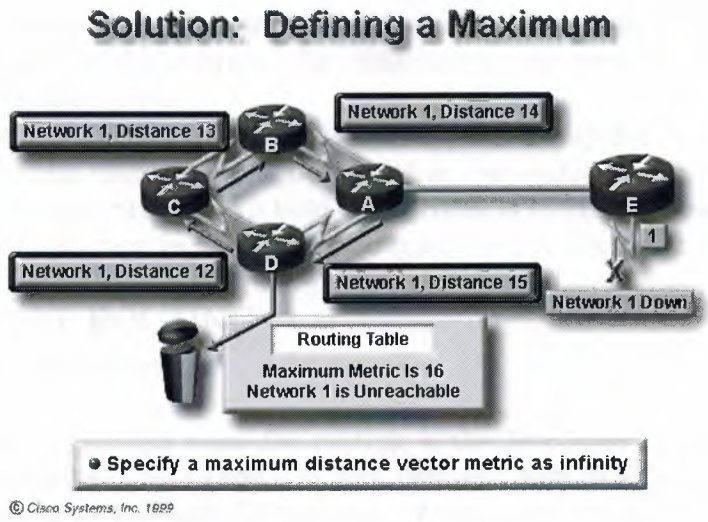


Figure: the solution

3.1.7 The solution of split horizon

Another possible source for a routing loop occurs when incorrect information that has been sent back to a router contradicts the correct information that it sent. Here is how this problem occurs:
1. Router A passes an update to Router B and Router D, indicating that Network 1 is down. Router C, however, transmits an update to Router B, indicating that Network 1 is available at a distance of 4, by way of Router D. This does not violate split-horizon rules.
2. Router B concludes, incorrectly, that Router C still has a valid path to Network 1, although at a much less favorable metric. Router B sends an update to Router A advising Router A of the new route to Network 1.
3. Router A now determines that it can send to Network 1 by way of Router B; Router B determines that it can send to Network 1 by way of Router C; and Router C determines that it can send to Network 1 by way of Router D. Any packet introduced into this environment will loop between routers.
4. Split-horizon attempts to avoid this situation. As shown in the Figure, if a routing update about Network 1 arrives from Router A, Router B or Router D cannot send information about Network 1 back to Router A. Split-horizon thus reduces incorrect routing information and reduces routing overhead.

3.1.8 The solution of hold-down timers

You can avoid a count to infinity problem by using hold-down timers that work as follows:
1.When a router receives an update from a neighbor indicating that a previously accessible network is now inaccessible, the router marks the route as inaccessible and starts a hold-down timer. If at any time before the hold-down timer expires an update is received from the same neighbor indicating that the network is again accessible, the router marks the network as accessible and removes the hold-down timer.
2.If an update arrives from a different neighboring router with a better metric than originally recorded for the network, the router marks the network as accessible and removes the hold-down timer.
3.If at any time before the hold-down timer expires an update is received from a different neighboring router with a poorer metric, the update is ignored. Ignoring an update with a poorer metric when a hold-down timer is in effect allows more time for the knowledge of a disruptive change to propagate through the entire network.

3.2 LINK-STATE ROUTING

3.2.1 Key characteristics

The second basic algorithm used for routing is the link-state algorithm. Link-state based routing algorithms, also known as SPF (shortest path first) algorithms, maintain a complex database of topology information. Whereas the distance-vector algorithm has nonspecific information about distant networks and no knowledge of distant routers, a link-state routing algorithm maintains full knowledge of distant routers and how they interconnect. Link-state routing uses:

- link-state advertisements (LSAs)
- a topological database
- the SPF algorithm, and the resulting SPF tree
- a routing table of paths and ports to each network

Engineers have implemented this link-state concept in OSPF (Open Shortest Path First) routing. RFC 1583 contains a description of OSPF link-state concepts and operations.

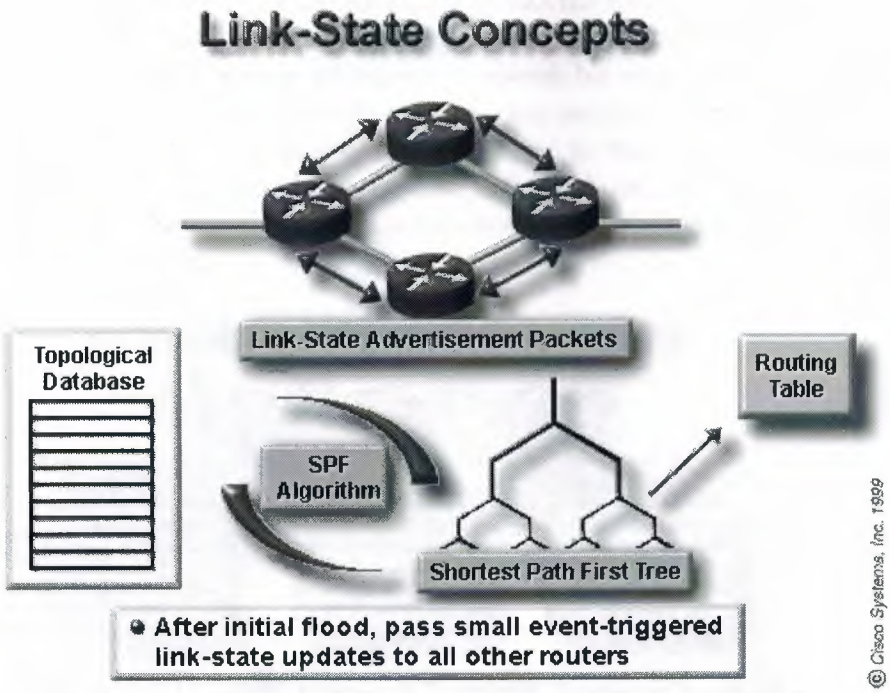


Figure: LINK – STATE CONCEPTS

3.2.2 How link-state protocols exchange routing tables?

Network discovery for link-state routing uses the following processes:
1. Routers exchange LSAs with each other. Each router begins with directly connected networks for which it has direct information.
2. Each router in parallel with the others constructs a topological database consisting of all the LSAs from the internetwork.
3. The SPF algorithm computes network reachability. The router constructs this logical topology as a tree, with itself as root, consisting of all possible paths to each network in the link-state protocol internetwork. It then sorts these paths shortest path first (SPF).
4. The router lists its best paths, and the ports to these destination networks, in the routing table. It also maintains other databases of topology elements and status details.

3.2.3 How topology changes propagate through the network of routers?

Link-state algorithms rely on using the same link-state updates. Whenever link-state topology changes, the routers that first become aware of the change send information to other routers or to a designated router that all other routers can use for updates. This involves sending common routing information to all routers in the internetwork. To achieve convergence, each router does the following:

- keeps track of its neighbors: each neighbor's name, whether the neighbor is up or down, and the cost of the link to the neighbor.
- constructs an LSA packet that lists its neighbor router names and link costs, including new neighbors, changes in link costs, and links to neighbors that have gone down.
- sends out this LSA packet so that all other routers receive it.

- when it receives an LSA packet, records the LSA packet in its database so that it updates the most recently generated LSA packet from each router.
- completes a map of the internetwork by using accumulated LSA packet data and then computes routes to all other networks by using the SPF algorithm.

Each time an LSA packet causes a change to the link-state database, the link-state algorithm (SPF) recalculates the best paths and updates the routing table. Then, every router takes the topology change into account as it determines the shortest path to use for packet routing.

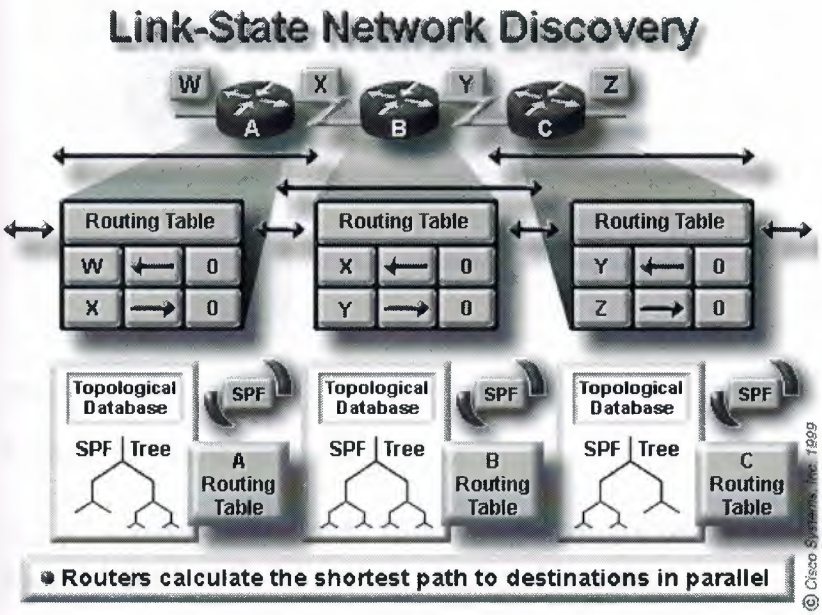


Figure: LINK-STATE NETWORK DISCOVERY

3.2.4 Two link-state concerns

There are two link-state concerns - processing and memory requirements, and bandwidth requirements.

1. Processing and memory requirements

Running link-state routing protocols in most situations requires that routers use more memory and perform more processing than distance-vector routing protocols. Network administrators must ensure that the routers they select are capable of providing these necessary resources.

Routers keep track of all other routers in a group and the networks that they can each reach directly. For link-state routing, their memory must be able to hold information from various databases, the topology tree, and the routing table. Using Dijkstra's algorithm to compute the SPF requires a processing task proportional to the number of links in the internetwork, multiplied by the number of routers in the internetwork.

2. Bandwidth requirements

Another cause for concern involves the bandwidth that must be consumed for initial link-state packet flooding. During the initial discovery process, all routers using link-

state routing protocols send LSA packets to all other routers. This action floods the internetwork as routers make their en masse demand for bandwidth, and temporarily reduce the bandwidth available for routed traffic that carries user data. After this initial flooding, link-state routing protocols generally require only minimal bandwidth to send infrequent or event-triggered LSA packets that reflect topology changes.

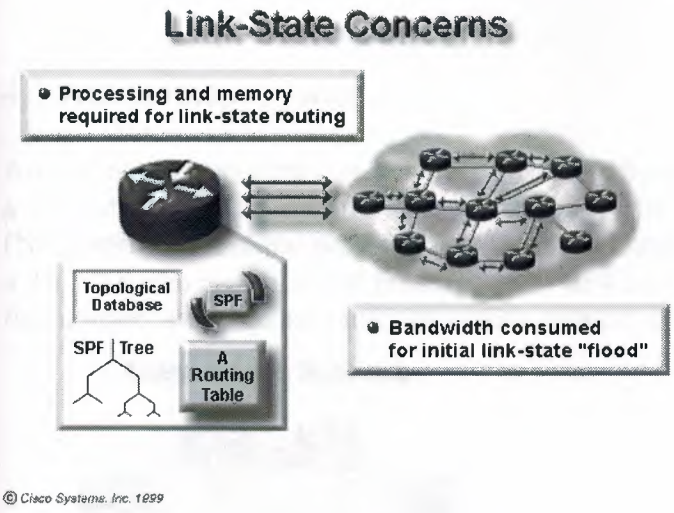


Figure: link-state concerns

3.2.5 Unsynchronized link-state advertisements (LSAs) leading to inconsistent path decisions amongst routers

The most complex and important aspect of link-state routing is making sure that all routers get all necessary LSA packets. Routers with different sets of LSAs calculate routes based on different topological data. Then, networks become unreachable as a result of a disagreement among routers about a link. Following is an example of inconsistent path information:

1. Between Routers C and D, Network 1 goes down. Both routers construct an LSA packet to reflect this unreachable status.
2. Soon afterward, Network 1 comes back up; another LSA packet reflecting this next topology change is needed.
3. If the original "Network 1, Unreachable" message from Router C uses a slow path for its update, that update comes later. This LSA packet can arrive at Router A after Router D's "Network 1, Back Up Now" LSA.
4. With unsynchronized LSAs, Router A can face a dilemma about which SPF tree to construct. Should it use paths that include Network 1, or paths without Network 1, which was most recently reported as unreachable?

If LSA distribution to all routers is not done correctly, link-state routing can result in invalid routes. Scaling up with link-state protocols on very large internetworks can expand the problem of faulty LSA packet distribution. If one part of the network comes up first with other parts coming up later, the order for sending and receiving LSA packets will vary. This variation can alter and impair convergence. Routers might learn about different versions of the topology before they construct their SPF trees and routing tables. On a large internetwork, parts that update more quickly can cause problems for parts that update more slowly.

CHAPTER FOUR

INTERIOR AND EXTERIOR ROUTING PROTOCOLS

4.1 Autonomous system

An autonomous system consists of routers, run by one or more operators, that present a consistent view of routing to the external world. The Network Information Center (NIC) assigns a unique autonomous system to enterprises. This autonomous system is a 16 bit number. A routing protocol such as Cisco's IGRP requires that you specify this unique, assigned autonomous system number in your configuration.

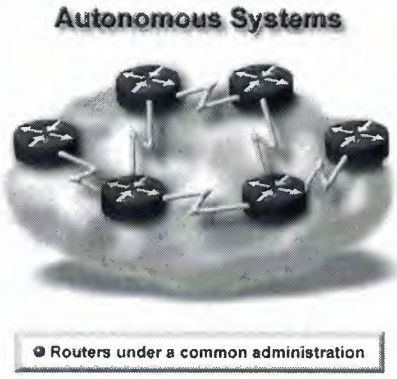


Figure: Autonomous system

4.2 Interior versus exterior routing protocols

Exterior routing protocols are used for communications between autonomous systems. Interior routing protocols are used within a single autonomous system.

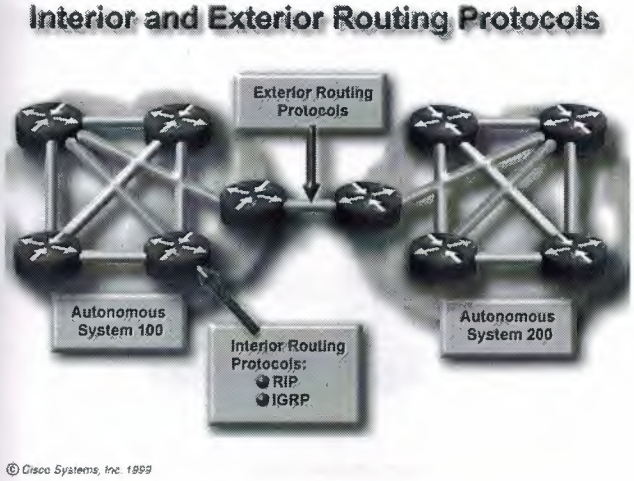


Figure: interior and exterior routing protocols

At the Internet layer of the TCP/IP suite of protocols, a router can use an IP routing protocol to accomplish routing through the implementation of a specific routing algorithm. Examples of IP routing protocols include:

- RIP-a distance-vector routing protocol
- IGRP-Cisco's distance-vector routing protocol
- OSPF-a link-state routing protocol
- EIGRP-a balanced hybrid routing protocol

The following sections show you how to configure the first two of these protocols.

4.3.1 RIP

The Routing Information Protocol (RIP) is a distance-vector protocol that uses hop count as its metric. RIP is widely used for routing traffic in the global Internet and is an interior gateway protocol (IGP), which means that it performs routing within a single autonomous system. Exterior gateway protocols, such as the Border Gateway Protocol (BGP), perform routing between different autonomous systems. The original incarnation of RIP was the Xerox protocol, GWINFO. A later version, known as routed (pronounced "route dee"), shipped with Berkeley Standard Distribution (BSD) Unix in 1982. RIP itself evolved as an Internet routing protocol, and other protocol suites use modified versions of RIP. The AppleTalk Routing Table Maintenance Protocol (RTMP) and the Banyan VINES Routing Table Protocol (RTP), for example, both are based on the Internet Protocol (IP) version of RIP. The latest enhancement to RIP is the RIP 2 specification, which allows more information to be included in RIP packets and provides a simple authentication mechanism.

IP RIP is formally defined in two documents: Request For Comments (RFC) 1058 and 1723. RFC 1058 (1988) describes the first implementation of RIP, while RFC 1723 (1994) updates RFC 1058. RFC 1058 enables RIP messages to carry more information and security features.

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.

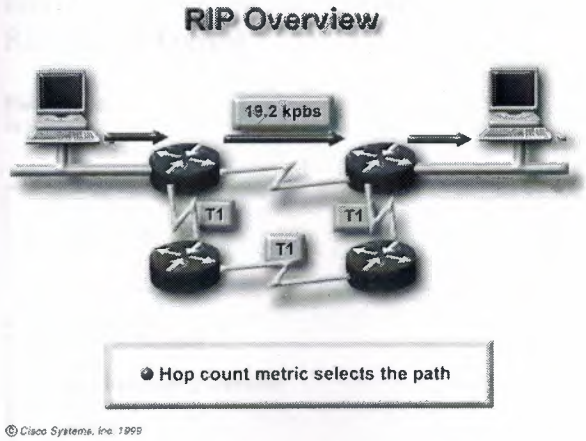


Figure: RIP Overview

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. The metric value for the path is increased by one, and the sender is indicated as the next hop. 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.

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 one 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.

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 one causes the metric to be infinity (that is, 16), the network destination is considered unreachable.

To adjust for rapid network-topology changes, RIP specifies a number of stability features that are common to many routing protocols. RIP, for example, implements the split-horizon and hold-down mechanisms to prevent incorrect routing information from being propagated. In addition, the RIP hop-count limit prevents routing loops from continuing indefinitely.

RIP uses numerous timers to regulate its performance. These include a routing-update timer, a route timeout, 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 number of seconds added each time the timer is reset to prevent collisions. 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.

RIP Packet Format

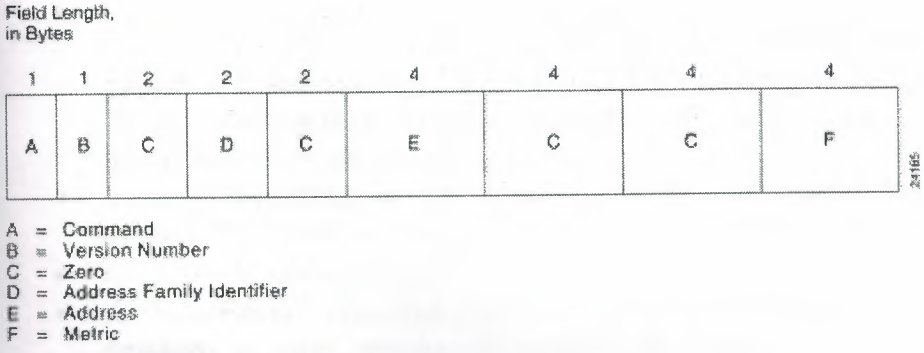


Figure 1: An IP RIP packet consists of nine fields.

The following descriptions summarize the IP RIP packet-format fields illustrated in **Figure -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---Not used.
- 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.

Note Up to 25 occurrences of the AFI, address, and metric fields are permitted in a single IP RIP packet. (Up to 25 destinations can be listed in a single RIP packet.)

The RIP 2 specification (described in RFC 1723) allows more information to be included in RIP packets and provides a simple authentication mechanism. **Figure -2** shows the IP RIP 2 packet format.

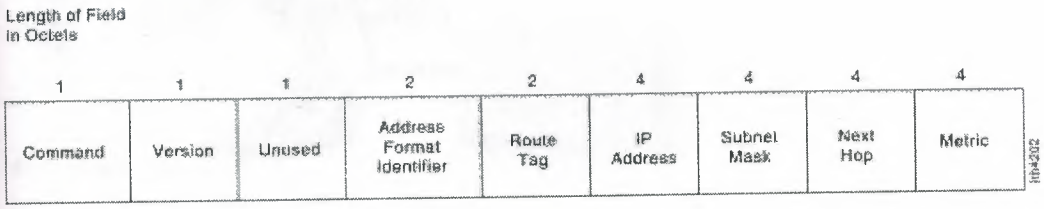


Figure -2: An IP RIP 2 packet consists of fields similar to those of an IP RIP packet.

The following descriptions summarize the IP RIP 2 packet format fields illustrated in Figure -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---Value set to 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 specified. The address-family identifier for IP is 2. 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.

Note Up to 25 occurrences of the AFI, address, and metric fields are permitted in a single IP RIP packet. That is, up to 25 routing table entries can be listed in a single RIP packet. If the AFI specifies an authenticated message, only 24 routing table entries can be specified.

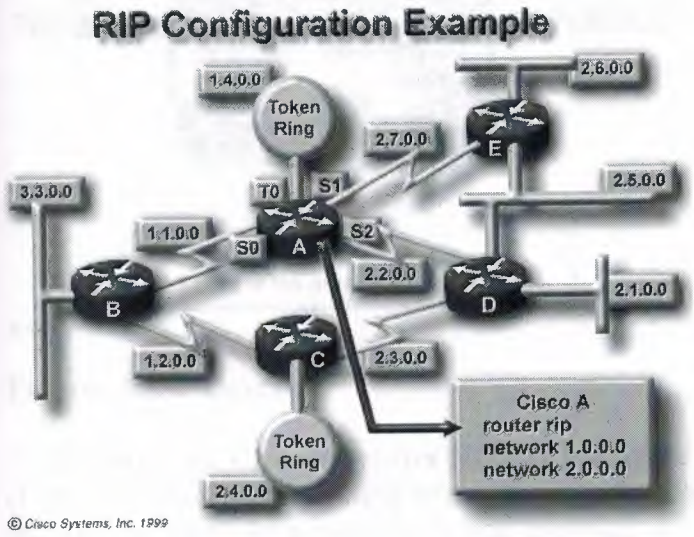


Figure: RIP Configuration Example

RIP was originally specified in RFC 1058. Its key characteristics include the following:

- It is a distance-vector routing protocol.
- Hop count is used as the metric for path selection.
- If the hop count is greater than 15, the packet will be discarded.
- By default, routing updates are broadcast every 30 seconds.

4.3.2 IGRP

The Interior Gateway Routing Protocol (IGRP) is a routing protocol that was developed in the mid-1980s by Cisco Systems, Inc. Cisco's principal goal in creating IGRP was to provide a robust protocol for routing within an autonomous system (AS). In the mid-1980s, the most popular intra-AS routing protocol was the Routing Information Protocol (RIP). Although RIP was quite useful for routing within small-to moderate-sized, relatively homogeneous internetworks, its limits were being pushed

by network growth. In particular, RIP's small hop-count limit (16) restricted the size of internetworks, and its single metric (hop count) did not allow for much routing flexibility in complex environments. The popularity of Cisco routers and the robustness of IGRP have encouraged many organizations with large internetworks to replace RIP with IGRP.

Cisco's initial IGRP implementation worked in Internet Protocol (IP) networks. IGRP was designed to run in any network environment, however, and Cisco soon ported it to run in OSI Connectionless-Network Protocol (CLNP) networks. Cisco developed Enhanced IGRP in the early 1990s to improve the operating efficiency of IGRP. This chapter discusses IGRP's basic design and implementation. Enhanced IGRP is discussed in **"Enhanced IGRP."**

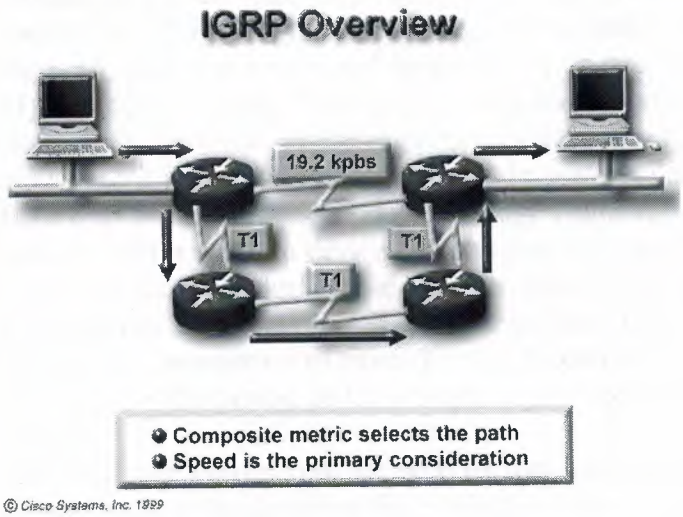


Figure: IGRP Overview

IGRP Protocol Characteristics IGRP is a distance-vector interior gateway protocol (IGP). Distance-vector routing protocols call for each router to send all or a portion of its routing table in a routing-update message at regular intervals to each of its neighboring routers. As routing information proliferates through the network, routers can calculate distances to all nodes within the internetwork.

Distance-vector routing protocols are often contrasted with link-state routing protocols, which send local connection information to all nodes in the internetwork. For a discussion of Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS), two popular link-state routing algorithms, see "Open Shortest Path First (OSPF)," and "Open System Interconnection (OSI) Protocols," respectively.

IGRP uses a combination (vector) of metrics. Internetwork delay, bandwidth, reliability, and load are all factored into the routing decision. Network administrators can set the weighting factors for each of these metrics. IGRP uses either the administrator-set or the default weightings to automatically calculate optimal routes.

IGRP provides a wide range for its metrics. Reliability and load, for example, can take on any value between 1 and 255; bandwidth can take on values reflecting speeds from 1,200 bps to 10 gigabits per second, while delay can take on any value from 1 to 2 to the 24th power. Wide metric ranges allow satisfactory metric setting in internetworks with widely varying performance characteristics. Most importantly, the

metric components are combined in a user-definable algorithm. As a result, network administrators can influence route selection in an intuitive fashion.
To provide additional flexibility, IGRP permits multipath routing. Dual equal-bandwidth lines can run a single stream of traffic in round-robin fashion, with automatic switchover to the second line if one line goes down. Also, multiple paths can be used even if the metrics for the paths are different. If, for example, one path is three times better than another because its metric is three times lower, the better path will be used three times as often. Only routes with metrics that are within a certain range of the best route are used as multiple paths.
IGRP provides a number of features that are designed to enhance its stability. These include hold-downs, split horizons, and poison-reverse updates.
Hold-downs are used to prevent regular update messages from inappropriately reinstating a route that might have gone bad. When a router goes down, neighboring routers detect this via the lack of regularly scheduled update messages. These routers then calculate new routes and send routing update messages to inform their neighbors of the route change. This activity begins a wave of triggered updates that filter through the network. These triggered updates do not instantly arrive at every network device, so it is therefore possible for a device that has yet to be informed of a network failure to send a regular update message (indicating that a route that has just gone down is still good) to a device that has just been notified of the network failure. In this case, the latter device would contain (and potentially advertise) incorrect routing information. Hold-downs tell routers to hold down any changes that might affect routes for some period of time. The hold-down period usually is calculated to be just greater than the period of time necessary to update the entire network with a routing change.
Split horizons derive from the premise that it is never useful to send information about a route back in the direction from which it came. Figure -1 illustrates the split-horizon rule. Router 1 (R1) initially advertises that it has a route to Network A. There is no reason for Router 2 (R2) to include this route in its update back to R1 because R1 is closer to Network A. The split-horizon rule says that R2 should strike this route from any updates it sends to R1. The split-horizon rule helps prevent routing loops. Consider, for example, the case where R1's interface to Network A goes down. to R1horizons, R2 continues to inform R1 that it can get to Network A (through R1). If R1 does not have sufficient intelligence, it actually might pick up R2's route as an alternative to its failed direct connection, causing a routing loop. Although hold-downs should prevent this, split horizons are implemented in IGRP because they provide extra algorithm stability.

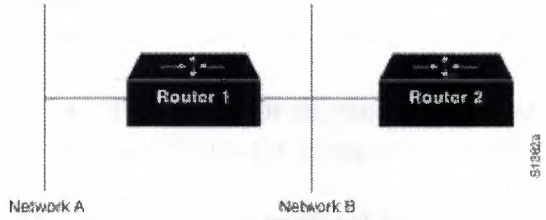


Figure -1: The split horizons rule helps protect against routing loops.

Split horizons should prevent routing loops between adjacent routers, but poison-reverse updates are necessary to defeat larger routing loops. Increases in routing metrics generally indicate routing loops. Poison-reverse updates then are sent to

remove the route and place it in hold-down. In Cisco's implementation of IGRP, poison-reverse updates are sent if a route metric has increased by a factor of 1.1 or greater.

Timers, IGRP maintain a number of timers and variables containing time intervals. These include an update timer, an invalid timer, a hold-time period, and a flush timer. The update timer specifies how frequently routing update messages should be sent. The IGRP default for this variable is 90 seconds. The invalid timer specifies how long a router should wait, in the absence of routing-update messages about a specific route before declaring that route invalid. The IGRP default for this variable is three times the update period. The hold-time variable specifies the hold-down period. The IGRP default for this variable is three times the update timer period plus 10 seconds. Finally, the flush timer indicates how much time should pass before a route should be flushed from the routing table. The IGRP default is seven times the routing update period.

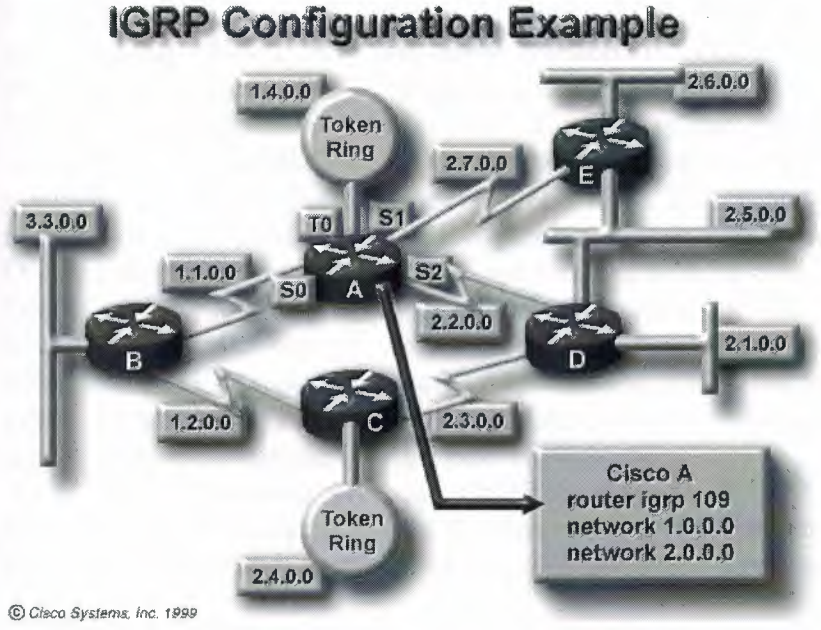


Figure: IGRP Configuration example

IGRP is a distance-vector routing protocol developed by Cisco. IGRP sends routing updates at 90 second intervals, advertising networks for a particular autonomous system. Some of the IGRP key design characteristics emphasize the following:

- versatility that enables it to automatically handle indefinite, complex topologies
- flexibility for segments that have different bandwidth and delay characteristics
- scalability for functioning in very large networks

The IGRP routing protocol by default uses two metrics, bandwidth and delay. IGRP can be configured to use a combination of variables to determine a composite metric. Those variables include:

- bandwidth
- delay
- load

- reliability

4.3.3 OSPF

Open Shortest Path First (OSPF) is a routing protocol developed for Internet Protocol (IP) networks by the interior gateway protocol (IGP) working group of the Internet Engineering Task Force (IETF). The working group was formed in 1988 to design an IGP based on the shortest path first (SPF) algorithm for use in the Internet. Similar to the Interior Gateway Routing Protocol (IGRP), OSPF was created because in the mid-1980s, the Routing Information Protocol (RIP) was increasingly unable to serve large, heterogeneous internetworks. This chapter examines the OSPF routing environment, underlying routing algorithm and general protocol components.
OSPF was derived from several research efforts, including Bolt, Beranek, Newman's (BBN's) SPF algorithm developed in 1978 for the ARPANET (a landmark packet-switching network developed in the early 1970s by BBN), Dr. Radia Perlman's research on fault-tolerant broadcasting of routing information (1988), BBN's work on area routing (1986), and an early version of OSI's Intermediate System-to-Intermediate System (IS-IS) routing protocol.
OSPF has two primary characteristics. The first is that the protocol is open, which means that its specification is in the public domain. The OSPF specification is published as Request For Comments (RFC) 1247. The second principal characteristic is that OSPF is based on the SPF algorithm, which sometimes is referred to as the Dijkstra algorithm, named for the person credited with its creation.
OSPF is a link-state routing protocol that calls for the sending of link-state advertisements (LSAs) to all other routers within the same hierarchical area. Information on attached interfaces, metrics used, and other variables is included in OSPF LSAs. As OSPF routers accumulate link-state information, they use the SPF algorithm to calculate the shortest path to each node.
As a link-state routing protocol, OSPF contrasts with RIP and IGRP, which are distance-vector routing protocols. Routers running the distance-vector algorithm send all or a portion of their routing tables in routing-update messages to their neighbors.

Unlike RIP, OSPF can operate within a **hierarchy**. The largest entity within the hierarchy is the autonomous system (AS), which is a collection of networks under a common administration that share a common routing strategy. OSPF is an intra-AS (interior gateway) routing protocol, although it is capable of receiving routes from and sending routes to other ASs.
An AS can be divided into a number of areas, which are groups of contiguous networks and attached hosts. Routers with multiple interfaces can participate in multiple areas. These routers, which are called area border routers, maintain separate topological databases for each area.

A topological database is essentially an overall picture of networks in relationship to routers. The topological database contains the collection of LSAs received from all routers in the same area. Because routers within the same area share the same information, they have identical topological databases.
The term domain sometimes is used to describe a portion of the network in which all routers have identical topological databases. Domain is frequently used interchangeably with AS.

An area's topology is invisible to entities outside the area. By keeping area topologies separate, OSPF passes less routing traffic than it would if the AS were not partitioned. Area partitioning creates two different types of OSPF routing, depending on whether the source and destination are in the same or different areas. Intra-area routing occurs when the source and destination are in the same area; interarea routing occurs when they are in different areas.
An OSPF backbone is responsible for distributing routing information between areas. It consists of all area border routers, networks not wholly contained in any area, and their attached routers. **Figure -1** shows an example of an internetwork with several areas.

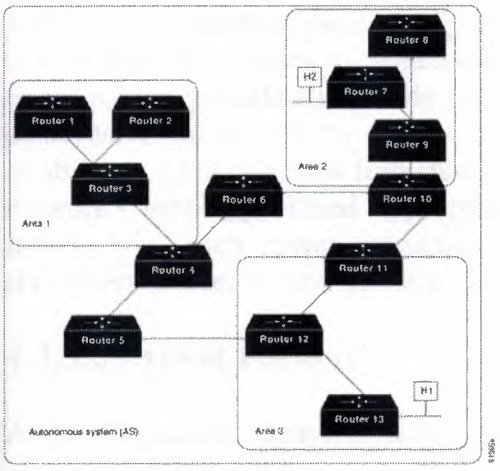


Figure -1: An OSPF AS consists of multiple areas linked by routers

In the figure, Routers 4, 5, 6, 10, 11, and 12 make up the backbone. If Host H1 in Area 3 wants to send a packet to Host H2 in area 2, the packet is sent to Router 13, which forwards the packet to Router 12, which sends the packet to Router 11. Router 11 then forwards the packet along the backbone to area border Router 10, which sends the packet through two intra-area routers (Router 9 and Router 7) to be forwarded to Host H2.
The backbone itself is an OSPF area, so all backbone routers use the same procedures and algorithms to maintain routing information within the backbone that any area router would. The backbone topology is invisible to all intra-area routers, as are individual area topologies to the backbone.
Areas can be defined in such a way that the backbone is not contiguous. In this case, backbone connectivity must be restored through virtual links. Virtual links are configured between any backbone routers that share a link to a nonbackbone area and function as if they were direct links.

4.3.3.1 SPF Algorithm

The shortest path first (SPF) routing algorithm is the basis for OSPF operations. When an SPF router is powered up, it initializes its routing-protocol data structures and then waits for indications from lower-layer protocols that its interfaces are functional.
After a router is assured that its interfaces are functioning, it uses the OSPF Hello protocol to acquire neighbors, which are routers with interfaces to a common network.

The router sends hello packets to its neighbors and receives their hello packets. In addition to helping acquire neighbors, hello packets also act as keep-alives to let routers know that other routers are still functional.
On multiaccess networks (networks supporting more than two routers), the Hello protocol elects a designated router and a backup designated router. Among other things, the designated router is responsible for generating LSAs for the entire multiaccess network. Designated routers allow a reduction in network traffic and in the size of the topological database.
When the link-state databases of two neighboring routers are synchronized, the routers are said to be adjacent. On multiaccess networks, the designated router determines which routers should become adjacent. Topological databases are synchronized between pairs of adjacent routers. Adjacencies control the distribution of routing-protocol packets, which are sent and received only on adjacencies.
Each router periodically sends an LSA to provide information on a router's adjacencies or to inform others when a router's state changes. By comparing established adjacencies to link states, failed routers can be detected quickly and the network's topology altered appropriately. From the topological database generated from LSAs, each router calculates a shortest-path tree, with itself as root. The shortest-path tree, in turn, yields a routing table.

4.3.3.2 Packet Format

All OSPF packets begin with a 24-byte header, as illustrated in Figure -2 .

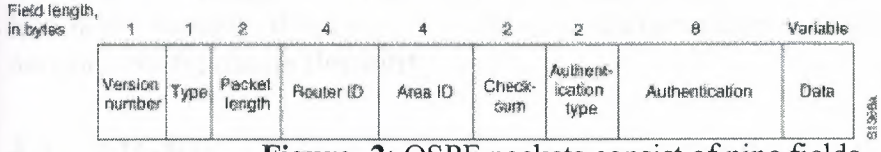


Figure -2: OSPF packets consist of nine fields.

The following descriptions summarize the header fields illustrated in figure 42-2.

- Version Number---Identifies the OSPF version used.
- Type---Identifies the OSPF packet type as one of the following:
  - Hello: Establishes and maintains neighbor relationships.
  - Database Description: Describes the contents of the topological database. These messages are exchanged when an adjacency is initialized.
  - Link-state Request: Requests pieces of the topological database from neighbor routers. These messages are exchanged after a router discovers (by examining database-description packets) that parts of its topological database are out of date.
  - Link-state Update: Responds to a link-state request packet. These messages also are used for the regular dispersal of LSAs. Several LSAs can be included within a single link-state update packet.
  - Link-state Acknowledgment: Acknowledges link-state update packets.
- Packet Length---Specifies the packet length, including the OSPF header, in bytes.
- Router ID---Identifies the source of the packet.

- Area ID---Identifies the area to which the packet belongs. All OSPF packets are associated with a single area.
- Checksum---Checks the entire packet contents for any damage suffered in transit.
- Authentication Type---Contains the authentication type. All OSPF protocol exchanges are authenticated. The Authentication Type is configurable on a per-area basis.
- Authentication---Contains authentication information.

Data---Contains encapsulated upper-layer information.

Additional OSPF **features** include equal-cost, multipath routing, and routing based on upper-layer type-of-service (TOS) requests. TOS-based routing supports those upper-layer protocols that can specify particular types of service. An application, for example, might specify that certain data is urgent. If OSPF has high-priority links at its disposal, these can be used to transport the urgent datagram.
OSPF supports one or more metrics. If only one metric is used, it is considered to be arbitrary, and TOS is not supported. If more than one metric is used, TOS is optionally supported through the use of a separate metric (and, therefore, a separate routing table) for each of the eight combinations created by the three IP TOS bits (the delay, throughput, and reliability bits). If, for example, the IP TOS bits specify low delay, low throughput, and high reliability, OSPF calculates routes to all destinations based on this TOS designation.
IP subnet masks are included with each advertised destination, enabling variable-length subnet masks. With variable-length subnet masks, an IP network can be broken into many subnets of various sizes. This provides network administrators with extra network-configuration flexibility.

4.3.4 EIGRP

The Enhanced Interior Gateway Routing Protocol (IGRP) represents an evolution from its predecessor IGRP This evolution resulted from changes in networking and the demands of diverse, large-scale internetworks. Enhanced IGRP integrates the capabilities of link-state protocols into distance-vector protocols. It incorporates the Diffusing-Update Algorithm (DUAL) developed at SRI International by Dr. J.J. Garcia-Luna-Aceves.
Enhanced IGRP provides compatibility and seamless interoperation with IGRP routers. An automatic-redistribution mechanism allows IGRP routes to be imported into Enhanced IGRP, and vice versa, so it is possible to add Enhanced IGRP gradually into an existing IGRP network. Because the metrics for both protocols are directly translatable, they are as easily comparable as if they were routes that originated in their own Autonomous Systems (ASs). In addition, Enhanced IGRP treats IGRP routes as external routes and provides a way for the network administrator to customize them.
This chapter provides an overview of the basic operations and protocol characteristics of Enhanced IGRP.

Key capabilities that distinguish **Enhanced** IGRP from other routing protocols include fast convergence, support variable-length subnet mask, support for partial updates, and support for multiple network-layer protocols.

A router running Enhanced IGRP stores all its neighbors' routing tables so that it can quickly adapt to alternate routes. If no appropriate route exists, Enhanced IGRP queries its neighbors to discover an alternate route. These queries propagate until an alternate route is found.
Its support for variable-length subnet masks permits routes to be automatically summarized on a network number boundary. In addition, Enhanced IGRP can be configured to summarize on any bit boundary at any interface.
Enhanced IGRP does not make periodic updates. Instead, it sends partial updates only when the metric for a route changes. Propagation of partial updates is automatically bounded so that only those routers that need the information are updated. As a result of these two capabilities, Enhanced IGRP consumes significantly less bandwidth than IGRP.
Enhanced IGRP includes support for AppleTalk, IP, and Novell NetWare. The AppleTalk implementation redistributes routes learned from the Routing Table Maintenance Protocol (RTMP). The IP implementation redistributes routes learned from OSPF, Routing Information Protocol (RIP), IS-IS, Exterior Gateway Protocol (EGP), or Border Gateway Protocol (BGP). The Novell implementation redistributes routes learned from Novell RIP or Service Advertisement Protocol (SAP).

4.3.4.1 Underlying Processes and Technologies

To provide superior routing performance, Enhanced IGRP employs four key technologies that combine to differentiate it from other routing technologies: neighbor discovery/recovery, reliable transport protocol (RTP), DUAL finite-state machine, and protocol-dependent modules.
Neighbor discovery/recovery is used by routers to dynamically learn about other routers on their directly attached networks. Routers also must discover when their neighbors become unreachable or inoperative. This process is achieved with low overhead by periodically sending small hello packets. As long as a router receives hello packets from a neighboring router, it assumes that the neighbor is functioning, and the two can exchange routing information.
Reliable Transport Protocol (RTP) is responsible for guaranteed, ordered delivery of Enhanced IGRP packets to all neighbors. It supports intermixed transmission of multicast or unicast packets. For efficiency, only certain Enhanced IGRP packets are transmitted reliably. On a multiaccess network that has multicast capabilities, such as Ethernet, it is not necessary to send hello packets reliably to all neighbors individually. For that reason, Enhanced IGRP sends a single multicast hello packet containing an indicator that informs the receivers that the packet need not be acknowledged. Other types of packets, such as updates, indicate in the packet that acknowledgment is required. RTP contains a provision for sending multicast packets quickly when unacknowledged packets are pending, which helps ensure that convergence time remains low in the presence of varying speed links.
DUAL finite-state machine embodies the decision process for all route computations by tracking all routes advertised by all neighbors. DUAL uses distance information to select efficient, loop-free paths and selects routes for insertion in a routing table based on feasible successors. A feasible successor is a neighboring router used for packet forwarding that is a least-cost path to a destination that is guaranteed not to be part of a routing loop. When a neighbor changes a metric, or when a topology change occurs, DUAL tests for feasible successors. If one is found, DUAL uses it to avoid

recomputing the route unnecessarily. When no feasible successors exist but neighbors still advertise the destination, a recomputation (also known as a diffusing computation) must occur to determine a new successor. Although recomputation is not processor-intensive, it does affect convergence time, so it is advantageous to avoid unnecessary recomputations.
Protocol-dependent modules are responsible for network-layer protocol-specific requirements. The IP-Enhanced IGRP module, for example, is responsible for sending and receiving Enhanced IGRP packets that are encapsulated in IP. Likewise, IP-Enhanced IGRP is also responsible for parsing Enhanced IGRP packets and informing DUAL of the new information that has been received. IP-Enhanced IGRP asks DUAL to make routing decisions, the results of which are stored in the IP routing table. IP-Enhanced IGRP is responsible for redistributing routes learned by other IP routing protocols.

4.3.4.2 **Routing Concepts**

Enhanced IGRP relies on four fundamental concepts: neighbor tables, topology tables, route states, and route tagging. Each of these is summarized in the discussions that follow.

4.3.4.3 Neighbor Tables

When a router discovers a new neighbor, it records the neighbor's address and interface as an entry in the neighbor table. One neighbor table exists for each protocol-dependent module. When a neighbor sends a hello packet, it advertises a hold time, which is the amount of time a router treats a neighbor as reachable and operational. If a hello packet is not received within the hold time, the hold time expires and DUAL is informed of the topology change.
The neighbor-table entry also includes information required by RTP. Sequence numbers are employed to match acknowledgments with data packets, and the last sequence number received from the neighbor is recorded so that out-of-order packets can be detected. A transmission list is used to queue packets for possible retransmission on a per-neighbor basis. Round-trip timers are kept in the neighbor-table entry to estimate an optimal retransmission interval.

4.3.4.3 Topology Tables

The topology table contains all destinations advertised by neighboring routers. The protocol-dependent modules populate the table, and the table is acted on by the DUAL finite-state machine. Each entry in the topology table includes the destination address and a list of neighbors that have advertised the destination. For each neighbor, the entry records the advertised metric, which the neighbor stores in its routing table.
An important rule that distance vector protocols must follow is that if the neighbor advertises this destination, it must use the route to forward packets.
The metric that the router uses to reach the destination is also associated with the destination. The metric that the router uses in the routing table, and to advertise to other routers, is the sum of the best advertised metric from all neighbors, plus the link cost to the best neighbor.

Route States, A topology-table entry for a destination can exist in one of two states: active or passive. A destination is in the passive state when the router is not performing a recomputation, or in the active state when the router is performing a recomputation. If feasible successors are always available, a destination never has to go into the active state, thereby avoiding a recomputation.
A recomputation occurs when a destination has no feasible successors. The router initiates the recomputation by sending a query packet to each of its neighboring routers. The neighboring router can send a reply packet, indicating it has a feasible successor for the destination, or it can send a query packet, indicating that it is participating in the recomputation. While a destination is in the active state, a router cannot change the destination's routing-table information. After the router has received a reply from each neighboring router, the topology-table entry for the destination returns to the passive state, and the router can select a successor.

Route Tagging , Enhanced IGRP supports internal and external routes. Internal routes originate within an Enhanced IGRP AS. Therefore, a directly attached network that is configured to run Enhanced IGRP is considered an internal route and is propagated with this information throughout the Enhanced IGRP AS. External routes are learned by another routing protocol or reside in the routing table as static routes. These routes are tagged individually with the identity of their origin.
External routes are tagged with the following information:

- Router ID of the Enhanced IGRP router that redistributed the route
- AS number of the destination
- Configurable administrator tag
- ID of the external protocol
- Metric from the external protocol
- Bit flags for default routing

Route tagging allows the network administrator to customize routing and maintain flexible policy controls. Route tagging is particularly useful in transit ASs, where Enhanced IGRP typically interacts with an interdomain routing protocol that implements more global policies, resulting in a very scalable, policy-based routing.

Enhanced IGRP uses the following **packet types**: hello and acknowledgment, update, and query and reply.
Hello packets are multicast for neighbor discovery/recovery and do not require acknowledgment. An acknowledgment packet is a hello packet that has no data. Acknowledgment packets contain a non-zero acknowledgment number and always are sent by using a unicast address.

Update packets are used to convey reachability of destinations. When a new neighbor is discovered, unicast update packets are sent so that the neighbor can build up its topology table. In other cases, such as a link-cost change, updates are multicast. Updates always are transmitted reliably.

Query and reply packets are sent when a destination has no feasible successors. Query packets are always multicast. Reply packets are sent in response to query packets to instruct the originator not to recompute the route because feasible successors exist. Reply packets are unicast to the originator of the query. Both query and reply packets are transmitted reliabl

4.4 BORDER GATEWAY PROTOCOL (BGP)

4.4.1 BGP

Routing involves two basic activities: determination of optimal routing paths and the transport of information groups (typically called packets) through an internetwork. The transport of packets through an internetwork is relatively straightforward. Path determination, on the other hand, can be very complex. One protocol that addresses the task of path determination in today's networks is the Border Gateway Protocol (BGP). This chapter summarizes the basic operations of BGP and provides a description of its protocol components.

BGP performs interdomain routing in Transmission-Control Protocol/Internet Protocol (TCP/IP) networks. BGP is an exterior gateway protocol (EGP), which means that it performs routing between multiple autonomous systems or domains and exchanges routing and reachability information with other BGP systems.

BGP was developed to replace its predecessor, the now obsolete Exterior Gateway Protocol (EGP), as the standard exterior gateway-routing protocol used in the global Internet. BGP solves serious problems with EGP and scales to Internet growth more efficiently.

Note EGP is a particular instance of an exterior gateway protocol (also EGP)---the two should not be confused.

Figure -1 illustrates core routers using BGP to route traffic between autonomous systems.

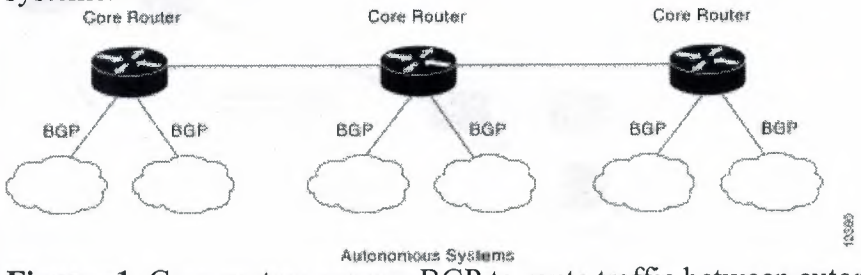


Figure -1: Core routers can use BGP to route traffic between autonomous systems.

BGP is specified in several Request For Comments (RFCs):

- RFC 1771 ---Describes BGP4, the current version of BGP
- RFC 1654---Describes the first BGP4 specification

RFC 1105, RFC 1163, and RFC 1267---Describes versions of BGP prior to BGP4

4.4.2 BGP Operation

BGP performs three types of routing: interautonomous system routing, intra-autonomous system routing, and pass-through autonomous system routing.

Interautonomous system routing occurs between two or more BGP routers in different autonomous systems. Peer routers in these systems use BGP to maintain a consistent view of the internetwork topology. BGP neighbors communicating between autonomous systems must reside on the same physical network. The Internet serves as an example of an entity that uses this type of routing because it is comprised of autonomous systems or administrative domains. Many of these domains represent the various institutions, corporations, and entities that make up the Internet. BGP is

frequently used to provide path determination to provide optimal routing within the Internet.

Intra-autonomous system routing occurs between two or more BGP routers located within the same autonomous system. Peer routers within the same autonomous system use BGP to maintain a consistent view of the system topology. BGP also is used to determine which router will serve as the connection point for specific external autonomous systems. Once again, the Internet provides an example of interautonomous system routing. An organization, such as a university, could make use of BGP to provide optimal routing within its own administrative domain or autonomous system. The BGP protocol can provide both inter- and intra-autonomous system routing services.

Pass-through autonomous system routing occurs between two or more BGP peer routers that exchange traffic across an autonomous system that does not run BGP. In a pass-through autonomous system environment, the BGP traffic did not originate within the autonomous system in question and is not destined for a node in the autonomous system. BGP must interact with whatever intra-autonomous system routing protocol is being used to successfully transport BGP traffic through that autonomous system. Figure -2 illustrates a pass-through autonomous system environment:

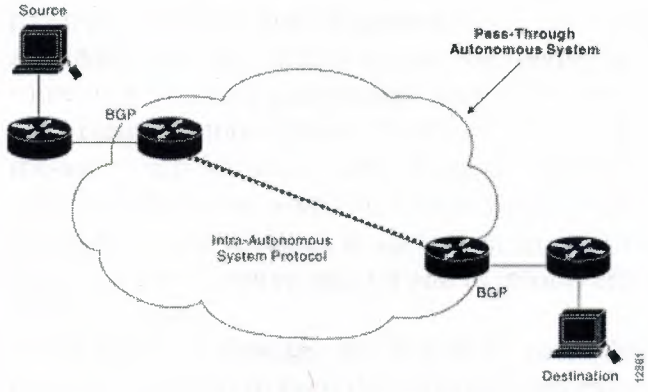


Figure -2: In pass-through autonomous system routing, BGP pairs with another intra-autonomous system-routing protocol.

4.4.3 BGP Routing

As with any routing protocol, BGP maintains routing tables, transmits routing updates, and bases routing decisions on routing metrics. The primary function of a BGP system is to exchange network-reachability information, including information about the list of autonomous system paths, with other BGP systems. This information can be used to construct a graph of autonomous system connectivity from which routing loops can be pruned and with which autonomous system-level policy decisions can be enforced.

Each BGP router maintains a routing table that lists all feasible paths to a particular network. The router does not refresh the routing table, however. Instead, routing information received from peer routers is retained until an incremental update is received.

BGP devices exchange routing information upon initial data exchange and after incremental updates. When a router first connects to the network, BGP routers exchange their entire BGP routing tables. Similarly, when the routing table changes, routers send the portion of their routing table that has changed. BGP routers do not send regularly scheduled routing updates, and BGP routing updates advertise only the optimal path to a network.
BGP uses a single routing metric to determine the best path to a given network. This metric consists of an arbitrary unit number that specifies the degree of preference of a particular link. The BGP metric typically is assigned to each link by the network administrator. The value assigned to a link can be based on any number of criteria, including the number of autonomous systems through which the path passes, stability, speed, delay, or cost.

4.4.4 BGP Message Types

Four BGP message types are specified in RFC 1771, A Border Gateway Protocol 4 (BGP-4): open message, update message, notification message, and keep-alive message.
The open message opens a BGP communications session between peers and is the first message sent by each side after a transport-protocol connection is established. Open messages are confirmed using a keep-alive message sent by the peer device and must be confirmed before updates, notifications, and keep-alives can be exchanged.
An update message is used to provide routing updates to other BGP systems, allowing routers to construct a consistent view of the network topology. Updates are sent using the Transmission-Control Protocol (TCP) to ensure reliable delivery. Update messages can withdraw one or more unfeasible routes from the routing table and simultaneously can advertise a route while withdrawing others.
The notification message is sent when an error condition is detected. Notifications are used to close an active session and to inform any connected routers of why the session is being closed.
The keep-alive message notifies BGP peers that a device is active. Keep-alives are sent often enough to keep the sessions from expiring.

4.4.5 BGP Packet Formats

The sections that follow summarize BGP open, updated, notification, and keep-alive message types, as well as the basic BGP header format. Each is illustrated with a format drawing, and the fields shown are defined.

All BGP message types use the basic packet header. Open, update, and notification messages have additional fields, but keep-alive messages use only the basic packet header. Figure -3 illustrates the fields used in the BGP header. The section that follows summarizes the function of each field.

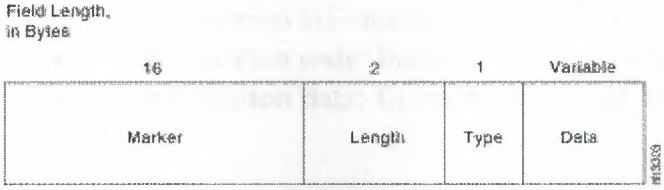


Figure -3: A BGP packet header consists of four fields.

Each BGP packet contains a header whose primary purpose is to identify the function of the packet in question. The following descriptions summarize the function of each field in the BGP header illustrated in Figure -3.

- Marker---Contains an authentication value that the message receiver can predict.
- Length---Indicates the total length of the message in bytes.
- Type---Type --- Specifies the message type as one of the following:
- Open
- Update
- Notification
- Keep-alive
- Data---Contains upper-layer information in this optional field.

BGP open messages are comprised of a BGP header and additional fields. Figure -4 illustrates the additional fields used in BGP open messages.

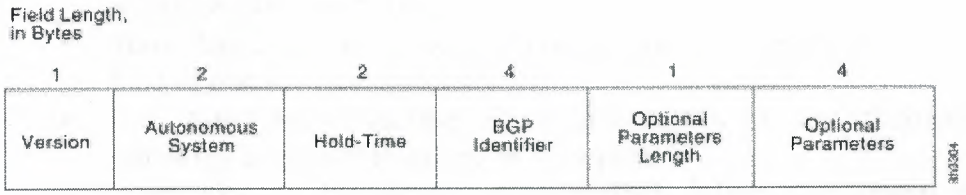


Figure -4: A BGP open message consists of six fields

BGP packets in which the type field in the header identifies the packet to be a BGP open message packet include the following fields. These fields provide the exchange criteria for two BGP routers to establish a peer relationship.

- Version---Provides the BGP version number so that the recipient can determine whether it is running the same version as the sender.
- Autonomous System---Provides the autonomous system number of the sender.
- Hold-Time---Indicates the maximum number of seconds that can elapse without receipt of a message before the transmitter is assumed to be nonfunctional.
- BGP Identifier---Provides the BGP identifier of the sender (an IP address), which is determined at startup and is identical for all local interfaces and all BGP peers.
- Optional Parameters Length---Indicates the length of the optional parameters field (if present).
- Optional Parameters---Contains a list of optional parameters (if any). Only one optional parameter type is currently defined: authentication information.
- Authentication information consists of the following two fields:
- Authentication code: Indicates the type of authentication being used.
- Authentication data: Contains data used by the authentication mechanism (if used).

BGP **update** messages are comprised of a BGP header and additional fields. Figure -5 illustrates the additional fields used in BGP update messages.

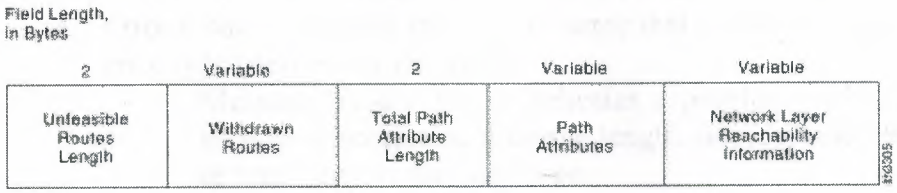


Figure -5: A BGP update message contains five fields.

BGP packets in which the type field in the header identifies the packet to be a BGP update message packet include the following fields. Upon receiving an update message packet, routers will be able to add or delete specific entries from their routing tables to ensure accuracy. Update messages consist of the following packets:

- Unfeasible Routes Length---Indicates the total length of the withdrawn routes field or that the field is not present.
- Withdrawn Routes---Contains a list of IP address prefixes for routes being withdrawn from service.
- Total Path Attribute Length---Indicates the total length of the path attributes field or that the field is not present.
- Path Attributes---Describes the characteristics of the advertised path. The following are possible attributes for a path:
    - Origin: Mandatory attribute that defines the origin of the path information
    - AS Path: Mandatory attribute composed of a sequence of autonomous system path segments
    - Next Hop: Mandatory attribute that defines the IP address of the border router that should be used as the next hop to destinations listed in the network layer reachability information field
    - Mult Exit Disc: Optional attribute used to discriminate between multiple exit points to a neighboring autonomous system
    - Local Pref: Discretionary attribute used to specify the degree of preference for an advertised route
    - Atomic Aggregate: Discretionary attribute used to disclose information about route selections
    - Aggregator: Optional attribute that contains information about aggregate routes

Network Layer Reachability Information---Contains a list of IP address prefixes for the advertised routes

Figure -6 illustrates the additional fields used in BGP notification messages.

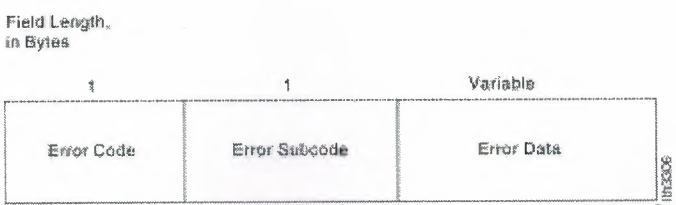


Figure -6: A BGP notification message consists of three fields

BGP packets in which the type field in the header identifies the packet to be a BGP notification message packet include the following fields. This packet is used to indicate some sort of error condition to the peers of the originating router.

- Error Code---Indicates the type of error that occurred. The following are the error types defined by the field:
    - Message Header Error: Indicates a problem with a message header, such as unacceptable message length, unacceptable marker field value, or unacceptable message type.
    - Open Message Error: Indicates a problem with an open message, such as unsupported version number, unacceptable autonomous system number or IP address, or unsupported authentication code.
    - Update Message Error: Indicates a problem with an update message, such as a malformed attribute list, attribute list error, or invalid next-hop attribute.
    - Hold Time Expired: Indicates that the hold-time has expired, after which time a BGP node will be considered nonfunctional.
    - Finite State Machine Error: Indicates an unexpected event.
    - Cease: Closes a BGP connection at the request of a BGP device in the absence of any fatal errors.
- Error Subcode---Provides more specific information about the nature of the reported error.
- Error Data---Contains data based on the error code and error subcode fields. This field is used to diagnose the reason for the notification message

CHAPTER FIVE

USING THE BORDER GATEWAY PROTOCOL FOR INTERDOMAIN ROUTING

5.1 Border Gateway Protocol

The Border Gateway Protocol (BGP), defined in RFC 1771, provides loop-free interdomain routing between autonomous systems. (An autonomous system [AS] is a set of routers that operates under the same administration.) BGP is often run among the networks of Internet service providers (ISPs). This case study examines how BGP works and how you can use it to participate in routing with other networks that run BGP. The following topics are covered:

- BGP Fundamentals
- BGP Decision Algorithm
- Controlling the Flow of BGP Updates
- Practical Design Example

5.1.1 BGP Fundamentals

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), and 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" later in this chapter), the commands for configuring EBGP and IBGP are the same. This case study 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 -1 shows a network that demonstrates the difference between EBGP and IBGP.

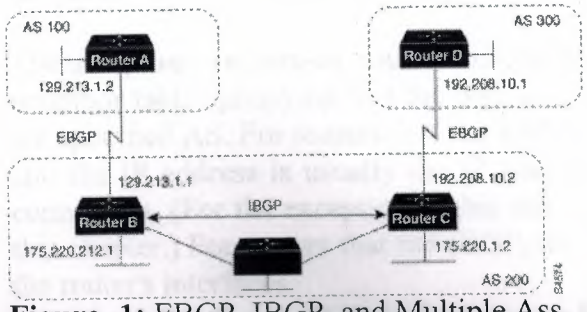


Figure -1: EBGP, IBGP, and Multiple Ass

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 -1, 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).

In Figure -1, the following commands configure BGP on Router A:
router bgp 100
neighbor 129.213.1.1 remote-as 200

The following commands configure BGP on Router B:
router bgp 200
neighbor 129.213.1.2 remote-as 100
neighbor 175.220.1.2 remote-as 200

The following commands configure BGP on Router C:
router bgp 200
neighbor 175.220.212.1 remote-as 200
neighbor 192.208.10.1 remote-as 300

The following commands configure BGP on Router D:
router bgp 300
neighbor 192.208.10.2 remote-as 200

The **router bgp** global configuration command enables a BGP routing process and assigns to it an AS number.

The **neighbor remote-as** router configuration command adds an entry to the BGP neighbor table specifying that the peer identified by a particular IP address belongs to the specified AS. For routers that run EBGP, neighbors are usually directly connected, and the IP address is usually the IP address of the interface at the other end of the connection. (For the exception to this rule, see the section "EBGP Multihop," later in this chapter.) For routers that run IBGP, the IP address can be the IP address of any of the router's interfaces.
Note the following about the ASs shown in Figure -1:

- Routers A and B are running EBGP, and Routers B and C are running IBGP. 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 one another, IBGP peers do not have to be directly connected.
- All BGP speakers within an AS must establish a peer relationship with each other. That is, the BGP speakers within an AS must be fully meshed logically. BGP4 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.

To verify that BGP peers are up, use the show ip bgp neighbors EXEC command. Following is the output of this command on Router A:

RouterA# show ip bgp neighbors
BGP neighbor is 129.213.1.1, remote AS 200, external link
BGP version 4, remote router ID 175.220.212.1
BGP state = established, table version = 3, up for 0:10:59
Last read 0:00:29, hold time is 180, keepalive interval is 60 seconds
Minimum time between advertisement runs is 30 seconds
Received 2828 messages, 0 notifications, 0 in queue
Sent 2826 messages, 0 notifications, 0 in queue
Connections established 11; dropped 10

Anything other than state = established indicates that the peers are not up. The remote router ID is the highest IP address on that router (or the highest loopback interface, if there is one). Notice the table version number: each time the table is updated by new incoming information, the table version number increments. A table version number that continually increments is an indication that a route is flapping, thereby causing routes to be updated continually.

Note When you make a configuration change with respect to a neighbor for which a peer relationship has been established, be sure to reset the BGP session with that neighbor. To reset the session, at the system prompt, issue the **clear ip bgp** EXEC command specifying the IP address of that neighbor.

5.1.1.1 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, provides more efficient ways of controlling the exchange of information within the AS, and presents a consistent view of the AS to external neighbors. For example, IBGP provides ways to control the exit point from an AS.
Figure -2 shows a topology that demonstrates IBGP.

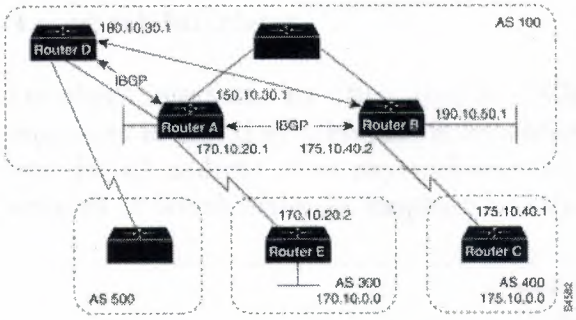


Figure 12-2: Internal BGP Example

The following commands configure Routers A and B in AS 100, and Router C in AS 400:

```
!Router A
router bgp 100
neighbor 180.10.30.1 remote-as 100
neighbor 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0

!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100
neighbor 175.10.40.1 remote-as 400
neighbor 180.10.30.1 remote-as 100
network 190.10.50.0

!Router C
router bgp 400
neighbor 175.10.40.2 remote-as 100
network 175.10.0.0

!Router D
router bgp 100
neighbor 150.10.30.1 remote-as 100
neighbor 190.10.50.1 remote as 100
network 190.10.0.0
```

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 -2, 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 shows a network in which using the loopback interface is advantageous.

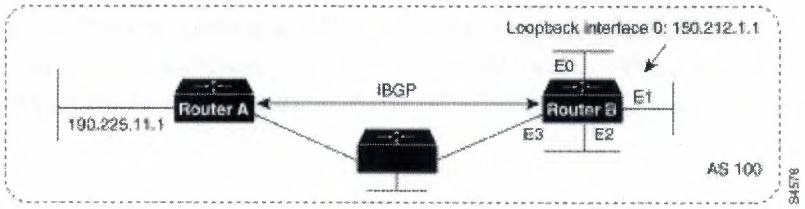


Figure -3: Use of Loopback Interfaces

In Figure -3, 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.

The following commands configure Router A for BGP:
!Router A
router bgp 100
neighbor 150.212.1.1 remote-as 100

The following commands configure Router B for BGP:
!Router B
loopback interface 0
ip address 150.212.1.1 255.255.0.0
!
router bgp 100
neighbor 190.225.11.1 remote-as 100
neighbor 190.225.11.1 update-source loopback 0

Router A specifies the IP address of the loopback interface (150.212.1.1) of Router B in the neighbor remote-as router configuration command. This use of the loopback interface requires that the configuration of Router B include the neighbor update-source router configuration command. When the neighbor update-source command is used, the source of BGP TCP connections for the specified neighbor is the IP address of the loopback interface instead of the IP address of a physical interface.

5.1.1.2 External BGP

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

solve configuration problems that arise when BGP routing updates are exchanged between different ASs:

- EBGP Multihop
- EBGP Load Balancing
- Synchronization

EBGP Multihop the two EBGP speakers are directly connected (for example, over a wide-area network [WAN] connection). Sometimes, however, they cannot be directly connected. In this special case, the neighbor ebgp-multihop router configuration command is used.

Figure -4: EBGP Multihop

The following commands configure Router A to run EBGP:

```
!Router A
loopback interface 0
ip address 129.213.1.1
!
router bgp 100
neighbor 180.225.11.1 remote-as 300
neighbor 180.225.11.1 ebgp-multihop
neighbor 180.225.11.1 update-source loopback 0
```

The neighbor remote-as router configuration command specifies the IP address of an interface that is an extra hop away (180.225.11.1 instead of 129.213.1.3), and the neighbor ebgp-multihop router configuration command enables EGBP multihop. Because Router A references an external neighbor by an address that is not directly connected, its configuration must include static routes or must enable an IGP so that the neighbors can reach each other.

The following commands configure Router B:

```
!Router B
loopback interface 0
ip address 180.225.11.1

router bgp 300
neighbor 129.213.1.1 remote-as 100
neighbor 129.213.1.1 ebgp-multihop
neighbor 129.213.1.1 update-source loopback 0
```

EBGP Load Balancing The neighbor ebgp-multihop router configuration command and loopback interfaces are also useful for configuring load balancing between two ASs over parallel serial lines, as shown in Figure -5.

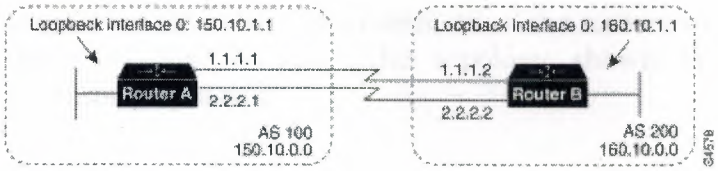


Figure -5: Load Balancing over Parallel Serial Lines

Without the neighbor ebgp-multihop command on each router, BGP would not perform load balancing in Figure -5, but with the neighbor ebgp-multihop command on each router, BGP uses both serial lines. The following commands configure load balancing for Router A:

```
!Router A
interface loopback 0
ip address 150.10.1.1 255.255.255.0
!
router bgp 100
neighbor 160.10.1.1 remote-as 200
neighbor 160.10.1.1 ebgp-multihop
neighbor 160.10.1.1 update-source loopback 0
network 150.10.0.0
!
ip route 160.10.0.0 255.255.0.0 1.1.1.2
ip route 160.10.0.0 255.255.0.0 2.2.2.2
```

The following commands configure load balancing for Router B:

```
!Router B
interface loopback 0
ip address 160.10.1.1 255.255.255.0
!
router bgp 200
neighbor 150.10.1.1 remote-as 100
neighbor 150.10.1.1 ebgp-multihop
neighbor 150.10.1.1 update-source loopback 0
network 160.10.0.0
!
ip route 150.10.0.0 255.255.0.0 1.1.1.1
ip route 150.10.0.0 255.255.0.0 2.2.2.1
```

The neighbor ebgp-multihop and neighbor update-source router configuration commands have the effect of making the loopback interface the next hop for EBGP, which allows load balancing to occur. Static routes are used to introduce two equal-cost paths to the destination. (The same effect could also be accomplished by using an IGP.) Router A can reach the next hop of 160.10.1.1 in two ways: via 1.1.1.2 and via 2.2.2.2. Likewise, Router B can reach the next hop of 150.10.1.1 in two ways: via 1.1.1.1 and via 2.2.2.1.

Synchronization, When an AS provides transit service to other ASs and if 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 -6 demonstrates the synchronization rule.

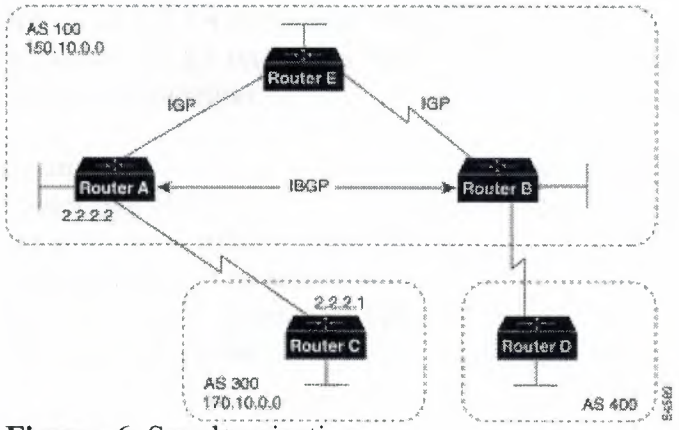


Figure -6: Synchronization

In Figure -6, 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, which states that if an AS (such as AS 100 in Figure -6) 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. 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.

Figure -7 shows a topology in which it is desirable to disable synchronization.

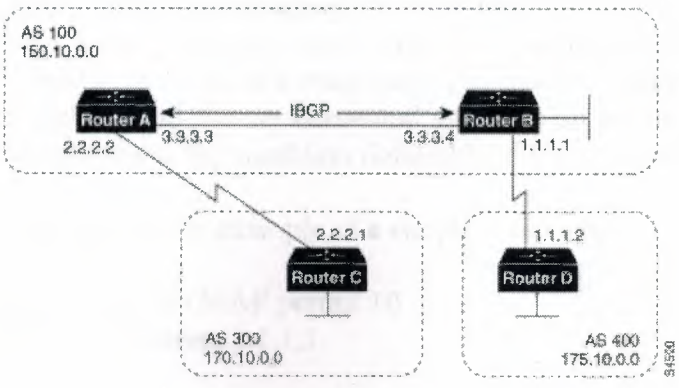


Figure -7: Disabled Synchronization

The following commands configure Routers A, B, and C:

```
!Router A
network 150.10.0.0
neighbor 3.3.3.4 remote-as 100
neighbor 2.2.2.1 remote-as 300
no synchronization

!Router B
router bgp 100
network 150.10.0.0
neighbor 1.1.1.2 remote-as 400
neighbor 3.3.3.3 remote-as 100
no synchronization

!Router D
router bgp 400
neighbor 1.1.1.1 remote-as 100
network 175.10.0.0
```

5.1.1.3 BGP and Route Maps

Route maps are used with BGP to control and modify routing information and to define the conditions by which routes are redistributed between routing domains. The format of a route map is as follows:

route-map map-tag [[permit | deny] | [sequence-number]]

The map tag is a name that identifies the route map, and the sequence number indicates the position that an instance of the route map is to have in relation to other instances of the same route map. (Instances are ordered sequentially.)

For example, you might use the following commands to define a route map named MYMAP:

```
Route-map MYMAP permit 10
! First set of conditions goes here.
Route-map MYMAP permit 20
! Second set of conditions goes here.
```

When BGP applies MYMAP to routing updates, it applies the lowest instance first (in this case, instance 10). If the first set of conditions is not met, the second instance is applied, and so on, until either a set of conditions has been met, or there are no more sets of conditions to apply.

The match and set route map configuration commands are used to define the condition portion of a route map. The match command specifies a criteria that must be matched, and the set command specifies an action that is to be taken if the routing update meets the condition defined by the match command.

Following is an example of a simple route map:

```
route-map MYMAP permit 10
match ip address 1.1.1.1
```

set metric 5

When an update matches IP address 1.1.1.1, BGP sets the metric for the update to 5, sends the update (because of the **permit** keyword), and breaks out of the list of route-map instances.
When an update does not meet the criteria of an instance, BGP applies the next instance of the route map to the update, and so on, until an action is taken, or there are no more route map instances to apply. If the update does not meet any criteria, the update is not redistributed or controlled.
When an update meets the match criteria, and the route map specifies the deny keyword, BGP breaks out of the list of instances, and the update is not redistributed or controlled.

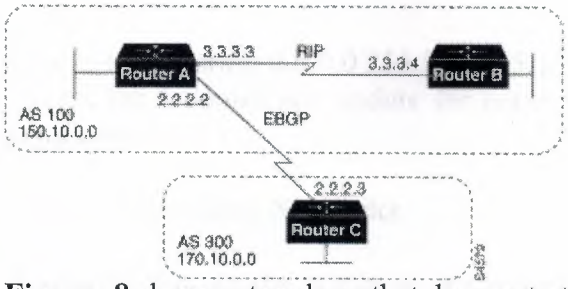


Figure -8 shows a topology that demonstrates the use of route maps.

In Figure -8, Routers A and B run RIP with each other, and Routers A and C run BGP with each other. If you want Router A to redistribute routes from 170.10.0.0 with a metric of 2 and to redistribute all other routes with a metric of 5, use the following commands for Router A:

```
! Router A
Router rip
Network 3.0.0.0
Network 2.0.0.0
Network 150.10.0.0
Passive-interface serial 0
Redistribute bgp 100 route-map SETMETRIC
!
Router bgp 100
Neighbor 2.2.2.3 remote-as 300
Network 150.10.0.0
!
Route-map SETMETRIC permit 10
Match ip-address 1
Set metric 2
!
Route-map SETMETRIC permit 20
Set metric 5
!
Access-list 1 permit 170.10.0.0 0.0.255.255
```

When a route matches the IP address 170.10.0.0, it is redistributed with a metric of 2. When a route does not match the IP address 170.10.0.0, its metric is set to 5, and the route is redistributed.

Assume that on Router C you want to set to 300 the community attribute of outgoing updates for network 170.10.0.0. The following commands apply a route map to outgoing updates on Router C:

```
! Router C
Router bgp 300
Network 170.10.0.0
Neighbor 2.2.2.2 remote-as 100
Neighbor 2.2.2.2 route-map SETCOMMUNITY out
!
Route-map SETCOMMUNITY permit 10
Match ip address 1
Set community 300
!
Access-list 1 permit 0.0.0.0 255.255.255.255
```

Access list 1 denies any update for network 170.10.0.0 and permits updates for any other network.

5.1.1.4 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

Note It is important to remember that routes advertised by the techniques described in this section are advertised in addition to other BGP routes that a BGP-configured router learns from its internal and external neighbors. BGP always passes on information that it learns from one peer to other peers. The difference is that routes generated by the network and redistribute router configuration commands specify the AS of the router as the originating AS for the network.
This section uses the topology shown in **Figure -9** to demonstrate how networks that originate from an AS can be advertised.

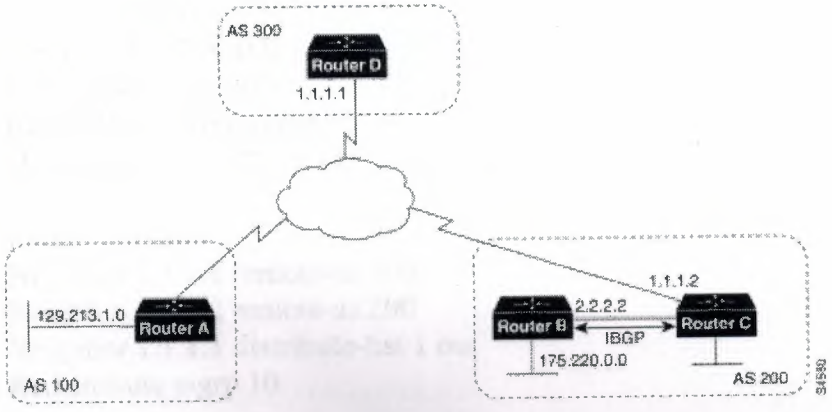


Figure -9: Network Advertisement Example 1

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 -9 to originate network 175.220.0.0 into BGP, use these commands:

```
! Router C
Router bgp 200
Neighbor 1.1.1.1 remote-as 300
Redistribute static
!
ip route 175.220.0.0 0.0.255.255 null 0
```

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 -9** 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 following commands configure Router C:

```
! Router C
Router eigrp 10
Network 175.220.0.0
Redistribute bgp 200
Redistributed connected
Default-metric 1000 100 250 100 1500
!
Router bgp 200
Neighbor 1.1.1.1 remote-as 300
Neighbor 2.2.2.2 remote-as 200
Neighbor 1.1.1.1 distribute-list 1 out
Redistribute eigrp 10
!
Access-list 1 permit 175.220.0.0 0.0.255.255
```

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:

```
!Router C
router bgp 200
neighbor 1.1.1.1 remote-as 300
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 -10 shows another topology that demonstrates the effects of the **network** command.

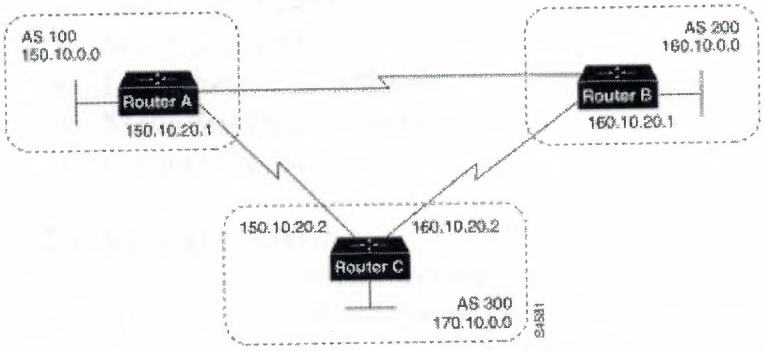


Figure -10: Network Advertisement Example 2

The following configurations use the network command to configure the routers shown in Figure -10:

```
!Router A
router bgp 100
neighbor 150.10.20.2 remote-as 300
network 150.10.0.0
```

```
!Router B
router bgp 200
neighbor 160.10.20.2 remote-as 300
network 160.10.0.0

!Router C
router bgp 300
neighbor 150.10.20.1 remote-as 100
neighbor 160.10.20.1 remote-as 200
network 170.10.0.0
```

To ensure a loop-free interdomain topology, BGP does not accept updates that originated from its own AS. For example, in **Figure -10**, 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.

5.2 BGP Decision Algorithm

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

5.2.1 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 -11.**

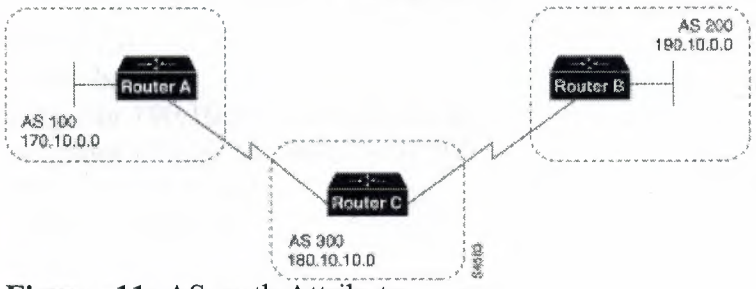


Figure -11: AS_path Attribute

In Figure -11, Router B advertises network 190.10.0.0 in AS 200 with an AS_path of 200. When the update for 190.10.0.0 traverses AS 300, Router C prepends its own AS number to it, so when the update reaches Router A, two AS numbers have been attached to it: 200 and then 300. That is, the AS_path attribute for reaching network 190.10.0.0 from Router A is 300, 200. Likewise, the AS_path attribute for reaching network 170.10.0.0 from Router B is 300, 100.

5.2.2 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 12 shows a network that demonstrates the value of the origin attribute.

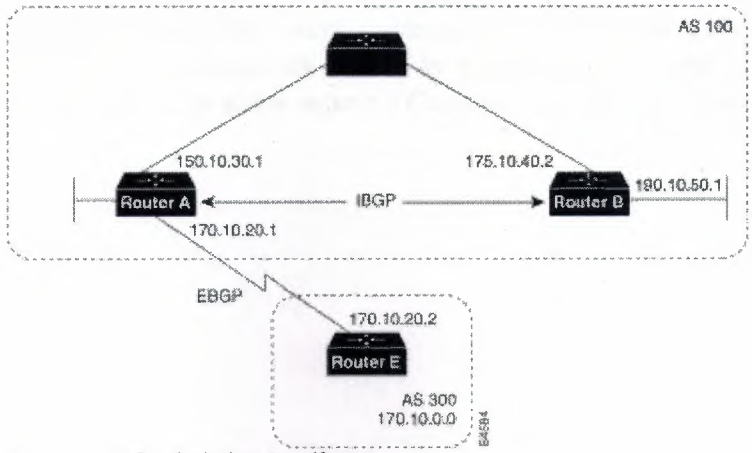


Figure -12: Origin Attribute

The following commands configure the routers shown in **Figure -12:**

```
!Router A
router bgp 100
neighbor 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0
redistribute static
!
ip route 190.10.0.0 255.255.0.0 null 0
```

```
!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100
network 190.10.50.0
```

```
!Router E
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```

Given these configurations, the following is true:

- From Router A, the route for reaching 170.10.0.0 has an AS_path of 300 and an origin attribute of IGP.
- From Router A, the route for reaching 190.10.50.0 has an empty AS_path (the route is in the same AS as Router A) and an origin attribute of IGP.
- From Router E, the route for reaching 150.10.0.0 has an AS_path of 100 and an origin attribute of IGP.

From Router E, the route for reaching 190.10.0.0 has an AS_path of 100 and an origin attribute of Incomplete (because 190.10.0.0 is a redistributed route

5.2.3 **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 -13.

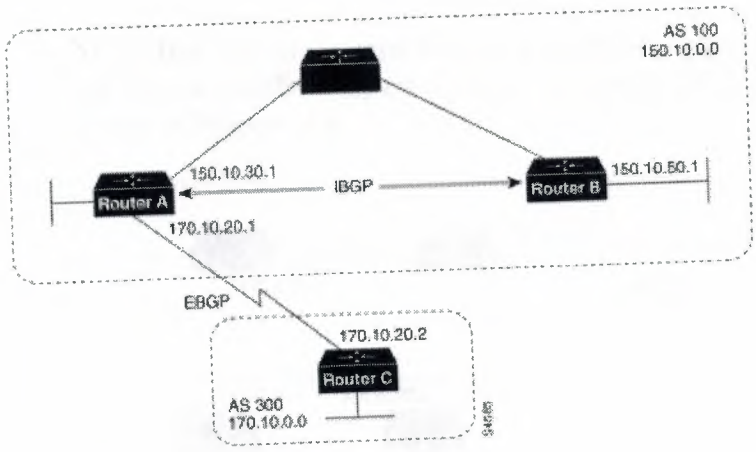


Figure -13: Next Hop Attribute

In **Figure -13**, 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.

The following commands configure the routers shown in **Figure -13**:

```
!Router A
router bgp 100
neighbor 170.10.20.2 remote-as 300
neighbor 150.10.50.1 remote-as 100
network 150.10.0.0

!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100

!Router C
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```

Note Router C advertises 170.10.0.0 to Router A with a next hop attribute of 170.10.20.2, and Router A advertises 170.10.0.0 to Router B with a next hop attribute of 170.10.20.2. The next hop of EBGP-learned routes is passed to the IBGP neighbor.

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 -14.**

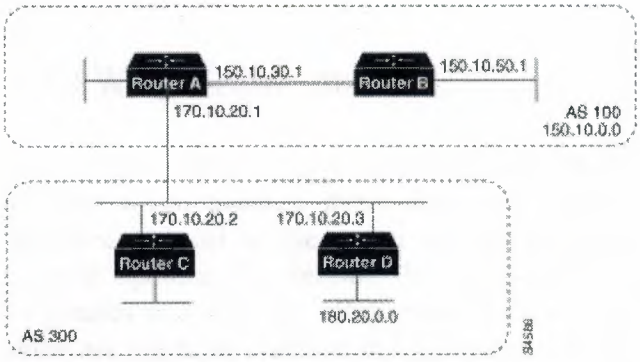


Figure -14: Next Hop Attribute and Multiaccess Media

In **Figure -14**, 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.

Next Hop Attribute and Nonbroadcast Media Access

In **Figure -15**, three networks are connected by a nonbroadcast media access (NBMA) cloud, such as Frame Relay.

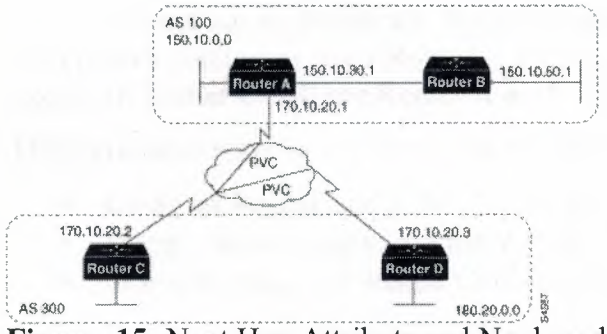


Figure -15: Next Hop Attribute and Nonbroadcast Media Access

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. To remedy this situation, use the **neighbor** next-hop-self router configuration command, as shown in the following configuration for Router C:

!Router C

router bgp 300
neighbor 170.10.20.1 remote-as 100
neighbor 170.10.20.1 next-hop-self

The neighbor next-hop-self command causes Router C to advertise 180.20.0.0 with the next hop attribute set to 170.10.20.2.

5.2.4 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 -16.

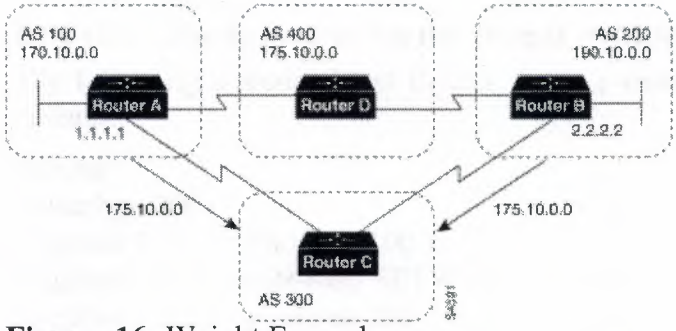


Figure -16: Weight Example

In **Figure -16**, 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

Using an Access List to Set the Weight Attribute

The following commands on Router C use access lists and the value of the AS_path attribute to assign a weight to route updates:

!Router C
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 filter-list 5 weight 2000
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 filter-list 6 weight 1000

```
!
ip as-path access-list 5 permit ^100$
ip as-path access-list 6 permit ^200$
```

In this example, 2000 is assigned to the weight attribute of updates from the neighbor at IP address 1.1.1.1 that are permitted by access list 5. Access list 5 permits updates whose AS_path attribute starts with 100 (as specified by ^) and ends with 100 (as specified by $). (The ^ and $ symbols are used to form regular expressions. For a complete explanation of regular expressions, see the appendix on regular expressions in the Cisco Internetwork Operating System (Cisco IOS) software configuration guides and command references.

This example also assigns 1000 to the weight attribute of updates from the neighbor at IP address 2.2.2.2 that are permitted by access list 6. Access list 6 permits updates whose AS_path attribute starts with 200 and ends with 200.

In effect, this configuration assigns 2000 to the weight attribute of all route updates received from AS 100 and assigns 1000 to the weight attribute of all route updates from AS 200.

By Using a Route Map to Set the Weight Attribute

The following commands on Router C use a route map to assign a weight to route updates:

```
!Router C
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-map SETWEIGHTIN in
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 route-map SETWEIGHTIN in
!
ip as-path access-list 5 permit ^100$
!
route-map SETWEIGHTIN permit 10
match as-path 5
set weight 2000
route-map SETWEIGHTIN permit 20
set weight 1000
```

This first instance of the setweightin route map assigns 2000 to any route update from AS 100, and the second instance of the setweightin route map assigns 1000 to route updates from any other AS.

By Using the neighbor weight Command to Set the Weight Attribute

The following configuration for Router C uses the **neighbor weight** router configuration command:

```
!Router C
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 weight 2000
neighbor 2.2.2.2 remote-as 200
```

neighbor 2.2.2.2 weight 1000
This configuration sets the weight of all route updates from AS 100 to 2000, and the weight of all route updates coming from AS 200 to 1000. The higher weight assigned to route updates from AS 100 causes Router C to send traffic through Router A.

5.2.5 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 only relevant 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 -17 demonstrates the local preference attribute.

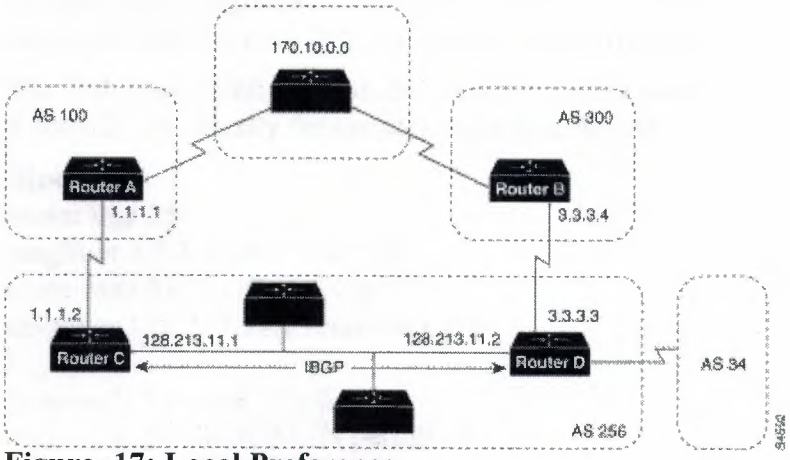


Figure -17: Local Preference

In Figure -17, 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

Using the bgp default local-preference Command

The following configurations use the **bgp default local-preference** router configuration command to set the local preference attribute on Routers C and D:

!Router C
router bgp 256
neighbor 1.1.1.1 remote-as 100
neighbor 128.213.11.2 remote-as 256
bgp default local-preference 150

!Router D

```
router bgp 256
neighbor 3.3.3.4 remote-as 300
neighbor 128.213.11.1 remote-as 256
bgp default local-preference 200
```

The configuration for Router C causes it to set the local preference of all updates from AS 300 to 150, and the configuration for Router D causes it to set the local preference for all updates from AS 100 to 200. Because local preference is exchanged within the AS, both Routers C and D determine that updates regarding network 170.10.0.0 have a higher local preference when they come from AS 300 than when they come from AS 100. As a result, all traffic in AS 256 destined for network 170.10.0.0 is sent to Router D as the exit point.

Using a Route Map to Set Local Preference

Route maps provide more flexibility than the bgp default local-preference router configuration command. When the bgp default local-preference command is used on Router D in **Figure -17**, the local preference attribute of all updates received by Router D will be set to 200, including updates from AS 34.

The following configuration uses a route map to set the local preference attribute on Router D specifically for updates regarding AS 300:

```
!Router D
router bgp 256
neighbor 3.3.3.4 remote-as 300
route-map SETLOCALIN in
neighbor 128.213.11.1 remote-as 256
!
ip as-path 7 permit ^300$
route-map SETLOCALIN permit 10
match as-path 7
set local-preference 200
!
Route-map SETLOCALIN permit 20
```

With this configuration, the local preference attribute of any update coming from AS 300 is set to 200. Instance 20 of the SETLOCALIN route map accepts all other routes.

5.2.6 **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.
Note In BGP Version 3, MED is known as Inter-AS_Metric.

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 -18 demonstrates the use of the MED attribute.

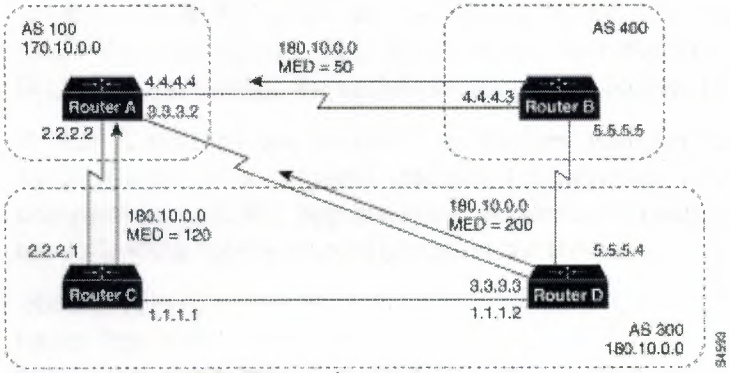


Figure -18: MED Example

In **Figure -18**, 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.

The following commands configure Routers A, B, C, and D:

```
!Router A
router bgp 100
neighbor 2.2.2.1 remote-as 300
neighbor 3.3.3.3 remote-as 300
neighbor 4.4.4.3 remote-as 400

!Router B
router bgp 400
neighbor 4.4.4.4 remote-as 100
neighbor 4.4.4.4 route-map SETMEDOUT out
neighbor 5.5.5.4 remote-as 300
!
route-map SETMEDOUT permit 10
set metric 50

!Router C
router bgp 300
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map SETMEDOUT out
neighbor 5.5.5.5 remote-as 400
neighbor 1.1.1.2 remote-as 300
!
route-map SETMEDOUT permit 10
set metric 120
```

```
!Router D
router bgp 300
neighbor 3.3.3.2 remote-as 100
neighbor 3.3.3.2 route map SETMEDOUT out
neighbor 1.1.1.1 remote-as 300
route-map SETMEDOUT permit 10
set metric 200
```

By default, BGP compares the MED attributes of routes coming from neighbors in the same external AS (such as AS 300 in Figure -18). Router A can only compare the MED attribute coming from Router C (120) to the MED attribute coming from Router D (200) even though the update coming from Router B has the lowest MED value.

Router A will choose Router C as the best path for reaching network 180.10.0.0. To force Router A to include updates for network 180.10.0.0 from Router B in the comparison, use the bgp always-compare-med router configuration command, as in the following modified configuration for Router A:

```
!Router A
router bgp 100
neighbor 2.2.2.1 remote-as 300
neighbor 3.3.3.3 remote-as 300
neighbor 4.4.4.3 remote-as 400
bgp always-compare-med
```

Router A will choose Router B as the best next hop for reaching network 180.10.0.0 (assuming that all other attributes are the same).

You can also set the MED attribute when you configure the redistribution of routes into BGP. For example, on Router B you can inject the static route into BGP with a MED of 50 as in the following configuration:

```
!Router B
router bgp 400
redistribute static
default-metric 50
!
ip route 160.10.0.0 255.255.0.0 null 0
```

The preceding configuration causes Router B to send out updates for 160.10.0.0 with a MED attribute of 50.

5.2.7 Community Attribute

The community attributes 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 -1.

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

Table -1: Predefined Communities

The following route maps set the value of the community attribute:

route-map COMMUNITYMAP
match ip address 1
set community no-advertise
!
route-map SETCOMMUNITY
match as-path 1
set community 200 additive

If you specify the additive keyword, the specified community value is added to the existing value of the community attribute. Otherwise, the specified community value replaces any community value that was set previously.

To send the community attribute to a neighbor, you must use the neighbor send-community router configuration command, as in the following example:

router bgp 100
neighbor 3.3.3.3 remote-as 300
neighbor 3.3.3.3 send-community
neighbor 3.3.3.3 route-map setcommunity out

For examples of how the community attribute is used to filter updates, see the section "Community Filtering," later in this chapter.

CONCLUSION

.

throughout the series chapter of the project, which is including and also will be illustrated the definitions, the structure and many application and to be more specific applications in network problems, applying the most important and essential technology among the other technologies which is routing , information about local area network (LAN) , some information about internet protocol and routing protocol . Network - A network is a group of computers connected together in a way that allows information to be exchanged between the computers, Local Area Network (LAN) - A LAN is a network of computers that are in the same general physical location, usually within a building or a campus. If the computers are far apart (such as across town or in different cities), then a Wide Area Network (WAN) is typically used. Routing is the act of moving information across an internetwork from a source to a destination. Along the way, at least one intermediate node typically is encountered. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer. The protocol is the pre-defined way that someone who wants to use a service talks with that service. The "someone" could be a person, but more often it is a computer program like a Web browser. Protocols are often text, and simply describe how the client and server will have their conversation.

REFERENCE

[1] Microsoft, networking essentials, Microsoft Corporation, Washington, 1996

[2] Thomas Robert M, introdction to local Area Network, sybex computer books Inc. U.S.A, 1996

[3] From the World Wide Web: www.cisco.com

[4] From Cisco Networking Academy Program: cisco.netacad.net

[5]TenebauM Andrew s., computer networks, 1996

[6] From world wide web:www.dti.gov.k/ent/coal