CCNA INTRODUCTION

CCNA INTRODUCTION

Ehsanullah Mohammadi

| BAND CHAK WARDAK AFGHANISTAN

Introduction

I have a book and I have prepared for you, my dear One who is interested in IT, especially the Network section, will not forget your best use of your prayers.

CCNA My main goal was to cover all the topics in this book. The one I made has some good things that can help you learn.

At the beginning of each chapter, a summary of the topics you will study in this chapter is written to help you become more familiar with the topics in this chapter and to help you learn with your loved ones.

This book also uses pictures and Configuration images to find out more.

The one book I've produced you can use its content I have tried my best to make a book for your friends that is used to the maximum.

And you friends can send us feedback on your good comments by email at eu.mohammadi2022@gamil.com.

I thank my father and my mother for their financial and spiritual encouragement to not forget their good prayers.

Thanks for your support my dear.

Best regards Ehsanullah Mohammadi

FRIST CHAPTER
WHAT IS NETWORK

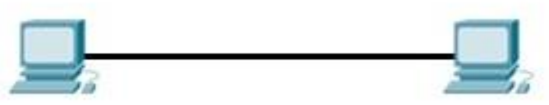
We Will Cover These Topics in This Chapter

- *What is Network?*
- *OSI and TCP/IP Model*
- *Type of Network*
- *Encapsulation*
- *Mac Address and IP Address*
- *The Type of Communication*
- *Network Device*
- *Half duplex and full duplex*
- *Unicast, multicast, and broadcast addresses*
- *IEEE Ethernet standards*

What is a network?

A computer network can be described as a system of interconnected devices that can communicate using some common standards (called protocols). These devices communicate to exchange resources (e.g. files and printers) and services.

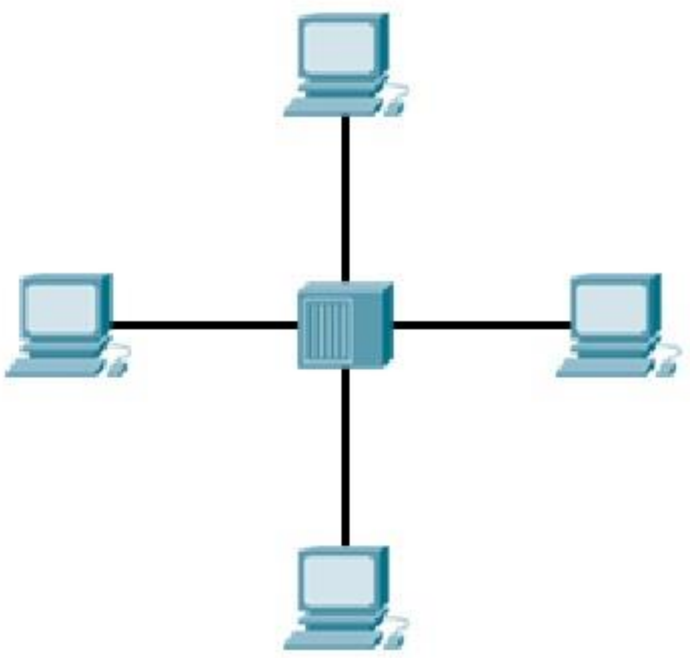
Here is an example network consisting of two computers connected together:



In the example above, the two computers are directly connected using a cable. This small network can be used to exchange data between just these two computers.

What if we want to expand our network?

Then we can use a network device, either a switch or a hub, to connect more than two computers together:



Now all of the devices on the network can communicate with each other.

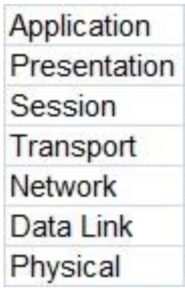
We'll talk more about hubs and switches in just a moment. For now, just remember that these devices serve as a central point to which all of the computers connect to.

OSI & TCP/IP models

OSI model

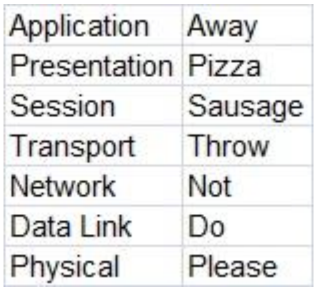
OSI (Open Systems Interconnection) model was created by the International Organization for Standardization (ISO), an international standard-setting body. It was designed to be a reference model for describing the functions of a communication system. The OSI model provides a framework for creating and implementing networking standards and devices and describes how network applications on different computers can communicate through the network media.

The OSI model has seven layers, with each layer describing a different function of data traveling through a network. Here is the graphical representation of these layers:



Application
Presentation
Session
Transport
Network
Data Link
Physical

The layers are usually numbered from the last one, meaning that the Physical layer is considered to be the first layer. It is useful to remember these layers, since there will certainly be a couple of questions on the CCNA exam regarding them. Most people learn the mnemonic „Please Do Not Throw Sausage Pizza Away“:



Application	Away
Presentation	Pizza
Session	Sausage
Transport	Throw
Network	Not
Data Link	Do
Physical	Please

So, what is the purpose of these layers?
They are most commonly used by vendors. They enable them to implement some functionality into a networking device, which then enables easier interoperability with devices from other vendors.

Here is a brief description of each of the layers of the OSI model.

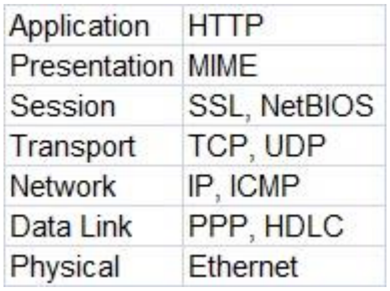
- *Physical – defines how to move bits from one device to another. It details how cables, connectors and network interface cards are supposed to work and how to send and receive bits.*
- *Data Link – encapsulates a packet in a frame. A frame contains a header and a trailer that enable devices to communicate. A header (most commonly) contains a source and destination MAC address. A trailer contains the Frame Check Sequence field, which is used to detect transmission errors. The data link layer has two sublayers:*

1. Logical Link Control – used for flow control and error detection.
2. Media Access Control – used for hardware addressing and for controlling the access method.

- *Network – defines device addressing, routing, and path determination. Device (logical) addressing is used to identify a host on a network (e.g. by its IP address).*
- *Transport – segments big chunks of data received from the upper layer protocols. Establishes and terminates connections between two computers. Used for flow control and data recovery.*
- *Session – defines how to establish and terminate a session between the two systems.*
- *Presentation – defines data formats. Compression and encryption are defined at this layer.*
- *Application – this layer is the closest to the user. It enables network applications to communicate with other network applications.*

It is a common practice to reference a protocol by the layer number or layer name. For example, HTTPS is referred to as an application (or Layer 7) protocol. Network devices are also sometimes described according to the OSI layer on which they operate – e.g. a Layer 2 switch or a Layer 7 firewall.

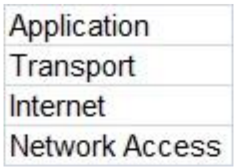
The following table shows which protocols reside on which layer of the OSI model:



Application	HTTP
Presentation	MIME
Session	SSL, NetBIOS
Transport	TCP, UDP
Network	IP, ICMP
Data Link	PPP, HDLC
Physical	Ethernet

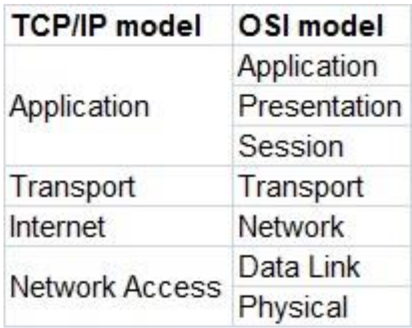
TCP/IP model

The TCP/IP model was created in the 1970s by the Defense Advance Research Project Agency (DARPA) as an open, vendor-neutral, public networking model. Just like the OSI model, it describes general guidelines for designing and implementing computer protocols. It consists of four layers: Network Access, Internet, Transport, and Application:



Application
Transport
Internet
Network Access

The following picture show the comparison between the TCP/IP model and OSI model:



TCP/IP model	OSI model
Application	Application
	Presentation
	Session
Transport	Transport
Internet	Network
Network Access	Data Link
	Physical

As you can see from the picture above, the TCP/IP model has fewer layers than the OSI model. The Application, Presentation, and Session layers of the OSI model are merged into a single layer in the TCP/IP model. Also, Physical and Data Link layers are called Network Access layer in the TCP/IP model. Here is a brief description of each layer:

- *Link – defines the protocols and hardware required to deliver data across a physical network.*
- *Internet – defines the protocols for the logical transmission of packets over the network.*
- *Transport – defines protocols for setting up the level of transmission service for applications. This layer is responsible for reliable transmission of data and the the error-free delivery of packets.*
- *Application – defines protocols for node-to-node application communication and provide services to the application software running on a computer.*

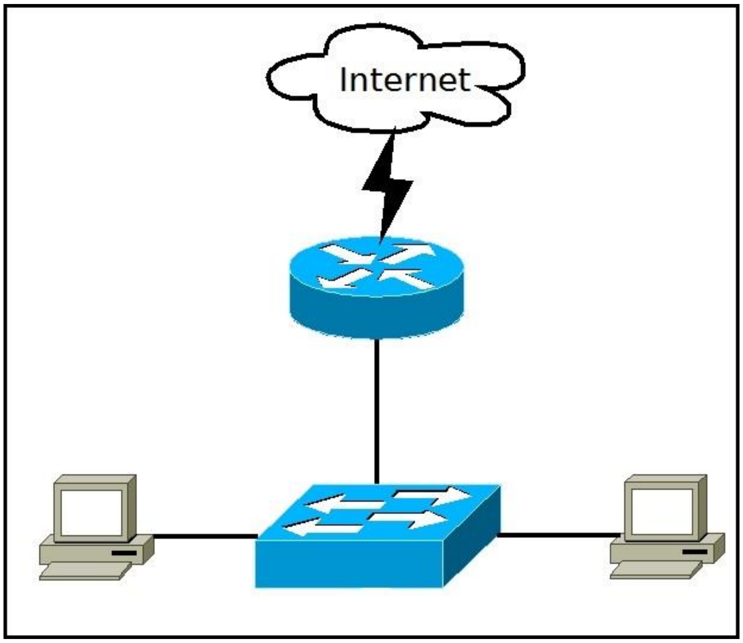
Differences between OSI and TCP/IP model

There are some other differences between these two models, besides the obvious difference in the number of layers. OSI model prescribes the steps needed to transfer data over a network and it is very specific in it, defining which protocol is used at each layer and how. The TCP/IP model is not that specific. It can be said that the OSI model prescribes and TCP/IP model describes.

Local area network (LAN)

The term local area network (LAN) is commonly used to describe a network of devices in a limited area (a house, office, building…). This type of network is usually capable of achieving high data transfer rate (up to 10 Gbps!) at low cost. Examples of this type of network are a small office network inside a single building or your home network.

A typical SOHO (small office/home office) LAN consist of PCs, printers, switches, routers, and cabling that connects all these devices together. The following figure shows a typical LAN:



In the picture above we have two computers that are connected to a switch. The switch is then connected to a router that provides the LAN with access to the Internet.

Some of the most popular LAN technologies are Ethernet, Token Ring and FDDI. Most LAN networks use TCP/IP to communicate. Twisted-pair cabling is usually used in a LAN.

Ethernet is by far the most popular wired LAN technology. It defines wiring, signaling, connectors, frame formats, protocol rules, etc. Most modern LANs also support the wireless LAN (WLAN) technology, defined by the IEEE 802.11 standards. WLANs use radio waves instead of wires or cables for links between devices.

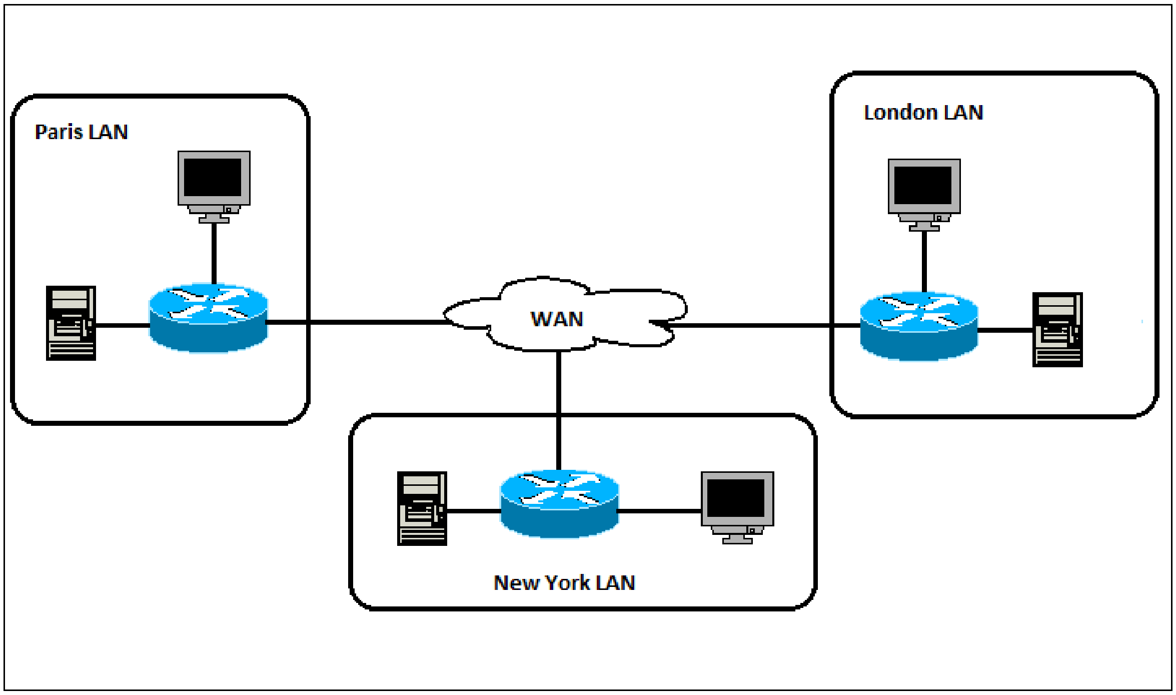
NOTE

The term metropolitan area network is used to describe a network in a single metropolitan area, hence the name. This type of network is usually bigger than a LAN and smaller than a WAN. An example of this type of network would be a network that connects two company offices inside the same city.

Wide area network

The term wide area network is used to describe a network that spans multiple geographic locations. Consider an example. A company has two offices, one in London and one in Berlin. Both offices have a LAN. If the company connects these two LANs together using WAN technology, a WAN is created.

The key difference between LANs and WANs is that the company usually doesn't own WAN infrastructure. A company usually leases WAN services from a service provider. A WAN spanning multiple cities could look something like this:



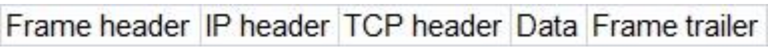
Frame Relay, ATM and X.25 are different types of WAN technologies. The Internet can also be considered a WAN.

Encapsulation

The term encapsulation is used to describe a process of adding headers and trailers around some data. This process can be explained with the four-layer TCP/IP model, with each step describing the role of the layer. For example, here is what happens when you send an email using your favorite email program (such as Outlook or Thunderbird):

1. *the email is sent from the Application layer to the Transport layer.*
2. *the Transport layer encapsulates the data and adds its own header with its own information, such as which port will be used and passes the data to the Internet layer*
3. *the Internet layer encapsulates the received data and adds its own header, usually with information about the source and destination IP addresses. The Internet layer than passes the data to the Network Access layer*
4. *the Network Access layer is the only layer that adds both a header and a trailer. The data is then sent through a physical network link.*

Here is a graphical representation of how each layer add its own information:



Frame header	IP header	TCP header	Data	Frame trailer

Each packet (header + encapsulated data) defined by a particular layer has a specific name:

- *Frame – encapsulated data defined by the Network Access layer. A frame can have both a header and a trailer.*
- *Packet – encapsulated data defined by the Network layer. A header contains the source and destination IP addresses.*
- *Segment – encapsulated data as defined by the Transport layer. Information such as the source and destination ports or sequence and acknowledgment numbers are included in the header.*

NOTE

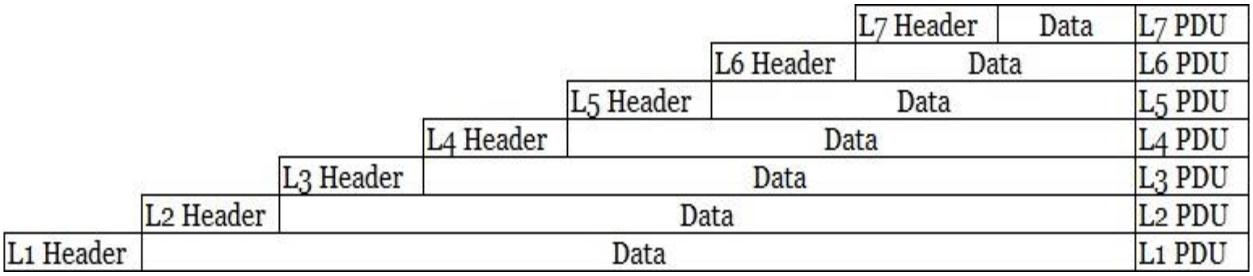
The term decapsulation refers to the process of removing headers and trailers as data passes from lower to upper layers. This process happens on the computer that is receiving data.

Data encapsulation in the OSI model

Just like with the TCP/IP layers, each OSI layer asks for services from the next lower layer. The lower layer encapsulates the higher layer's data between a header (Data Link protocols also add a trailer).

While the TCP/IP model uses terms like segment, packet and frame to refer to a data packet defined by a particular layer, the OSI model uses a different term: protocol data unit (PDU). A PDU represent a unit of data with headers and trailers for the particular layer, as well as the encapsulated data. Since the OSI model has 7 layers, PDUs are numbered from 1 to 7, with the Physical layer being the first one. For example, the term Layer 3 PDU refers to the data encapsulated at the Network layer of the OSI model.

Here is a graphical representation of all the PDUs in the OSI model:



						L7 Header	Data	L7 PDU
					L6 Header	Data		L6 PDU
				L5 Header	Data			L5 PDU
			L4 Header	Data				L4 PDU
		L3 Header	Data					L3 PDU
	L2 Header	Data						L2 PDU
L1 Header	Data							L1 PDU

Ethernet explained

Ethernet is the most used networking technology for LANs today. It defines wiring and signaling for the Physical layer of the OSI model. For the Data Link layer, it defines frame formats and protocols.

Ethernet is described as IEEE 802.3 standard. It uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and supports speeds up to 100 Gbps. It can use coaxial, twisted pair and fiber optic cables. Ethernet uses frames to with source and destination MAC addresses to deliver data.

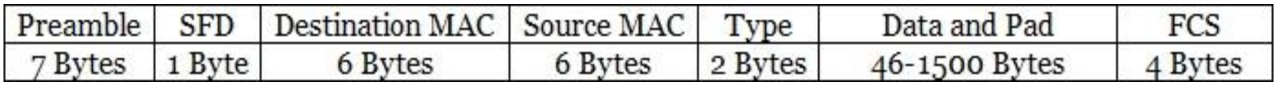
NOTE

The term Ethernet LAN refers to a combination of computers, switches, and different kinds of cables that use the Ethernet standard to communicate over the network. It is by far the most popular LAN technology today.

Ethernet frame

We have already learned that encapsulated data defined by the Network Access layer is called an Ethernet frame. An Ethernet frame starts with a header, which contains the source and destination MAC addresses, among other data. The middle part of the frame is the actual data. The frame ends with a field called Frame Check Sequence (FCS).

The Ethernet frame structure is defined in the IEEE 802.3 standard. Here is a graphical representation of an Ethernet frame and a description of each field in the frame:



Preamble	SFD	Destination MAC	Source MAC	Type	Data and Pad	FCS
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	46-1500 Bytes	4 Bytes

- *Preamble – informs the receiving system that a frame is starting and enables synchronization.*
- *SFD (Start Frame Delimiter) – signifies that the Destination MAC Address field begins with the next byte.*
- *Destination MAC – identifies the receiving system.*
- *Source MAC – identifies the sending system.*
- *Type – defines the type of protocol inside the frame, for example IPv4 or IPv6.*
- *Data and Pad – contains the payload data. Padding data is added to meet the minimum length requirement for this field (46 bytes).*
- *FCS (Frame Check Sequence) – contains a 32-bit Cyclic Redundancy Check (CRC) which allows detection of corrupted data.*

The FCS field is the only field present in the Ethernet trailer. It allows the receiver to discover whether errors occurred in the frame. Note that Ethernet only detects in-transit corruption of data – it does not attempt to recover a lost frame. Other higher level protocols (e.g. TCP) perform error recovery.

MAC & IP addresses

MAC address

A Media Access Control (MAC) address is a 48-bit (6 bytes) address that is used for communication between two hosts in an Ethernet environment. It is a hardware address, which means that it is stored in the firmware of the network card.

Every network card manufacturer gets a universally unique 3-byte code called the Organizationally Unique Identifier (OUI). Manufacturers agree to give all NICs a MAC address that begins with the assigned OUI. The manufacturer then assigns a unique value for the last 3 bytes, which ensures that every MAC address is globally unique.

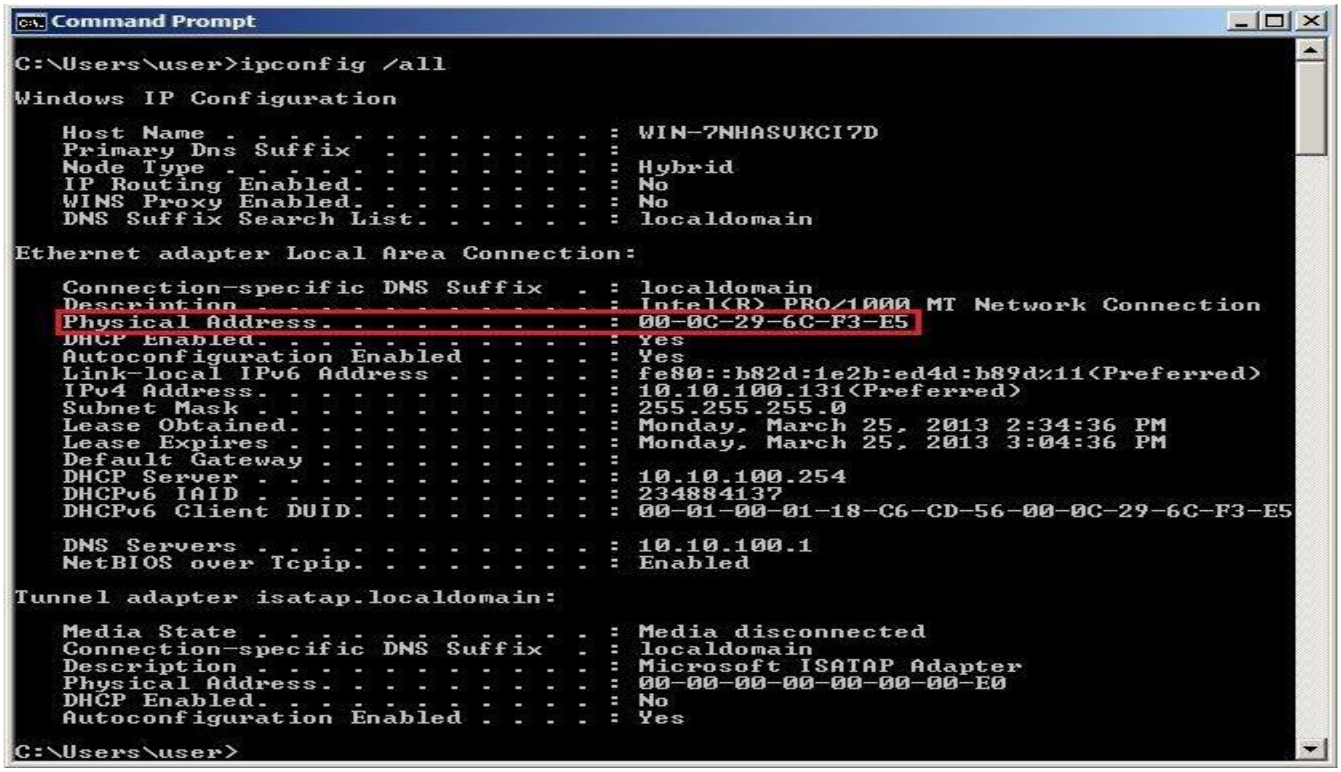
MAC addresses are usually written in the form of 12 hexadecimal digits. For example, consider the following MAC address:

```
D8-D3-85-EB-12-E3
```

Every hexadecimal character represents 4 bits, so the first six hexadecimal characters represent the vendor (Hewlett Packard in this case).

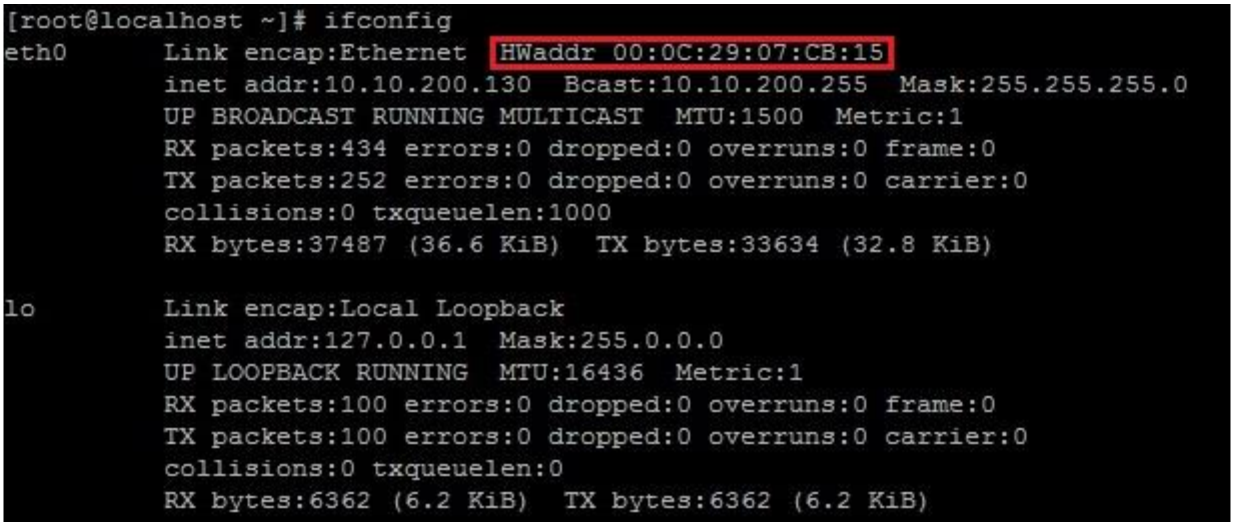
How to find out your own MAC address?

If you are using Windows, start the Command Prompt (Start – Programs – Accessories – Command Prompt). Type the ipconfig/all command and you should see a field called Physical Address under the Ethernet adapter settings:



```
C:\Users\user>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : WIN-7NHASVKCI7D
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : localdomain

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : localdomain
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-6C-F3-E5
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::b82d:1e2b:ed4d:b89d%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.10.100.131(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Monday, March 25, 2013 2:34:36 PM
   Lease Expires . . . . . . . . . . : Monday, March 25, 2013 3:04:36 PM
   Default Gateway . . . . . . . . . :
   DHCP Server . . . . . . . . . . . : 10.10.100.254
   DHCPv6 IAID . . . . . . . . . . . : 234884137
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-18-C6-CD-56-00-0C-29-6C-F3-E5

   DNS Servers . . . . . . . . . . . : 10.10.100.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter isatap.localdomain:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : localdomain
   Description . . . . . . . . . . . : Microsoft ISATAP Adapter
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes

C:\Users\user>
```

If you are using Linux, type the ifconfig command. You should see your MAC address referred to as HW address.



```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:07:CB:15
          inet addr:10.10.200.130  Bcast:10.10.200.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:434 errors:0 dropped:0 overruns:0 frame:0
          TX packets:252 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:37487 (36.6 KiB)  TX bytes:33634 (32.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:100 errors:0 dropped:0 overruns:0 frame:0
          TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6362 (6.2 KiB)  TX bytes:6362 (6.2 KiB)
```

IP address

An IP address is a 32-bit number that identifies a host on a network. Each device that wants to communicate with other devices on a TCP/IP network needs to have an IP address configured. For example, in order to access the Internet, your computer will need to have an IP address assigned (usually obtained by your router from the ISP).

An IP address is usually written in the form of four decimal numbers separated by periods (e.g. 10.0.50.1). The first part of the address represents the network the device is on (e.g. 10.0.0.0), while the second part of the address identifies the host device (e.g. 10.0.50.1).

In contrast to MAC address, an IP address is a logical address. It can be configured manually or it can be obtained from a DHCP server.

NOTE

The term IP address is usually used for IPv4, which is the fourth version of the IP protocol. A newer version exists, IPv6, and uses 128-bit addressing.

Private IP addresses

There are three ranges of addresses that can be used in a private network (e.g. your home LAN). These addresses are not routable through the Internet.

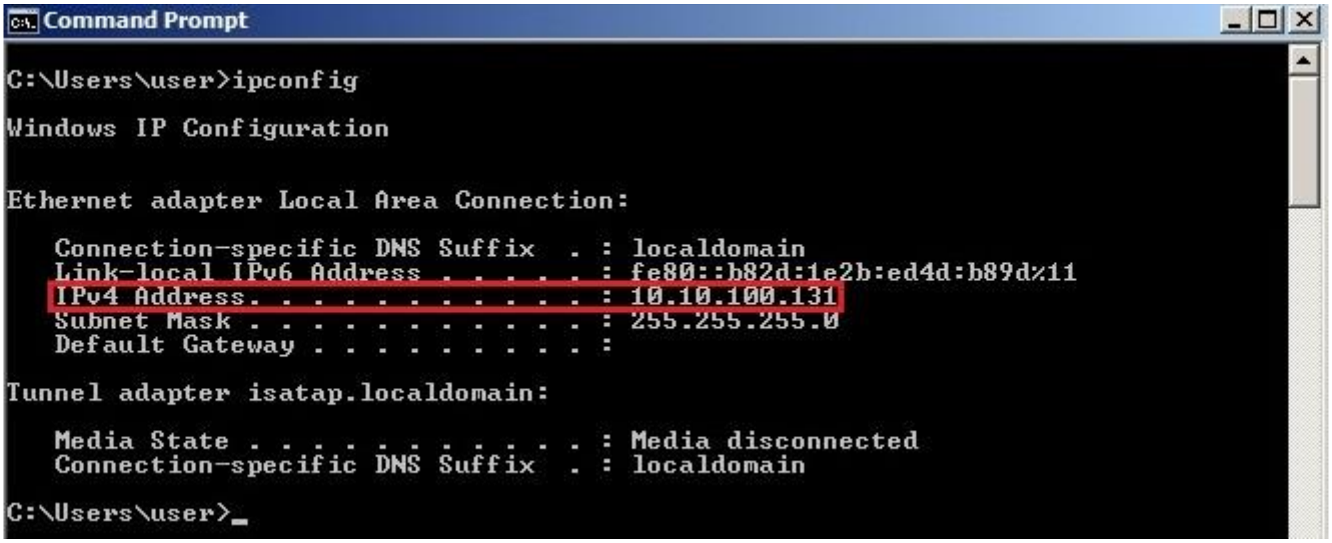
Private addresses ranges are:

- *10.0.0.0 – 10.255.255.255*

- *172.16.0.0 – 172.31.255.255*
- *192.168.0.0 – 192.168.255.255*

How to find out your IP address

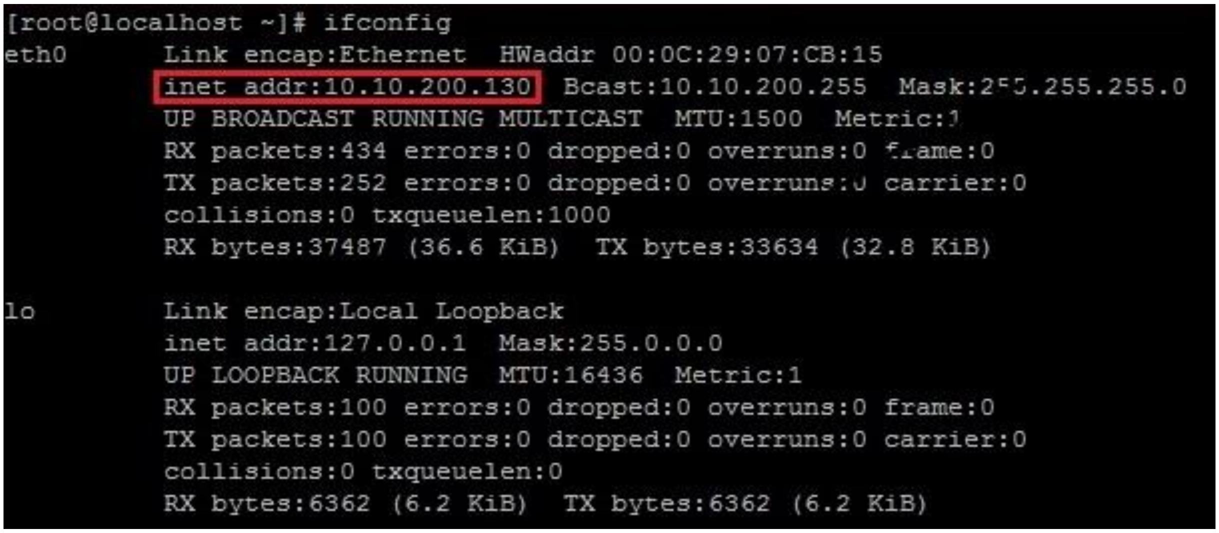
If you are using Windows, start the Command Prompt (Start – Programs – Accessories – Command Prompt). Enter the ipconfig command. You should see a field called IP Address:



```
C:\Users\user>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::b82d:1e2b:ed4d:b89d%11
   IPv4 Address. . . . . . . . . . . : 10.10.100.131
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Tunnel adapter isatap.localdomain:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : localdomain

C:\Users\user>_
```

Linux users:

Enter ifconfig. You should see a field called inetaddr:



```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:07:CB:15
          inet addr:10.10.200.130  Bcast:10.10.200.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:434 errors:0 dropped:0 overruns:0 frame:0
          TX packets:252 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:37487 (36.6 KiB)  TX bytes:33634 (32.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:100 errors:0 dropped:0 overruns:0 frame:0
          TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6362 (6.2 KiB)  TX bytes:6362 (6.2 KiB)
```

Unicast, multicast, and broadcast addresses

There are three types of Ethernet addresses:

- *unicast addresses – represent a single LAN interface. A unicast frame will be sent to a specific device, not to a group of devices on the LAN.*
- *multicast addresses – represent a group of devices in a LAN. A frame sent to a multicast address will be forwarded to a group of devices on the LAN.*
- *broadcast addresses – represent all device on the LAN. Frames sent to a broadcast address will be delivered to all devices on the LAN.*

The unicast address will have the value of the MAC address of the destination device.

Multicast frames have a value of 1 in the least-significant bit of the first octet of the destination address. This helps a network switch to distinguish between unicast and multicast addresses. One example of an Ethernet multicast address would be 01:00:0C:CC:CC:CC, which is an address used by CDP (Cisco Discovery Protocol).

The broadcast address has the value of FFFF.FFFF. FFFF (all binary ones). The switch will flood broadcast frames out all ports except the port that it was received on.

Network devices

Let's take a look at the network devices commonly found in today's LANs.

Hubs

A hub serves as a central point to which all of the hosts in a network connect to. A Hub is an OSI Layer 1 device and has no concept of Ethernet frames or addressing. It simply receives a signal from one port and sends it out to all other ports. Here is an example 4-port Ethernet hub (source: Wikipedia):

Today, hubs are considered obsolete and switches are commonly used instead. Hubs have numerous disadvantages. They are not aware of the traffic that passes through them. They create only one large collision domain. A hub typically operates in half duplex. There is also a security issue with hubs since the traffic is forwarded to all ports (except the source port), which makes it possible to capture all traffic on a network with a network sniffer!

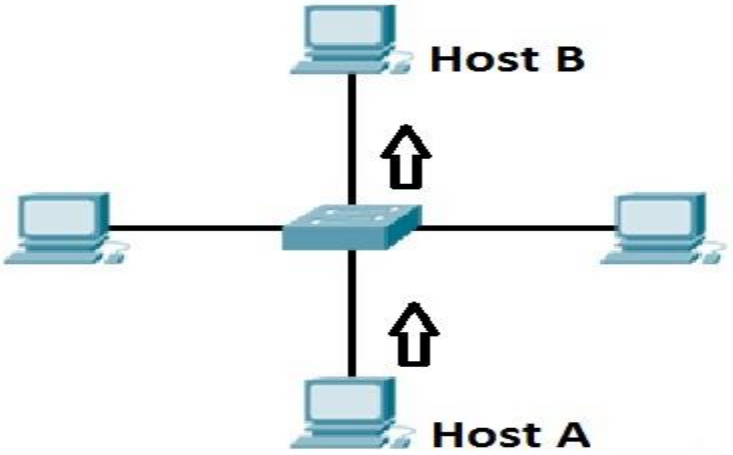
> *NOTE*
>
> *Hubs are sometimes called multiport repeaters.*

Switches

Like hubs, a switch is used to connect multiple hosts together, but it has many advantages over a hub. Switch is an OSI Layer 2 device, which means that it can inspect received traffic and make forwarding decisions. Each port on a switch is a separate collision domain and can run in a full duplex mode (photo credit: Wikipedia).

How switches work

Let's take a look at the following example:

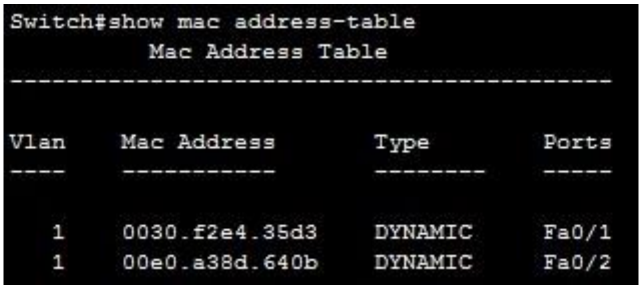


Host A is trying to communicate with Host B and sends a packet. A packet arrives at the switch, which looks at the destination MAC address. The switch then searches that address in its MAC address table. If the MAC address is found, the switch then forwards the packet only to the port that connected to the frame's destination. If the MAC address is not found, the switch will flood the frame out all other ports. To learn which MAC address is associated with which port, switches examine the source MAC addresses of the receiving packet and store that MAC addresses in their MAC address table.

What is a MAC address table?

A MAC address table lists which MAC address is connected to which port. It is used by switches to make forwarding decisions. The table is populated by examining the source MAC address of the incoming packet. If the source MAC address of a packet is not present in the table, the switch adds an entry to it's MAC address table.

The picture below show how a MAC address table on a switch looks like:



```
Switch#show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0030.f2e4.35d3    DYNAMIC     Fa0/1
   1    00e0.a38d.640b    DYNAMIC     Fa0/2
```

Routers

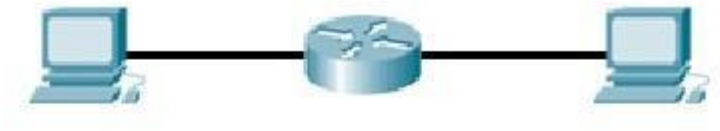
A router is a device that routes packets from one network to another. A router is most commonly an OSI Layer 3 device. Routers divide broadcast domains and have traffic filtering capabilities.

The picture below shows a typical home router:

How routers work

A router uses IP addresses to figure out where to send packets. If two hosts from different networks want to communicate, they will need a router between them to route packets

For example, consider the following example network:

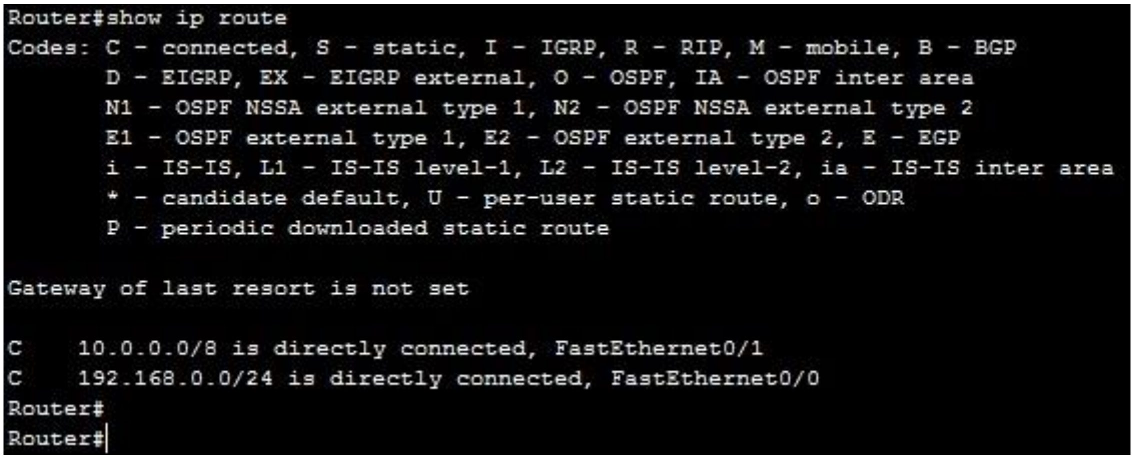


Host A and host B are on different networks. If host A wants to communicate with host B, it will have to send a packet to the router. The router receives the packet and checks the destination IP address. If the destination IP address is in the routing table, the router will forward the packet out the interface associated with that network.

What is a routing table?

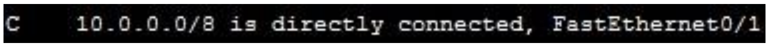
A routing table lists a route for every network that a router can reach. It can be statically configured (using IOS commands) or dynamically learned (using a routing protocol). It is used by routers when deciding where to forward packets.

The picture below shows how a routing table looks like:



```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/1
C    192.168.0.0/24 is directly connected, FastEthernet0/0
Router#
Router#
```

The command to display an IP routing table is show ip route. In the picture above, you can see that this router has two directly connected subnets. Let's take a closer look at the first entry in the routing table:



```
C    10.0.0.0/8 is directly connected, FastEthernet0/1
```

C means that the route is a directly connected route. The network in question is 10.0.0.0/8, and the router will forward each packet destined for that network out interface FastEthernet0/1.

NOTE

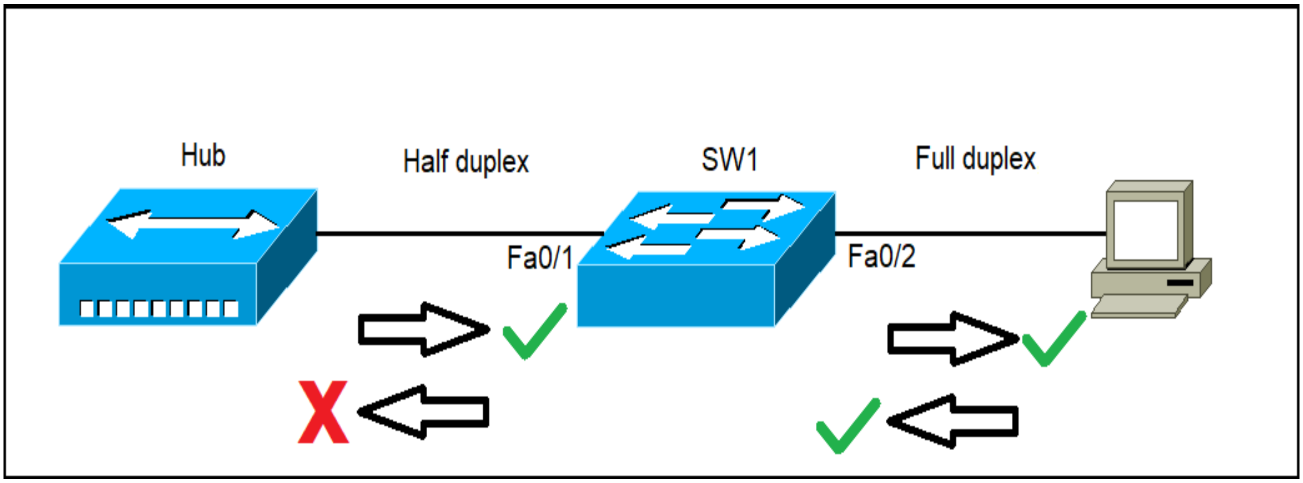
In Windows, you can use the netstat -r command to display the routing table of your system.

Half duplex and full duplex

In telecommunication, a duplex communication system is a point-to-point system of two devices that can communicate with each other in both direction. These two types of duplex communication systems exist in Ethernet environments:

- *half-duplex – a port can send data only when it is not receiving data. In other words, it cannot send and receive data at the same time. Network hubs run in half-duplex mode in order to prevent collisions. Since hubs are rare in modern LANs, the half-duplex system is not widely used in Ethernet networks anymore.*
- *full-duplex – all nodes can send and receive on their port at the same time. There are no collisions in full-duplex mode, but the host NIC and the switch port must support the full-duplex mode. Full-duplex Ethernet uses two pairs of wires at the same time instead of a single wire pair like half-duplex.*

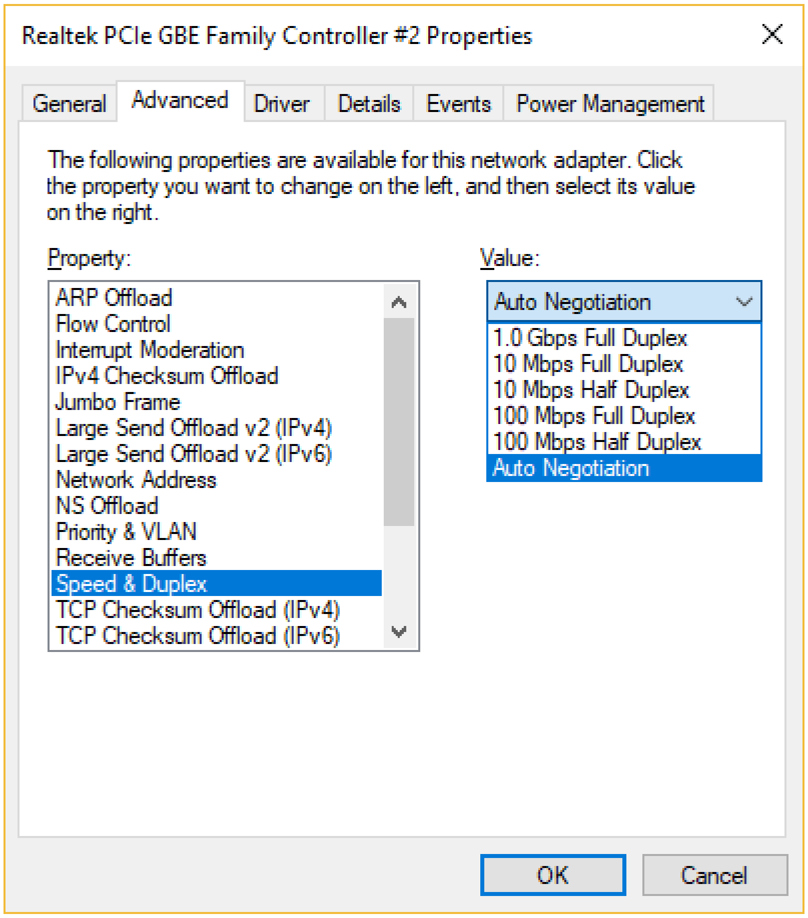
The following picture illustrates the concept:



Because hubs can only operate in half duplex, the switch and hub will negotiate to use half-duplex, which means that only one device can send data at the time. The workstation on the right supports full duplex, so the link between the switch and the workstation will use full duplex, with both devices sending data simultaneously.

Each NIC and switch port has a duplex setting. For all links between hosts and switches, or between switches, the full-duplex mode should be used. However, for all links connected to a LAN hub, the half-duplex mode should be used in order to prevent a duplex mismatch that could decrease network performance.

In Windows, you can set up duplex settings in the Properties window of your network adapter:



The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

ARP Offload
Flow Control
Interrupt Moderation
IPv4 Checksum Offload
Jumbo Frame
Large Send Offload v2 (IPv4)
Large Send Offload v2 (IPv6)
Network Address
NS Offload
Priority & VLAN
Receive Buffers
Speed & Duplex
TCP Checksum Offload (IPv4)
TCP Checksum Offload (IPv6)

Value:

Auto Negotiation

1.0 Gbps Full Duplex
10 Mbps Full Duplex
10 Mbps Half Duplex
100 Mbps Full Duplex
100 Mbps Half Duplex
Auto Negotiation

IEEE Ethernet standards

Ethernet is defined in a number of IEEE 802.3 standards. These standards define the physical and data-link layer specifications for Ethernet. The most important 802.3 standards are:

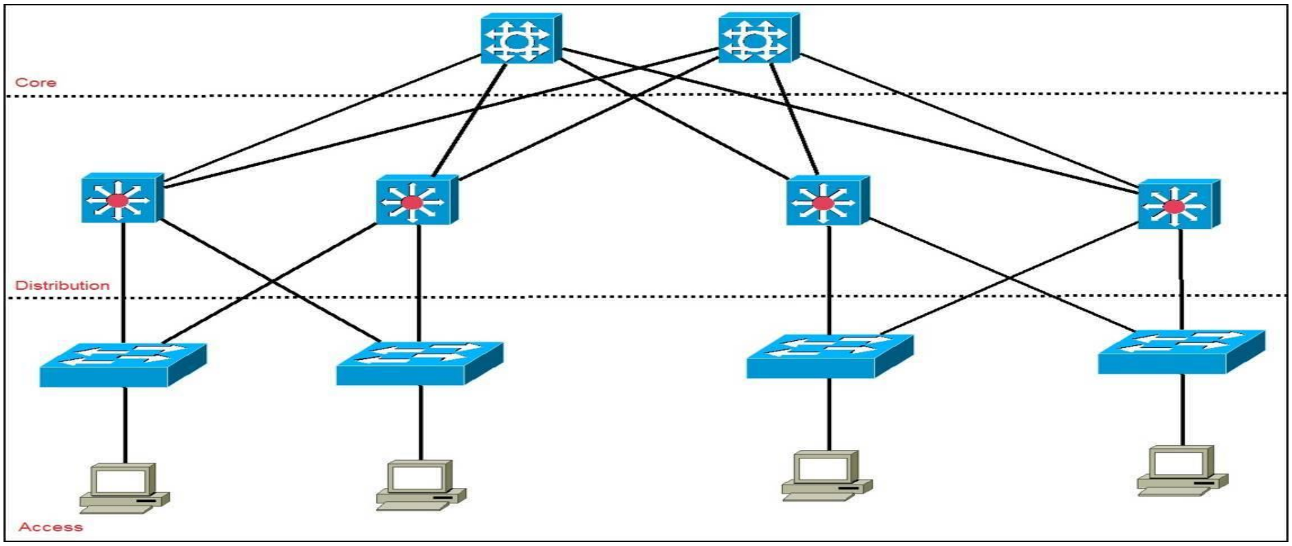
- *10Base-T (IEEE 802.3) – 10 Mbps with category 3 unshielded twisted pair (UTP) wiring, up to 100 meters long.*
- *100Base-TX (IEEE 802.3u) – known as Fast Ethernet, uses category 5, 5E, or 6 UTP wiring, up to 100 meters long.*
- *100Base-FX (IEEE 802.3u) – a version of Fast Ethernet that uses multi-mode optical fiber. Up to 412 meters long.*
- *1000Base-CX (IEEE 802.3z) – uses copper twisted-pair cabling. Up to 25 meters long.*
- *1000Base-T (IEEE 802.3ab) – Gigabit Ethernet that uses Category 5 UTP wiring. Up to 100 meters long.*
- *1000Base-SX (IEEE 802.3z) – 1 Gigabit Ethernet running over multimode fiber-optic cable.*
- *1000Base-LX (IEEE 802.3z) – 1 Gigabit Ethernet running over single-mode fiber.*
- *10GBase-T (802.3.an) – 10 Gbps connections over category 5e, 6, and 7 UTP cables.*

Notice how the first number in the name of the standard represents the speed of the network in megabits per second. The word base refers to baseband, meaning that the signals are transmitted without modulation. The last part of the standard name refers to the cabling used to carry signals. For example, 1000Base-T means that the speed of the network is up to 1000 Mbps, baseband signaling is used, and the twisted-pair cabling will be used (T stands for twisted-pair).

Cisco three-layer hierarchical model

Because networks can be extremely complicated, with multiple protocols and diverse technologies, Cisco has developed a layered hierarchical model for designing a reliable network infrastructure. This three-layer model helps you design, implement, and maintain a scalable, reliable, and cost-effective network. Each of layers has its own features and functionality, which reduces network complexity.

Here is an example of the Cisco hierarchical model:



Here is a description of each layer:

- *Access – controls user and workgroup access to the resources on the network. This layer usually incorporates Layer 2 switches and access points that provide connectivity between workstations and servers. You can manage access control and policy, create separate collision domains, and implement port security at this layer.*
- *Distribution – serves as the communication point between the access layer and the core. Its primary functions are to provide routing, filtering, and WAN access and to determine how packets can access the core. This layer determines the fastest way that network service requests are accessed – for example, how a file request is forwarded to a server – and, if necessary, forwards the request to the core layer. This layer usually consists of routers and multilayer switches.*
- *Core – also referred to as the network backbone, this layer is responsible for transporting large amounts of traffic quickly. The core layer provides interconnectivity between distribution layer*

devices it usually consists of high speed devices, like high end routers and switches with redundant links.

SECOND CHAPTER
TYPES OF ETHERNET CABLING

We Will Cover These Topics in This Chapter

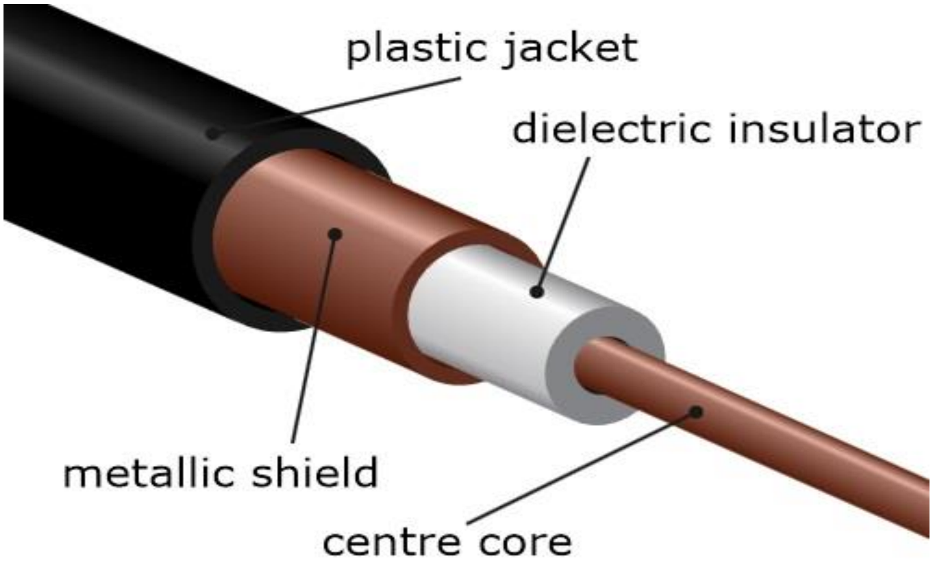
- ***Types of Ethernet cables straight-through and crossover***
- ***Types of Ethernet cabling***

Type OF Ethernet Cabling

There are three cable types commonly used for Ethernet cabling: coaxial, twisted pair, and fiber-optic cabling. In today's LANs, the twisted pair cabling is the most popular type of cabling, but the fiber-optic cabling usage is increasing, especially in high performance networks. Coaxial cabling is generally used for cable Internet access. Let's explain all three cable types in more detail.

Coaxial cabling

A coaxial cable has an inner conductor that runs down the middle of the cable. The conductor is surrounded by a layer of insulation which is then surrounded by another conducting shield, which makes this type of cabling resistant to outside interference. This type of cabling comes in two types – thinnet and thicknet. Both types have maximum transmission speed of 10 Mbps. Coaxial cabling was previously used in computer networks, but today are largely replaced by twisted-pair cabling (Photo credit: Wikipedia)

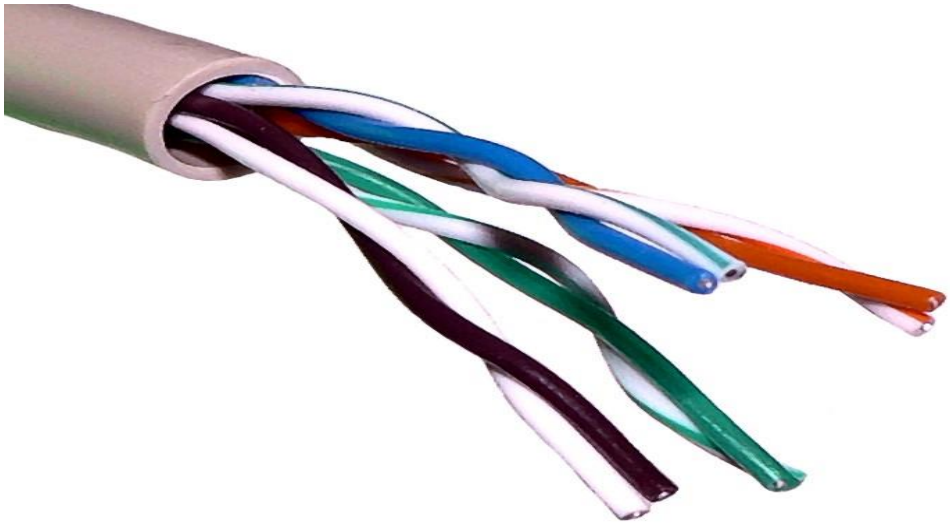


Twisted-pair cabling

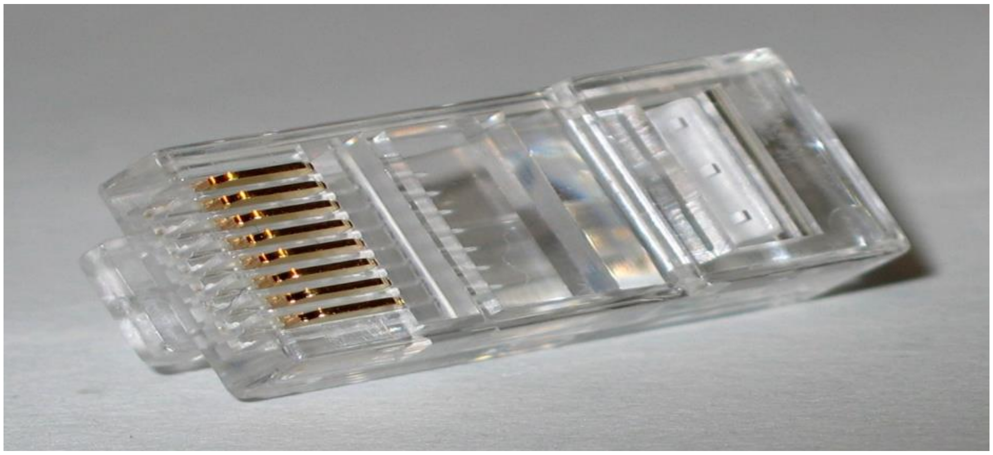
A twisted-pair cable has four pair of wires. These wires are twisted around each other to reduce crosstalk and outside interference. This type of cabling is common in current LANs.

Twisted-pair cabling can be used for telephone and network cabling. It comes in two versions, UTP (Unshielded Twisted-Pair) and STP (Shielded Twisted-Pair). The difference between these two is that an STP cable has an additional layer of insulation that protects data from outside interferences.

Here you can see how a twisted pair cable looks like (Photo credit: Wikipedia):

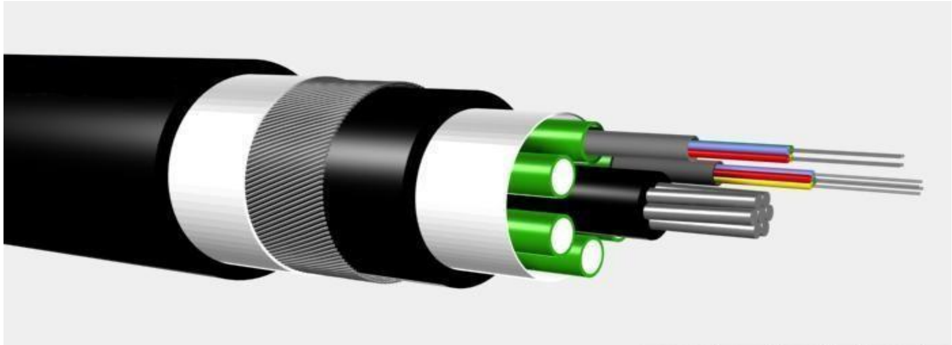


A twisted-pair cable uses 8P8C connector, sometimes wrongly referred to as RJ45 connector (Photo credit: Wikipedia).



Fiber-optic cabling

This type of cabling uses optical fibers to transmit data in the form of light signals. The cables have strands of glass surrounded by a cladding material (Photo credit: Wikipedia).



This type of cabling can support greater cable lengths than any other cabling type (up to a couple of miles). The cables are also immune to electromagnetic interference. As you can see, this cabling method has many advantages over other methods but its main drawback is that it is more expensive.

There are two types of fiber-optic cables:

- *Single-mode fiber (SMF) – uses only a single ray of light to carry data.*
- *Multi-mode fiber (MMF) – uses multiple rays of light to carry data.*

Two types of connectors are commonly used:

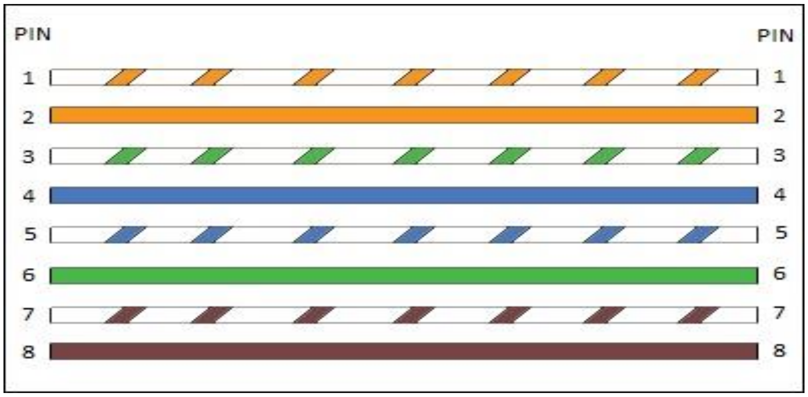
- *ST (Straight-tip connector)*
- *SC (Subscriber connector)*

Types of Ethernet cables – straight-through and crossover

Ethernet cables can come in two forms when it comes to wiring:

1. Straight-through cable

This cable type has identical wiring on both ends (pin 1 on one end of the cable is connected to pin 1 at the other end of the cable, pin 2 is connected to pin 2 etc.):



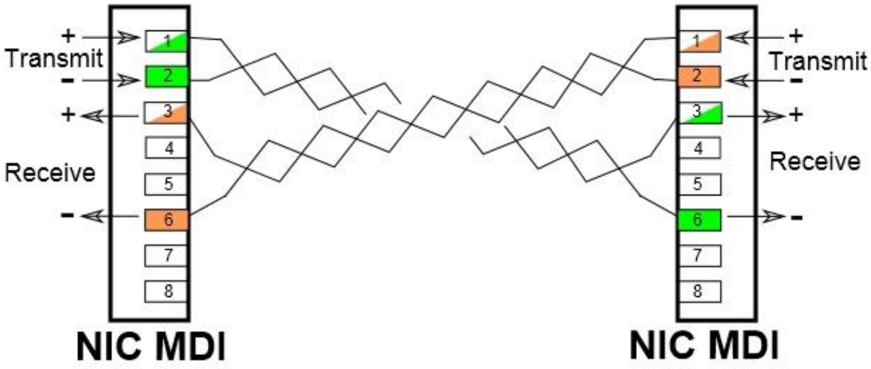
This type of cable is used to connect the following devices:

- *computer to hub*
- *computer to switch*
- *router to hub*
- *router to switch*

Computers and routers use wires 1 and 2 to transmit data and wires 3 and 6 to receive data. Hubs and switches use wires 1 and 2 to receive data and wires 3 and 6 to send data. That is why, if you want to connect two computers together, you will need a crossover cable.

2. Crossover cable

With the crossover cable, the wire pairs are swapped, which means that different pins are connected together – pin 1 on one end of the cable is connected to pin 3 on the other end, pin 2 on one end is connected to pin 6 on the other end (Photo credit: Wikipedia):



This type of cable is used when you need to connect two devices that use same wires to send and receive data. For example, consider connecting two computers together. If you use straight-through cable, with identical wiring in both ends, both computers will use wires 1 and 2 to send data. If computer A sends some packets to computer B, computer A will send that data using wires 1 and 2. That will cause a problem because computers expect packets to be received on wires 3 and 6, and your network will not work properly. This is why you need to use a crossover cable for such connections.

NOTE

Newer devices support the Auto MDI-X capability to automatically detect and configure the required cable connection type. This removes the need for a specific cable type between certain devices. Also, note that the Gigabit Ethernet and faster standards use all four wire pairs to transfer data in both direction simultaneously.

THIRD CHAPTER
WHAT IS IP ADDRESS

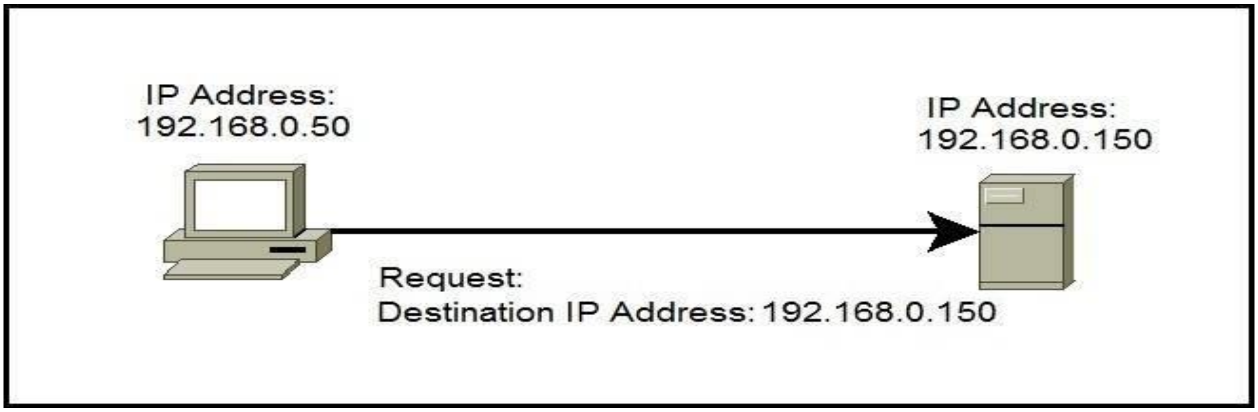
We Will Cover These Topics in This Chapter

- *Types of IP addresses*
- *Classes of IP addresses*
- *Subnetting explained*
- *Subnet mask*
- *Create subnets*
- *CIDR (Classless inter-domain routing)*

Types of IP addresses

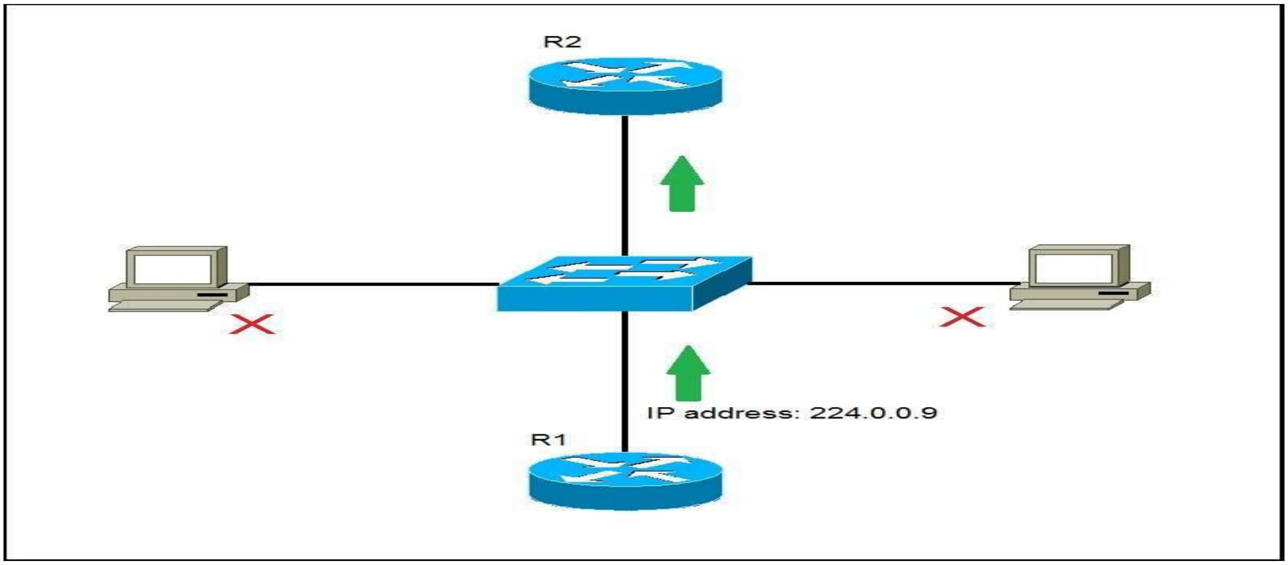
The IP addresses are divided into three different types, based on their operational characteristics:

1. unicast IP addresses – an address of a single interface. The IP addresses of this type are used for one-to-one communication. Unicast IP addresses are used to direct packets to a specific host. Here is an example:



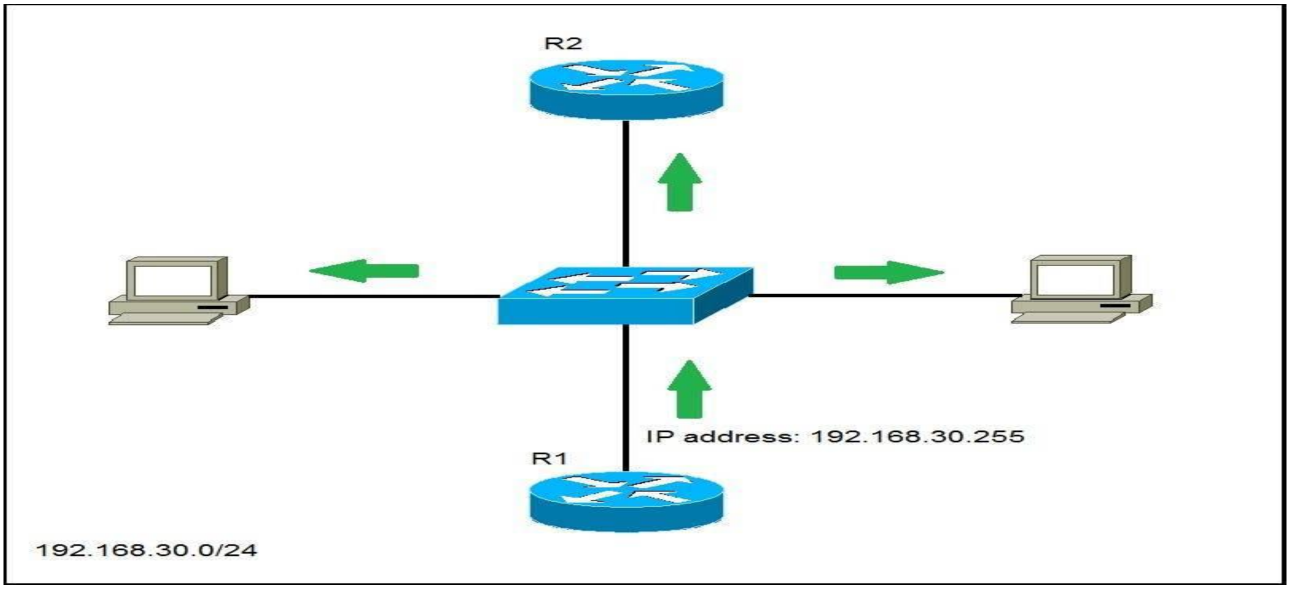
In the picture above you can see that the host wants to communicate with the server. It uses the (unicast) IP address of the server (192.168.0.150) to do so.

2. multicast IP addresses – used for one-to-many communication. Multicast messages are sent to IP multicast group addresses. Routers forward copies of the packet out to every interface that has hosts subscribed to that group address. Only the hosts that need to receive the message will process the packets. All other hosts on the LAN will discard them. Here is an example:



R1 has sent a multicast packet destined for 224.0.0.9. This is an RIPv2 packet, and only routers on the network should read it. R2 will receive the packet and read it. All other hosts on the LAN will discard the packet.

3. broadcast IP addresses – used to send data to all possible destinations in the broadcast domain (the one-to-everybody communication). The broadcast address for a network has all host bits on. For example, for the network 192.168.30.0 255.255.255.0 the broadcast address would be 192.168.0.255. Also, the IP address of all 1's (255.255.255.255) can be used for local broadcast. Here's an example:



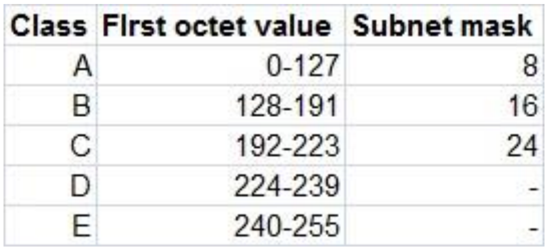
R1 wants to communicate with all hosts on the network and has sent a broadcast packet to the broadcast IP address of 192.168.30.255. All hosts in the same broadcast domain will receive and process the packet.

Classes of IP addresses

TCP/IP defines five classes of IP addresses: class A, B, C, D, and E. Each class has a range of valid IP addresses. The value of the first octet determines the class. IP addresses from the first three classes (A, B and C) can be used for host addresses. The other two classes are used for other purposes – class D for multicast and class E for experimental purposes.

The system of IP address classes was developed for the purpose of Internet IP addresses assignment. The classes created were based on the network size. For example, for the small number of networks with a very large number of hosts, the Class A was created. The Class C was created for numerous networks with small number of hosts.

Classes of IP addresses are:



Class	First octet value	Subnet mask
A	0-127	8
B	128-191	16
C	192-223	24
D	224-239	-
E	240-255	-

For the IP addresses from Class A, the first 8 bits (the first decimal number) represent the network part, while the remaining 24 bits represent the host part. For Class B, the first 16 bits (the first two numbers) represent the network part, while the remaining 16 bits represent the host part. For Class C, the first 24 bits represent the network part, while the remaining 8 bits represent the host part.

Consider the following IP addresses:

- *10.50.120.7 – because this is a Class A address, the first number (10) represents the network part, while the remainder of the address represents the host part (50.120.7). This means that, in order for devices to be on the same network, the first number of their IP addresses has to be the same for both devices. In this case, a device with the IP address of 10.47.8.4 is on the same network as the device with the IP address listed above. The device with the IP address 11.5.4.3 is not on the same network, because the first number of its IP address is different.*
- *172.16.55.13 – because this is a Class B address, the first two numbers (172.16) represent the network part, while the remainder of the address represents the host part (55.13). A device with the IP address of 172.16.254.3 is on the same network, while a device with the IP address of 172.55.54.74 isn't.*

NOTE
The system of network address ranges described here is generally bypassed today by use of the Classless Inter-Domain Routing (CIDR) addressing.

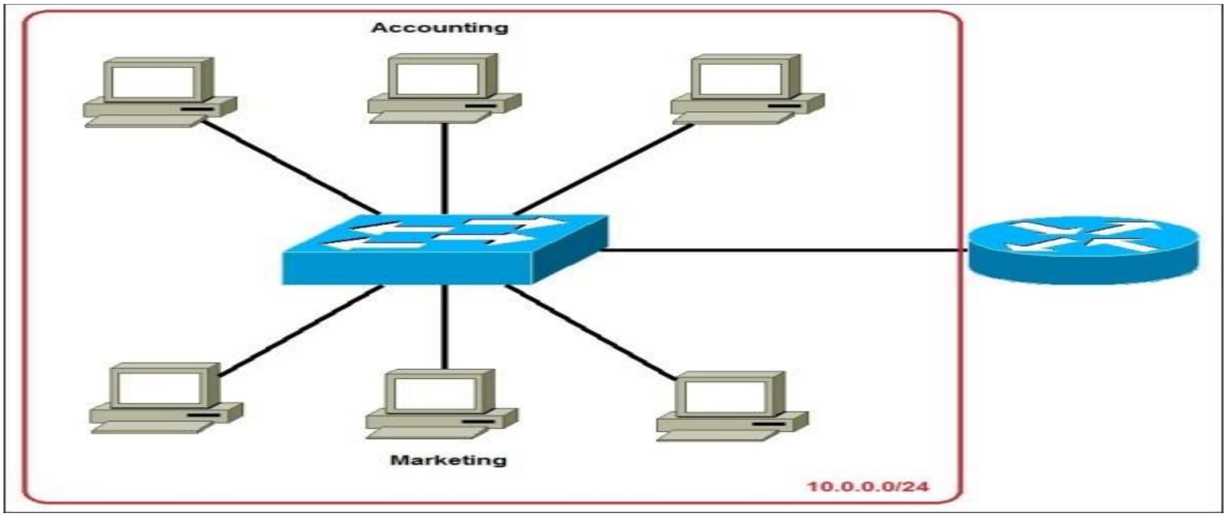
Special IP address ranges that are used for special purposes are:

- *0.0.0.0/8 – addresses used to communicate with the local network*
- *127.0.0.0/8 – loopback addresses*
- *169.254.0.0/16 – link-local addresses (APIPA)*

Subnetting explained

Subnetting is the practice of dividing a network into two or more smaller networks. It increases routing efficiency, enhances the security of the network and reduces the size of the broadcast domain.

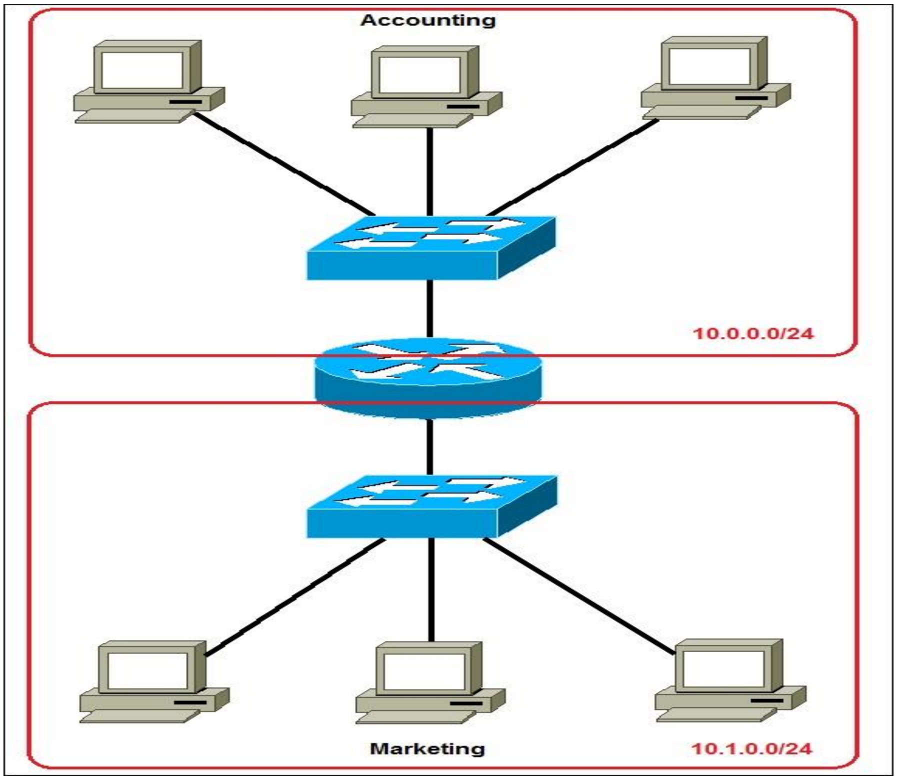
Consider the following example:



In the picture above we have one huge network: 10.0.0.0/24. All hosts on the network are in the same subnet, which has following disadvantages:

- *a single broadcast domain – all hosts are in the same broadcast domain. A broadcast sent by any device on the network will be processed by all hosts, creating lots of unnecessary traffic.*
- *network security – each device can reach any other device on the network, which can present security problems. For example, a server containing sensitive information shouldn't be in the same network as an user workstation.*
- *organizational problems – in a large networks, different departments are usually grouped into different subnets. For example, you can group all devices from the Accounting department in the same subnet and then give access to sensitive financial data only to hosts from that subnet.*

The network above could be submitted like this:

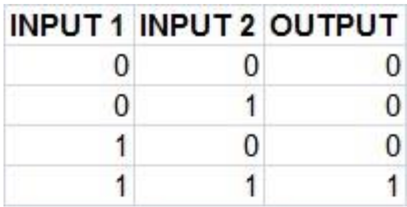


Now, two subnets were created for different departments: 10.0.0.0/24 for Accounting and 10.1.0.0/24 for Marketing. Devices in each subnet are now in a different broadcast domain. This will reduce the amount of traffic flowing on the network and allow us to implement packet filtering on the router.

Subnet mask

An IP address is divided into two parts: network and host parts. For example, an IP class A address consists of 8 bits identifying the network and 24 bits identifying the host. This is because the default subnet mask for a class A IP address is 8 bits long. (or, written in dotted decimal notation, 255.0.0.0). What does it mean? Well, like an IP address, a subnet mask also consists of 32 bits. Computers use it to determine the network part and the host part of an address. The 1s in the subnet mask represent a network part, the 0s a host part.

Computers works only with bits. The math used to determine a network range is binary AND.

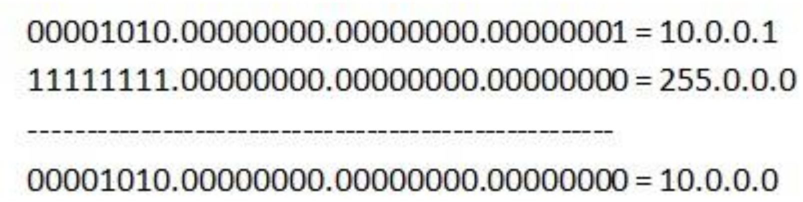


INPUT 1	INPUT 2	OUTPUT
0	0	0
0	1	0
1	0	0
1	1	1

Let's say that we have the IP address of 10.0.0.1 with the default subnet mask of 8 bits (255.0.0.0).
First, we need to convert the IP address to binary:

IP address: 10.0.0.1 = 00001010.00000000.00000000.00000001
Subnet mask 255.0.0.0 = 11111111.00000000.00000000.0000000

Computers then use the AND operation to determine the network number:



```
00001010.00000000.00000000.00000001 = 10.0.0.1
11111111.00000000.00000000.00000000 = 255.0.0.0
--------------------------------------------------------
00001010.00000000.00000000.00000000 = 10.0.0.0
```

The computer can then determine the size of the network. Only IP addresses that begins with 10 will be in the same network. So, in this case, the range of addresses in this network is 10.0.0.0 – 10.255.255.255.

NOTE

A subnet mask must always be a series of 1s followed by a series of 0s.

Create subnets

There are a couple of ways to create subnets. In this article we will subnet a class C address 192.168.0.0 that, by default, has 24 subnet bits and 8 host bits.

Before we start subnetting, we have to ask ourselves these two questions:

1. How many subnets do we need?

2x = number of subnets. x is the number of 1s in the subnet mask. With 1 subnet bit, we can have 21 or 2 subnets. With 2 bits, 22 or 4 subnets, with 3 bits, 23 or 8 subnets, etc.

2. How many hosts per subnet do we need?

2y – 2 = number of hosts per subnet. y is the number of 0s in the subnet mask.

Subnetting example

An example will help you understand the subnetting concept. Let's say that we need to subnet a class C address192.168.0.0/24. We need two subnets with 50 hosts per subnet. Here is our calculation:

1. Since we need only two subnets, we need 21 subnet bits. In our case, this means that we will take one bit from the host part. Here is the calculation:

First, we have a class C address 192.168.0.0 with the subnet mask of 24. Let's convert them to binary:

192.168.0.0 = 11000000.10101000.00000000.00000000
255.255.255.0 = 11111111.11111111.11111111.00000000

We need to take covert a single zero from the host part of the subnet mask. Here is our new subnet mask:

255.255.255.128 = 11111111.11111111.11111111.10000000

Remember, the ones in the subnet mask represent the network.

2. We need 50 hosts per subnet. Since we took one bit from the host part, we are left with seven bits for the hosts. Is it enough for 50 hosts? The formula to calculate the number of hosts is 2y – 2, with y representing the number of host bits. Since 27 – 2 is 126, we have more than enough bits for our hosts.

3. Our network will look like this:

192.168.0.0/25 – the first subnet has the subnet number of 192.168.0.0. The range of IP addresses in this subnet is 192.168.0.0 – 192.168.0.127.

192.168.0.128/25 – the second subnet has the subnet number of 192.168.0.128. The range of IP addresses in this subnet is 192.168.0.128 – 192.168.0.255.

CIDR (Classless inter-domain routing)

CIDR (Classless inter-domain routing) is a method of public IP address assignment. It was introduced in 1993 by Internet Engineering Task Force with the following goals:

- to deal with the IPv4 address exhaustion problem
- to slow down the growth of routing tables on Internet routers

Before CIDR, public IP addresses were assigned based on the class boundaries:

- Class A – the classful subnet mask is /8. The number of possible IP addresses is 16,777,216 (2 to the power of 24).
- Class B – the classful subnet mask is /16. The number of addresses is 65,536
- Class C – the classful subnet mask is /24. Only 256 addresses available.

Some organizations were known to have gotten an entire Class A public IP address (for example, IBM got all the addresses in the 9.0.0.0/8 range). Since these addresses can't be assigned to other companies, there was a shortage of available IPv4 addresses. Also, since IBM probably didn't need more than 16 million IP addresses, a lot of addresses were unused.

To combat this, the classful network scheme of allocating the IP address was abandoned. The new system was classless – a classful network was split into multiple smaller networks. For example, if a company needs 12 public IP addresses, it would get something like this: 190.5.4.16/28.

The number of usable IP addresses can be calculated with the following formula:

2 to the power of host bits – 2

In the example above, the company got 14 usable IP addresses from the 190.5.4.16 – 190.5.4.32 range because there are 4 host bits and 2 to the power of 4 minus 2 is 14 The first and the last address are the network address and the broadcast address, respectively. All other addresses inside the range could be assigned to Internet hosts.

FORTH CHAPTER
NETWORK TOOLS

We Will Cover These Topics in This Chapter

- ***Ping explained***
- ***Traceroute explained***

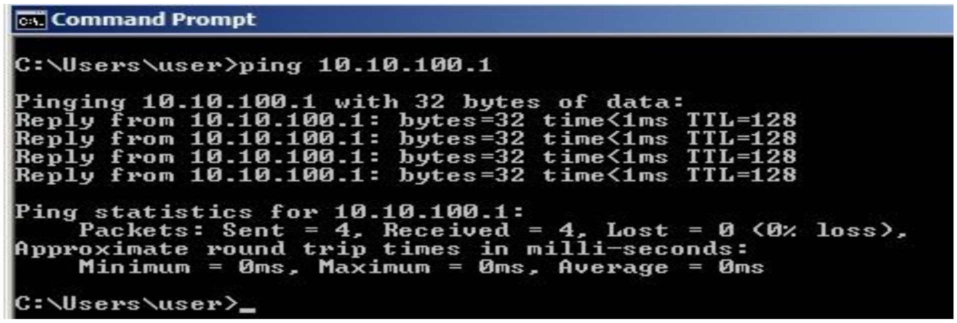
Network Tools

Ping explained

ping is perhaps the most commonly used tool to troubleshoot a network. Ping (Packet Internet Groper) is included with most operating systems. It is invoked using a ping command and uses ICMP (Internet Control Message Protocol) to reports errors and provides information related to IP packet processing. Ping works by sending an ICMP echo request message to the specified IP address. If the computer with the destination IP address is reachable, it responds with an ICMP echo reply message.

A ping command usually outputs some other information about a network performance, e.g. a round-trip time, a time to send an ICMP request packet and receive an ICMP reply packet.

Here is an output of the ping command from Windows 7:



```
C:\Users\user>ping 10.10.100.1

Pinging 10.10.100.1 with 32 bytes of data:
Reply from 10.10.100.1: bytes=32 time<1ms TTL=128
Reply from 10.10.100.1: bytes=32 time<1ms TTL=128
Reply from 10.10.100.1: bytes=32 time<1ms TTL=128
Reply from 10.10.100.1: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.100.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\user>_
```

In the example above we have pinged the ip address 10.10.100.1. By default, ping on Windows sends four ICMP request packets. As you can see from the output above, the host with the IP address of 10.10.100.1 is reachable and has replied with four ICMP reply packets. You can also see that the remote host has replied within 1 MS (time<1ms), which indicates that the network is not congested.

Traceroute explained

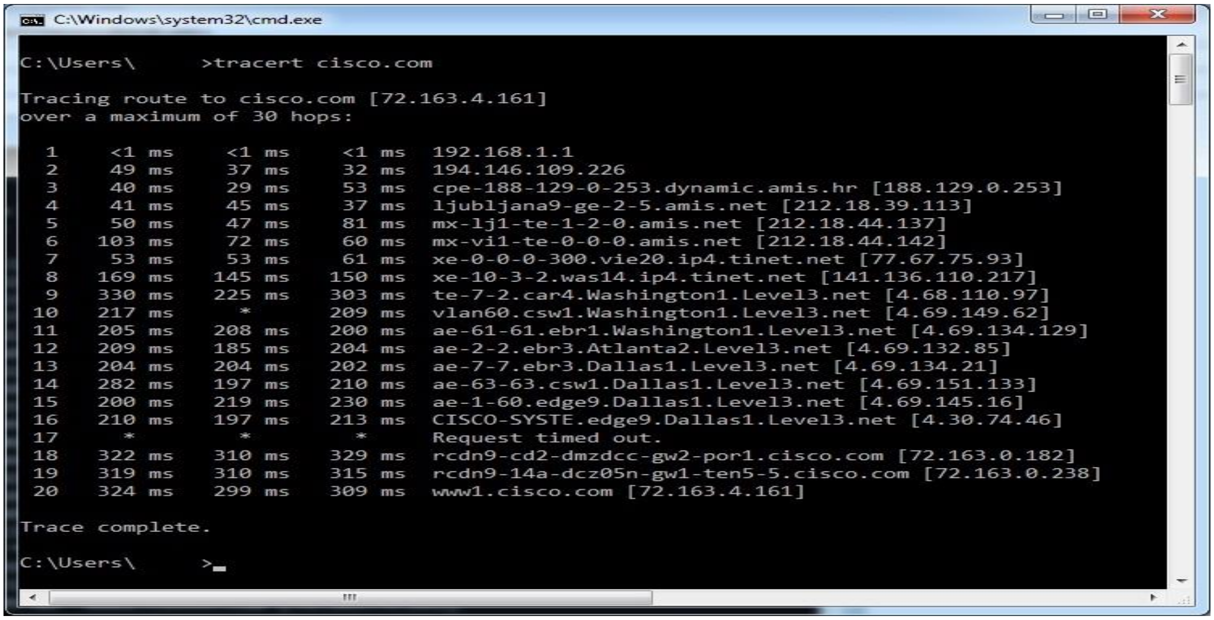
Traceroute is a command-line interface based tool used to identify the path used by a packet to reach its target. This tool also uses ICMP messages, but unlike ping, it identifies every router in a path taken by the packets. Traceroute is useful when troubleshooting network problems because it can help identify where exactly the problem is. You can figure out which router in the path to an unreachable target should be examined more closely as the probable cause of the network's failure.

Traceroute sends a series of ICMP echo request packets to a destination. First series of messages has a Time to Live (TTL) parameter set to 1, which means that the first router in a path will discard the packet and send an ICMP Time Exceeded message. TTL is then increased by one until the destination host is reached and an ICMP echo reply message is received. Originating host can then use received ICMP messages to identify all routers in a path.

NOTE

The traceroute command on Windows is named tracert. On Unix and Cisco IOS traceroute it is invoked using the traceroute command.

Here is an example of using the tracert command in Windows:



```
C:\Users\      >tracert cisco.com

Tracing route to cisco.com [72.163.4.161]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.1.1
  2    49 ms    37 ms    32 ms  194.146.109.226
  3    40 ms    29 ms    53 ms  cpe-188-129-0-253.dynamic.amis.hr [188.129.0.253]
  4    41 ms    45 ms    37 ms  ljubljana9-ge-2-5.amis.net [212.18.39.113]
  5    50 ms    47 ms    81 ms  mx-lj1-te-1-2-0.amis.net [212.18.44.137]
  6   103 ms    72 ms    60 ms  mx-vi1-te-0-0-0.amis.net [212.18.44.142]
  7    53 ms    53 ms    61 ms  xe-0-0-0-300.vie20.ip4.tinet.net [77.67.75.93]
  8   169 ms   145 ms   150 ms  xe-10-3-2.was14.ip4.tinet.net [141.136.110.217]
  9   330 ms   225 ms   303 ms  te-7-2.car4.Washington1.Level3.net [4.68.110.97]
 10   217 ms     *      209 ms  vlan60.csw1.Washington1.Level3.net [4.69.149.62]
 11   205 ms   208 ms   200 ms  ae-61-61.ebr1.Washington1.Level3.net [4.69.134.129]
 12   209 ms   185 ms   204 ms  ae-2-2.ebr3.Atlanta2.Level3.net [4.69.132.85]
 13   204 ms   204 ms   202 ms  ae-7-7.ebr3.Dallas1.Level3.net [4.69.134.21]
 14   282 ms   197 ms   210 ms  ae-63-63.csw1.Dallas1.Level3.net [4.69.151.133]
 15   200 ms   219 ms   230 ms  ae-1-60.edge9.Dallas1.Level3.net [4.69.145.16]
 16   210 ms   197 ms   213 ms  CISCO-SYSTE.edge9.Dallas1.Level3.net [4.30.74.46]
 17     *        *        *     Request timed out.
 18   322 ms   310 ms   329 ms  rcdn9-cd2-dmzdcc-gw2-por1.cisco.com [72.163.0.182]
 19   319 ms   310 ms   315 ms  rcdn9-14a-dcz05n-gw1-ten5-5.cisco.com [72.163.0.238]
 20   324 ms   299 ms   309 ms  www1.cisco.com [72.163.4.161]

Trace complete.

C:\Users\      >_
```

In the output above you can see that the traceroute command has listed the IP addresses of all of the routers in the path.

Traceroute on Unix-like operating systems

Traceroute command on Unix works slightly different than the Windows version. It uses UDP packets with a large destination port number (33434 to 33534) that is unlikely to be used by any application at the destination host. Like the Windows version of the command, traceroute on Unix uses TTL to get the IP addresses of the intermediary routers. When a destination host is reached, it replies with an ICMP port unreachable message.

CHAPTER FIVE
TCP/IP SUITE OF PROTOCOLS

We Will Cover These Topics in This Chapter

- ***What is Protocols***
- ***TCP explained***
- ***UDP explained***
- ***Ports explained***
- ***ARP (Address Resolution Protocol) explained***
- ***DHCP & DNS***
- ***Telnet & SSH***
- ***FTP & TFTP***
- ***SNMP (Simple Network Management Protocol)***
- ***HTTP and HTTPS explained***
- ***NTP (Network Time Protocol)***
- ***APIPA (Automatic Private IP Addressing)***
- ***ICMP (Internet Control Message Protocol)***
- ***IP header***

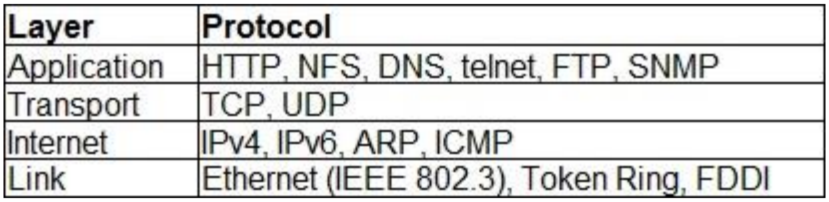
TCP/IP suite of protocols

The TCP/IP suite is a set of protocols used on computer networks today (most notably on the Internet). It provides an end-to-end connectivity by specifying how data should be packetized, addressed, transmitted, routed and received on a TCP/IP network. This functionality is organized into four abstraction layers and each protocol in the suite resides in a particular layer.

The TCP/IP suite is named after its most important protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP). Some of the protocols included in the TCP/IP suite are:

- *ARP (Address Resolution Protocol) – used to associate an IP address with a MAC address.*
- *IP (Internet Protocol) – used to deliver packets from the source host to the destination host based on the IP addresses.*
- *ICMP (Internet Control Message Protocol) – used to detects and reports network error conditions. Used in ping.*
- *TCP (Transmission Control Protocol) – a connection-oriented protocol that enables reliable data transfer between two computers.*
- *UDP (User Datagram Protocol) – a connectionless protocol for data transfer. Since a session is not created before the data transfer, there is no guarantee of data delivery.*
- *FTP (File Transfer Protocol) – used for file transfers from one host to another.*
- *Telnet (Telecommunications Network) – used to connect and issue commands on a remote computer.*
- *DNS (Domain Name System) – used for host names to the IP address resolution.*
- *HTTP (Hypertext Transfer Protocol) – used to transfer files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.*

The following table shows which protocols reside on which layer of the TCP/IP model:



Layer	Protocol
Application	HTTP, NFS, DNS, telnet, FTP, SNMP
Transport	TCP, UDP
Internet	IPv4, IPv6, ARP, ICMP
Link	Ethernet (IEEE 802.3), Token Ring, FDDI

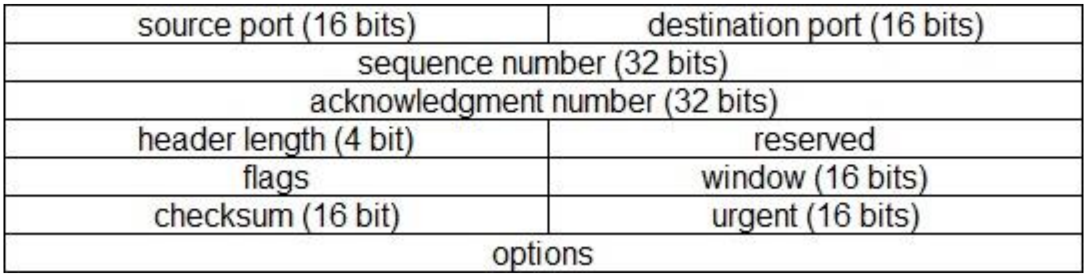
TCP explained

One of the main protocols in the TCP/IP suite is Transmission Control Protocol (TCP). TCP provides reliable and ordered delivery of data between applications running on hosts on a TCP/IP network. Because of its reliable nature, TCP is used by applications that require high reliability, such as FTP, SSH, SMTP, HTTP, etc.

TCP is connection-oriented, which means that, before data is sent, a connection between two hosts must be established. The process used to establish a TCP connection is known as the three-way handshake. After the connection has been established, the data transfer phase begins. After the data is transmitted, the connection is terminated.

One other notable characteristic of TCP is its reliable delivery. TCP uses sequence numbers to identify the order of the bytes sent from each computer so that the data can be reconstructed in order. If any data is lost during the transmission, the sender can retransmit the data.

Because of all of its characteristics, TCP is considered to be complicated and costly in terms of network usage. The TCP header is up to 24 bytes long and consists of the following fields:



source port (16 bits)	destination port (16 bits)
sequence number (32 bits)	
acknowledgment number (32 bits)	
header length (4 bit)	reserved
flags	window (16 bits)
checksum (16 bit)	urgent (16 bits)
options	

- source port – the port number of the application on the host sending the data.
- destination port – the port number of the application on the host receiving the data.
- sequence number – used to identify each byte of data.
- acknowledgment number – the next sequence number that the receiver is expecting.
- header length – the size of the TCP header.
- reserved – always set to 0.
- flags – used to set up and terminate a session.
- window – the window size the sender is willing to accept.
- checksum – used for error-checking of the header and data.
- urgent – indicates the offset from the current sequence number, where the segment of non-urgent data begins.
- options – various TCP options, such as Maximum Segment Size (MSS) or Window Scaling.

NOTE

TCP is a Transport layer protocol (Layer 4 of the OSI model).

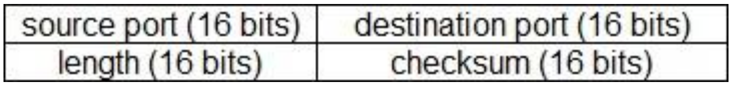
UDP explained

One other important protocol in the TCP/IP site is User Datagram Protocol (UDP). This protocol is basically a scaled-down version of TCP. Just like TCP, this protocol provides delivery of data between applications running on hosts on a TCP/IP network, but, unlike TCP, it does not sequence the data and *does not care about the order in which the segments arrive at the destination. Because of this it is considered to be an unreliable protocol. UDP is also considered to be a connectionless protocol, since no virtual circuit is established between two endpoints before the data transfer takes place.*

Because it does not provide many features that TCP does, UDP uses much less network resources than TCP. UDP is commonly used with two types of applications:

- *applications that are tolerant of the lost data – VoIP (Voice over IP) uses UDP because if a voice packet is lost, by the time the packet would be retransmitted, too much delay would have occurred, and the voice would be unintelligible.*
- *applications that have some application mechanism to recover lost data – Network File System (NFS) performs recovery with application layer code, so UDP is used as a transport-layer protocol.*

The UDP header is 8 bytes long and consists of the following fields:



source port (16 bits)	destination port (16 bits)
length (16 bits)	checksum (16 bits)

Here is a description of each field:

- *source port – the port number of the application on the host sending the data.*
- *destination port – the port number of the application on the host receiving the data.*
- *length – the length of the UDP header and data.*
- *checksum – checksum of both the UDP header and UDP data fields.*

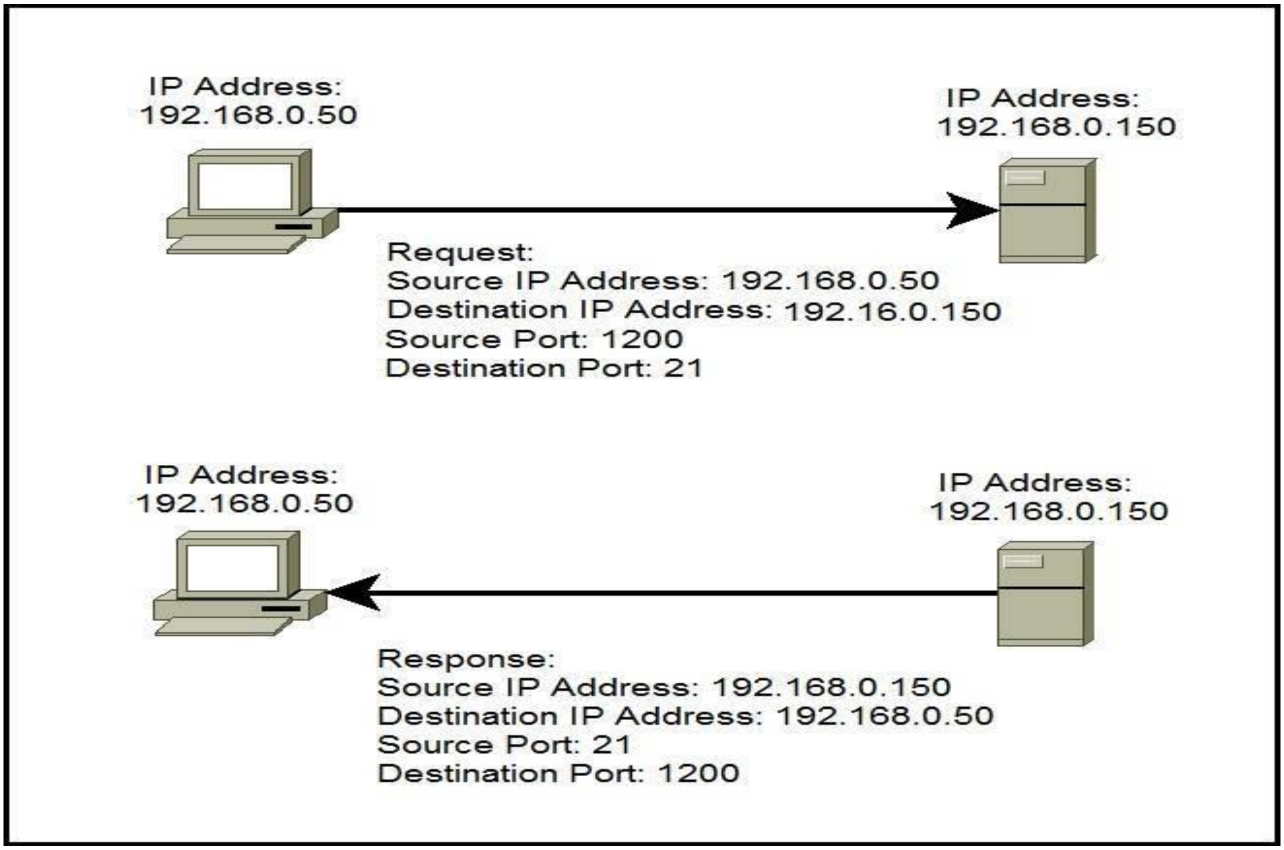
NOTE

UDP is a Transport layer protocol (Layer 4 of the OSI model).

Ports explained

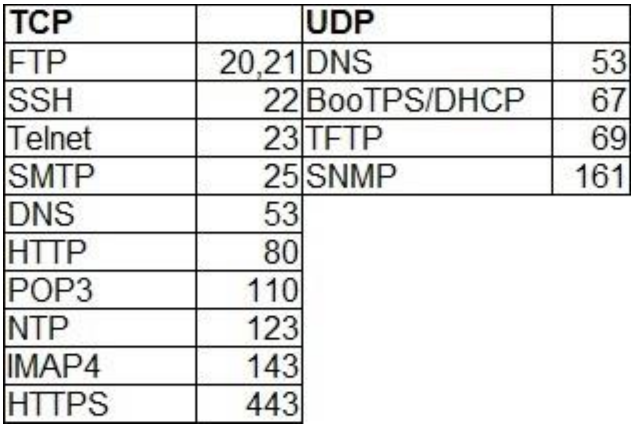
A port is a 16-bit number used to identify specific applications and services. TCP and UDP specify the source and destination port numbers in their packet headers and that information, along with the source and destination IP addresses and the transport protocol (TCP or UDP), enables applications running on hosts on a TCP/IP network to communicate.

Applications that provide a service (such as FTP and HTTP servers) open a port on the local computer and listen for connection requests. A client can request the service by pointing the request to the application's IP address and port. A client can use any locally unused port number for communication. Consider the following example:



In the picture above you can see that a host with an IP address of 192.168.0.50 wants to communicate with the FTP server. Because FTP servers use, by default, the well-known port 21, the host generates the request and sends it to the FTP server's IP address and port. The host use the locally unused port of 1200 for communication. The FTP server receives the request, generates the response, and sends it to the host's IP address and port.

Port numbers are from 0 to 65535. The first 1024 ports are reserved for use by certain privileged services:



TCP		**UDP**	
FTP	20,21	DNS	53
SSH	22	BooTPS/DHCP	67
Telnet	23	TFTP	69
SMTP	25	SNMP	161
DNS	53		
HTTP	80		
POP3	110		
NTP	123		
IMAP4	143		
HTTPS	443		

NOTE

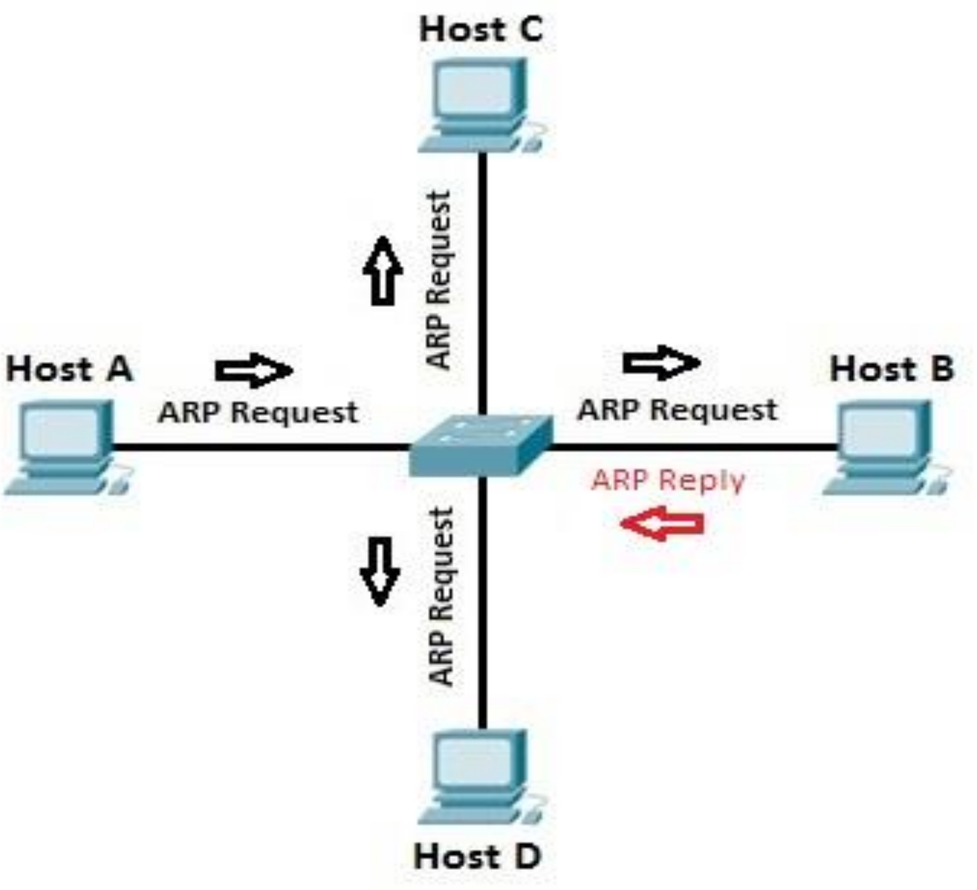
The combination of an IP address and a port number is called a socket. In our example the socket would be 192.168.0.50:1200.

ARP (Address Resolution Protocol) explained

ARP (Address Resolution Protocol) is a network protocol used to find out the hardware (MAC) address of a device from an IP address. It is used when a device wants to communicate with some other device on a local network (for example on an Ethernet network that requires physical addresses to be known before sending packets). The sending device uses ARP to translate IP addresses to MAC addresses. The device sends an ARP request message containing the IP address of the receiving device. All devices on a local network segment see the message, but only the device that has that IP address responds with the ARP reply message containing its MAC address. The sending device now has enough information to send the packet to the receiving device.

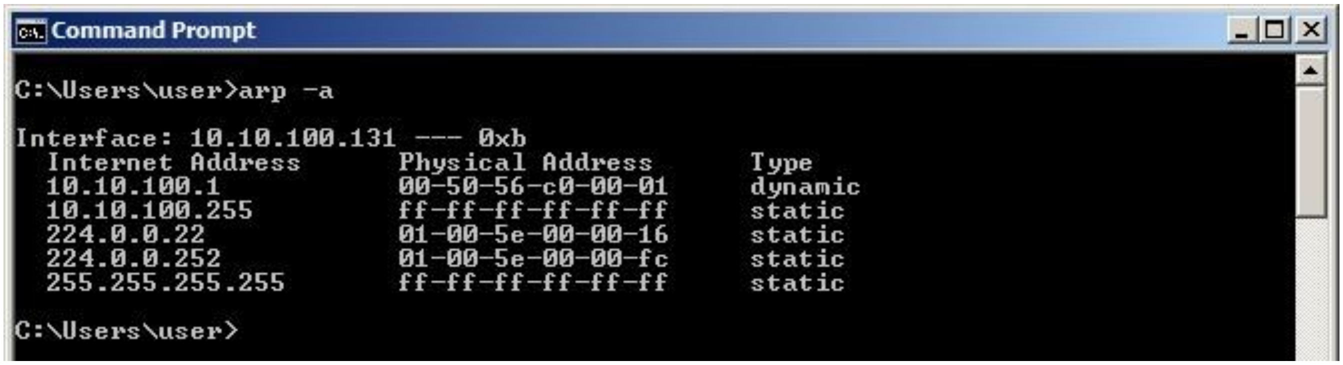
ARP request packets are sent to the broadcast addresses (FF:FF:FF:FF:FF:FF for the Ethernet broadcasts and 255.255.255.255 for the IP broadcast).

Here is the explanation of the ARP process:



Let's say that Host A wants to communicate with host B. Host A knows the IP address of host B, but it doesn't know the host B's MAC address. In order to find out the MAC address of host B, host A sends an ARP request, listing the host B's IP address as the destination IP address and the MAC address of FF:FF:FF:FF:FF:FF (Ethernet broadcast). Switch will forward the frame out all interfaces (except the incoming interface). Each device on the segment will receive the packet, but because the destination IP address is host B's IP address, only host B will reply with the ARP reply packet, listing its MAC address. Host A now has enough information to send the traffic to host B.

All operating systems maintain ARP caches that are checked before sending an ARP request message. Each time a host needs to send a packet to another host on the LAN, it first checks its ARP cache for the correct IP address and matching MAC address. The addresses will stay in the cache for a couple of minutes. You can display ARP entries in Windows by using the arp -a command:



```
C:\Users\user>arp -a

Interface: 10.10.100.131 --- 0xb
  Internet Address      Physical Address      Type
  10.10.100.1           00-50-56-c0-00-01     dynamic
  10.10.100.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\user>
```

DHCP & DNS

DHCP (Dynamic Host Configuration Protocol)

DHCP is a network protocol that is used to assign various network parameters to a device. This greatly simplifies administration of a network, since there is no need to assign static network parameters for each device.

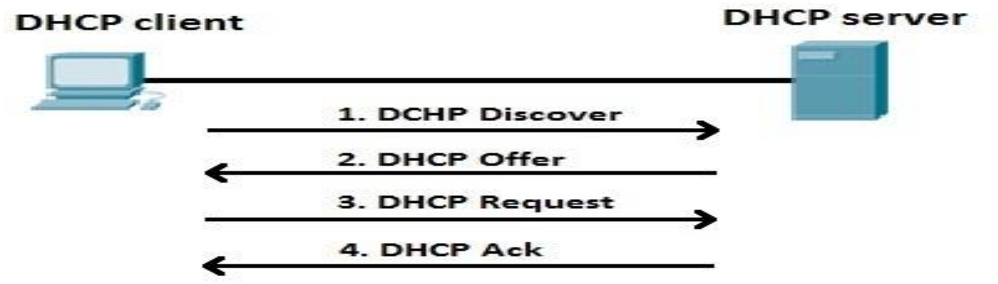
DHCP is a client-server protocol. A client is a device that is configured to use DHCP to request network parameters from a DHCP server. DHCP server maintains a pool of available IP addresses and assigns one of them to the host. A DHCP server can also provide some other parameters, such as:

- *subnet mask*
- *default gateway*
- *domain name*
- *DNS server*

Cisco routers can be configured as both DHCP client and DHCP server.

DHCP process explained:

DHCP client goes through the four step process:



1: A DHCP client sends a broadcast packet (DHCP Discover) to discover DHCP servers on the LAN segment.

2: The DHCP servers receive the DHCP Discover packet and respond with DHCP Offer packets, offering IP addressing information.

3: If the client receives the DHCP Offer packets from multiple DHCP servers, the first DHCP Offer packet is accepted. The client responds by broadcasting a DHCP Request packet, requesting the network parameters from the server that responded first.

4: The DHCP server approves the lease with a DHCP Acknowledgement packet. The packet includes the lease duration and other configuration information.

NOTE

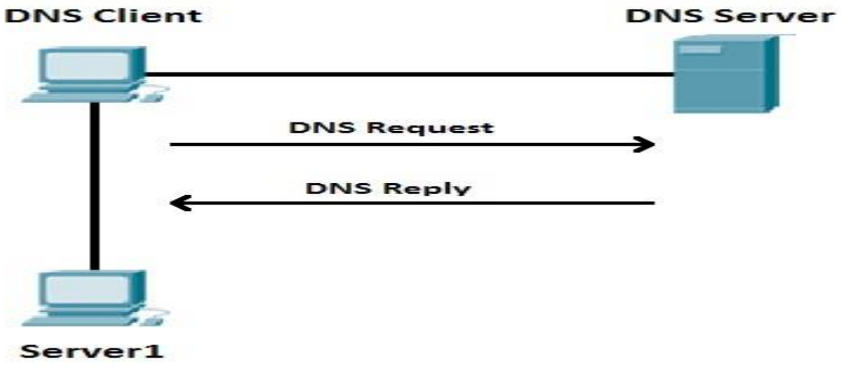
DHCP uses a well-known UDP port number 67 for the DHCP server, and the UDP port number 68 for the client.

DNS (Domain Name System)

DNS is a network protocol used to translate hostnames into IP addresses. DNS is not required to establish a network connection, but it is much more user friendly for human users than the numeric addressing scheme. Consider this example – you can access the Google homepage by typing 216.58.207.206, but it's much easier just to type www.google.com!

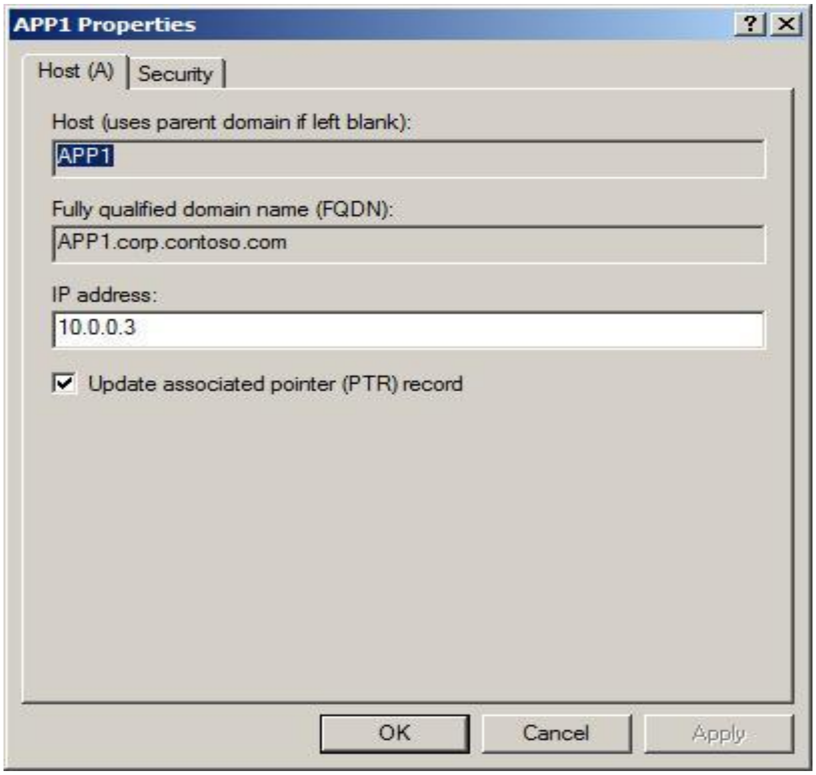
To use DNS, you must have a DNS server configured to handle the resolution process. A DNS server has a special-purpose application installed. The application maintains a table of dynamic or static hostname-to-IP address mappings. When a user request some network resource using a hostname, (e.g. by typing www.google.com in a browser), a DNS request is sent to the DNS *server asking for the IP address of the hostname. The DNS server then replies with the IP address. The user's browser can now use that IP address to access www.google.com.*

The figure below explains the concept:



Suppose that the DNS Client wants to communicate with the server named Server1. Since the DNS Client doesn't know the IP address of Server1, it sends a DNS Request to the DNS Server, asking for Server1's IP address. The DNS Server replies with the IP address of Server1 (DNS Reply).

The picture below shows a sample DNS record, taken from a DNS server:



Host (uses parent domain if left blank):
APP1

Fully qualified domain name (FQDN):
APP1.corp.contoso.com

IP address:
10.0.0.3

☑ Update associated pointer (PTR) record

Here you can see that the host with the hostname APP1 is using the IP address of 10.0.0.3.

NOTE

DNS uses a well-known UDP port 53.

Telnet & SSH

Telnet

Telnet is a network protocol that allows a user to communicate with a remote device. It is a virtual terminal protocol used mostly by network administrators to remotely access and manage devices. Administrator can access the device by telnetting to the IP address or hostname of a remote device.

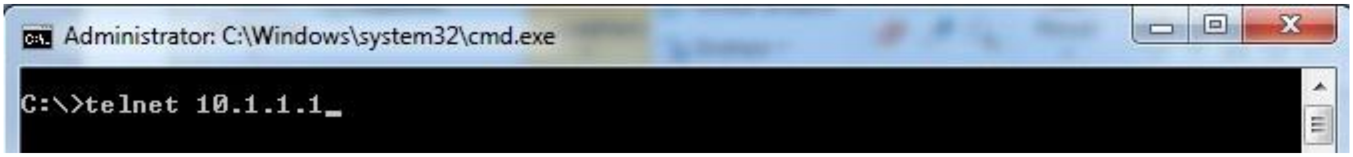
To use telnet, you must have a software (Telnet client) installed. On a remote device, a Telnet server must be installed and running. Telnet uses the TCP port 23 by default.

One of the greatest disadvantages of this protocol is that all data, including usernames and passwords, is sent in clear text, which is a potential security risk. This is the main reason why Telnet is rarely used today and is being replaced by a much secure protocol called SSH. Here you can find information about setting up Telnet access on your Cisco device.

NOTE

The word telnet can also refer to the software that implements the telnet protocol.

On Windows, you can start a Telnet session by typing the telnet IP_ADDRESS or HOSTNAME command:



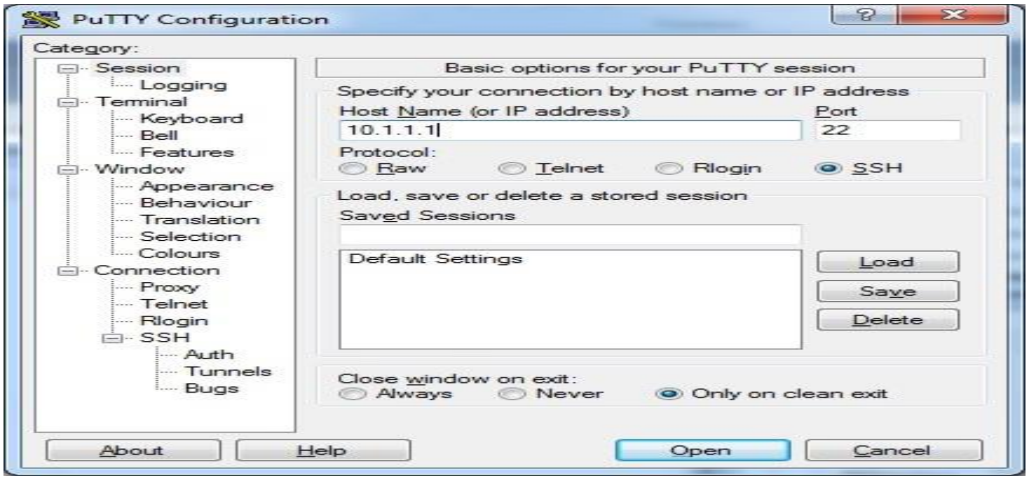
```
C:\>telnet 10.1.1.1_
```

SSH (Secure Shell)

SSH is a network protocol used to remotely access and manage a device. The key difference between Telnet and SSH is that SSH uses encryption, which means that all data transmitted over a network is secure from eavesdropping. SSH uses the public key encryption for such purposes.

Like Telnet, a user accessing a remote device must have an SSH client installed. On a remote device, an SSH server must be installed and running. SSH uses the TCP port 22 by default.

Here is an example of creating an SSH session using Putty, a free SSH client:



PuTTY Configuration

Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): 10.1.1.1

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

NOTE

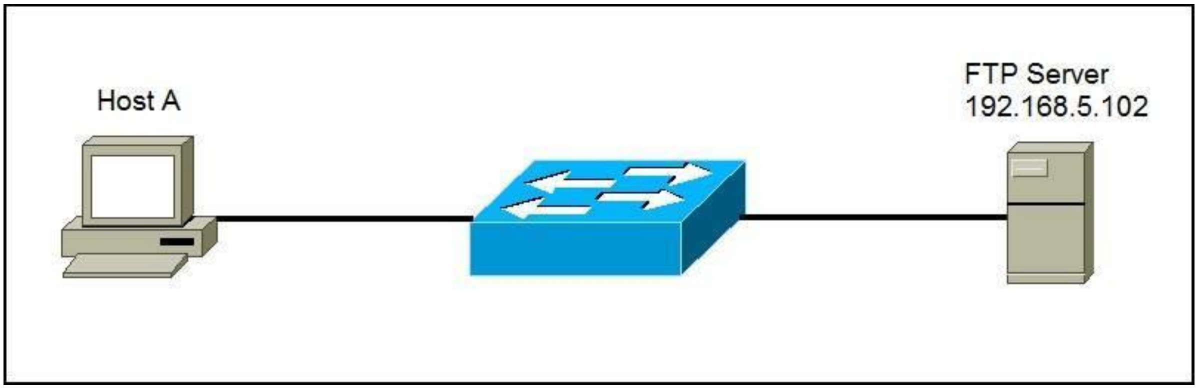
SSH is the most common way to remotely access and manage a Cisco device. Here you can find information about setting up SSH access on your Cisco device.

FTP & TFTP

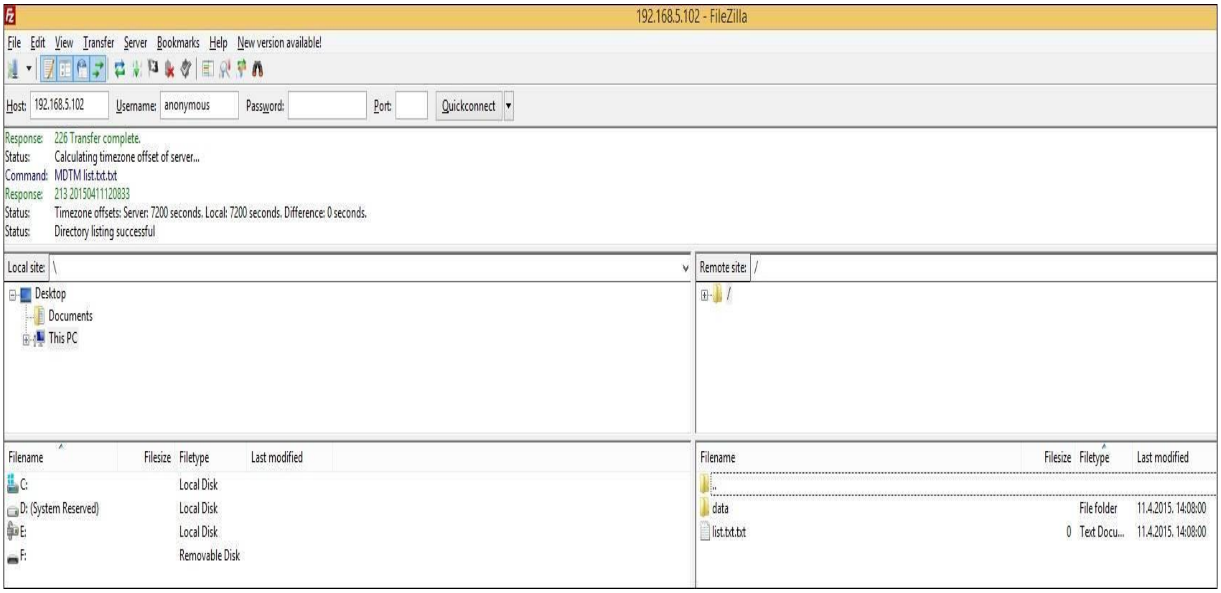
FTP (File Transfer Protocol)

FTP is a network protocol used to transfer files from one computer to another over a TCP network. Like Telnet, it uses a client-network architecture, which means that a user has to have an FTP client installed to access the FTP server running on a remote machine. After establishing the FTP connection, the user can download or upload files to and from the FTP server.

Consider the following example:



A user wants to transfer files from Host A to the FTP server. The user will start an FTP client program (in this example, FileZilla), and initiate the connection:


192.168.5.102 - FileZilla
File Edit View Transfer Server Bookmarks Help New version available!
Quickconnect

Host: 192.168.5.102 Username: anonymous Password: Port:

```
Response:   226 Transfer complete.
Status:     Calculating timezone offset of server...
Command:    MDTM list.txt.txt
Response:   213 20150411120833
Status:     Timezone offsets: Server: 7200 seconds. Local: 7200 seconds. Difference: 0 seconds.
Status:     Directory listing successful
```

Local site: \

- Desktop
  - Documents
  - This PC

Remote site: /

- /

Filename	Filesize	Filetype	Last modified
C:		Local Disk	
D: (System Reserved)		Local Disk	
E:		Local Disk	
F:		Removable Disk	

Filename	Filesize	Filetype	Last modified
..			
data		File folder	11.4.2015. 14:08:00
list.txt.txt	0	Text Docu...	11.4.2015. 14:08:00

In the example above, the anonymous authentication was used, so the user was not asked to provide the password. The client can now transfer files from and to the FTP server using the graphical interface.

NOTE

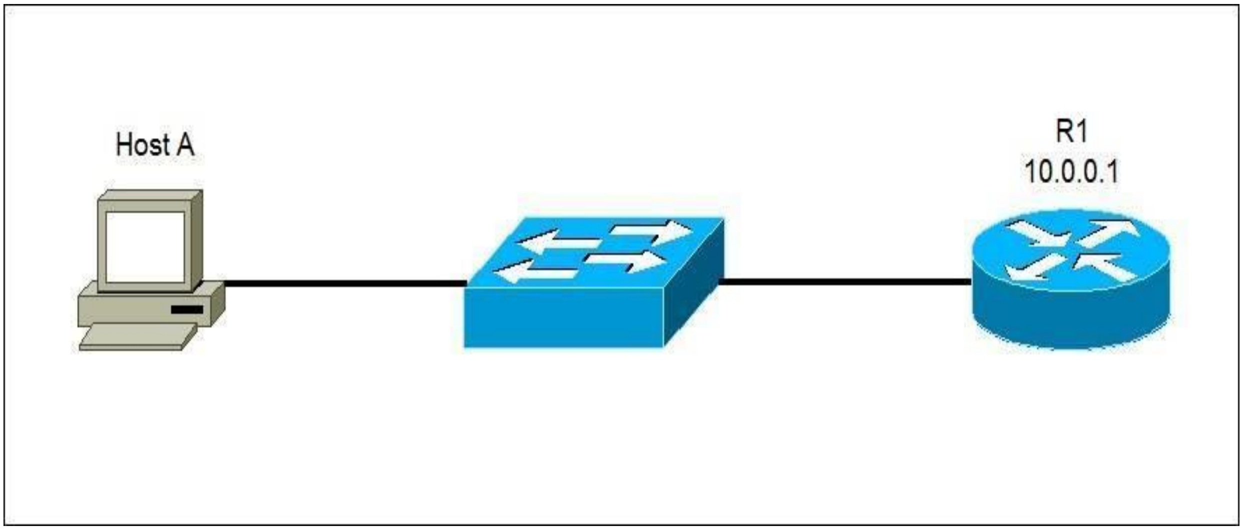
FTP uses two TCP ports: port 20 for sending data and port 21 for sending control commands. The protocol supports the use of authentication, but like Telnet, all data is sent in clear text, including usernames and passwords.

TFTP (Trivial File Protocol)

TFTP is a network protocol used to transfer files between remote machines. It is a simple version of FTP, lacking some of the more advanced features FTP offers, but requiring less resources than FTP.

Because of it's simplicity TFTP can be used only to send and receive files. This protocol is not widely used today, but it still can be used to save and restore a router configuration or to backup an IOS image.

Consider the following example:



A user wants to transfer files from Host A to the router R1. R1 is a Cisco device and it has a TFTP server installed. The user will start an TFTP client program and initiate the data transfer.

NOTE

TFTP doesn't support user authentication and sends all data in clear text. It uses UDP port 69 for communication.

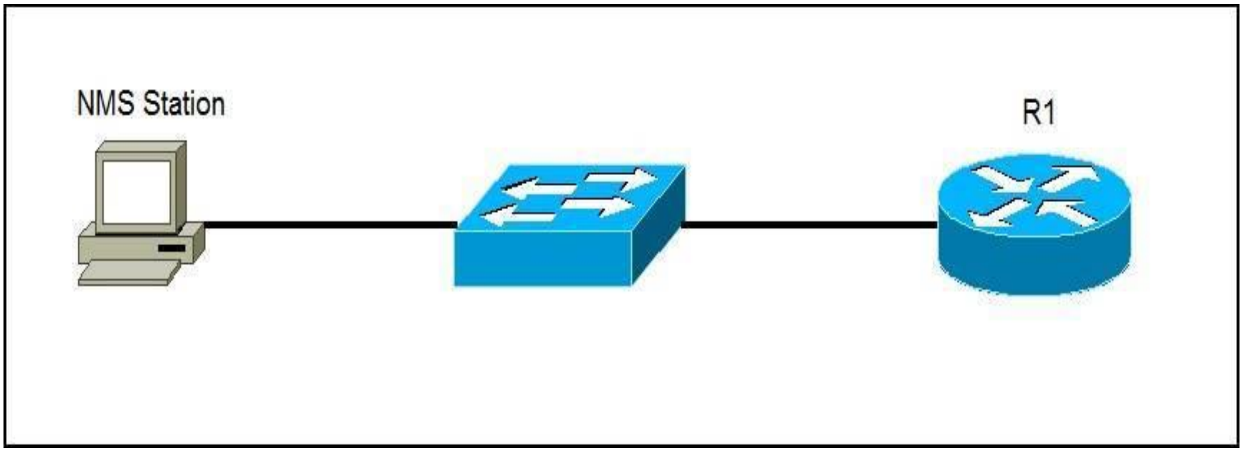
SNMP (Simple Network Management Protocol)

Simple Network Management Protocol (SNMP) is an application layer protocol that is used for network device management. This protocol can collects and manipulate valuable network information from switches, routers, servers, printers, and other network-attached devices.

An SNMP-managed network consists of two components:

- *Network management station (NMS) – the software which runs on the administrative computer. This software gathers SNMP data by requiring the devices on the network to disclose certain information. Devices can also inform the NMS about problems they are experiencing by sending an SNMP alert (called a trap).*
- *Agent – the software which runs on managed devices and reports information via SNMP to the NMS.*

Consider the following example:



The router R1 is configured to send SNMP traps to the NMS Station. If a problem occurs, the router will send an SNMP trap to Host A. For example, if there is a port security violation on R1, the router will send the SNMP trap, notifying that there has been a potential security breach on the network.

NOTE

SNMP agents use a UDP port 161, while the manager uses a UDP port 162. The current SNMP version is SNMPv3. The prior versions, SNMPv1 and SNMPv2 are considered obsolete and should not be used.

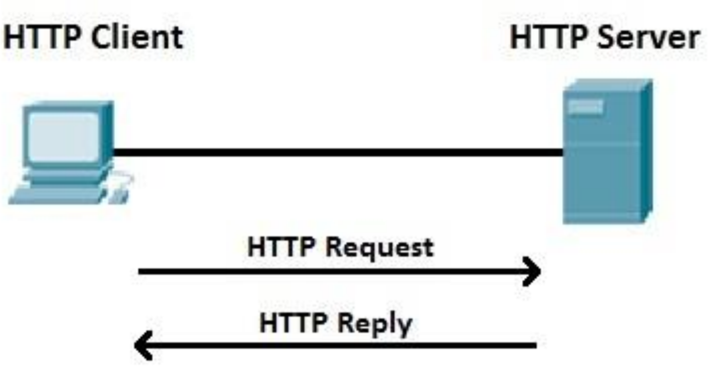
HTTP and HTTPS explained

HTTP (Hypertext Transfer Protocol)

HTTP is an client-server protocol that allows clients to request web pages from web servers. It is an application level protocol widely used on the Internet. Clients are usually web browsers. When a user wants to access a web page, a browser sends an HTTP Request message to the web server. The server responds with the requested web page. By default, web servers use the TCP port 80.

Clients and web servers use request-response method to communicate with each other, with clients sending the HTTP Requests and servers responding with the HTTP Responses. Clients usually send their requests using GET or POST methods, for example GET /homepage.html. Web servers responds with a status message (200 if the request was successful) and sends the requested resource.

An example will clarify this process:



The client wants to access http://google.com and points his browser to the URL http://google.com (this is an example of an HTTP Request message). The web server hosting http://google.com receives the request and responds with the content of the web page (the HTTP response message).

Web servers usually use a well-known TCP port 80. If the port is not specified in a URL, browsers will use this port when sending HTTP request. For example, you will get the same result when requesting http://google.com and http://google.com:80.

NOTE

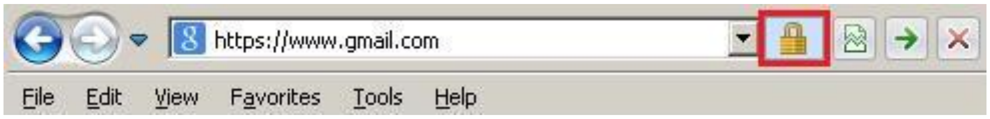
The version of HTTP most commonly used today is HTTP/1.1. A newer version, HTTP/2, is available and supported by most browser.

HTTPS (Hypertext Transfer Protocol Secure)

Hypertext Transfer Protocol Secure is a secure version of HTTP. This protocol enables secure communication between a client (e.g. web browser) and a server (e.g. web server) by using encryption. HTTPS uses Transport Layer Security (TLS) protocol or its predecessor Secure Sockets Layer (SSL) for encryption.

HTTPS is commonly used to create a secure channel over some insecure network, e.g. Internet. A lot of traffic on the Internet is unencrypted and susceptible to sniffing attacks. HTTPS encrypts sensitive information, which makes a connection secure.

HTTPS URLs begin with https instead of http. In Internet Explorer, you can immediately recognize that a web site is using HTTPS because a lock appears to the right of the address bar:


https://www.gmail.com

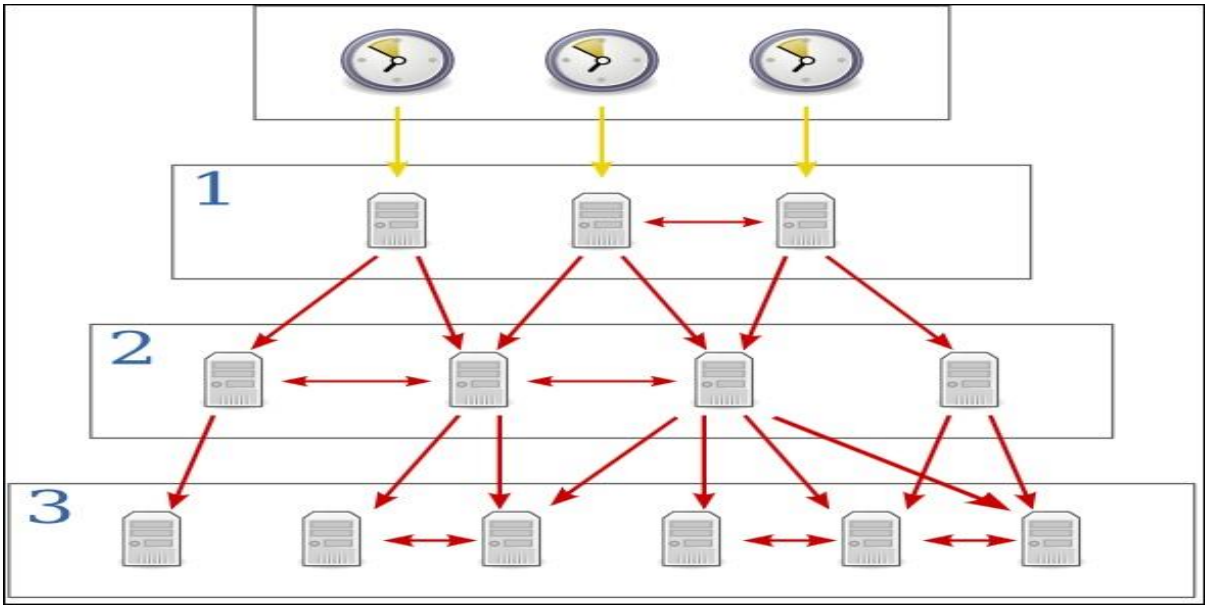
NOTE

HTTPS uses a well-known TCP port 443. If the port is not specified in a URL, browsers will use this port when sending HTTPS request. For example, you will get the same result when requesting https://gmail.com andhttps://gmail.com:443.

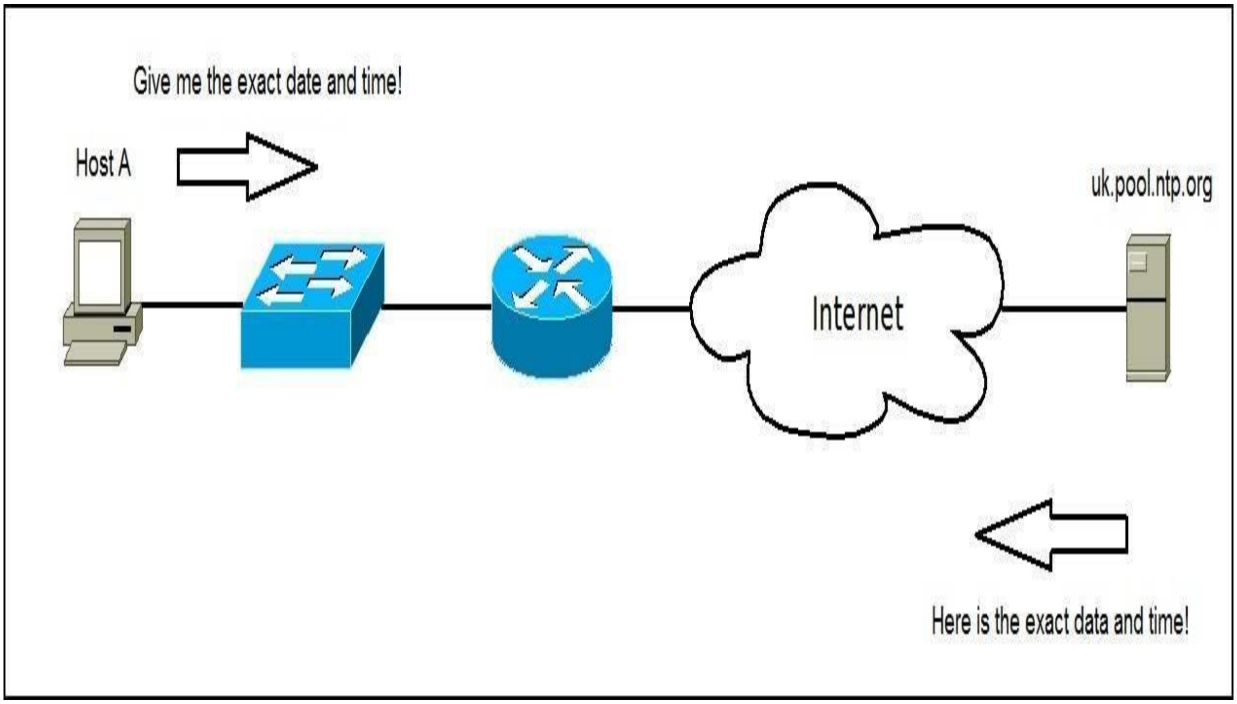
NTP (Network Time Protocol)

Network Time Protocol (NTP) is an application layer protocol used for clock synchronization between hosts on a TCP/IP network. The goal of NTP is to ensure that all computers on a network agree on the time, since even a small difference can create problems. For example, if there is more than 5 minutes difference on your host and the Active Directory domain controller, you will not be able to login into your AD domain.

NTP uses a hierarchical system of time sources. At the top of the structure are highly accurate time sources – typically atomic or GPS clocks. These clocks are known as stratum 0 servers. Stratum 1 servers are directly linked to stratum 0 servers and computers run NTP servers that deliver the time to stratum 2 servers, and so on (image source: Wikipedia):



NTP uses a client-server architecture; one host is configured as the NTP server and all other hosts on the network are configured as NTP clients. Consider the following example:


Give me the exact date and time!
Host A
uk.pool.ntp.org
Internet
Here is the exact data and time!

Host A is configured to use a public NTP server uk.pool.ntp.org. Host A will periodically send an NTP request to the NTP server. The NTP server will provide the accurate data and time, so Host A can synchronize its clock.

NOTE

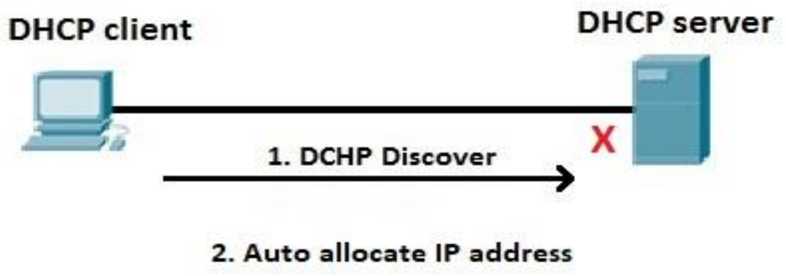
NTP uses a well-known UDP port 123. The current version is NTPv4, and it is backward compatible with NTPv3.

APIPA (Automatic Private IP Addressing)

Automatic Private IP Addressing (APIPA) is a feature in operating systems (such as Windows) that enables computers to automatically self-configure an IP address and subnet mask when their DHCP server isn't reachable. The IP address range for APIPA is 169.254.0.1-169.254.255.254, with the subnet mask of 255.255.0.0.

When a DHCP client boots up, it looks for a DHCP server in order to obtain network parameters. If the client can't communicate with the DHCP server, it uses APIPA to configure itself with an IP address from the APIPA range. This way, the host will still be able to communicate with other hosts on the local network segment that are also configured for APIPA.

Consider the following example:



The host on the left is configured as DHCP client. The host boots up and looks for DHCP servers on the network. However, the DHCP server is down and can't respond to the host. After some time (from a couple of seconds to a couple of minutes, depending on the operating system) the client auto-configures itself with an address from the APIPA range (e.g. 169.254.154.22).

NOTE

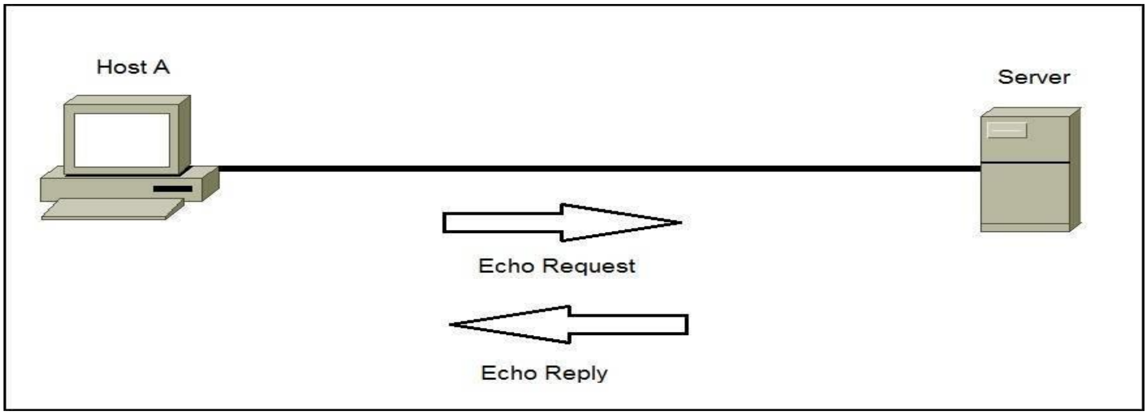
If your host is using an IP address from the APIPA range, there is usually a problem on the network. Check the network connectivity of your host and the status of the DHCP server.

The APIPA service also checks regularly for the presence of a DHCP server (every three minutes). If it detects a DHCP server on the network, the DHCP server replaces the APIPA networking addresses with dynamically assigned addresses.

ICMP (Internet Control Message Protocol)

ICMP (Internet Control Message Protocol) is a network layer protocol that reports errors and provides information related to IP packet processing. ICMP is used by network devices to send error messages indicating, for example, that a requested service is not available or that a host isn't reachable.

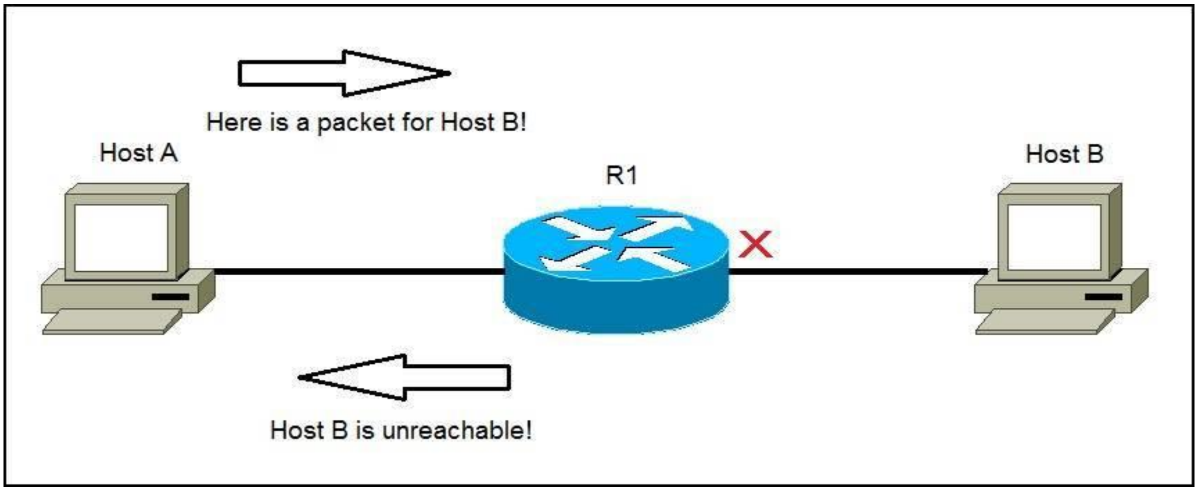
ICMP is commonly used by network tools such as ping or traceroute. Consider the following example that illustrates how ping can be used to test the reachability of a host:



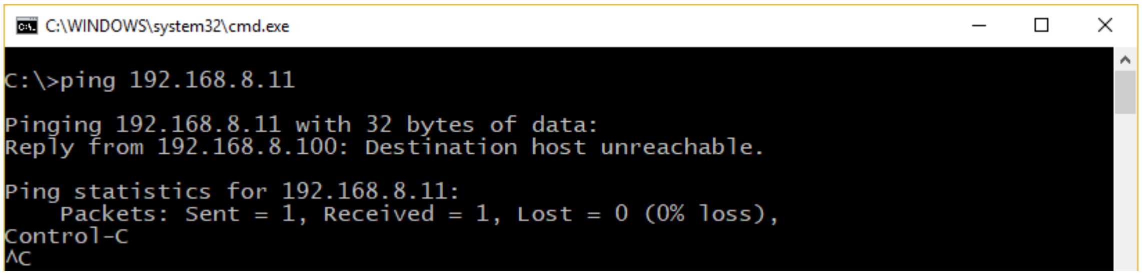
Host A wants to test whether it can reach Server over the network. Host A will start the ping utility that will send ICMP Echo Request packets to Server. If Server is reachable, it will respond with ICMP Echo Reply packets. If Host A receives no response from Server, there might be a problem on the network.

> *NOTE*
>
> *ICMP messages are encapsulated in IP datagrams, which means that they don't use higher level protocols (such as TCP or UDP) for transmission.*

One other common ICMP message is the Destination unreachable message. Here is an example:



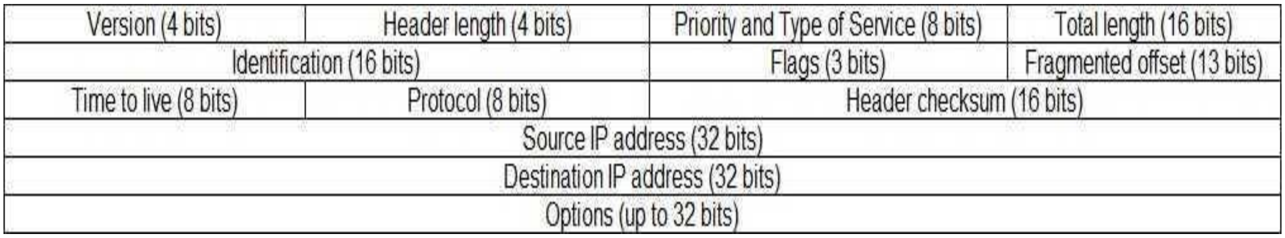
Host A sends a packet to Host B. Because the Host B is down, the router will send an ICMP Destination host unreachable message to Host A, informing it that the destination host is unreachable, e.g.:



```
C:\>ping 192.168.8.11

Pinging 192.168.8.11 with 32 bytes of data:
Reply from 192.168.8.100: Destination host unreachable.

Ping statistics for 192.168.8.11:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Control-C
^C
```

IP header

An IP header is a prefix to an IP packet that contains information about the IP version, length of the packet, source and destination IP addresses, etc. It consists of the following fields:

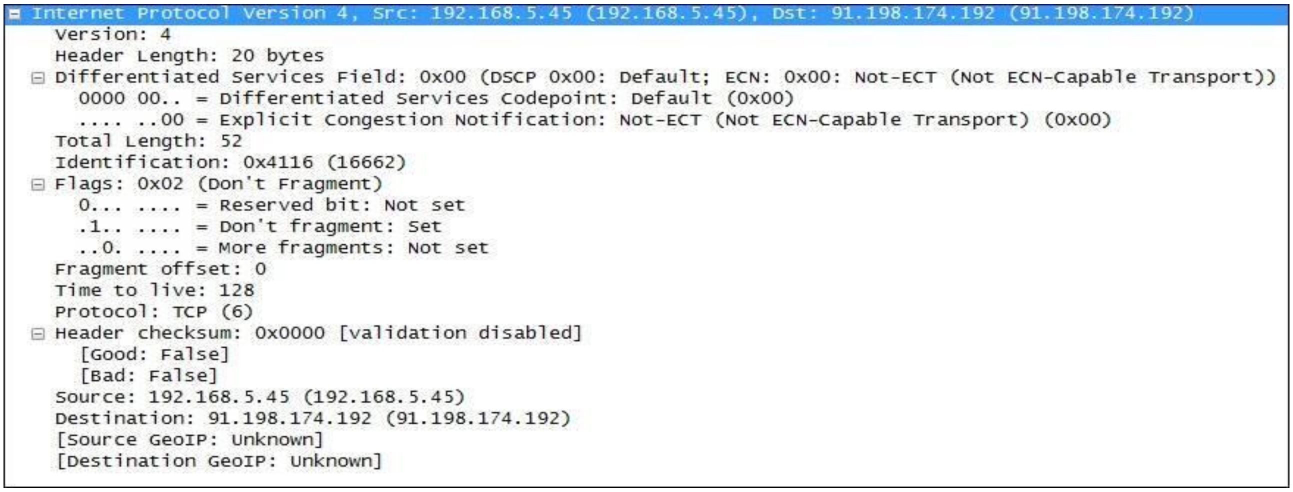


Version (4 bits)	Header length (4 bits)	Priority and Type of Service (8 bits)	Total length (16 bits)
Identification (16 bits)		Flags (3 bits)	Fragmented offset (13 bits)
Time to live (8 bits)	Protocol (8 bits)	Header checksum (16 bits)	
Source IP address (32 bits)			
Destination IP address (32 bits)			
Options (up to 32 bits)			

Here is a description of each field:

- *Version – the version of the IP protocol. For IPv4, this field has a value of 4.*
- *Header length – the length of the header in 32-bit words. The minimum value is 20 bytes, and the maximum value is 60 bytes.*
- *Priority and Type of Service – specifies how the datagram should be handled. The first 3 bits are the priority bits.*
- *Total length – the length of the entire packet (header + data). The minimum length is 20 bytes, and the maximum is 65,535 bytes.*
- *Identification – used to differentiate fragmented packets from different datagrams.*
- *Flags – used to control or identify fragments.*
- *Fragmented offset – used for fragmentation and reassembly if the packet is too large to put in a frame.*
- *Time to live – limits a datagram's lifetime. If the packet doesn't get to its destination before the TTL expires, it is discarded.*
- *Protocol – defines the protocol used in the data portion of the IP datagram. For example, TCP is represented by the number 6 and UDP by 17.*
- *Header checksum – used for error-checking of the header. If a packet arrives at a router and the router calculates a different checksum than the one specified in this field, the packet will be discarded.*
- *Source IP address – the IP address of the host that sent the packet.*
- *Destination IP address – the IP address of the host that should receive the packet.*
- *Options – used for network testing, debugging, security, and more. This field is usually empty.*

Consider the following IP header, captured with Wireshark:



```
⊟ Internet Protocol Version 4, Src: 192.168.5.45 (192.168.5.45), Dst: 91.198.174.192 (91.198.174.192)
    Version: 4
    Header Length: 20 bytes
  ⊟ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)
    Total Length: 52
    Identification: 0x4116 (16662)
  ⊟ Flags: 0x02 (Don't Fragment)
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
  ⊟ Header checksum: 0x0000 [validation disabled]
      [Good: False]
      [Bad: False]
    Source: 192.168.5.45 (192.168.5.45)
    Destination: 91.198.174.192 (91.198.174.192)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
```

Notice the fields in the header: the IP version is IPv4, the header length is 20 bytes, the upper-level protocol used is TCP, the TTL value is set tu 128, source and destination IP addresses are listed, etc.

CHAPTER SIX
CISCO IOS OVERVIEW

We Will Cover These Topics in This Chapter

- *Cisco IOS overview*
- *Power on a Cisco device*
- *Get help in IOS*
- *Running & startup configuration*
- *IOS basic commands*
- *show command*
- *Configure descriptions*
- *Run privileged commands within global config mode*
- *Ports on an IOS device*
- *Pipe character in IOS*
- *Backing up IOS configuration*
- *IOS boot sequence*
- *IOS command modes*

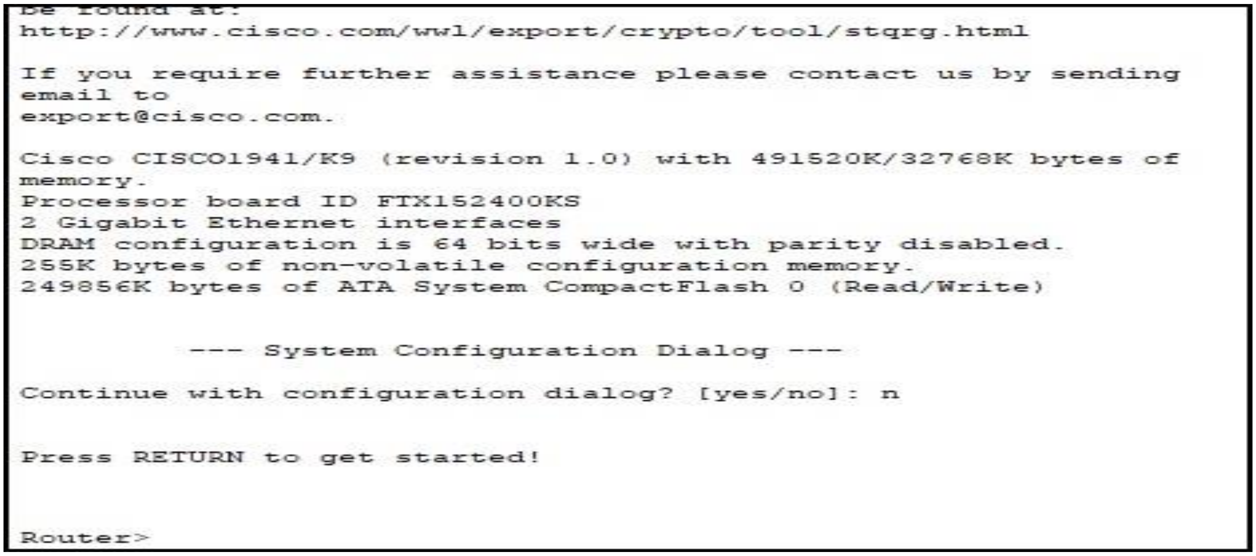
Cisco IOS overview

IOS (Internetwork Operating System) is a multitasking operating system used on most Cisco routers and switches. IOS has a command-line interface with the predetermined number of multiple-word commands. This operating system is used to configure routing, switching, internetworking and other features supported by a Cisco device.

NOTE

Previous versions of Cisco switches ran Cat OS, a discounted version of a CLI-based operating system.

Below you can see how IOS looks like when a Cisco device is started for the first time:



```
be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending
email to
export@cisco.com.

Cisco CISCO1941/K9 (revision 1.0) with 491520K/32768K bytes of
memory.
Processor board ID FTX152400KS
2 Gigabit Ethernet interfaces
DRAM configuration is 64 bits wide with parity disabled.
255K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)


         --- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: n


Press RETURN to get started!



Router>
```

Accessing the IOS

There are three most common ways to access the IOS:

1. Console access – this type of access is usually used to configure newly acquired devices. These devices usually don't have an IP address configured, and therefore can not be accessed through the network. Most of the Cisco devices have a physical console port. This port can be connected to a computer using a rollover cable, a special type of cable with pins on one end reversed on the other end of the cable. The rollover cable is a serial cable, which means that you can't just plug it in an Ethernet port on your computer. You will need an adapter that converts an interface on your computer (usually a 9-pin serial interface) into RJ-45.

2. Telnet access – this type of access used to be a common way to access network devices. Telnet is an terminal emulation program that enables you to access IOS through the network and configure the device remotely. The device that is being configured needs to have an Telnet server installed and an IP address configured.

Telnet uses a well known TCP port 23. One of the biggest disadvantages of this protocol is that is sends all data as clear-text, which includes the passwords! This is the reason why this type of access is usually not used anymore. Instead, SSH is usually used.

3. SSH access – like Telnet, this access type enables you to configure devices remotely, but it adds an extra layer of security by encrypting all communications using public-key cryptography. SSH uses well known TCP port 22.

IOS modes

IOS has many different modes. There are three main modes any many sub modes. We will describe the three main modes and one sub mode.

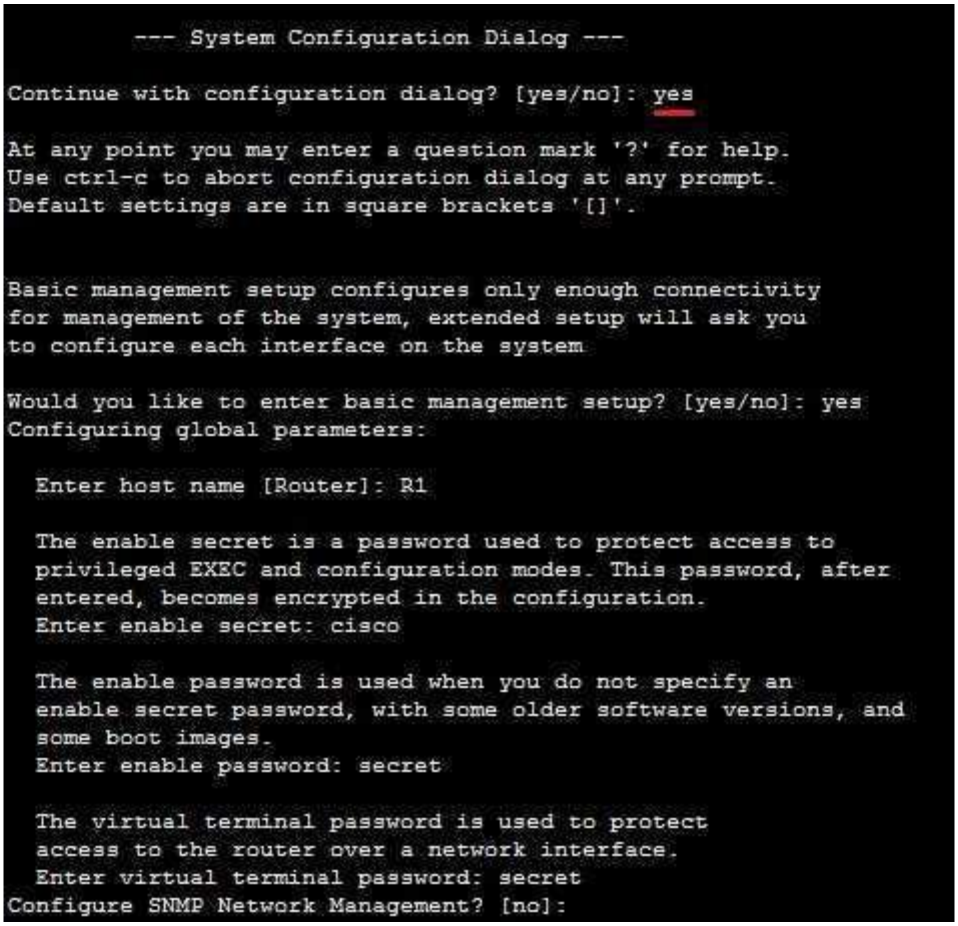
- *user EXEC mode – the default mode for the IOS CLI. This is the mode that a user is placed in after accessing the IOS. Only basic commands (like ping or telnet) are available in this mode.*
- *privileged EXEC Mode – this mode is accessed by typing the enable command from the user EXEC mode. This mode can be password protected. In this mode a user can view and change a device's configuration.*
- *global configuration mode – this mode can be accessed by typing the configure terminal command from the privileged EXEC mode. It is used to change the device's configuration.*

A global configuration mode can have many sub modes. For example, when a user wants to configure an interface, he will have to enter the interface sub mode by entering the interface INTERFACE_TYPE INTERFACE_NUMBER command (e.g. interface Fast Ethernet 0/1) from the global configuration mode. This sub mode can have many commands that are specific for the interface.

Let's describe each of the modes mentioned above in more detail.

Power on a Cisco device

When you first power-on a newly purchased Cisco device, it will perform a power-on self-test (POST) to discover the hardware components and verify that all components work properly. If the POST is successful, the device will enter the setup mode. This mode presents a step-by-step dialog to help you configure some basic parameters, such as the device hostname, passwords, interface IP address, etc. To enter the setup mode, power on your device and type yes when prompted to make a selection:



```
        --- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: yes

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.


Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

Would you like to enter basic management setup? [yes/no]: yes
Configuring global parameters:

  Enter host name [Router]: R1

  The enable secret is a password used to protect access to
  privileged EXEC and configuration modes. This password, after
  entered, becomes encrypted in the configuration.
  Enter enable secret: cisco

  The enable password is used when you do not specify an
  enable secret password, with some older software versions, and
  some boot images.
  Enter enable password: secret

  The virtual terminal password is used to protect
  access to the router over a network interface.
  Enter virtual terminal password: secret
Configure SNMP Network Management? [no]:
```

The wizard guides you through the initial configuration of your device and will create an initial configuration file. The setup mode is useful when you are unfamiliar with the IOS CLI, but once you learn the basics of CLI, you probably won't use this mode ever again.

NOTE

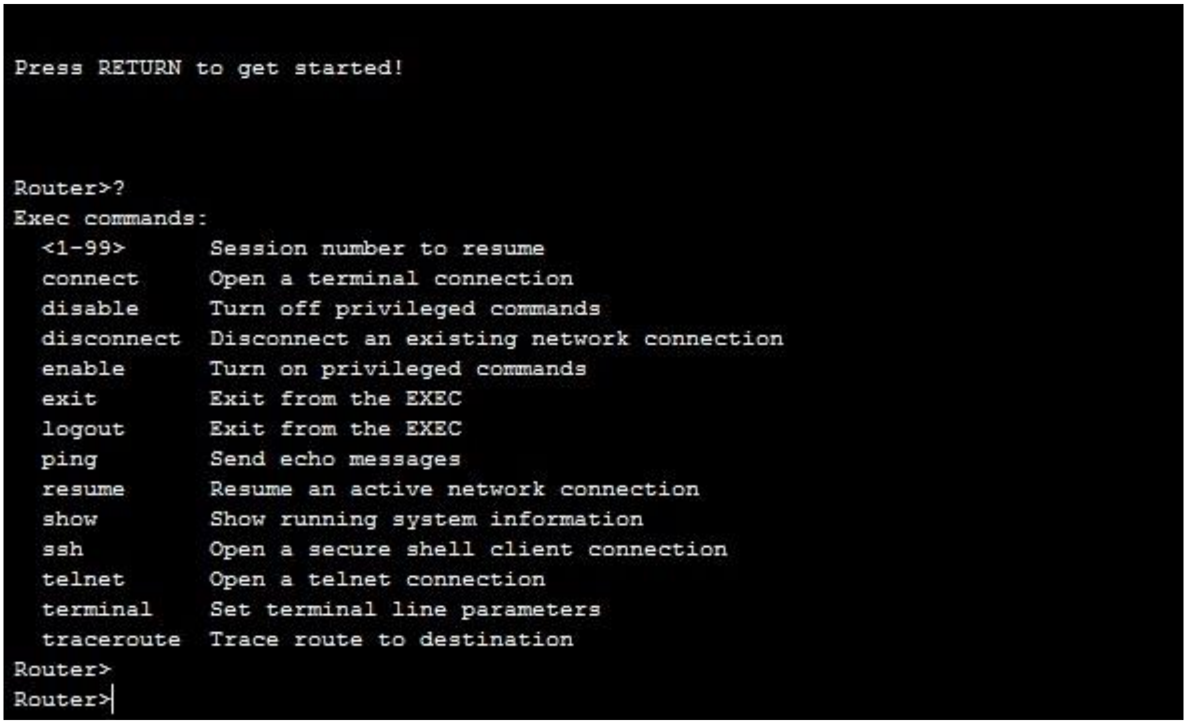
You can enter the setup mode at any time from the command line by typing the setup command from the privileged mode. To exit the setup mode without saving any changes, press CRTL+C.

IOS command modes

We've already learned that IOS has three main command modes: the user exec, privileged exec, and the global configuration modes. Each of these modes serves a different purpose and has its own set of commands. In this lesson we will describe each of this modes in more detail.

User EXEC mode commands

Initially, a user logs into the User Exec mode. This is the mode with the least number of commands. You can get a list of all available commands by typing the character ?.



```
Press RETURN to get started!



Router>?
Exec commands:
  <1-99>      Session number to resume
  connect     Open a terminal connection
  disable     Turn off privileged commands
  disconnect  Disconnect an existing network connection
  enable      Turn on privileged commands
  exit        Exit from the EXEC
  logout      Exit from the EXEC
  ping        Send echo messages
  resume      Resume an active network connection
  show        Show running system information
  ssh         Open a secure shell client connection
  telnet      Open a telnet connection
  terminal    Set terminal line parameters
  traceroute  Trace route to destination
Router>
Router>
```

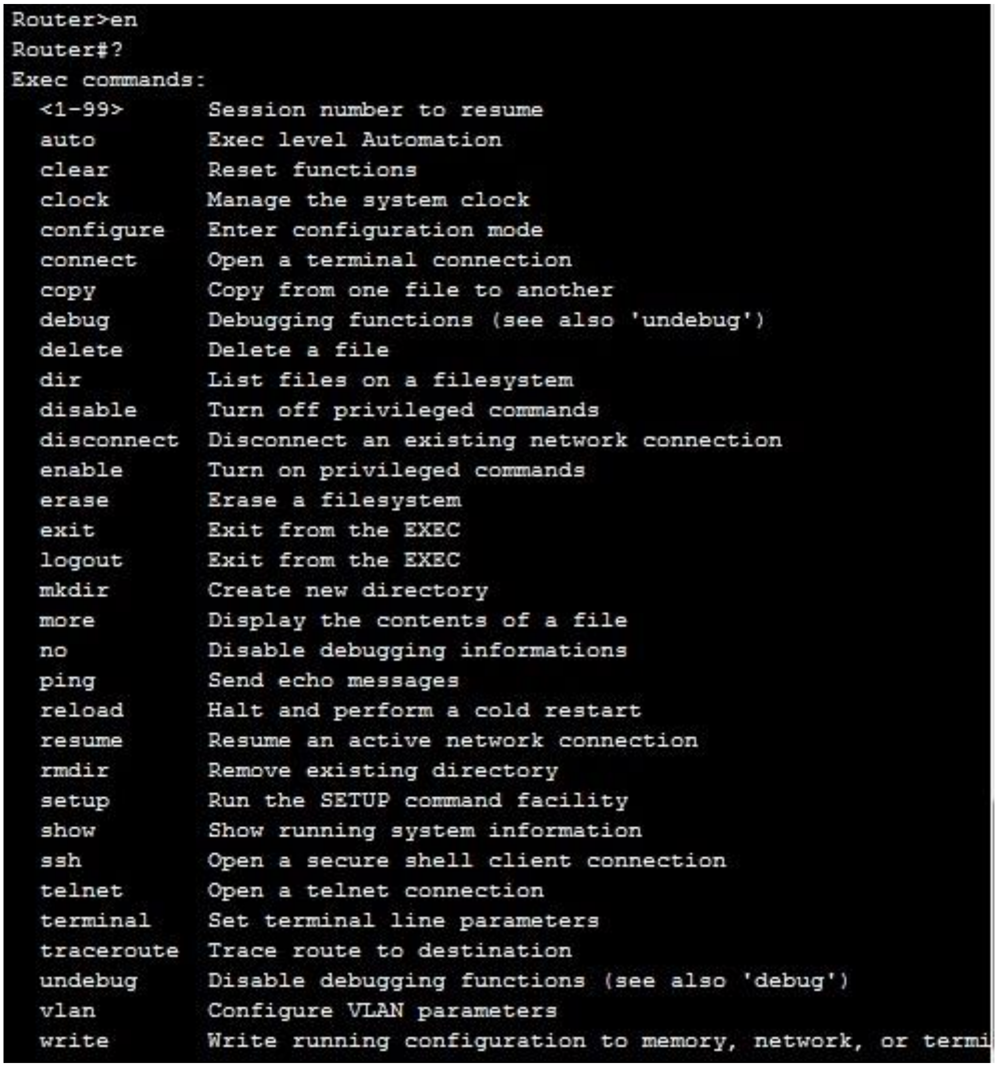
As you can see, most of the commands available are used to show statistics and perform some basic troubleshooting. The prompt on the left side of the screen always displays the device hostname (R1 in this case), followed by the character >.

All commands can be abbreviated to their first letters of the command name. For example, you can abbreviate ping by typing pin, because no other command in the User EXEC mode IOS mode begins with these letters.

Privileged EXEC mode commands

This IOS mode is also called enable mode because you must enter the enable command from a user EXEC mode if you want to access this mode. You can use more commands in the privileged EXEC mode than you were able to use in the user EXEC mode. You can save a device configuration or reload a device in this mode. You can also enter a third mode, the configuration mode. The access to the privileged EXEC mode is usually protected with a password.

The prompt for this mode shows # after the device hostname.



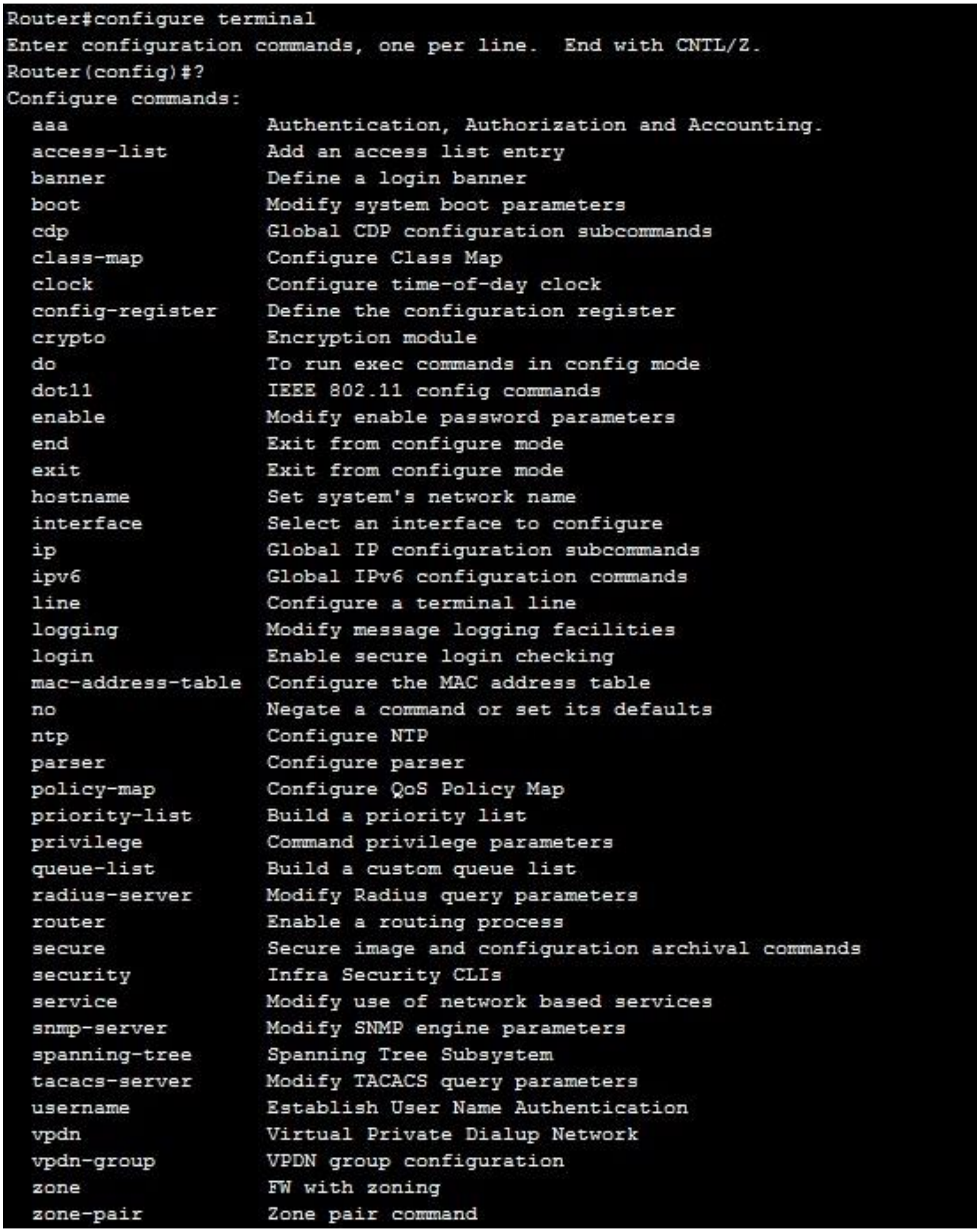
```
Router>en
Router#?
Exec commands:
  <1-99>      Session number to resume
  auto        Exec level Automation
  clear       Reset functions
  clock       Manage the system clock
  configure   Enter configuration mode
  connect     Open a terminal connection
  copy        Copy from one file to another
  debug       Debugging functions (see also 'undebug')
  delete      Delete a file
  dir         List files on a filesystem
  disable     Turn off privileged commands
  disconnect  Disconnect an existing network connection
  enable      Turn on privileged commands
  erase       Erase a filesystem
  exit        Exit from the EXEC
  logout      Exit from the EXEC
  mkdir       Create new directory
  more        Display the contents of a file
  no          Disable debugging informations
  ping        Send echo messages
  reload      Halt and perform a cold restart
  resume      Resume an active network connection
  rmdir       Remove existing directory
  setup       Run the SETUP command facility
  show        Show running system information
  ssh         Open a secure shell client connection
  telnet      Open a telnet connection
  terminal    Set terminal line parameters
  traceroute  Trace route to destination
  undebug     Disable debugging functions (see also 'debug')
  vlan        Configure VLAN parameters
  write       Write running configuration to memory, network, or termi
```

Global configuration mode commands

To change a device configuration, you need to enter the global configuration mode. This mode can be accessed by typing configure terminal (or conf t, the abbreviated version of the command) from the enable mode. The prompt for this mode is hostname(config).

Global configuration mode commands are used to configure a device. You can set a hostname, configure authentication, set an IP address for an interface, etc. From this mode you can also access sub modes, for example the interface mode, from where you can configure interface options.

You can get back to a privileged EXEC mode by typing the end command. You can also type CTRL + C to exit the configuration mode.



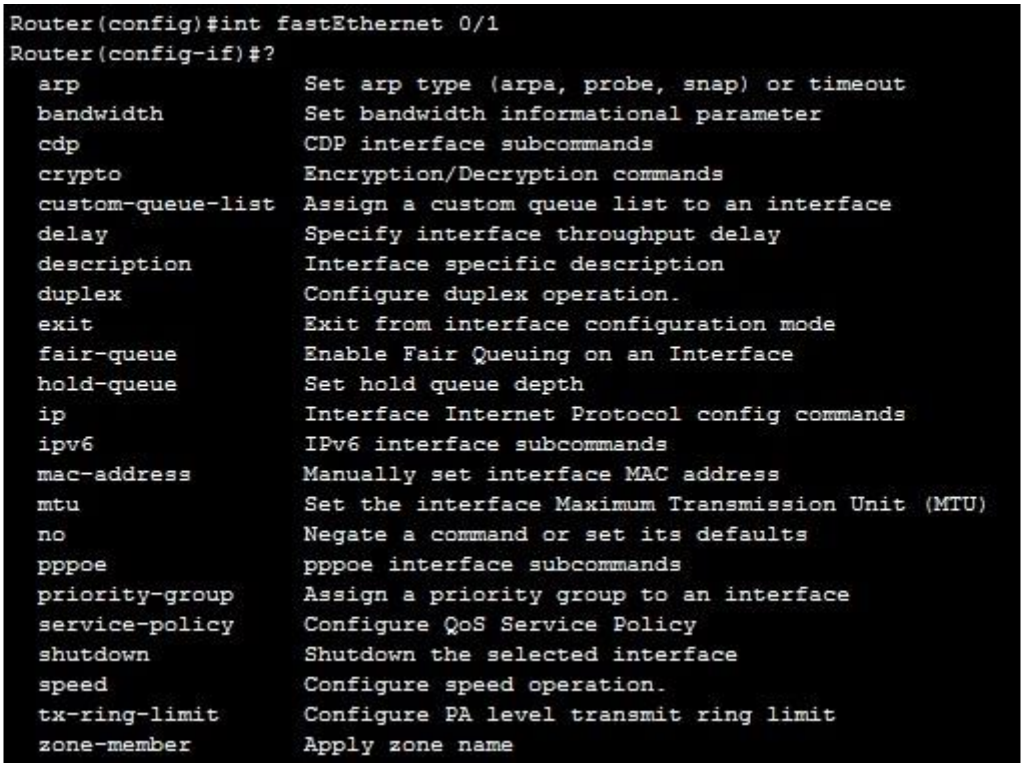
```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#?
Configure commands:
  aaa                   Authentication, Authorization and Accounting.
  access-list           Add an access list entry
  banner                Define a login banner
  boot                  Modify system boot parameters
  cdp                   Global CDP configuration subcommands
  class-map             Configure Class Map
  clock                 Configure time-of-day clock
  config-register       Define the configuration register
  crypto                Encryption module
  do                    To run exec commands in config mode
  dot11                 IEEE 802.11 config commands
  enable                Modify enable password parameters
  end                   Exit from configure mode
  exit                  Exit from configure mode
  hostname              Set system's network name
  interface             Select an interface to configure
  ip                    Global IP configuration subcommands
  ipv6                  Global IPv6 configuration commands
  line                  Configure a terminal line
  logging               Modify message logging facilities
  login                 Enable secure login checking
  mac-address-table     Configure the MAC address table
  no                    Negate a command or set its defaults
  ntp                   Configure NTP
  parser                Configure parser
  policy-map            Configure QoS Policy Map
  priority-list         Build a priority list
  privilege             Command privilege parameters
  queue-list            Build a custom queue list
  radius-server         Modify Radius query parameters
  router                Enable a routing process
  secure                Secure image and configuration archival commands
  security              Infra Security CLIs
  service               Modify use of network based services
  snmp-server           Modify SNMP engine parameters
  spanning-tree         Spanning Tree Subsystem
  tacacs-server         Modify TACACS query parameters
  username              Establish User Name Authentication
  vpdn                  Virtual Private Dialup Network
  vpdn-group            VPDN group configuration
  zone                  FW with zoning
  zone-pair             Zone pair command
```

Sub mode commands

A global configuration mode contains many sub modes. For example, if you want to configure an interface you have to enter that interface configuration mode. Each sub mode contains only commands that pertain to the resource that is being configured.

To enter the interface configuration mode you need to specify which interface you would like to configure. This is done by using the interface

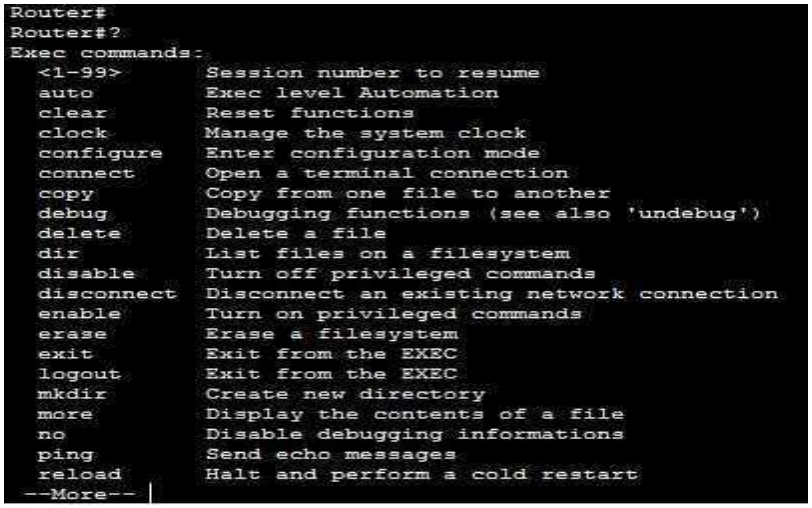
INTERFACE_TYPE/INTERFACE_NUMBER global configuration command, where INTERFACE_TYPE represents the type of an interface (Ethernet, Fast Ethernet, Serial…) and INTERFACE_NUMBER represents the interface number, since Cisco devices usually have more than one physical interface. Once inside the interface configuration mode, you can get a list of available commands by typing the "?" character. Each sub mode has its own prompt. Notice how the command prompt was changed to Router(config-if) after I've entered the interface sub mode:



```
Router(config)#int fastEthernet 0/1
Router(config-if)#?
  arp                  Set arp type (arpa, probe, snap) or timeout
  bandwidth            Set bandwidth informational parameter
  cdp                  CDP interface subcommands
  crypto               Encryption/Decryption commands
  custom-queue-list    Assign a custom queue list to an interface
  delay                Specify interface throughput delay
  description          Interface specific description
  duplex               Configure duplex operation.
  exit                 Exit from interface configuration mode
  fair-queue           Enable Fair Queuing on an Interface
  hold-queue           Set hold queue depth
  ip                   Interface Internet Protocol config commands
  ipv6                 IPv6 interface subcommands
  mac-address          Manually set interface MAC address
  mtu                  Set the interface Maximum Transmission Unit (MTU)
  no                   Negate a command or set its defaults
  pppoe                pppoe interface subcommands
  priority-group       Assign a priority group to an interface
  service-policy       Configure QoS Service Policy
  shutdown             Shutdown the selected interface
  speed                Configure speed operation.
  tx-ring-limit        Configure PA level transmit ring limit
  zone-member          Apply zone name
```

Get help in IOS

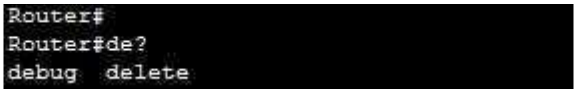
You can use the question mark to display a list of commands available in the prompt you are in:



```
Router#
Router#?
Exec commands:
  <1-99>       Session number to resume
  auto         Exec level Automation
  clear        Reset functions
  clock        Manage the system clock
  configure    Enter configuration mode
  connect      Open a terminal connection
  copy         Copy from one file to another
  debug        Debugging functions (see also 'undebug')
  delete       Delete a file
  dir          List files on a filesystem
  disable      Turn off privileged commands
  disconnect   Disconnect an existing network connection
  enable       Turn on privileged commands
  erase        Erase a filesystem
  exit         Exit from the EXEC
  logout       Exit from the EXEC
  mkdir        Create new directory
  more         Display the contents of a file
  no           Disable debugging informations
  ping         Send echo messages
  reload       Halt and perform a cold restart
 --More--
```

If the output spans more than one page, press the spacebar to display the following page of commands, or press Enter to go one command at a time. To quit the output, press q.

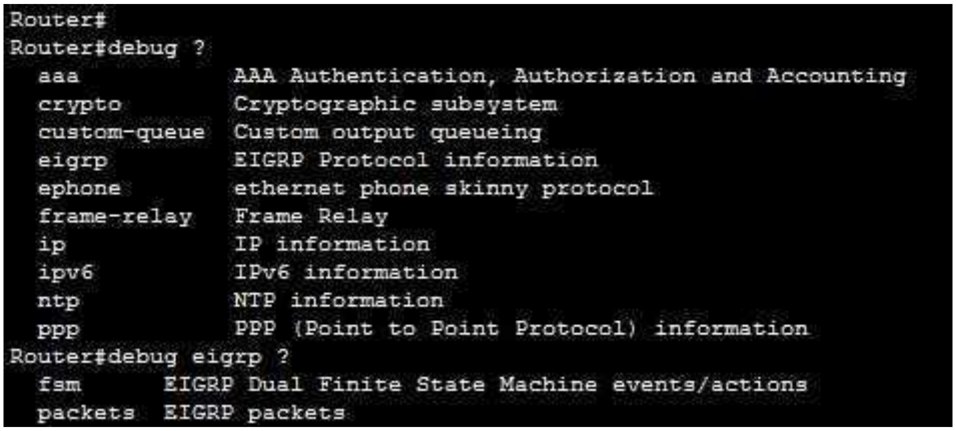
To display only commands that start with a particular character or a string of characters, type the letters and then press the question mark:



```
Router#
Router#de?
debug  delete
```

In the picture above you can see that we've displayed all commands that start with de.

If the command is more than one word long, you can use the question mark to display the next command in a string:

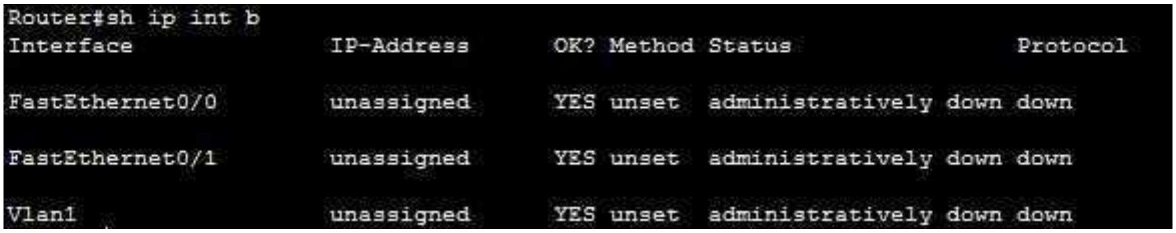


```
Router#
Router#debug ?
  aaa           AAA Authentication, Authorization and Accounting
  crypto        Cryptographic subsystem
  custom-queue  Custom output queueing
  eigrp         EIGRP Protocol information
  ephone        ethernet phone skinny protocol
  frame-relay   Frame Relay
  ip            IP information
  ipv6          IPv6 information
  ntp           NTP information
  ppp           PPP (Point to Point Protocol) information
Router#debug eigrp ?
  fsm      EIGRP Dual Finite State Machine events/actions
  packets  EIGRP packets
```

In the picture above you can see that we've displayed all commands that can follow the command debug. We then displayed all commands that can follow the commands debug eigrp.

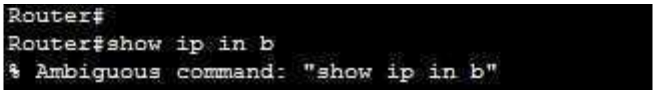
You can also autocomplete a command. Just type the first few characters and press Tab. If there is only a single match, IOS will complete the command.

You don't have to type an entire word to finish a command. Just can type just the first letter or a couple of letters, and if there is only a single match, IOS will understand what are you trying to accomplish. For example, you can type sh ip int b instead of a longer version, show ip interface brief:



```
Router#sh ip int b
Interface              IP-Address      OK? Method Status                Protocol

FastEthernet0/0        unassigned      YES unset  administratively down down

FastEthernet0/1        unassigned      YES unset  administratively down down

Vlan1                  unassigned      YES unset  administratively down down
```

Note that we were able to execute the command above because each set of characters had only one match in the list of commands. If we've typed sh ip in b instead, IOS would not have understood our intention:



```
Router#
Router#show ip in b
% Ambiguous command: "show ip in b"
```

The % Ambiguous command: "show ip in b" message was displayed because the third keyword, in, has more than one meaning (inspect or interface).

Running & startup configuration

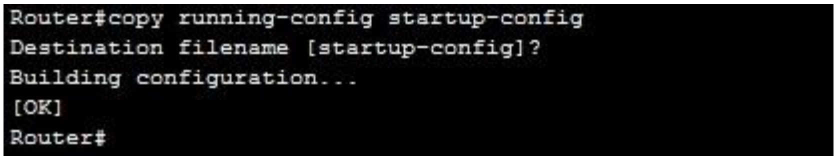
Cisco devices store commands in two configuration files:

- *startup configuration*
- *running configuration*

Immediately after you type a command in the global configuration mode, it will be stored in the running configuration. A running configuration resides in a device's RAM, so if a device loses power, all configured commands will be lost.

To avoid this scenario, you need to copy your current configuration into the startup configuration. A startup configuration is stored in the nonvolatile memory of a device, which means that all configuration changes are saved even if the device loses power.

To copy your running configuration into the startup configuration you need to type the command copy running-configuration startup-configuration.



```
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
```

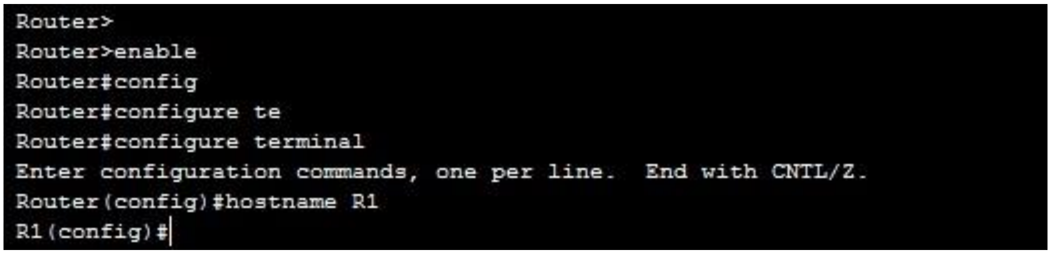
IOS basic commands

In this article we will go through some basic IOS commands.

Hostname command

The hostname command is used to configure the device hostname. Because this command changes a device configuration, it must be entered in the global configuration mode. After typing the command, the prompt will change and display the new hostname.

Here is an example that shows you how to change a hostname of a device.
First, enter the global configuration mode by typing the enable command in the user EXEC mode and the configuration terminal command in the privileged EXEC mode. Once inside the global configuration mode, type the command hostname R1. Notice how the prompt was changed to reflect the configured value.



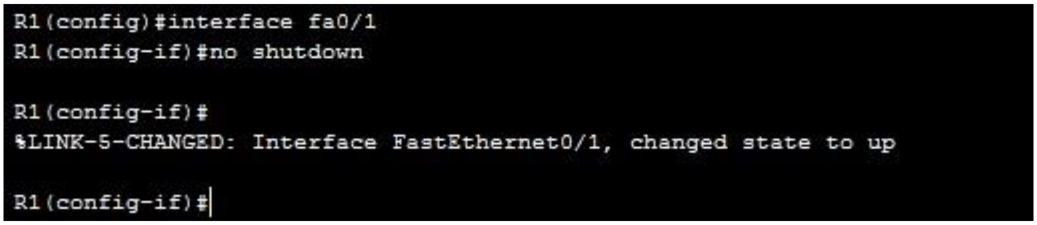
```
Router>
Router>enable
Router#config
Router#configure te
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#
```

No shutdown command

By default, all interfaces on a Cisco router are turned off. To enable an interface, the no shutdown command is used. You first need to enter the sub mode of the interface that you want to configure. You can do that by using the global configuration mode command interface INTERFACE_TYPE/ INTERFACE_NUMBER. You can get a list of available interfaces by typing the '?' character after the interface command.

You may notice that the prompt has changed to reflect the mode you are currently in. For the interface mode the HOSTNAME#(config-if) prompt is shown.

Once inside the interface mode, you can enable an interface by typing the no shutdown command.



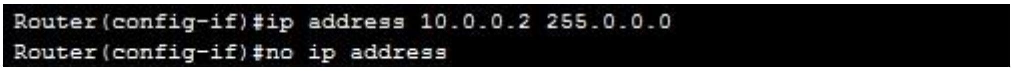
```
R1(config)#interface fa0/1
R1(config-if)#no shutdown

R1(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

R1(config-if)#
```

IP address command

The ip address interface mode command is used to assign an IP address to an interface. The syntax of this command is ip address IP_ADDRESS SUBNET_MASK. For example, if we want to assign an IP address of 10.0.0.1 with the subnet mask 255.0.0.0 to a interface, we would use the following command:

```
ip address 10.0.0.1 255.0.0.0
```

What if you had made a mistake and written the ip address 10.0.0.2 255.0.0.0 command instead of the command above? Well, you can remove the wrong IP address by typing the same command, but this time with the no keyword in front of it, in our case no ip address. You can remove any command from your IOS configuration by using the no keyword in front of the command.



```
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no ip address
```

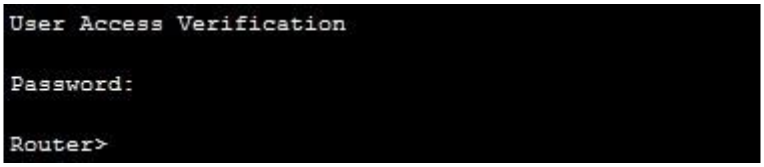
Setting up passwords

Each Cisco IOS device has the built-in authentication features. There are three basic ways to configure authentication on a device:

- *Configure a password for the console access – by default, the console access doesn't requires a password. You can configure a password for the console access by using the following set of commands:*

```
HOSTNAME(config) line console 0

HOSTNAME(config-line) password PASSWORD

HOSTNAME(config-line) login
```

This will force a user to type the password when trying to access the device through the console port.

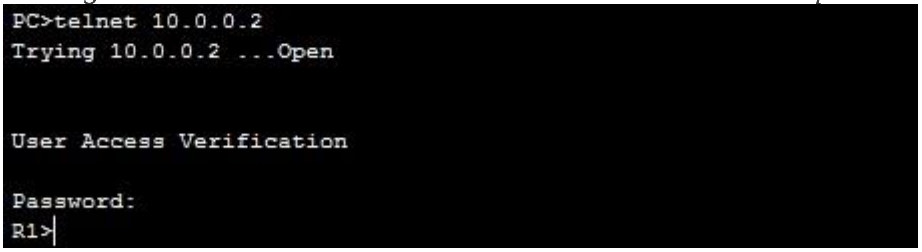


```
User Access Verification

Password:

Router>
```

- *Configure a password for the telnet access – by default, the telnet access is disabled. You need to enable it. This is done using the following sequence of commands:*

```
HOSTNAME (config) line vty FIRST_VTY LAST_VTY

HOSTNAME (config-line) password PASSWORD

HOSTNAME (config-line) login
```

The first command defines a range of virtual terminal sessions that you would like to configure. A virtual session can be a telnet or SSH session. Cisco devices usually supports 16 concurrent VTY sessions. So, this command usually looks like this: line vty 0 15.
The login command allows a remote access to a device. It is required in order for telnet to work.

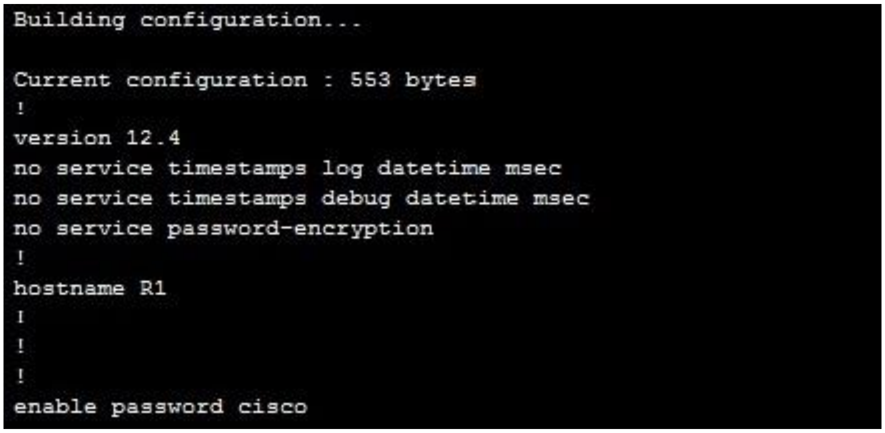


```
PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open


User Access Verification

Password:
R1>
```

- *Configure a password for the privileged EXEC mode – from the privileged EXEC mode you can enter the global configuration mode and change the configuration of a device. Therefore it is important to prevent an unauthorized user from entering the global configuration mode. You can do that by setting up a password to enter the privileged EXEC mode. This can be done in two ways:*

```
HOSTNAME(config) enable password PASSWORD

HOSTNAME(config) enable secret PASSWORD
```

Both of the commands above accomplish the same thing, but with one major difference. The enable secret PASSWORD commands encrypts the password, while the enable password PASSWORD command doesn't, which means that an unauthorized user could just read a password from the device configuration:



```
Building configuration...

Current configuration : 553 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname R1
!
!
!
enable password cisco
```

Notice how the password (cisco) is visible in the device's configuration.

Service password-encryption command

By default, passwords configured using the enable password command and passwords for the console or telnet access are stored in clear text in the configuration file. This presents a security risk because an attacker could easily find out passwords. The global configuration service password-encryption command encrypts all passwords configured.

It is important to note that this type of password encryption is not consider especially secure, since the algorithm used can be easily cracked. Cisco recommends using this command only with additional security measures.

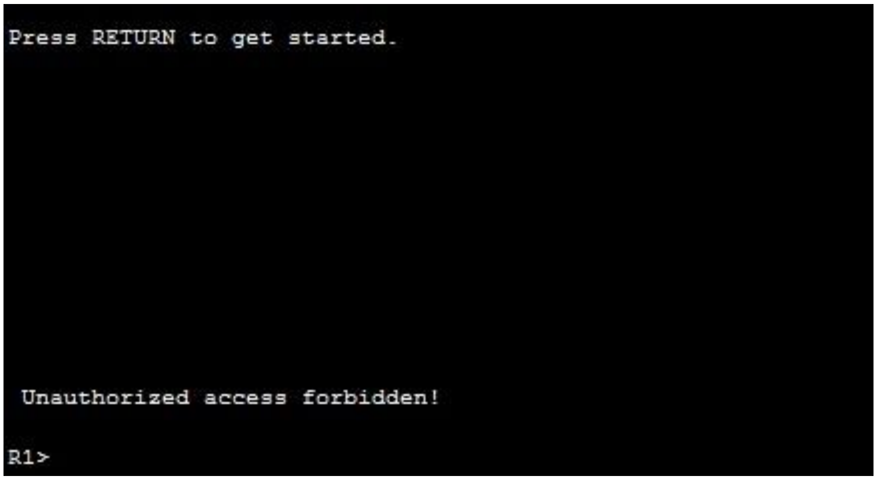
Configuring banners

You can display a banner on a Cisco device. A banner is usually shown before the login prompt. It is usually some text that appears on the screen when a user connect to the device (e.g. some legal information).

The most commonly used banner is the Message Of The Day (MOTD) banner. This banner, if configured, is shown before the login prompt to every user that is trying to establish a session with the device. The following global configuration command is used to configure a MOTD banner:

```
hostname(config) banner motd DELIMITING_CHARACTER TEXT DELIMITING_CHARACTER
```

A delimiting character is a character of your choice. Its purpose is to signify the start and end of a text that will appear in the banner. For example, the command banner motd # Unauthorized access forbidden! # will show the following text: Unauthorized access forbidden!.



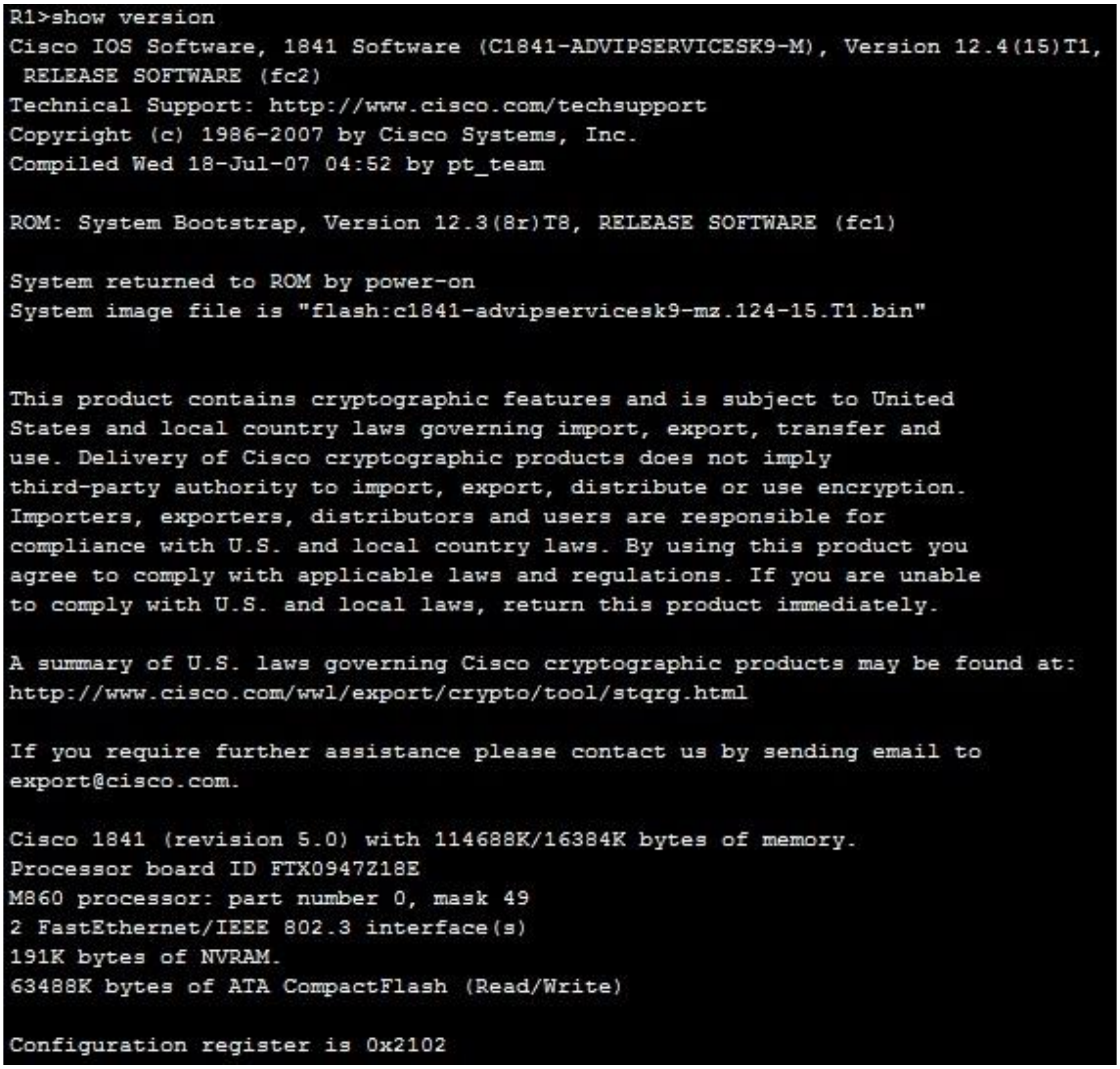
```
Press RETURN to get started.



 Unauthorized access forbidden!

R1>
```

Show version command

The show version command is used to display information about a Cisco device. The command can be entered in both the user EXEC and privileged EXEC mode. By using this command you can find out many useful information about your Cisco device, such as:

- *Software Version – IOS software version*
- *System up-time – time since last reboot*
- *Software image name – IOS filename stored in flash*
- *Hardware Interfaces – interfaces available on device*
- *Configuration Register value – bootup specifications, console speed setting, etc.*
- *Amount of RAM memory – amount of RAM memory*
- *Amount of NVRAM memory*
- *Amount of Flash memory*

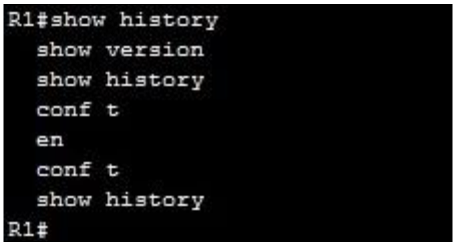
The following example shows the output of the command:



```
R1>show version
Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 12.4(15)T1,
 RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Wed 18-Jul-07 04:52 by pt_team

ROM: System Bootstrap, Version 12.3(8r)T8, RELEASE SOFTWARE (fc1)

System returned to ROM by power-on
System image file is "flash:c1841-advipservicesk9-mz.124-15.T1.bin"


This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

Cisco 1841 (revision 5.0) with 114688K/16384K bytes of memory.
Processor board ID FTX0947Z18E
M860 processor: part number 0, mask 49
2 FastEthernet/IEEE 802.3 interface(s)
191K bytes of NVRAM.
63488K bytes of ATA CompactFlash (Read/Write)

Configuration register is 0x2102
```

Show history command

An IOS device stores, by default, 10 last commands you have entered in your current EXEC session. You can use the show history command from the user EXEC or privileged EXEC mode to display them.



```
R1#show history
  show version
  show history
  conf t
  en
  conf t
  show history
R1#
```

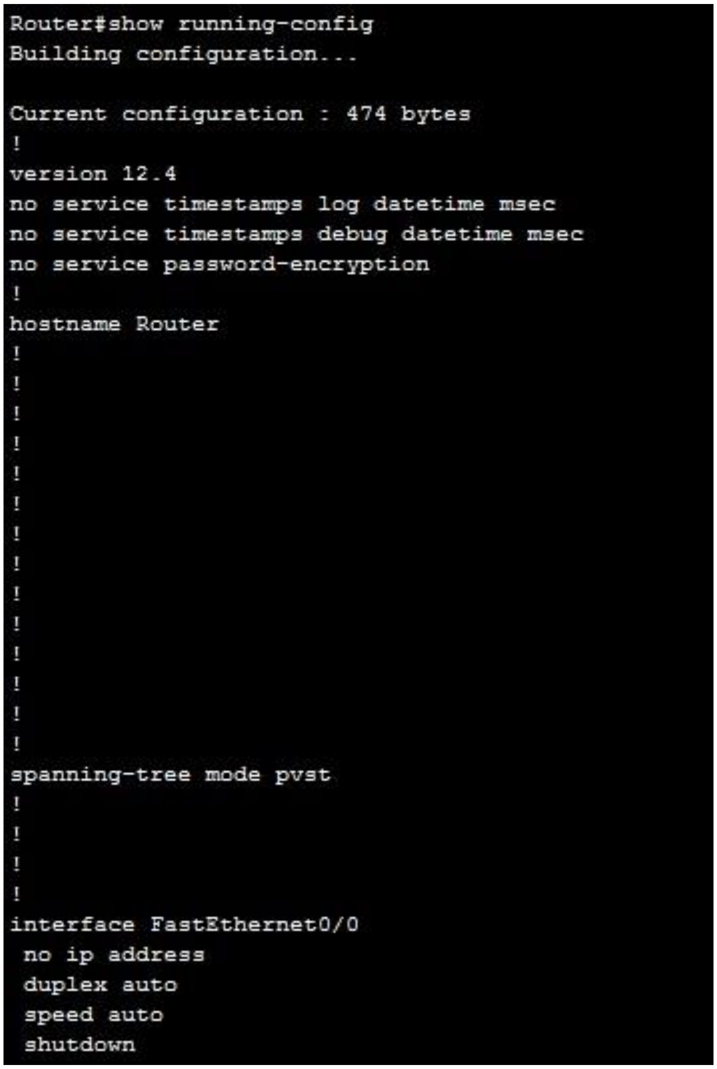
You can set a number of command saved in the buffer for the current terminal session by using the terminal history size NUMBER command from the user EXEC or privileged EXEC mode.

NOTE

Another way to recall your command from the history buffer is by using the up arrow key on your keyboard. Most recent command is recalled first.

Show running-configuration & show startup-configuration commands

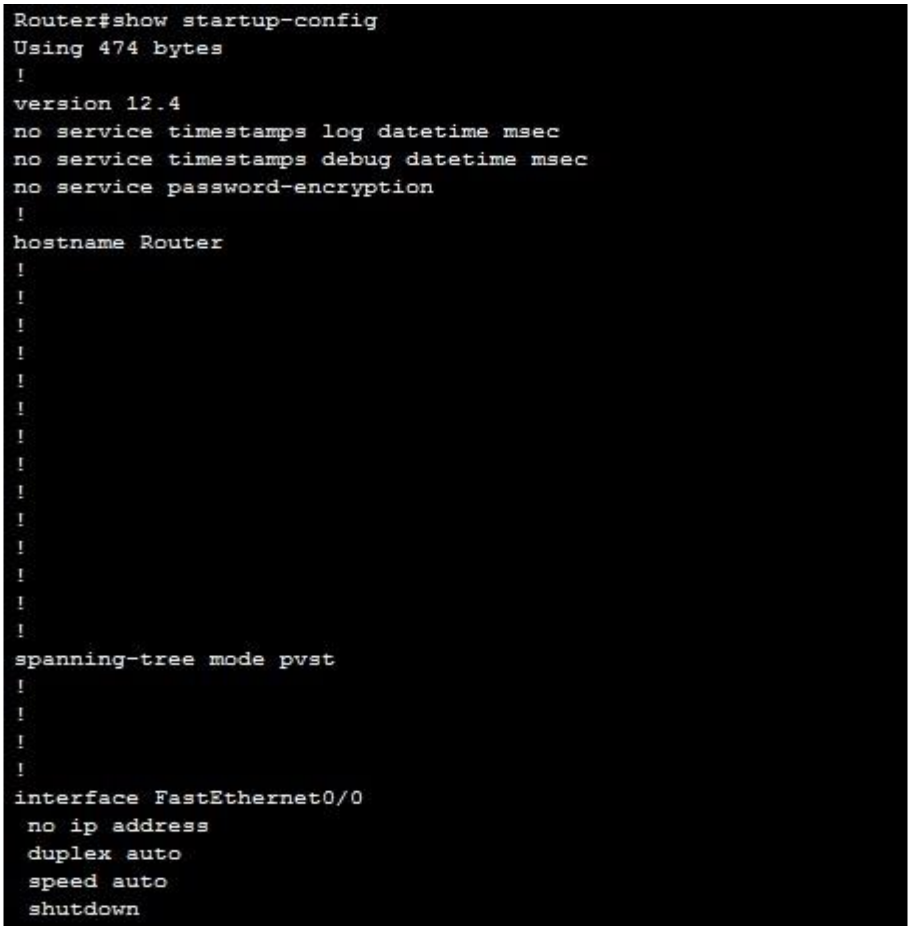
After you have changed the configuration of your device you can verify its configuration. To display the current configuration, type show running-configuration from the privileged EXEC mode. This show the configuration that is stored in a device's RAM.



```
Router#show running-config
Building configuration...

Current configuration : 474 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
!
!
!
!
!
!
!
!
!
!
!
spanning-tree mode pvst
!
!
!
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
```

After you have stored your running configuration into the startup configuration, you can view the saved configuration using the show startup-config command from the privileged EXEC mode.

This command shows the configuration that is currently stored in the device's NVRAM. This configuration will be loaded next time the device is restarted.



```
Router#show startup-config
Using 474 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
!
!
!
!
!
!
!
!
!
!
!
spanning-tree mode pvst
!
!
!
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
```

show command

We’ve already mentioned a couple of show commands in the previous sections, so you should already we somewhat aware of this command. This command is used to display the device’s configuration, statistics, command history, interface status… The show command is invoked from the enable mode and can accept a lot of parameters:

```
Floor1#show ?

aaa Show AAA values

access-lists List access lists

arp Arp table

cdp CDP information

class-map Show QoS Class Map

clock Display the system clock

controllers Interface controllers status
```

```
crypto Encryption module

debugging State of each debugging option

dhcp Dynamic Host Configuration Protocol status

dot11 IEEE 802.11 show information

file Show filesystem information

flash: display information about flash: file system

…

terminal Display terminal configuration parameters

users Display information about terminal lines

version System hardware and software status

vlan-switch VTP VLAN status

vtp Configure VLAN database
```

Here is a brief description of the most popular show commands:

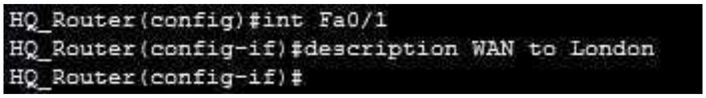
- *show running-config – displays the running (current) configuration of your device:*
- *show startup-config – displays the startup configuration of your device:*
- *show ip interface brief – provides information about the interfaces on a router, including the logical (IP) address and status:*
- *show history – shows the command history:*
- *show interface INTERFACE – displays the status of the specified interface:*
- *show version – shows information about the device, such as the IOS version running on the device, number of interfaces, device model, time of the last reboot, amount of memory available on the device, etc.*

Configure descriptions

Adding a description to an interface on a Cisco device doesn't provide any extra functionality, but it is useful for administrative purposes, since it will help you to remember the interface function. A description of an interface is locally significant and can be up to 240 characters long. It can be set using the description command from the interface sub mode:

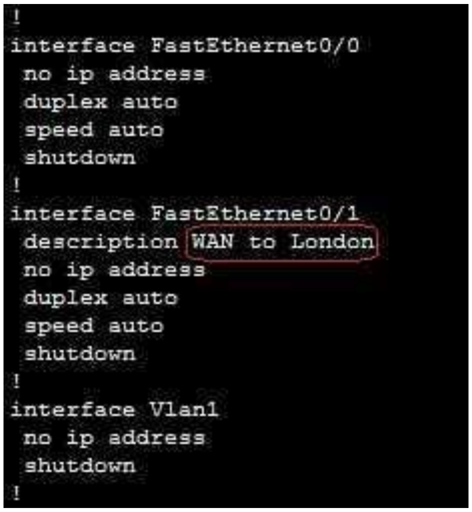
```
DEVICE(config) interface Fa0/1

DEVICE(config-if) description WAN to London
```

Example configuration:



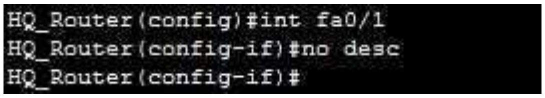
```
HQ_Router(config)#int Fa0/1
HQ_Router(config-if)#description WAN to London
HQ_Router(config-if)#
```

The description is displayed in the output of the show running-config command:



```
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface FastEthernet0/1
 description WAN to London
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
```

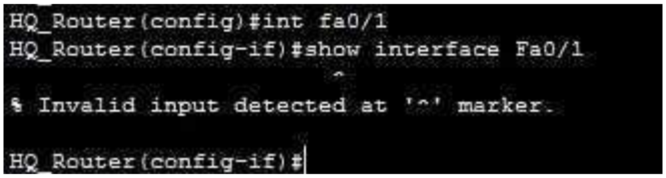
To erase the description, use the no description interface mode command (or the shortcut no desc):



```
HQ_Router(config)#int fa0/1
HQ_Router(config-if)#no desc
HQ_Router(config-if)#
```

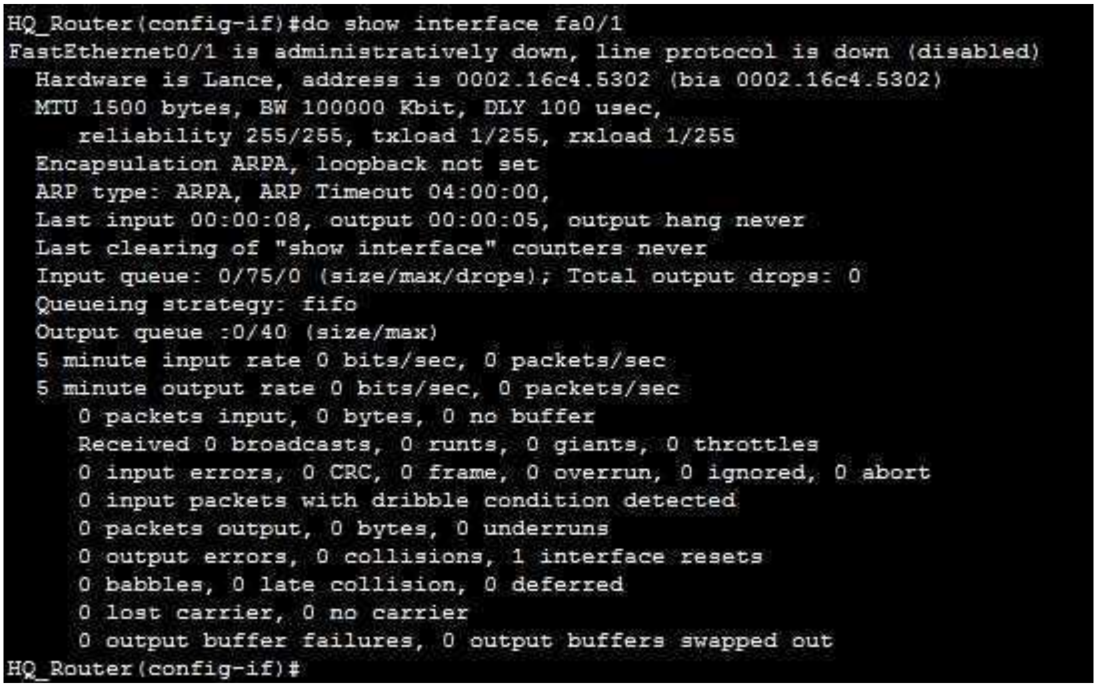
Run privileged commands within global config mode

Beginning with the IOS 12.3, the privileged-exec mode commands (such as show running-configuration, show interface status, etc.) can be executed within the global configuration mode and its sub modes. This allows you to execute privileged-exec mode commands without needing to exit the current configuration mode. Here is an example that explains the usefulness of this feature:



```
HQ_Router(config)#int fa0/1
HQ_Router(config-if)#show interface Fa0/1
                       ^
% Invalid input detected at '^' marker.

HQ_Router(config-if)#
```

In the example above you can see that we're currently in the interface sub mode. We want to get more information about the interface with the show interface Fa0/1 command, but we got an error because the command is not available in this mode. However, if we use the do keyword in front of the command, the command will succeed:

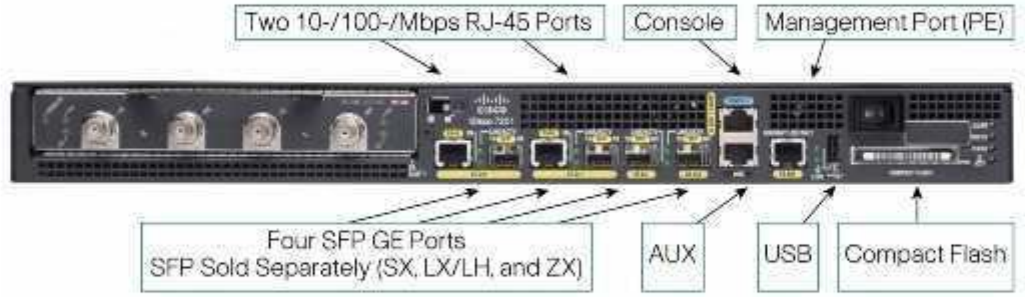


```
HQ_Router(config-if)#do show interface fa0/1
FastEthernet0/1 is administratively down, line protocol is down (disabled)
  Hardware is Lance, address is 0002.16c4.5302 (bia 0002.16c4.5302)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00,
  Last input 00:00:08, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
HQ_Router(config-if)#
```

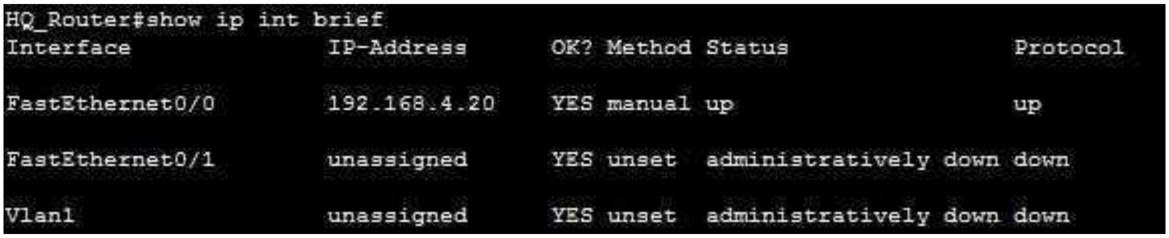
The command was now executed because of the do keyword. Notice that we're still in the interface sub mode and we can continue with the interface configuration.

Ports on an IOS device

Cisco uses the term interface to refer to physical ports on an IOS device. Interfaces can be configured with different settings, depending on the type of the interface and whether you are configuring an interface on a router or on a switch. For example, the Cisco 7201 Router has four GE physical ports (image source: Cisco):


Two 10-/100-/Mbps RJ-45 Ports
Console
Management Port (PE)
Four SFP GE Ports
SFP Sold Separately (SX, LX/LH, and ZX)
AUX
USB
Compact Flash

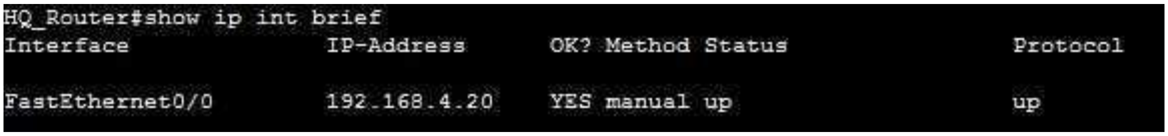
To display the router interfaces in IOS, use the show ip int brief command from the privileged exec mode:



```
HQ_Router#show ip int brief
Interface              IP-Address      OK? Method Status                Protocol

FastEthernet0/0        192.168.4.20    YES manual up                    up

FastEthernet0/1        unassigned      YES unset  administratively down down

Vlan1                  unassigned      YES unset  administratively down down
```

In the output above we can see that this router has 2 physical interfaces – FastEthernet0/0 and FastEthernet0/1.

Consider the output for the Fa0/0 interface:

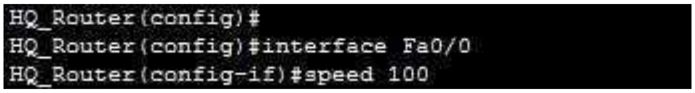


```
HQ_Router#show ip int brief
Interface              IP-Address      OK? Method Status                Protocol

FastEthernet0/0        192.168.4.20    YES manual up                    up
```

Here is a brief description of each column:

- *Interface – displays the type of the interface, in this case Fast Ethernet 0/0. The first zero specifies the physical slot on the router, while the second zero specifies the port number.*
- *IP-Address – displays the interface's IP address.*
- *OK? – YES in this column signifies that the IP address is currently valid.*
- *Method – manual in this column means that the interface has been manually configured. DHCP means that the interface has been configured using DHCP.*
- *Status – up indicates that the interface is administratively up.*
- *Protocol – up indicates that the interface is operational.*

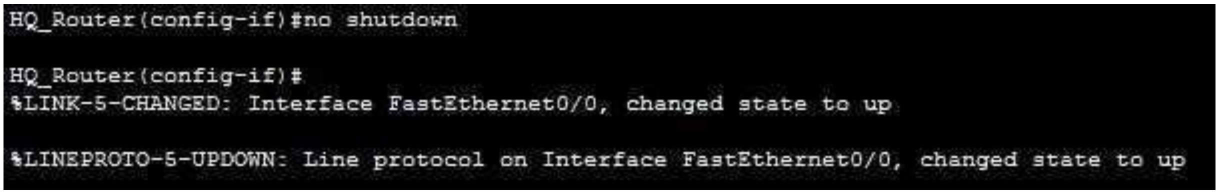
To configure a specific interface, use the interface TYPE SLOT/PORT command from the global config mode. This puts us in the interface sub mode, where we can configure various interface options:



```
HQ_Router(config)#
HQ_Router(config)#interface Fa0/0
HQ_Router(config-if)#speed 100
```

In the example above you can see that we've configured the speed option for the interface Fast Ethernet 0/0.

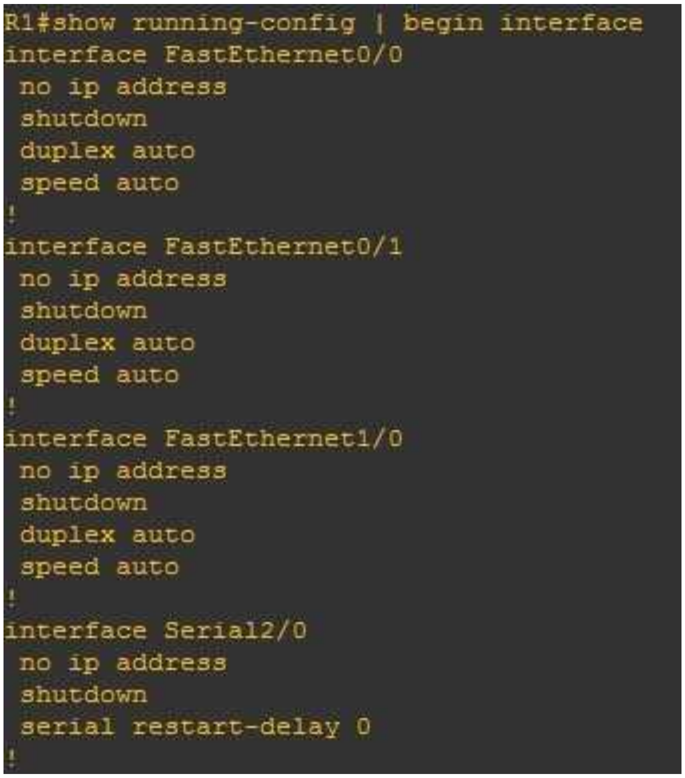
By default, all ports on a Cisco switch are up and running as soon as you power-on the device. This means that all you need is to connect your devices and the switch and you are good to go. This isn't the case with Cisco routers, however. You need to manually enable each interface on a router with the no shutdown interface mode command:



```
HQ_Router(config-if)#no shutdown

HQ_Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

Pipe character in IOS

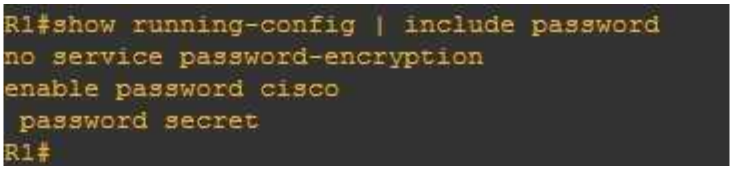
IOS supports the use of the pipe character (represented with the | character) to filter the output of the show commands. The pipe function takes the output of the command and sends it to another function, such as begin or include. This way, you can filter the output to find the section of the output that interests you. Here are a couple of examples:



```
R1#show running-config | begin interface
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial2/0
 no ip address
 shutdown
 serial restart-delay 0
!
```

In the picture above you can see that we've entered the show running-config | begin interface command. This command starts the output from the first occurrence of the word interface.

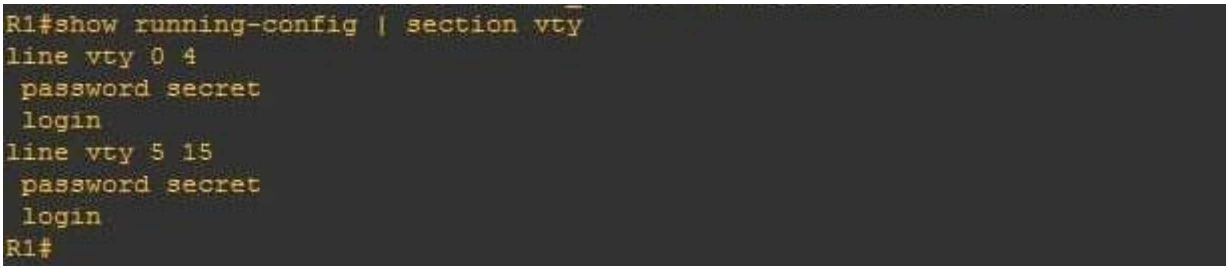
Another example, this time with include:



```
R1#show running-config | include password
no service password-encryption
enable password cisco
 password secret
R1#
```

As you can see from the example above, the include function displays only lines that include the word password.

To display only the section of the output about a certain feature, use the section function:



```
R1#show running-config | section vty
line vty 0 4
 password secret
 login
line vty 5 15
 password secret
 login
R1#
```

You can see in the example above that the command displayed only the vty section of the running configuration.

NOTE

Cisco Packet Tracer doesn’t support the pipe function. The examples above were created in GNS3.

IOS boot sequence

The IOS boot sequence is a process performed after an Cisco IOS device is powered on. The IOS device performs a power-on self-test (POST) to test its hardware components and choose an IOS image to load. The boot sequence consists of the following steps:

1. The device performs the power-on self-test (POST) process to discover and verify its hardware components.

2. If the POST test is successful, the bootstrap program is copied from ROM into RAM.

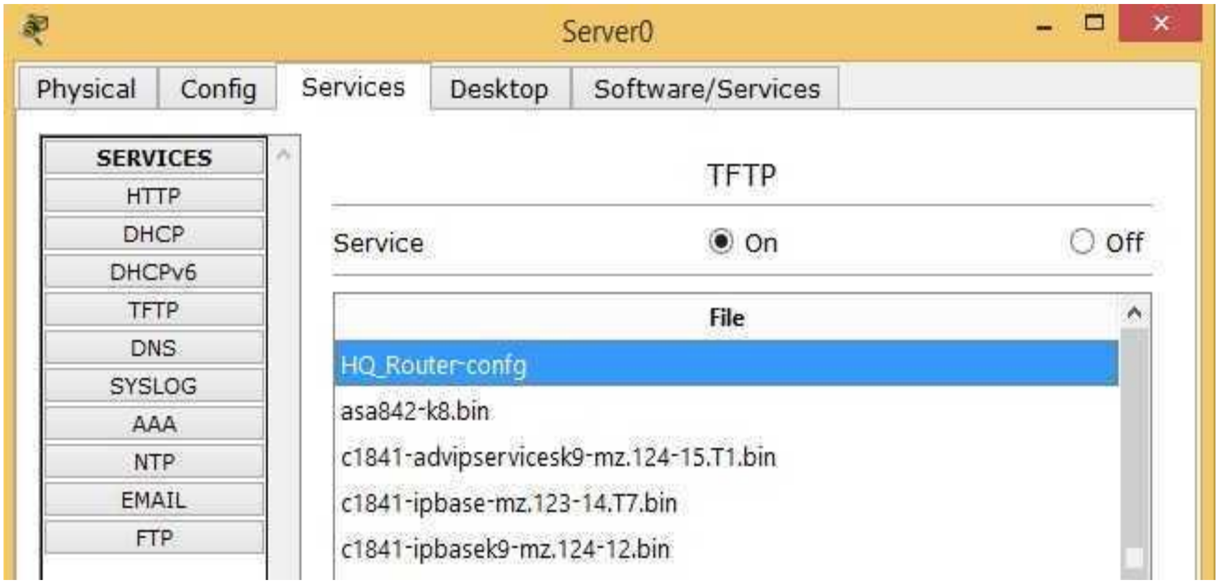
3. The bootstrap program decides which IOS image to load from the flash memory into RAM, and then loads the chosen IOS.

4. IOS finds the startup configuration file, usually located in NVRAM, and loads it into RAM as the running configuration.

Backing up IOS configuration

It is always a good idea to have a backup copy of the configuration of your IOS device. IOS configurations are usually copied to a TFTP server using the copy command. You can backup both the startup configuration and the running configuration of your device. The copy commands accepts two parameters: the first parameter is the from location, and the second it the to location.

TFTP is a client-server network protocol used to send and receive files. To backup files to a TFTP server, you will have to set it up first. You can use the Packet Tracer to do so; just add a Server to your topology, assign it an IP address and enable the TFTP service:

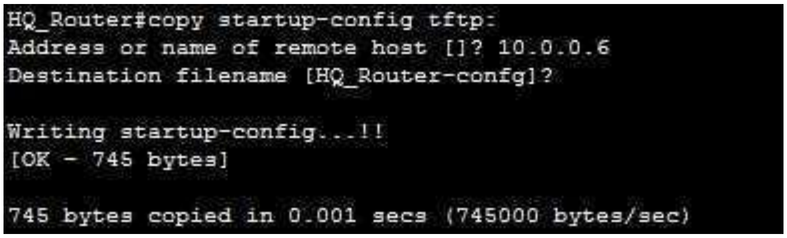


TFTP

Service ● On ○ Off

File
HQ_Router-confg
asa842-k8.bin
c1841-advipservicesk9-mz.124-15.T1.bin
c1841-ipbase-mz.123-14.T7.bin
c1841-ipbasek9-mz.124-12.bin

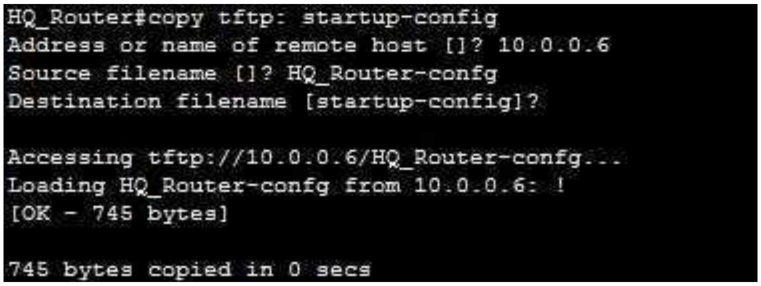
To backup the startup configuration to a TFTP server, you can use the copy startup-config tftp: command:



```
HQ_Router#copy startup-config tftp:
Address or name of remote host []? 10.0.0.6
Destination filename [HQ_Router-confg]?

Writing startup-config...!!
[OK - 745 bytes]

745 bytes copied in 0.001 secs (745000 bytes/sec)
```

Remember, the first parameter after the copy keyword is the from location, while the second one is the to location. In our case, the from location is the current startup-config, and the to location is the remote TFTP server.

To restore the configuration, just switch the order of the parameters – copy tftp startup-config:



```
HQ_Router#copy tftp: startup-config
Address or name of remote host []? 10.0.0.6
Source filename []? HQ_Router-confg
Destination filename [startup-config]?

Accessing tftp://10.0.0.6/HQ_Router-confg...
Loading HQ_Router-confg from 10.0.0.6: !
[OK - 745 bytes]

745 bytes copied in 0 secs
```

Notice that we had to specify the filename, along with the IP address of the TFTP server.

CHAPTER SEVEN
WHAT IS IP ROUTING

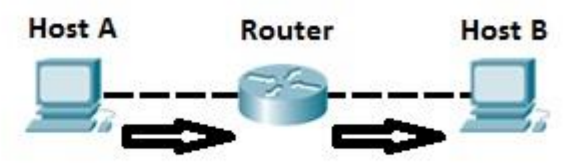
We Will Cover These Topics in This Chapter

- ***Routing protocols***
- ***Type Routing Protocols***
- ***Connected, static & dynamic routes***
- ***Administrative distance & metric***
- ***Difference between distance vector and link state routing protocols***

What is IP routing?

IP routing is the process of sending packets from a host on one network to another host on a different remote network. This process is usually done by routers. Routers examine the destination IP address of a packet, determine the next-hop address, and forward the packet. Routers use routing tables to determine a next hop address to which the packet should be forwarded.

Consider the following example of IP routing:

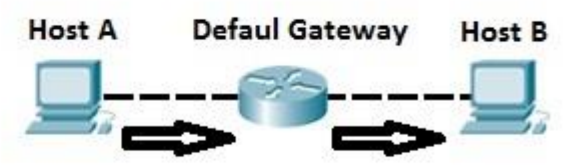


Host A wants to communicate with host B, but host B is on another network. Host A is configured to send all packets destined for remote networks to router R1. Router R1 receives the packets, examines the destination IP address and forwards the packet to the outgoing interface associated with the destination network.

Default gateway

A default gateway is a router that hosts use to communicate with other hosts on remote networks. A default gateway is used when a host doesn't have a route entry for the specific remote network and doesn't know how to reach that network. Hosts can be configured to send all packets destined to remote networks to a default gateway, which has a route to reach that network.

The following example explains the concept of a default gateway more thoroughly.



Host A has an IP address of the router R1 configured as the default gateway address. Host A is trying to communicate with host B, a host on another, remote network. Host A looks up in its routing table to check if there is an entry for that destination network. If the entry is not found, the host sends all data to the router R1. Router R1 receives the packets and forwards them to host B.

Routing table

Each router maintains a routing table and stores it in RAM. A routing table is used by routers to determine the path to the destination network. Each routing table consists of the following entries:

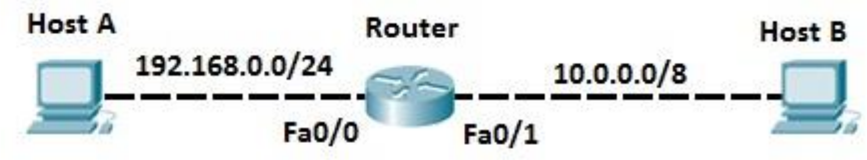
- network destination and subnet mask – specifies a range of IP addresses.
- remote router – IP address of the router used to reach that network.
- outgoing interface – outgoing interface the packet should go out to reach the destination network.

There are three different methods for populating a routing table:

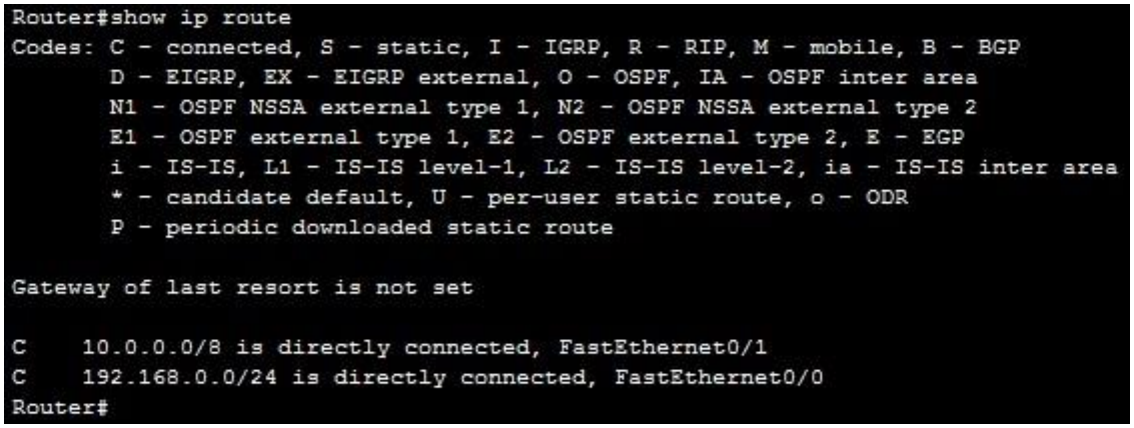
- directly connected subnets
- using static routing
- using dynamic routing

Each of this method will be described in the following chapters.

Consider the following example. Host A wants to communicate with host B, but host B is on another network. Host A is configured to send all packets destined for remote networks to the router. The router receives the packets, checks the routing table to see if it has an entry for the destination address. If it does, the router forwards the packet out the appropriate interface port. If the router doesn't find the entry, it discards the packet.



We can use the show ip route command from the enabled mode to display the router's routing table.



```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/1
C    192.168.0.0/24 is directly connected, FastEthernet0/0
Router#
```

As you can see from the output above, this router has two directly connected routes to the subnets 10.0.0.0/8 and 192.168.0.0/24. The character C in the routing table indicates that a route is a directly connected route. So when host A sends the packet to host B, the router will look up into its routing table and find the route to the 10.0.0.0/8 network on which host B resides. The router will then use that route to route packets received from host A to host B.

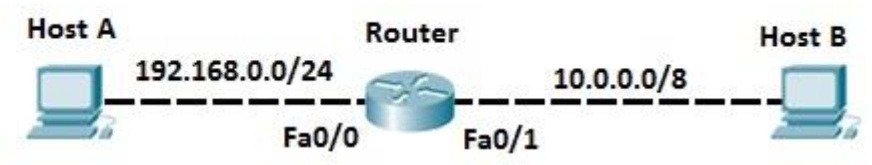
Connected, static & dynamic routes

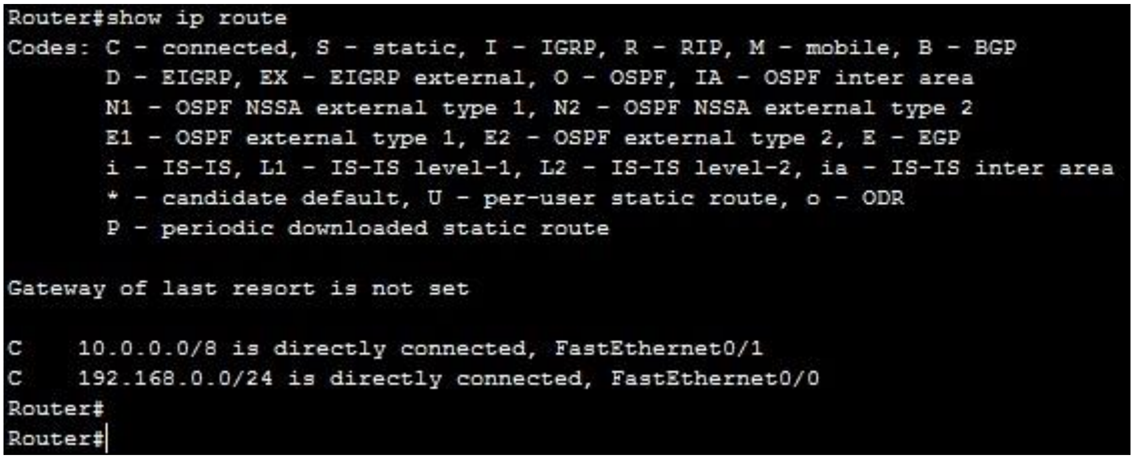
Let's explain the types of routes that can be found in a router's routing table.

Connected routes

Subnets directly connected to a router's interface are added to the router's routing table. Interface has to have an IP address configured and both interface status codes must be in the up and up state. A router will be able to route all packets destined for all hosts in subnets directly connected to its active interfaces.

Consider the following example. The router has two active interfaces, Fa0/0 and Fa0/1. Each interface has been configured with an IP address and is currently in the up-up state, so the router adds these subnets to its routing table.





```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/1
C    192.168.0.0/24 is directly connected, FastEthernet0/0
Router#
Router#
```

As you can see from the output above, the router has two directly connected routes to the subnets 10.0.0.0/8 and 192.168.0.0/24. The character C in the routing table indicates that a route is a directly connected route.

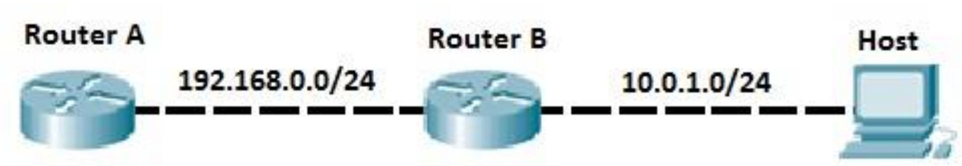
NOTE

You can see only connected routes in a router's routing table by typing the show ip route connected command.

Static routes

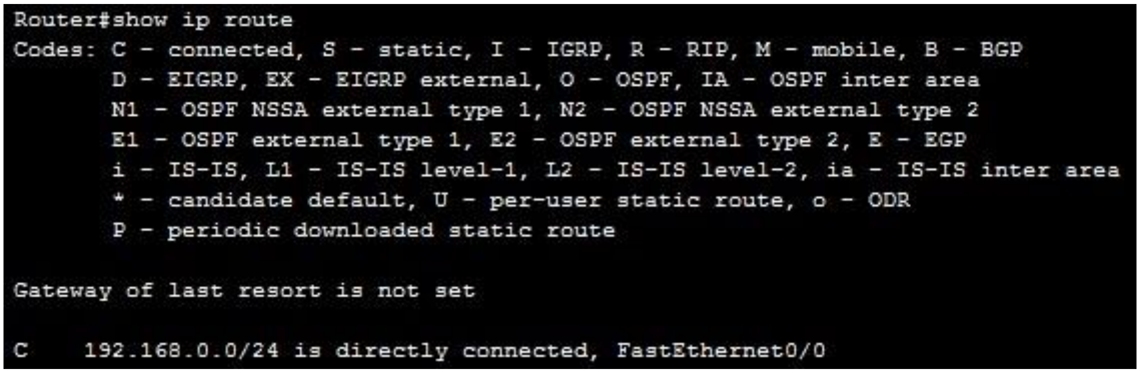
By adding static routes, a router can learn a route to a remote network that is not directly connected to one of its interfaces. Static routes are configured manually by typing the global configuration mode command ip route DESTINATION_NETWORK SUBNET_MASK NEXT_HOP_IP_ADDRESS. This type of configuration is usually used in smaller networks because of scalability reasons (you have to configure each route on each router).

A simple example will help you understand the concept of static routes.



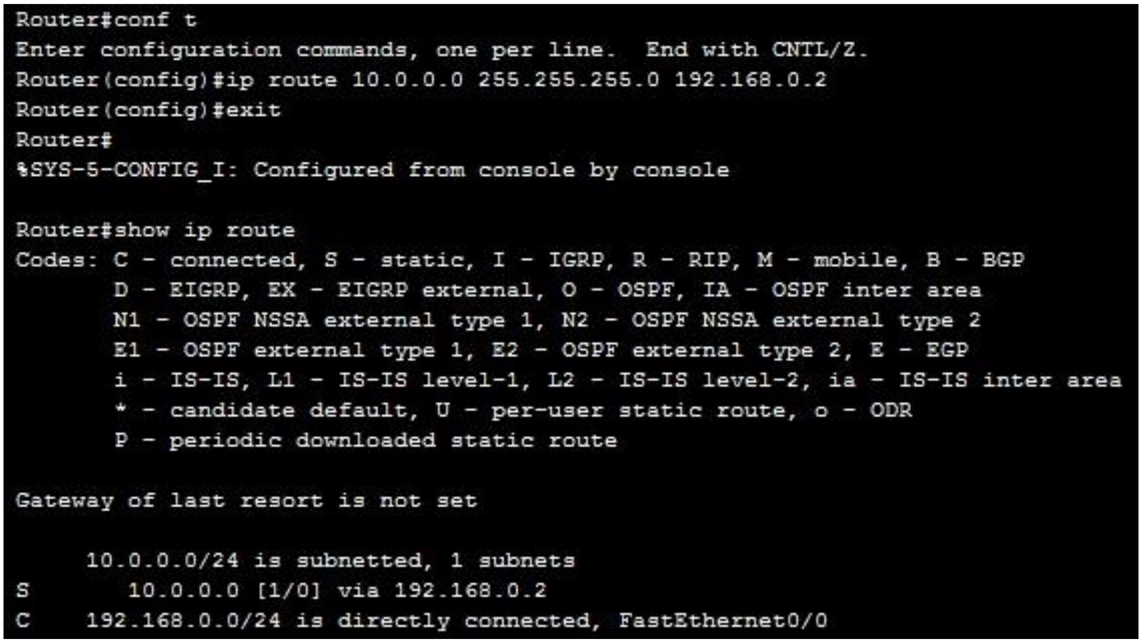
Router A is directly connected to router B. Router B is directly connected to the subnet 10.0.1.0/24. Since that subnet is not directly connected to Router A, the router doesn't know how to route packets destined for that subnet. However. you can configure that route manually on router A.

First, consider the router A's routing table before we add the static route:



```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.0.0/24 is directly connected, FastEthernet0/0
```

Now, we'll use the static route command to configure router A to reach the subnet 10.0.0.0/24. The router now has the route to reach the subnet.



```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.255.255.0 192.168.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
S       10.0.0.0 [1/0] via 192.168.0.2
C    192.168.0.0/24 is directly connected, FastEthernet0/0
```

The character S in the routing table indicates that a route is a statically configured route.

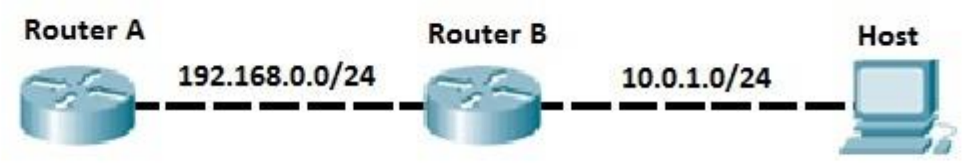
Another version of the ip route command exists. *You don't have to specify the* next-hop IP *address. You can rather specify the exit interface of the local router. In the example above we could have typed the ip route DEST_NETWORK NEXT_HOP_INTERFACE command to instruct router A to send all traffic destined for the subnet out the right interface. In our case, the command would be ip route 10.0.0.0 255.255.255.0 Fa0/0.*

Dynamic routes

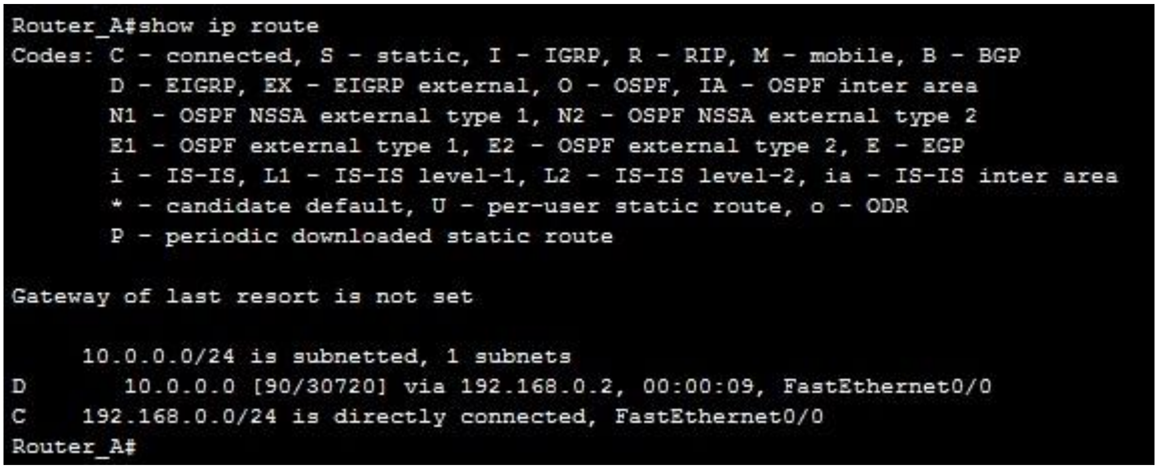
A router can learn dynamic routes if a routing protocol is enabled. A routing protocol is used by routers to exchange routing information with each other. Every router in the network can then use information to build its routing table. A routing protocol can dynamically choose a different route if a link goes down, so this type of routing is fault-tolerant. Also, unlike with static routing, there is no need to manually configure every route on every router, which greatly reduces the administrative overhead. You only need to define which routes will be advertised on a router that connect directly to the corresponding subnets – routing protocols take care of the rest.

The disadvantage of dynamic routing is that it increases memory and CPU usage on a router, because every router has to process received routing information and calculate its routing table.

To better understand the advantages that dynamic routing protocols bring, consider the following example:



Both routers are running a routing protocol, namely EIGRP. There is no static routes on Router A, so R1 doesn't know how to reach the subnet 10.0.0.0/24 that is directly connected to Router B. Router B then advertises the subnet to Router A using EIGRP. Now Router A has the route to reach the subnet. This can be verified by typing the show ip route command:



```
Router_A#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
D       10.0.0.0 [90/30720] via 192.168.0.2, 00:00:09, FastEthernet0/0
C    192.168.0.0/24 is directly connected, FastEthernet0/0
Router_A#
```

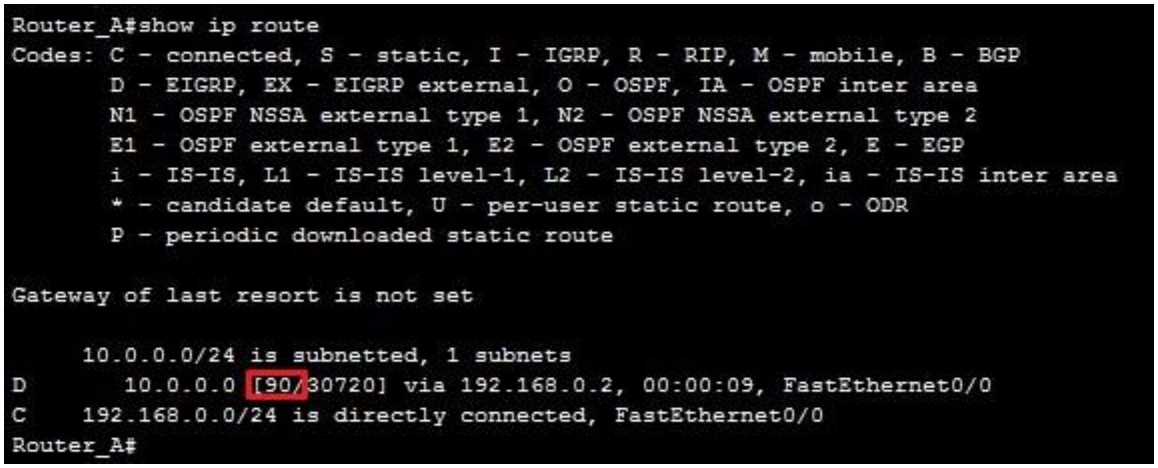
You can see that Router A has learned the subnet from EIGRP. The letter D in front of the route indicates that the route has been learned through EIGRP. If the subnet 10.0.0.0/24 fails, Router B can immediately inform Router A that the subnet is no longer reachable.

Administrative distance & metric

Administrative distance

A network can use more than one routing protocol, and routers on the network can learn about a route from multiple sources. Routers need to find a way to select a better path when there are multiple paths available. Administrative distance number is used by routers to find out which route is better (lower number is better). For example, if the same route is learned from RIP and EIGRP, a Cisco router will choose the EIGRP route and stores it in the routing table. This is because EIGRP routes have (by default) the administrative distance of 90, while RIP route have a higher administrative distance of 120.

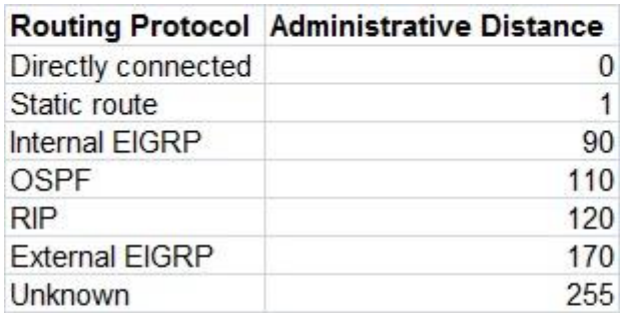
You can display the administrative distance of all routes on your router by typing the show ip route command:



```
Router_A#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
D       10.0.0.0 [90/30720] via 192.168.0.2, 00:00:09, FastEthernet0/0
C    192.168.0.0/24 is directly connected, FastEthernet0/0
Router_A#
```

In the case above, the router has only a single route in its routing table learned from a dynamic routing protocols – the EIGRP route.

The following table lists the administrative distance default values:

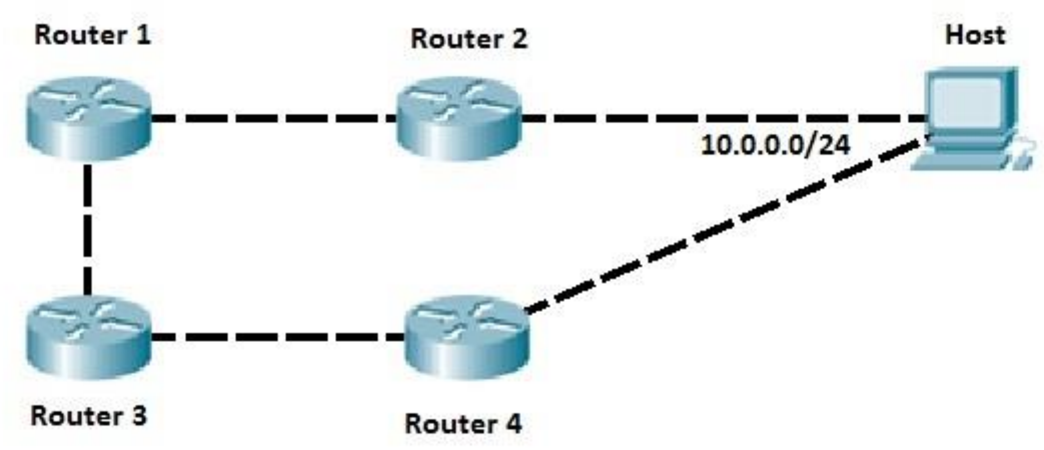


Routing Protocol	Administrative Distance
Directly connected	0
Static route	1
Internal EIGRP	90
OSPF	110
RIP	120
External EIGRP	170
Unknown	255

Metric

If a router learns two different paths for the same network from the same routing protocol, it has to decide which route is better and will be placed in the routing table. Metric is the measure used to decide which route is better (lower number is better). Each routing protocol uses its own metric. For example, RIP uses hop counts as a metric, while OSPF uses cost.

The following example explains the way RIP calculates its metric and why it chooses one path over another.

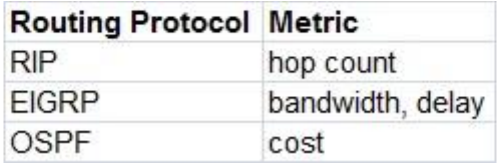


RIP has been configured on all routers. Router 1 has two paths to reach the subnet 10.0.0.0/24. One path is goes through Router 2, while the other path goes through Router 3 and then Router 4. Because RIP uses the hop count as its metric, the path through Router 1 will be used to reach the 10.0.0.0/24 subnet. This is because that subnet is only one router away on the path. The other path will have a higher metric of 2, because the subnet is two routers away.

NOTE

The example above can be used to illustrate a disadvantage of using RIP as a routing protocol. Imagine if the first path through R2 was the 56k modem link, while the other path (R3-R4) is a high speed WAN link. Router R1 would still chose the path through R2 as the best route, because RIP uses only the hop count as its metric.

The following table lists the parameters that various routing protocols use to calculate the metric:



Routing Protocol	Metric
RIP	hop count
EIGRP	bandwidth, delay
OSPF	cost

Routing protocols

Dynamic routes are routes learned via routing protocols. Routing protocols are configured on routers with the purpose of exchanging routing information. There are many benefits of using routing protocols in your network, such as:

- *unlike static routing, you don't need to manually configure every route on each router in the network. You just need to configure the networks to be advertised on a router directly connected to them.*
- *if a link fails and the network topology changes, routers can advertise that some routes have failed and pick a new route to that network.*

Types of routing protocols

There are two types of routing protocols:

1. Distance vector (RIP, IGRP)
2. Link state (OSPF, IS-IS)

Cisco has created its own routing protocol – EIGRP. EIGRP is considered to be an advanced distance vector protocol, although some materials erroneously state that EIGRP is a hybrid routing protocol, a combination of distance vector and link state.

All of the routing protocols mentioned above are interior routing protocols (IGP), which means that they are used to exchange routing information within one autonomous system. BGP (Border Gateway Protocol) is an example of an exterior routing protocol (EGP) which is used to exchange routing information between autonomous systems on the Internet.

Distance vector protocols

As the name implies, distance vector routing protocols use distance to determine the best path to a remote network. The distance is something like the number of hops (routers) to the destination network.

Distance vector protocols usually send the complete routing table to each neighbor (a neighbor is directly connected router that runs the same routing protocol). They employ some version of Bellman-Ford algorithm to calculate the best routes. Compared with link state routing protocols, distance vector protocols are easier to configure and require little management, but are susceptible to routing loops and converge slower than the link state routing protocols. Distance vector protocols also use more bandwidth because they send complete routing table, while the link state protocols send specific updates only when topology changes occur.

RIP and EIGRP are examples of distance vector routing protocols.

Link state protocols

Link state routing protocols are the second type of routing protocols. They have the same basic purpose as distance vector protocols, to find a best path to a destination, but use different methods to do so. Unlike distance vector protocols, link state protocols don't advertise the entire routing table. Instead, they advertise information about a network topology (directly connected links, neighboring routers…), so that in the end all routers running a link state protocol have the same topology database. Link state routing protocols converge much faster than distance vector routing protocols, support classless routing, *send updates using multicast addresses and use triggered routing updates. They also require more router CPU and memory usage than distance-vector routing protocols and can be harder to configure.*

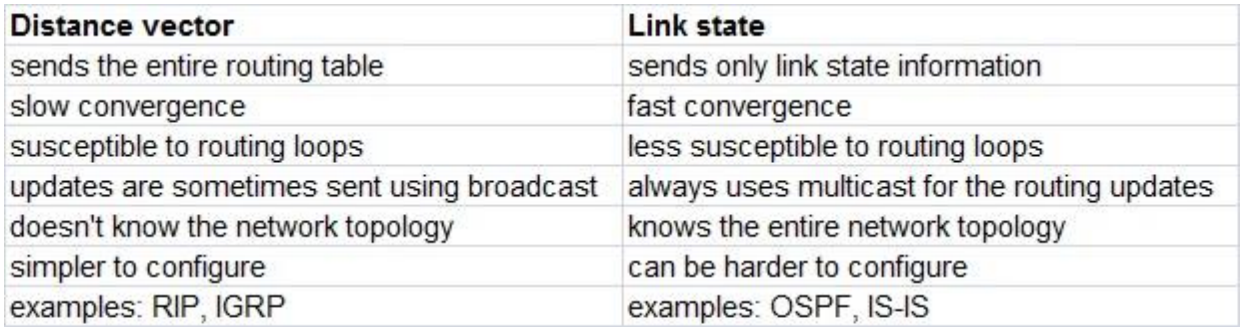
Each router running a link state routing protocol creates three different tables:

- *neighbor table – the table of neighboring routers running the same link state routing protocol.*
- *topology table – the table that stores the topology of the entire network.*
- *routing table – the table that stores the best routes.*

Shortest Path First algorithm is used to calculate the best route. OSPF and IS-IS are examples of link state routing protocols.

Difference between distance vector and link state routing protocols

The following table summarizes the differences:



Distance vector	**Link state**
sends the entire routing table	sends only link state information
slow convergence	fast convergence
susceptible to routing loops	less susceptible to routing loops
updates are sometimes sent using broadcast	always uses multicast for the routing updates
doesn't know the network topology	knows the entire network topology
simpler to configure	can be harder to configure
examples: RIP, IGRP	examples: OSPF, IS-IS

CHAPTER EIGHT
RIP OVERVIEW

We Will Cover These Topics in This Chapter

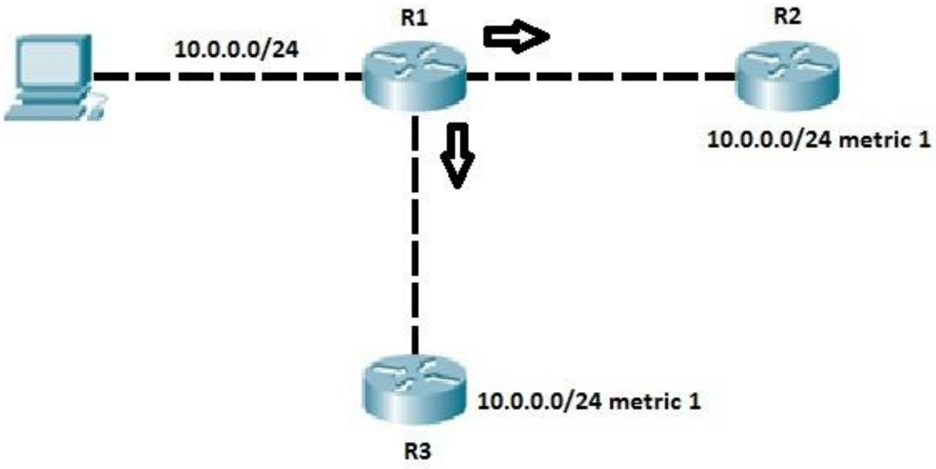
- ***RIP Overview***
- ***Configuration of RIP v2***
- ***Passive Interface Command***
- ***RIP loop Prevention***
- ***Advertise Default Route Using RIP***
- ***And, More Topics…***

RIP overview

RIP (Routing Information Protocol) is one of the oldest distance vector routing protocols. It is usually used on small networks because it is very simple to configure and maintain, but lacks some advanced features of routing protocols like OSPF or EIGRP. Two versions of the protocol exists: version 1 and version 2. Both versions use hop count as a metric and have the administrative distance of 120. RIP version 2 is capable of advertising subnet masks and uses multicast to send routing updates, while version 1 doesn't advertise subnet masks and uses broadcast for updates. Version 2 is backwards compatible with version 1.

RIPv2 sends the entire routing table every 30 seconds, which can consume a lot of bandwidth. RIPv2 uses multicast address of 224.0.0.9 to send routing updates, supports authentication and triggered updates (updates that are sent when a change in the network occurs).

For example of how RIP works, consider the following figure.



Router R1 directly connects to the subnet 10.0.0.0/24. Network engineer has configured RIP on R1 to advertise the route to this subnet. R1 sends routing updates to R2 and R3. The routing updates list the subnet, subnet mask and metric for this route. Each router, R2 and R3, receives this update and adds the route to their respective routing tables. Both routers list the metric of 1 because the network is only one hop away.

NOTE

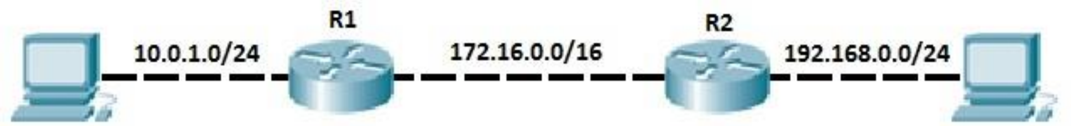
Maximum hop count for a RIP route is 15. Any route with a higher hop count is considered to be unreachable.

Configuring RIPv2

Configuring RIPv2 is a pretty straightforward process. Only three steps are required:

1. enabling RIP by using the router rip global configuration command
2. instructing the router to use RIPv2 by typing the version 2 command
3. telling RIP which networks to advertise by using one or more network commands.

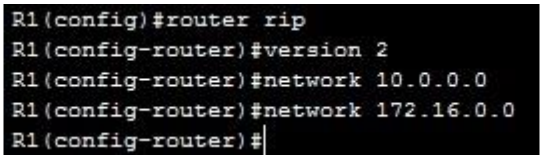
The first two commands are easy to comprehend, but the last command requires a little bit more thought. With the network command you specify which interfaces will participate in the routing process. This command takes a classful network as a parameter and enables RIP on the corresponding interfaces. Let's configure our sample network to use RIP.



Router R1 and R2 have directly connected subnets. We want to include these subnets in the RIP routing process. To do that, we first need to enable RIP on both routers and then advertise these subnets using the network command.

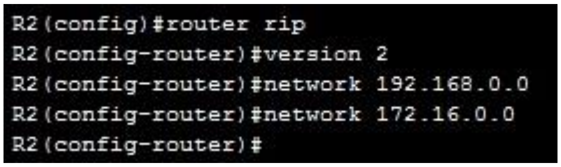
On router R1, in the global configuration mode, enter the router rip command to enable RIP. In the RIP configuration mode, change the version of the protocol to 2 by using the version 2 command. Next, use the network 10.0.0.0 command to include the Fa0/1 interface on the router R1 in the routing process. Remember, the network command takes a classful network number as a parameter, so in this case every interface that has an IP address that begins with 10 will be included in the RIP process (IP addresses that begins with 10 are, by default, the class A addresses and have the default subnet mask of 255.0.0.0). For instance, if another interface on the router had the IP address of 10.1.0.1 it would also be included in the routing process with the network command. You also need to include the link between the two routers in the RIP routing process. This is done by adding another network statement, network 172.16.0.0.

So, the configuration on R1 should look like this:



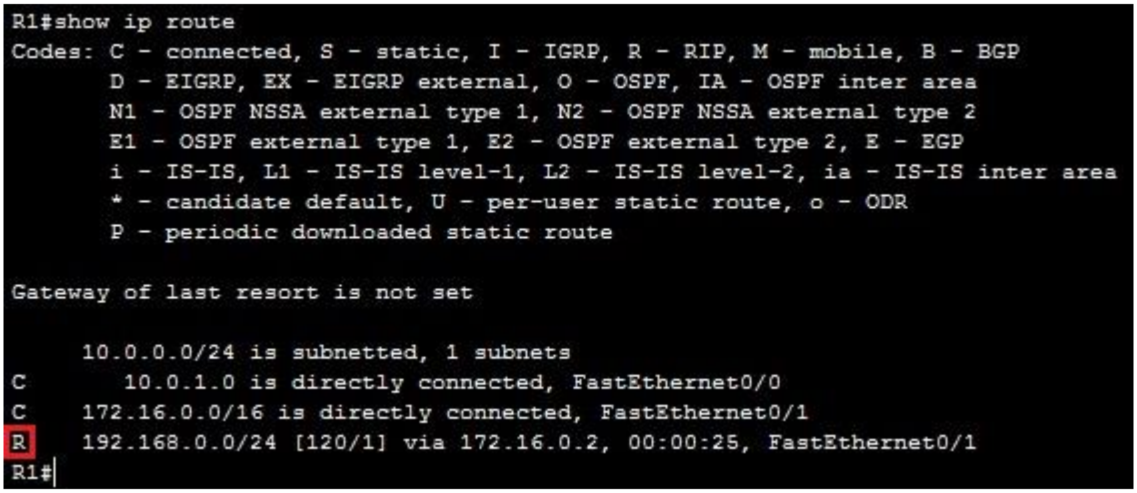
```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 10.0.0.0
R1(config-router)#network 172.16.0.0
R1(config-router)#
```

The configuration on R2 looks similar, but with different network number for the directly connected subnet:



```
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 192.168.0.0
R2(config-router)#network 172.16.0.0
R2(config-router)#
```

You can verify that router R1 have a route to the R2's directly connected subnet by typing the show ip route command:

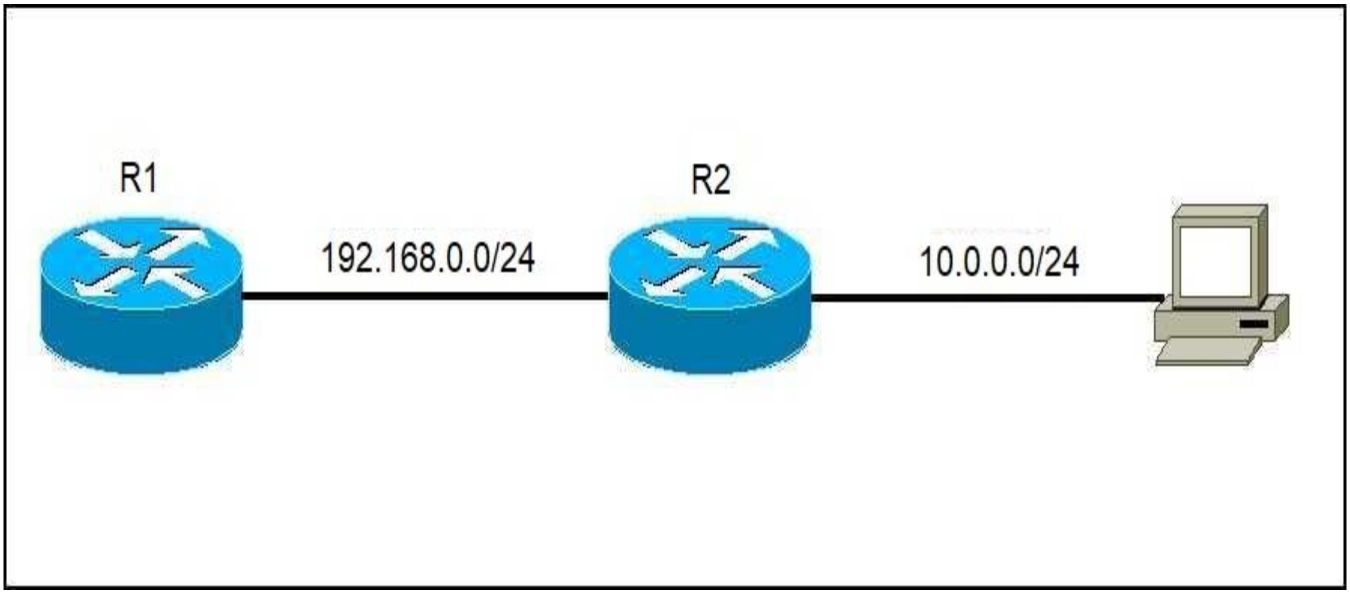


```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.1.0 is directly connected, FastEthernet0/0
C    172.16.0.0/16 is directly connected, FastEthernet0/1
R    192.168.0.0/24 [120/1] via 172.16.0.2, 00:00:25, FastEthernet0/1
R1#
```

The legend lists R for all RIP routes in the routing table. Also note that the administrative distance of 120 is shown, together with the metric of 1.

passive-interface command

Consider the following example network with RIP turned on:



The RIP configuration on R2 looks like this:

```
router rip

 version 2

 network 10.0.0.0
network 192.168.0.0
```

As we've already mentioned, the network command does two things:

- *advertises the defined network in RIP.*
- *activates RIP on the interfaces whose addresses fall within the specified classful networks.*

So in the example network above, RIP will also be activated on the interface connected to the workstation on the right. This means that the workstation will also receive RIP updates, which is pointless. To prevent this from happening, the passive interface command is used:

```
R2(config)#router rip

R2(config-router) #passive-interface Gi0/1
```

Now, the RIP process will no longer send RIP updates out the Gi0/1 interface. However, all received RIP updates will be processed and the subnet 10.0.0.0/24 will still be advertised.

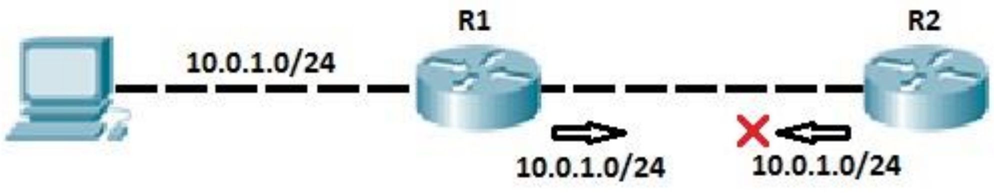
RIP loop prevention

Distance vector protocols are susceptible to routing loops. Routing loops occur when a packet is continually routed through the same routers over and over again, in an endless circle. Because they can render a network unusable, distance vector routing protocols (such as RIP and EIGRP) employ several different mechanisms to prevent routing loops. We will describe them in this article.

Split Horizon

Split horizon is one of the features of distance vector routing protocols that prevents routing loops. This feature prevents a router from advertising a route back onto the interface from which it was learned.

Consider the following network topology:

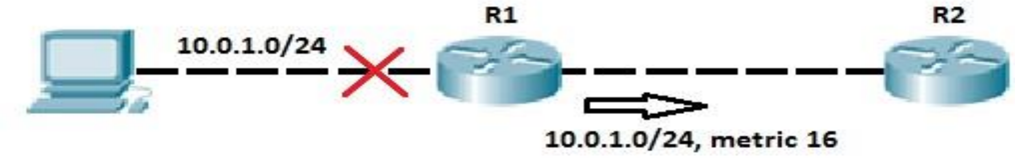


Router R1 has a route to the subnet 10.0.1.0/24 that is advertised to router R2 by using RIP. Router R2 receives the update and stores the route in its routing table. Router R2 knows that the routing update for that route has come from R1, so it won't advertise the route back to router R1. Otherwise, if the network 10.0.1.0/24 goes down, router R1 could receive a route to the subnet 10.0.1.0/24 from R2. Router R1 would think that R2 has the route to reach the subnet, and would send packets designated for the 10.0.1.0/24 to R2. R2 would receive the packets from R1 and *sends them back to R1, because R2 thinks that R1 has a route to reach the subnet, thereby creating a routing loop.*

Route poisoning

Route poisoning is another method for preventing routing loops employed by distance vector routing protocols. When a router detects that one of its directly connected routes has failed, it sends the advertisement for that route with an infinite metric (poisoning the route). A router that receives the update knows that the route has failed and doesn't use it anymore.

Consider the following example:

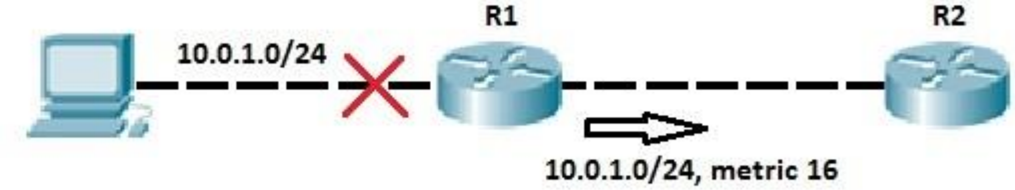


Router R1 is directly connected to the 10.0.1.0/24 subnet. Router R1 runs RIP and the subnet is advertised to R2. When the R1's Fa0/1 interface fails, the route advertisement is sent by R1 to R2 indicating that the route has failed. The route has a metric of 16, which is more than the RIP's maximum hop count of 15, so R1 considers the route to be unreachable.

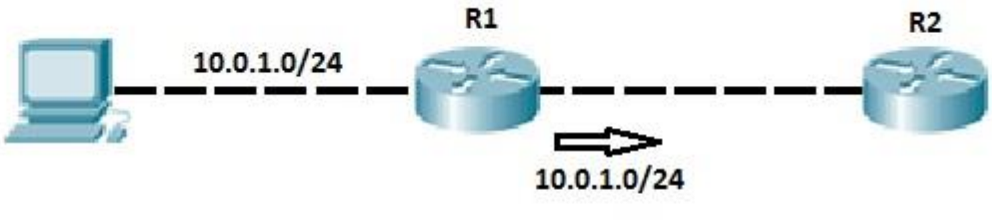
Holddown timer

Holddown is another loop-prevention mechanism employed by distance vector routing protocol. This feature prevents a router from learning new information about a failed route. When a router receives the information about the unreachable route, the holddown timer is started. The router ignores all routing updates for that route until the timer expires (by default, 180 seconds in RIP). Only updates allowed during that period are updates sent from the router that originally advertised the route. If that router advertise the update, the holddown timer is stopped and the routing information is processed.

An example will help you understand the concept better. Consider the following network topology.

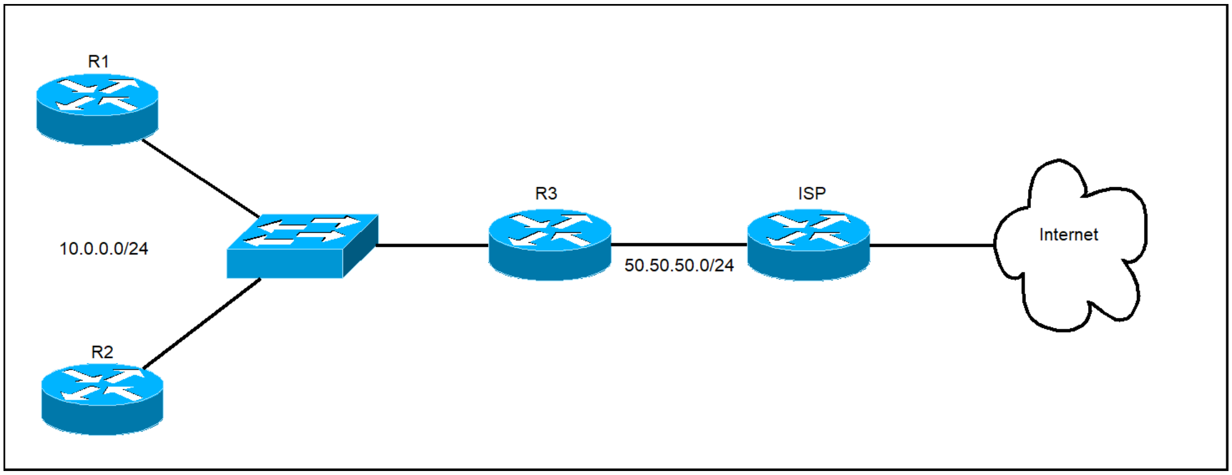


Router R1 has advertised its directly connected subnet 10.0.1.0/24 through RIP. After some period of time, the interface Fa0/1 on R1 fails and the router R1 sends the poisoned route to R2. R2 receives the routing update, marks the route as unreachable and starts the holddown timer. During that time all updates from any other routers about that route are ignored to prevent routing loops. If interface Fa0/1 on R1 comes back up, R1 again advertises the route. R2 process that update even if the holddown timer is still running, because the update is sent by the same router that originally advertised the route.



Advertise default routes using RIP

Consider the following example network:



In the network above we have three routers running RIP. Router R3 is connected to the ISP's internet router and and has a static default route that points to it. It is possible to advertise that default route using RIP to other routers in the local network. On R3, we simply need to enter the default-information originate command in the RIP configuration mode.

Here is the configuration on R3:

```
R3(config)#ip route 0.0.0.0 0.0.0.0 50.50.50.1

R3(config)#router rip

R3(config-router)#default-information originate
```

R1 and R2 don’t need any additional configuration – they learn the default route just like any other RIP route:

```
R1#show ip route rip

R*   0.0.0.0/0 [120/1] via 10.0.0.1, 00:00:04, GigabitEthernet0/0
```

As you can see from the output above, R1 learned about the default route via RIP. The route is marked with an asterisk (*), indicating that the route is a candidate to be the default route.

CHAPTER NINE
EIGRP OVERVIEW

We Will Cover These Topics in This Chapter

- ***EIGRP overview***
- ***EIGRP Configuration***
- ***EIGRP automatic & manual summarization***
- ***EIGRP authentication & load balancing***
- ***EIGRP Reliable Transport Protocol (RTP)***
- ***EIGRP Diffusing Update Algorithm (DUAL)***
- ***EIGRP summary***

EIGRP overview

EIGRP (Enhanced Interior Gateway Routing Protocol) is an advanced distance vector routing protocol. This protocol is an evolution of an earlier Cisco protocol called IGRP, which is now considered obsolete. EIGRP supports classless routing and VLSM, route summarization, incremental updates, load balancing and many other useful features. It is a Cisco proprietary protocol, so all routers in a network that is running EIGRP must be Cisco routers.

Routers running EIGRP must become neighbors before exchanging routing information. To dynamically discover neighbors, EIGRP routers use the multicast address of 224.0.0.10. Each EIGRP router stores routing and topology information in three tables:

- Neighbor table – stores information about EIGRP neighbors
- Topology table – stores routing information learned from neighboring routers
- Routing table – stores the best routes

Administrative distance of EIGRP is 90, which is less than both the administrative distance of RIP and the administrative distance of OSPF, so EIGRP routes will be preferred over these routes. EIGRP uses Reliable Transport Protocol (RTP) for sending messages.

EIGRP calculates its metric by using bandwidth, delay, reliability and load. By default, only bandwidth and delay are used when calculating metric, while reliability and load are set to zero.

EIGPR uses the concept of autonomous systems. An autonomous system is a set of EIGRP enabled routers that should become EIGRP neighbors. Each router inside an autonomous system must have the same autonomous system number configured, otherwise routers will not become neighbors.

EIGRP Neighbors

EIGRP must establish neighbor relationships with other EIGRP neighboring routers before exchanging routing information. To establish a neighbor relationships, routers send hello packets every couple of seconds. Hello packets are sent to the multicast address of 224.0.0.10.

NOTE

On LAN interfaces hellos are sent every 5 seconds. On WAN interfaces every 60 seconds.

The following fields in a hello packet must be the identical in order for routers to become neighbors:

- ASN (autonomous system number)
- subnet number
- K values (components of metric)

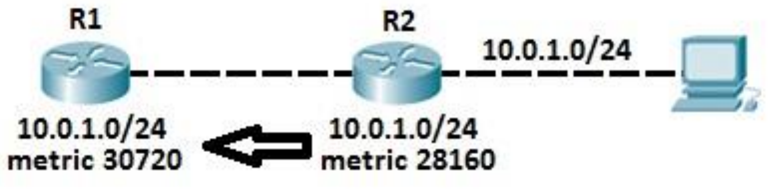
Routers send hello packets every couple of seconds to ensure that the neighbor relationship is still active. By default, routers considers the neighbor to be down after a hold-down timer has expired. Hold-down timer is, by default, three times the hello interval. On LAN network the hold-down timer is 15 seconds.

Feasible and reported distance

Two terms that you will often encounter when working with EIGRP are feasible and reported distance. Let's clarify these terms:

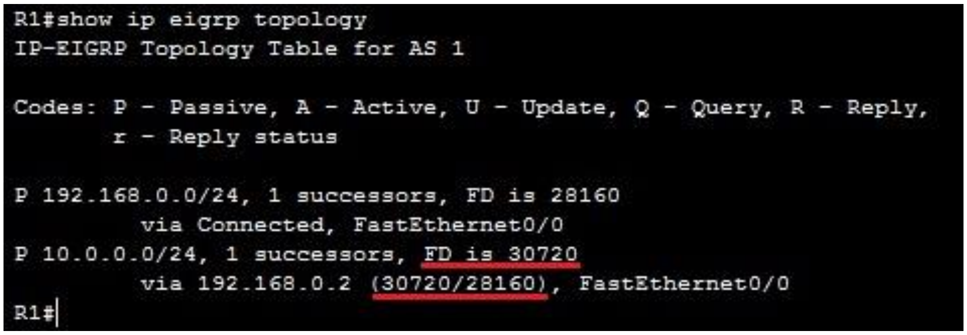
- *Feasible distance (FD) – the metric of the best route to reach a network. That route will be listed in the routing table.*
- *Reported distance (RD) – the metric advertised by a neighboring router for a specific route. It other words, it is the metric of the route used by the neighboring router to reach the network.*

To better understand the concept, consider the following example.



EIGRP has been configured on R1 and R2. R2 is directly connected to the subnet 10.0.1.0/24 and advertises that subnet into EIGRP. Let's say that R2's metric to reach that subnet is 28160. When the subnet is advertised to R1, R2 informs R1 that its metric to reach 10.0.1.0/24 is 10. From the R1's perspective that metric is considered to be the reported distance for that route. R1 receives the update and adds the metric to the neighbor to the reported distance. That metric is called the feasible distance and is stored in R1's routing table (30720 in our case).

The feasible and reported distance are displayed in R1's EIGRP topology table:



```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS 1

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 192.168.0.0/24, 1 successors, FD is 28160
         via Connected, FastEthernet0/0
P 10.0.0.0/24, 1 successors, FD is 30720
         via 192.168.0.2 (30720/28160), FastEthernet0/0
R1#
```

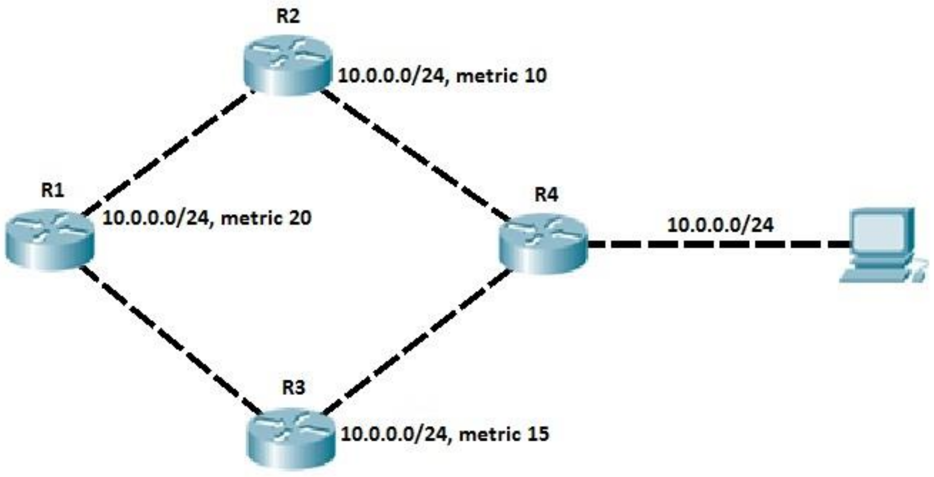
Successor and feasible successor

Another two terms that appear often in the EIGRP world are successor and feasible successor. A successor is the route with the best metric to reach a destination. That route is stored in the routing table. A feasible successor is a backup path to reach that same destination that can be used immediately if the successor route fails. These backup routes are stored in the topology table.

For a route to be chosen as a feasible successor, one condition must be met:

- *the neighbor's advertised distance (AD) for the route must be less than the successor's feasible distance (FD).*

The following example explains the concept of a successor and a feasible successor.



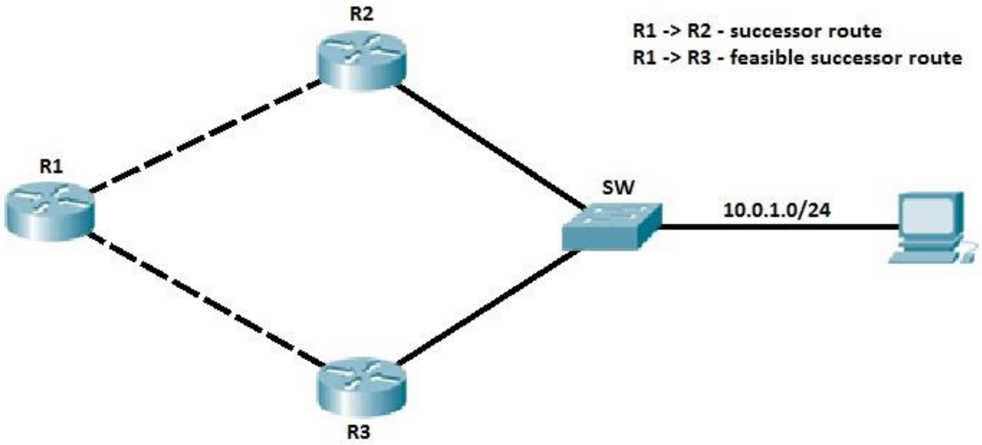
R1 has two paths to reach the subnet 10.0.0.0/24. The path through R2 has the best metric (20) and it is stored in the R1's routing table. The other route, through R3, is a feasible successor route, because the feasibility condition has been met (R3's advertised distance of 15 is less than R1's feasible distance of 20). R1 stores that route in the topology table. This route can be immediately used if the primary route fails.

EIGRP topology table

EIGRP topology table contains all learned routes to a destination. The table holds all routes received from a neighbor, successors and feasible successors for every route, and interfaces on which updates were received. The table also holds all locally connected subnets included in an EIGRP process.

Best routes (the successors) from the topology table are stored in the routing table. Feasible successors are only stored in the topology table and can be used immediately if the primary route fails.

Consider the following network topology.


R2
R1 -> R2 - successor route
R1 -> R3 - feasible successor route
R1
SW
10.0.1.0/24
R3

EIGRP is running on all three routers. Routers R2 and R3 both connect to the subnet 10.0.1.0/24 and advertise that subnet to R1. R1 receives both updates and calculates the best route. The best path goes through R2, so R1 stores that route in the routing table. Router R1 also calculates the metric of the route through R3. Let's say that advertised distance of that route is less then feasible distance of the best route. The feasibility condition is met and router R1 stores that route in the topology table as a feasible successor route. The route can be used immediately if the primary route fails.

EIGRP configuration

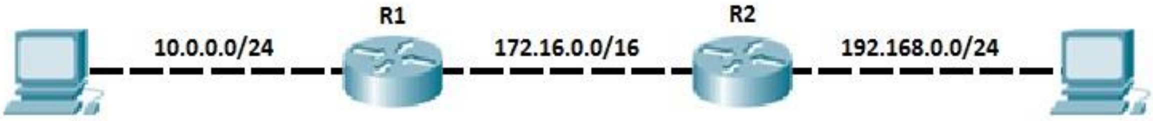
Configuring EIGRP 1

EIGRP configuration closely resembles RIP configuration. Only two steps are required:

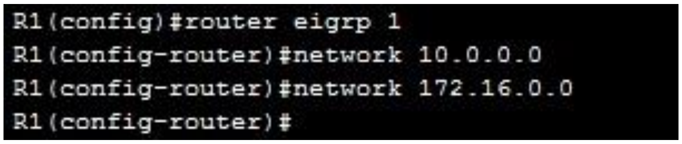
- *enabling EIGRP by using the router eigrp ASN_NUMBER command*
- *telling EIGRP which networks to advertise by using one or more network statements*

The first command, router eigrp ASN_NUMBER, enables EIGRP on a router. ASN_NUMBER represents an autonomous system number and has to be the same on all routers running EIGRP, otherwise routers won't become neighbors. The second command, network SUBNET, enables EIGRP on selected interfaces and specifies which networks will be advertised. By default, the network command takes a classful network number as the parameter.

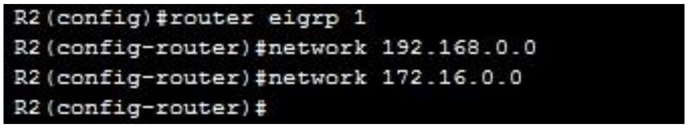
To illustrate a configuration of EIGRP, we will use the following network:



The network depicted above consists of only two routers. Each router has a directly connected subnet that needs to be advertised through EIGRP. The following figure show the EIGRP configuration on R1 and R2:

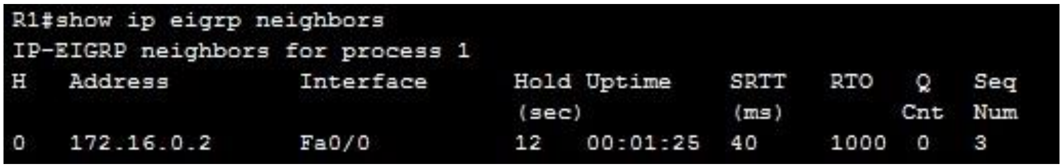


```
R1(config)#router eigrp 1
R1(config-router)#network 10.0.0.0
R1(config-router)#network 172.16.0.0
R1(config-router)#
```



```
R2(config)#router eigrp 1
R2(config-router)#network 192.168.0.0
R2(config-router)#network 172.16.0.0
R2(config-router)#
```

You can verify that routers have become neighbors by using the show ip eigrp neighbors command on either router:

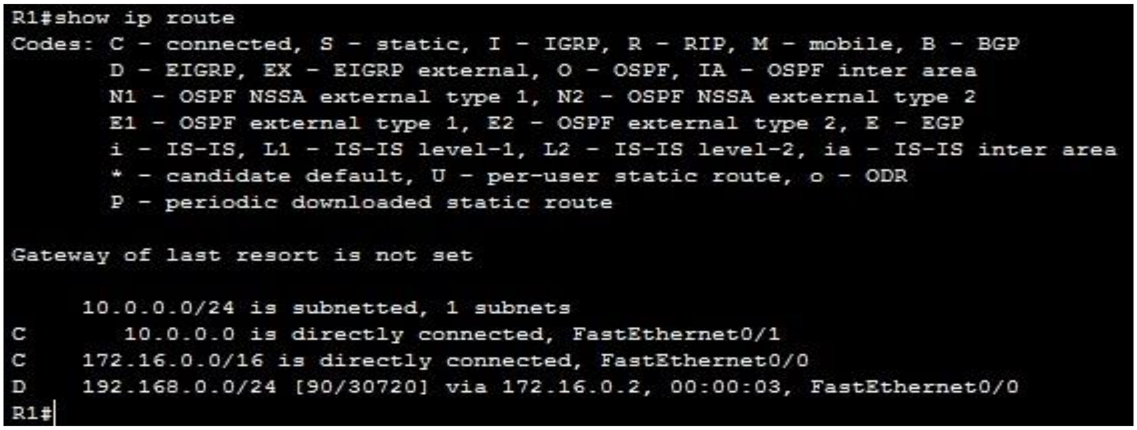


```
R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address         Interface       Hold Uptime    SRTT   RTO   Q   Seq
                                    (sec)          (ms)         Cnt Num
0   172.16.0.2      Fa0/0           12   00:01:25  40     1000  0   3
```

The command above lists all EIGRP neighbors. The address field lists the neighboring router RID (router ID). The interface field shows on which local interface the neighbor relationship has been formed.

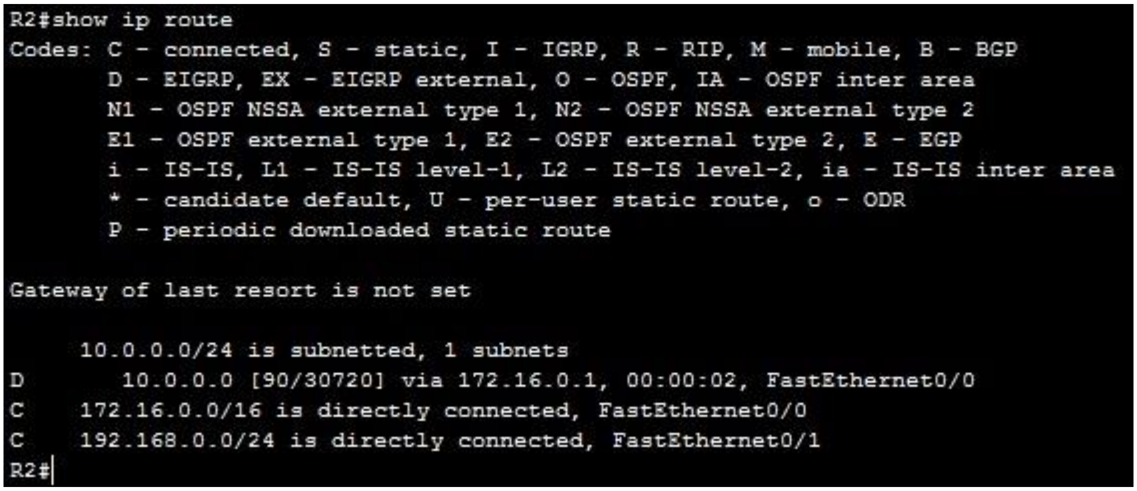
You can verify that routes are indeed being exchanged by using the show ip route command on both routers:

R1:



```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, FastEthernet0/1
C    172.16.0.0/16 is directly connected, FastEthernet0/0
D    192.168.0.0/24 [90/30720] via 172.16.0.2, 00:00:03, FastEthernet0/0
R1#
```

R2:



```
R2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
D       10.0.0.0 [90/30720] via 172.16.0.1, 00:00:02, FastEthernet0/0
C    172.16.0.0/16 is directly connected, FastEthernet0/0
C    192.168.0.0/24 is directly connected, FastEthernet0/1
R2#
```

NOTE

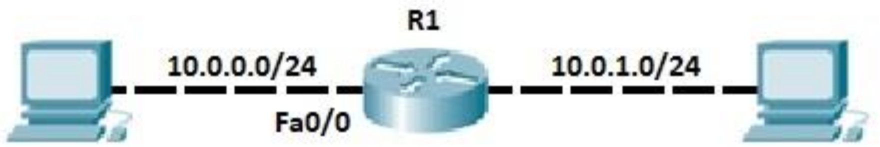
The D character at the beginning of a line in a routing table indicates that the route has been learned via EIGRP.

Configuring EIGRP 2

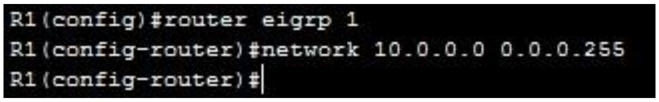
By default, the network command uses a classful network as the parameter. All interfaces inside that classful network will participate in the EIGRP process.To enable EIGRP only on specific interfaces, a wildcard mask can be used. The syntax of the command is:

(router-eigrp) network WILDCARD_MASK

Consider the following example.

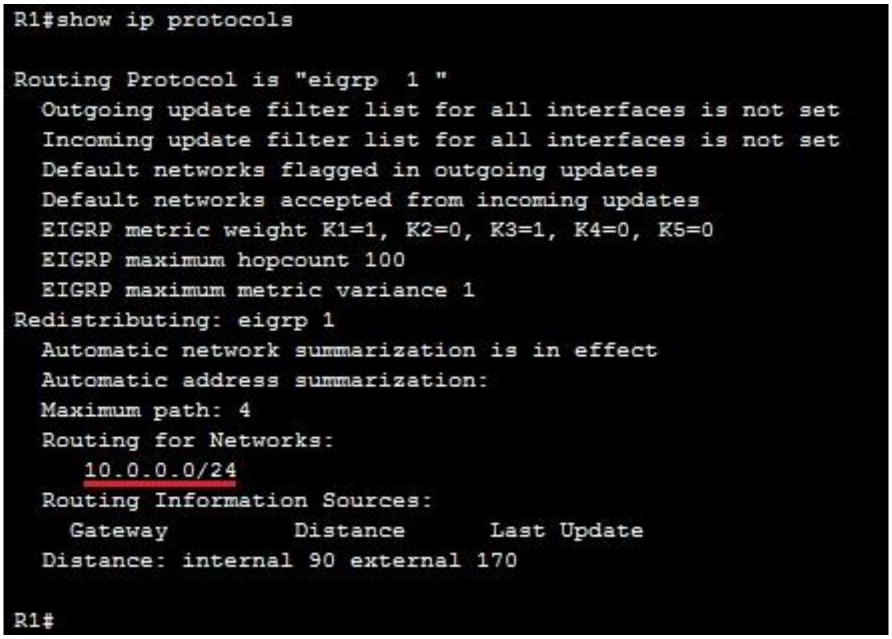


Router R1 has two directly connected subnets, 10.0.0.0/24 and 10.0.1.0/24. We want to enable EIGRP only on the subnet connected to the interface Fa0/0. If we enter the network 10.0.0.0 command under the EIGRP configuration mode, both subnets will be included in EIGRP process because we've used a classful network number in the network command. To configure EIGRP only on interface Fa0/0, the network 10.0.0.0 0.0.0.255 command can be used. This will enable EIGRP only on interfaces starting with 10.0.0.X.



```
R1(config)#router eigrp 1
R1(config-router)#network 10.0.0.0 0.0.0.255
R1(config-router)#
```

By using the command show ip protocols, you can verify that only the network 10.0.0.0/24 is included in EIGRP:



```
R1#show ip protocols

Routing Protocol is "eigrp  1 "
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
Redistributing: eigrp 1
  Automatic network summarization is in effect
  Automatic address summarization:
  Maximum path: 4
  Routing for Networks:
     10.0.0.0/24
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: internal 90 external 170

R1#
```

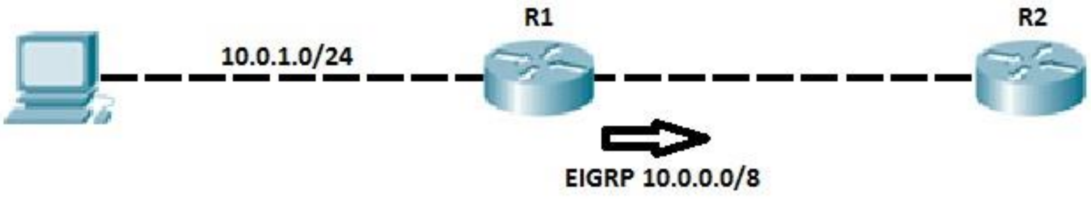
EIGRP automatic & manual summarization

Route summarization is a method of representing multiple networks with a single summary address. It is often use in large networks with many subnets because it reduces the number of routes that a router must maintain and minimizes the traffic used for routing updates. Two methods for summarizing routes exist: automatic summarization and manual summarization.

EIGRP automatic summarization

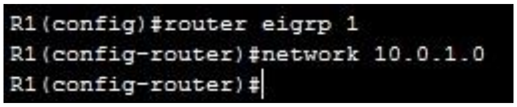
By default, EIGRP has the auto summary feature enabled. Because of this, routes are summarized to classful address at network boundaries in the routing updates.

To better understand the concept of auto-summarization, consider the following example.



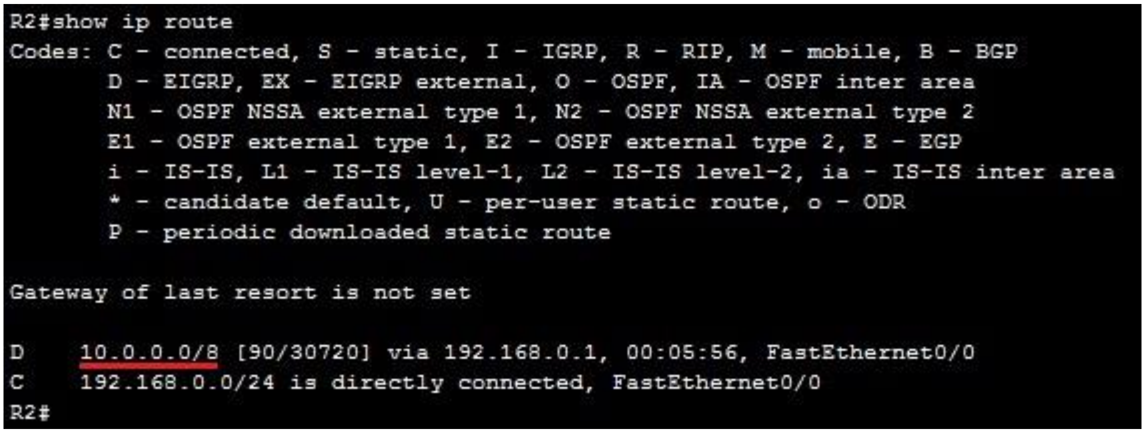
Router R1 and R2 are running EIGRP. Router R1 has the locally connected subnet 10.0.1.0/24 that is advertised to the router R2. Because of the auto summary feature, the router R1 *summarizes the network 10.0.1.0/24 before sending the route to R2. With the auto summary feature turned on, R1 sends the classful route 10.0.0.0/8 to R2 instead of the more specific 10.0.1.0/24 route.*

On R1, we have configured the following network statement:



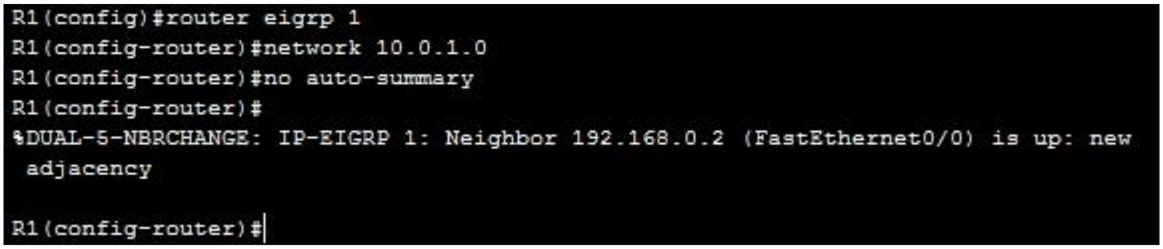
```
R1(config)#router eigrp 1
R1(config-router)#network 10.0.1.0
R1(config-router)#
```

But, because of the auto-summary feature, R2 receives the route to the classful network 10.0.0.0/8:



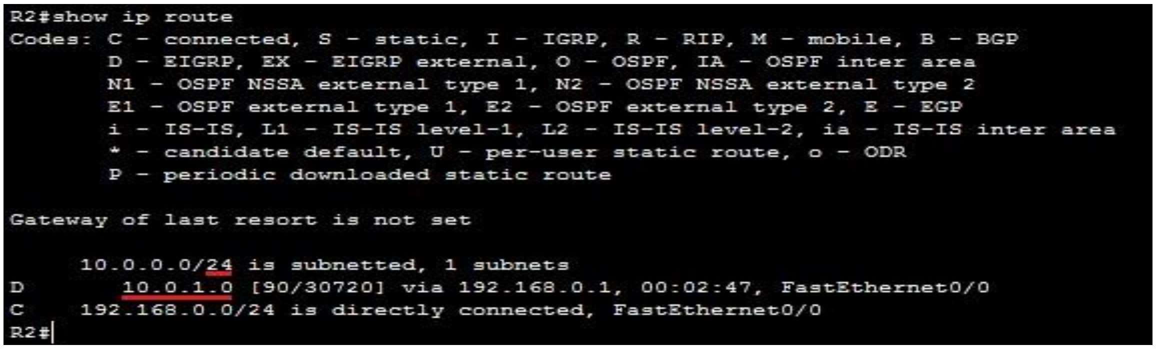
```
R2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

D    10.0.0.0/8 [90/30720] via 192.168.0.1, 00:05:56, FastEthernet0/0
C    192.168.0.0/24 is directly connected, FastEthernet0/0
R2#
```

The auto summary feature can cause problems with discontinuous networks. This is why this feature is usually turned off. This is done by using the no auto-summary command:



```
R1(config)#router eigrp 1
R1(config-router)#network 10.0.1.0
R1(config-router)#no auto-summary
R1(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.0.2 (FastEthernet0/0) is up: new
 adjacency

R1(config-router)#
```

Now R2 has the classless route to reach the subnet 10.0.1.0/24:



```
R2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
D       10.0.1.0 [90/30720] via 192.168.0.1, 00:02:47, FastEthernet0/0
C    192.168.0.0/24 is directly connected, FastEthernet0/0
R2#
```

NOTE

After typing the no auto-summary command, the neighbor relationship will be re-established.

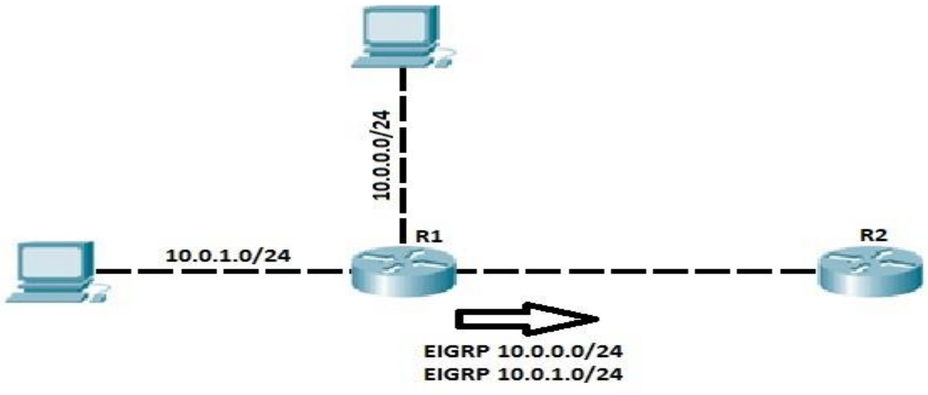
EIGRP manual summarization

One of the advantages of EIGRP over some other routing protocols (like OSPF) is that manual summarization can be done on any router within a network. A single route can be used to represent multiple routes, which reduces the size of routing tables in a network.

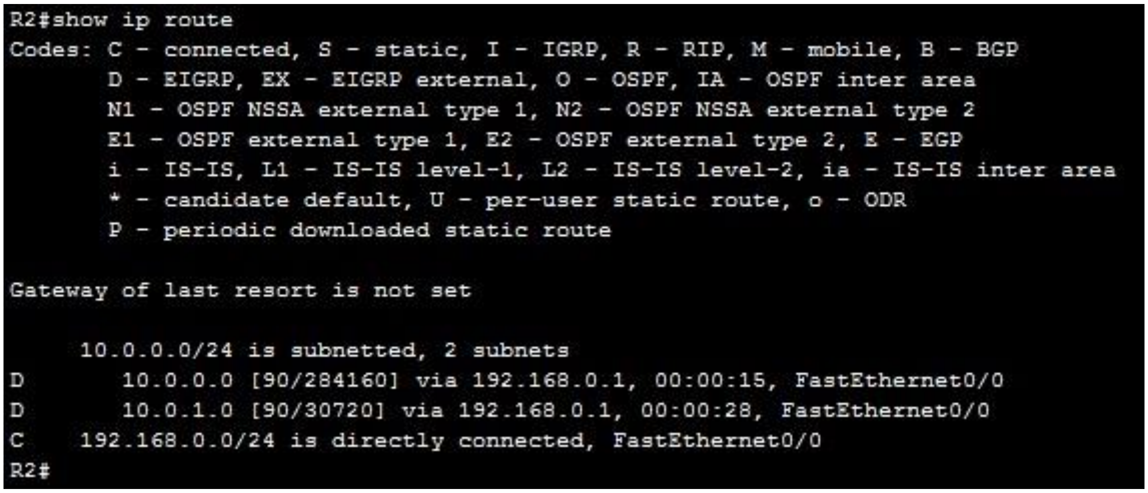
Manual summarization is configured on a per-interface basis. The syntax of the command is:

```
(config-if) ip summary-address eigrp ASN SUMMARY_ADDRESS SUBNET_MASK
```

An example will help you to understand the concept of manual summarization:

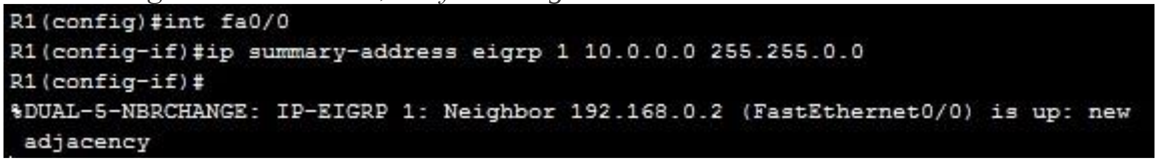


Router R1 and R2 are running EIGRP. Router R1 (on the left) has two directly connected subnets: 10.0.0.0/24 and 10.0.1.0/24. EIGRP advertises these subnets as two separate routes. R2 now has two routes for two subnets, which can be confirmed by using the show ip route command on R2:



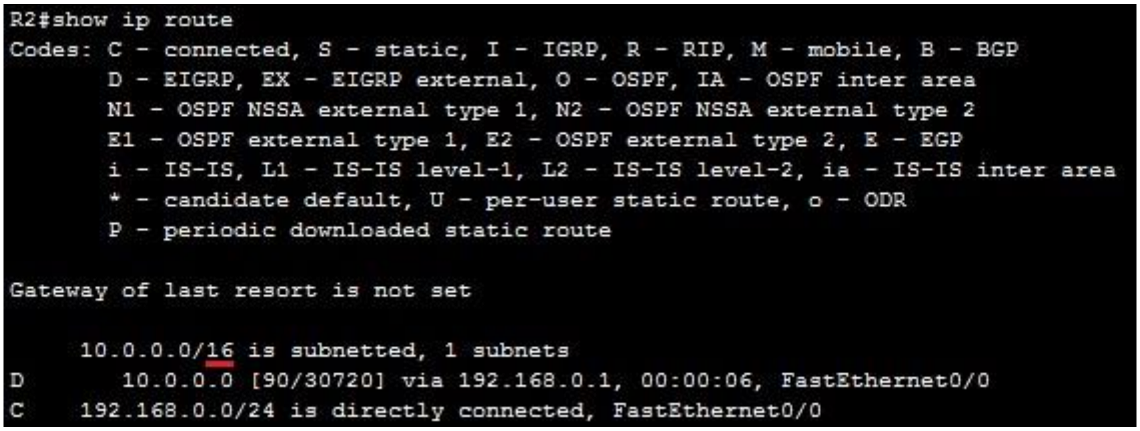
```
R2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 2 subnets
D       10.0.0.0 [90/284160] via 192.168.0.1, 00:00:15, FastEthernet0/0
D       10.0.1.0 [90/30720] via 192.168.0.1, 00:00:28, FastEthernet0/0
C    192.168.0.0/24 is directly connected, FastEthernet0/0
R2#
```

We could configure R1 to advertise only one summary route for both subnets, which helps reduce R2's routing table. To do this, the following command can be used:



```
R1(config)#int fa0/0
R1(config-if)#ip summary-address eigrp 1 10.0.0.0 255.255.0.0
R1(config-if)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.0.2 (FastEthernet0/0) is up: new
 adjacency
```

Now, R1 is sending only one route to reach both subnets to R2. We can verify that by using the show ip route command on R2:



```
R2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/16 is subnetted, 1 subnets
D       10.0.0.0 [90/30720] via 192.168.0.1, 00:00:06, FastEthernet0/0
C    192.168.0.0/24 is directly connected, FastEthernet0/0
```

Now R2 has only one route to reach both subnets on R1.

NOTE

In the example above, the ip summary command included two subnets on R1, but also some other addresses that are not in these subnets. The range of the summarized addresses is 10.0.0.0 – 10.0.255.255, so R2 thinks that R1 has the routes for all addresses inside that range. That could cause some problems if these addresses exist somewhere else in the network.

EIGRP authentication & load balancing

EIGRP authentication

EIGRP authentication is used to prevent an attacker from forming the EIGRP neighbor relationship with your router and advertising incorrect routing information. By using the same preshared key (PSK) on all routers you can force EIGRP to authenticate each EIGRP message. That way you can ensure that your router accepts routing updates only from the trusted sources. To authenticate every message, the MD5 (Message Digest 5) algorithm is used.

Three steps are required to configure EIGRP authentication:

- *1. creating a keychain*
- *2. specifying a key string for a key*
- *3. configuring EIGRP to use authentication*

EIGRP uses the concept of key chains. Each key chain can have many keys, just like in real life. You can specify a different lifetime interval of each key. That way the second key in a key chain can be used after the first one is expired, the third one after the second and so on. After you have created a key chain with the corresponding keys, you need to enable EIGRP authentication for a particular interface.

To configure a router to use EIGRP configuration the following commands are used:

1. (global-config) key chain NAME – creates a keychain
2. (config-keychain) key NUMBER – identifies the key number
3. (config-keychain-key) key-string STRING – specifies the key string for the key

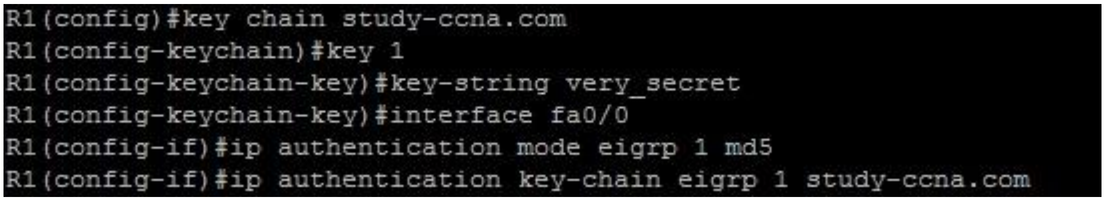
Next, we need to enable EIGRP authentication on an interface. From the interface mode, the following commands are used:

4. (config-if) ip authentication mode eigrp ASN md5 – enables EIGRP authentication on the interface
5. (config-if) ip authentication key-chain eigrp ASN KEY_CHAIN_NAME – specifies the name of the key chain that will be used for authentication

NOTE

For the authentication to work, the key number and the key string have to match on both routers! The key chain name doesn't have to be the same on both routers.

The following example shows how EIGRP authentication is configured:



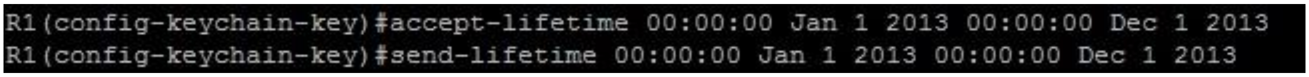
```
R1(config)#key chain study-ccna.com
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string very_secret
R1(config-keychain-key)#interface fa0/0
R1(config-if)#ip authentication mode eigrp 1 md5
R1(config-if)#ip authentication key-chain eigrp 1 study-ccna.com
```

To establish a time frame for the validity of a key, you need to configure the accept-lifetime and the send-lifetime parameters. The syntax of the commands is:

```
(config-keychain-key) accept-lifetime start_time {infinite | end_time | duration seconds}

(config-keychain-key) send-lifetime start_time {infinite | end_time | duration seconds}
```

The first command specifies the time period during which the key will be accepted. The second command specifies the time period during which the key will be sent.

For example, if we want to use a key only from January 1st, 2013 to December 1st, 2013, the following commands are used:

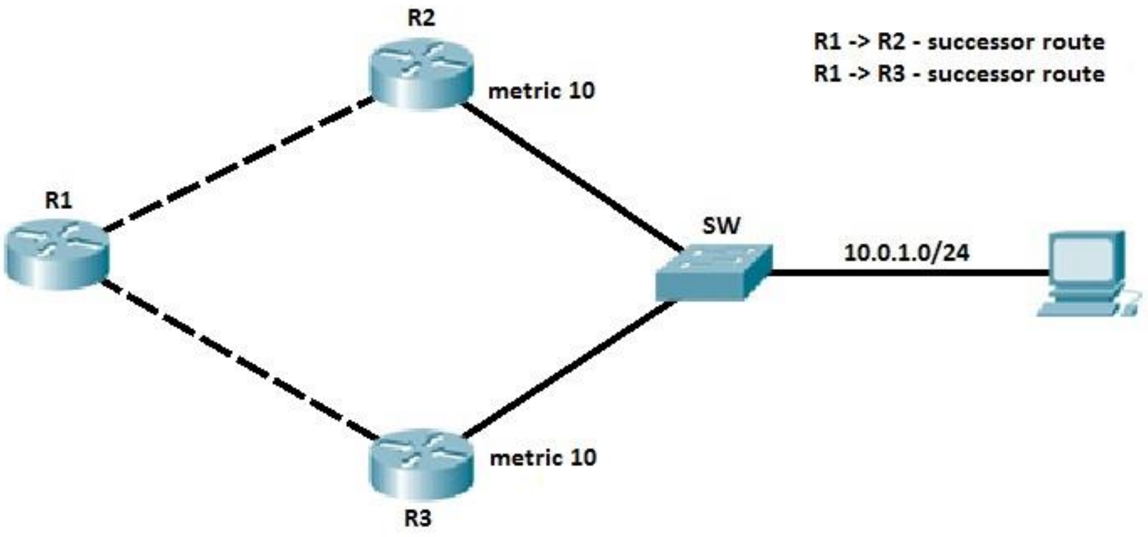


```
R1(config-keychain-key)#accept-lifetime 00:00:00 Jan 1 2013 00:00:00 Dec 1 2013
R1(config-keychain-key)#send-lifetime 00:00:00 Jan 1 2013 00:00:00 Dec 1 2013
```

EIGRP load balancing

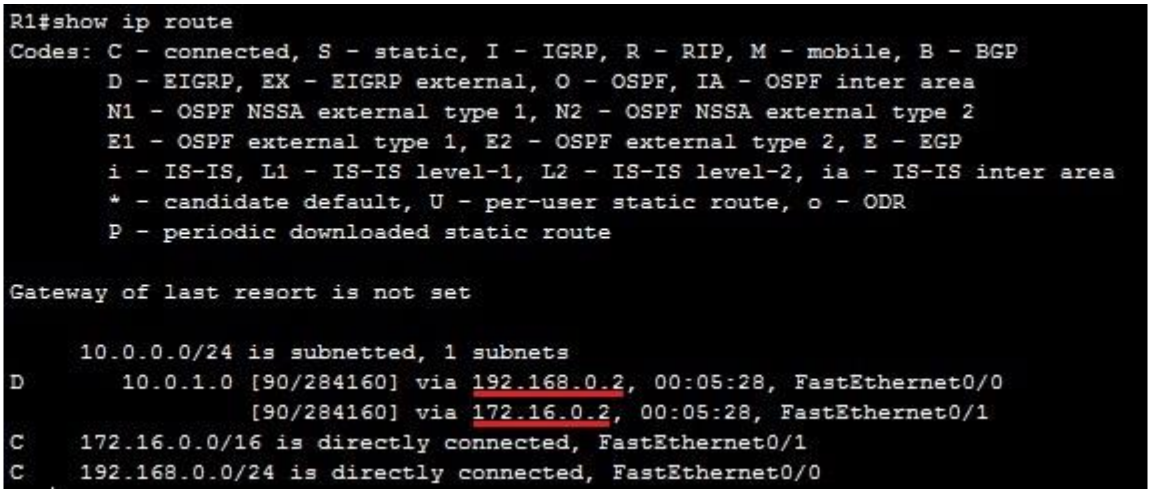
By default, EIGRP supports equal-cost load balancing over four links. Equal-cost means that multiple routes must have the same metric to reach a destination, so that router can choose to load balance across equal cost links.

To better understand the equal-cost load balancing concept, consider the following example.



All three routers are running EIGRP. Routers R2 and R3 are connected to the subnet 10.0.1.0/24. Both routers advertise the route to reach that subnet to R1. Router R1 receives the two routing updates for the subnet 10.0.1.0/24 with the same metric (the metric is the same because both routers connect to the subnet 10.0.1.0/24 and R1 across the links with the same bandwidth and delay values). Router R1 places both routes in the routing table and load balances across three links.

You can verify that R1 is indeed using both paths by typing the show ip route command:



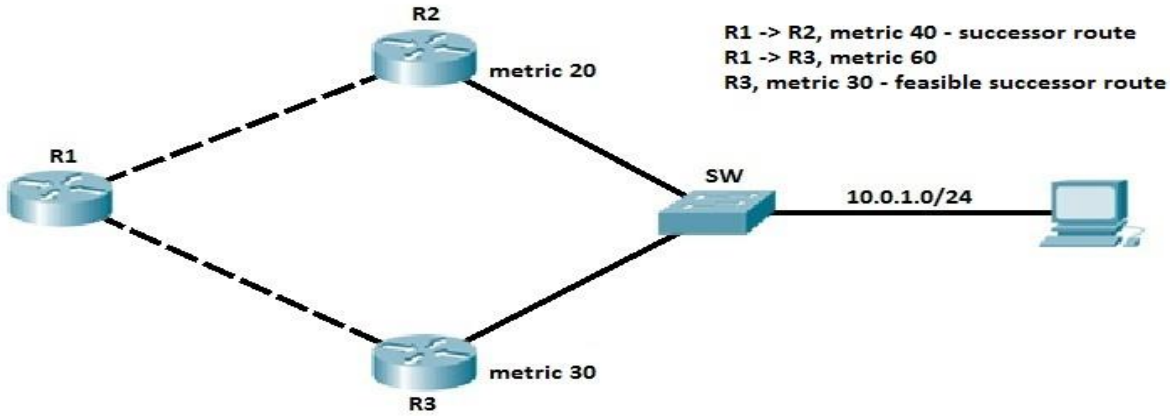
```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
D       10.0.1.0 [90/284160] via 192.168.0.2, 00:05:28, FastEthernet0/0
                 [90/284160] via 172.16.0.2, 00:05:28, FastEthernet0/1
C    172.16.0.0/16 is directly connected, FastEthernet0/1
C    192.168.0.0/24 is directly connected, FastEthernet0/0
```

One of the advantages of EIGRP is that, unlike OSPF and many other routing protocols, EIGRP also supports unequal-cost load balancing. You can set up your router to load balance over links with different metric to reach a destination. To accomplish unequal-cost load balancing, the variance command is used. The command takes one parameter, the multiplier, which tells the router to load balance across each link with the metric for the destination less than the feasible distance multiplied by the multiplier value.

NOTE

The multiplier value, by default, is 1. The maximum value is 128.

Consider the following example.


R2
metric 20
R1 -> R2, metric 40 - successor route
R1 -> R3, metric 60
R3, metric 30 - feasible successor route
R1
SW
10.0.1.0/24
metric 30
R3

All three routers are running EIGRP. Routers R2 and R3 are connected to the subnet 10.0.1.0/24. Both routers advertise the route to reach that subnet to R1. Router R1 chooses the route from R2 as the best route. Let's say that R1 calculated the metric of 40 for the path through R2. That route is placed in the R1's routing table. But what if we want to load balance traffic across the other link? The route through R3 has a feasible distance of 30, which is less than the metric of the successor route, so the feasibility condition has been met and that route has been placed in the R1's topology table. Let's say that R1 *calculated the metric of 60 for the route through R3. To enable load balancing across that link, you need to use the variance command:*

```
(router-eigrp) variance MULTIPLIER
```

In our example, the variance 2 command can be used. This tells the router to load balance across any links with the metric less then 80 (because 40 times 2 is 80). The route through R3 is added to the routing table.

NOTE
A path has to be a feasible successor route to be used in unequal load balancing.

EIGRP Reliable Transport Protocol (RTP)

EIGRP doesn't send messages with UDP or TCP; instead, a Cisco's protocol called Reliable Transport Protocol (RTP) is used for communication between EIGRP-speaking routers. As the name implies, reliability is a key feature of this protocol, and it is designed to enable quick delivery of updates and tracking of data reception.

Five different packets types are used by EIGRP:

- *Update – contains route information. When routing updates are sent in response to the metric or topology changes, reliable multicasts are used. In the event that only one router needs an update, for example when a new neighbor is discovered, unicasts are used.*
- *Query – a request for specific routes that always uses the reliable multicast method. Routers send queries when they realize they've lost the path to a particular network and are looking for alternative paths.*
- *Reply – sent in response to a query via the unicast method. Replies can include a specific route to the queried destination or declare that there is no known route.*
- *Hello – used to discover EIGRP neighbors. It is sent via unreliable multicast and no acknowledgment is required.*
- *Acknowledgment (ACK) – sent in response to an update and is always unicast. ACKs are not sent reliably.*

NOTE
The acronym RTP is also used for a different protocol – Real-time Transport Protocol (RTP), used for VoIP communication.

EIGRP Diffusing Update Algorithm (DUAL)

Diffusing Update Algorithm (DUAL) is an algorithm used by EIGRP to select and maintain the best route to each remote network. DUAL is also used for the following purposes:

- *discover a backup route if there is one available.*
- *support for variable length subnet masks (VLSMs).*
- *perform dynamic route recoveries.*
- *query neighbors for unknown alternate routes.*
- *send out queries for alternate routes.*

EIGRP stores all routes advertised by all EIGRP neighbors. The metric of these routes is used by DUAL to select the efficient and loop free paths. DUAL selects routes that will be inserted into the routing table. If a route fails, and there is no feasible successor, DUAL chooses a replacement route, which usually takes a couple of seconds.

The following requirements must be met in order for DUAL to work properly:

- *EIGRP neighbors must discovered.*
- *all transmitted EIGRP messages should be received correctly.*
- *all changes and messages should be processed in the order in which they're detected.*

EIGRP summary

Here is a list of the most important EIGRP features:

- *advanced distance vector routing protocol*
- *classless routing protocol*
- *supports VLSM (Variable Length Subnet Mask)*
- *converges fast*
- *supports multiple Network layer protocols (IPv4, IPv6, IPX, AppleTalk…)*
- *uses multicast address of 224.0.0.10 for routing updates*
- *sends partial routing updates*
- *supports equal and unequal-cost load balancing*
- *supports manual summarization on any router within a network*
- *by default, uses bandwidth and delay to calculate its metric*
- *Cisco proprietary*
- *supports MD5 authentication*

CHAPTER TEN
OSPF OVERVIEW

We Will Cover These Topics in This Chapter

- ***OSPF overview***
- ***OSPF configuration***
- ***Designated & Backup Designated Router***
- ***OSPF authentication***
- ***OSPF summarization***
- ***Differences between OSPF and EIGRP***

OSPF overview

OSPF (Open Shortest Path First) is a link state routing protocol. Because it is an open standard, it is implemented by a variety of network vendors. OSPF will run on most routers that doesn't necessarily have to be Cisco routers (unlike EIGRP which can be run only on Cisco routers).

Here are the most important features of OSPF:

- *a classless routing protocol*
- *supports VLSM, CIDR, manual route summarization, equal cost load balancing*
- *incremental updates are supported*
- *uses only one parameter as the metric – the interface cost.*
- *the administrative distance of OSPF routes is, by default, 110.*
- *uses multicast addresses 224.0.0.5 and 224.0.0.6 for routing updates.*

Routers running OSPF have to establish neighbor relationships before exchanging routes. Because OSPF is a link state routing protocol, neighbors don't exchange routing tables. Instead, they exchange information about network topology. Each OSFP router then runs SFP algorithm to calculate the best routes and adds those to the routing table. Because each router knows the entire topology of a network, the chance for a routing loop to occur is minimal.

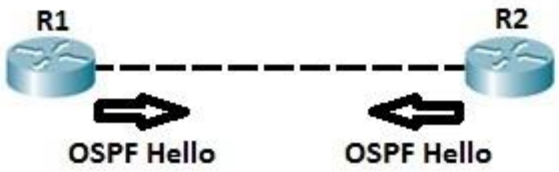
Each OSPF router stores routing and topology information in three tables:

- *Neighbor table – stores information about OSPF neighbors*
- *Topology table – stores the topology structure of a network*
- *Routing table – stores the best routes*

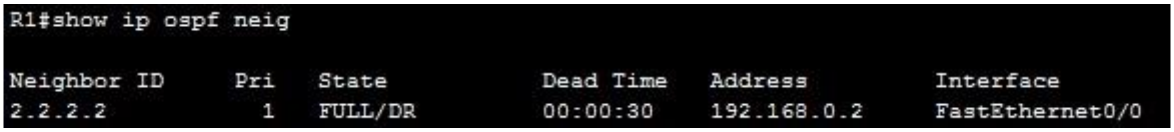
OSPF neighbors

OSPF routers need to establish a neighbor relationship before exchanging routing updates. OSPF neighbors are dynamically discovered by sending Hello packets out each OSPF-enabled interface on a router. Hello packets are sent to the multicast IP address of 224.0.0.5.

The process is explained in the following figure:



Routers R1 and R2 are directly connected. After OSFP is enabled both routers send Hellos to each other to establish a neighbor relationship. You can verify that the neighbor relationship has indeed been established by typing the show ip ospf neighbors command.



```
R1#show ip ospf neig

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   FULL/DR         00:00:30    192.168.0.2     FastEthernet0/0
```

In the example above, you can see that the router-id of R2 is 2.2.2.2. Each OSPF router is assigned a router ID. A router ID is determined by using one of the following:

1. using the router-id command under the OSPF process.
2. using the highest IP address of the router's loopback interfaces.
3. using the highest IP address of the router's physical interfaces.

The following fields in the Hello packets must be the same on both routers in order for routers to become neighbors:

- subnet
- area id
- hello and dead interval timers
- authentication
- area stub flag
- MTU

By default, OSPF sends hello packets every 10 second on an Ethernet network (Hello interval). A dead timer is four times the value of the hello interval, so if a routers on an Ethernet network doesn't receive at least one Hello packet from an OSFP neighbor for 40 seconds, the routers declares that neighbor to be down.

OSPF neighbor states

Before establishing a neighbor relationship, OSPF routers need to go through several state changes. These states are explained below.

1. Init state – a router has received a Hello message from the other OSFP router
2. 2-way state – the neighbor has received the Hello message and replied with a Hello message of his own
3. Exstart state – beginning of the LSDB exchange between both routers. Routers are starting to exchange link state information.
4. Exchange state – DBD (Database Descriptor) packets are exchanged. DBDs contain LSAs headers. Routers will use this information to see what LSAs need to be exchanged.
5. Loading state – one neighbor sends LSRs (Link State Requests) for every network it doesn't know about. The other neighbor replies with the LSUs (Link State Updates) which contain information about

requested networks. After all the requested information have been received, other neighbor goes through the same process
6. Full state – both routers have the synchronized database and are fully adjacent with each other.

OSPF areas

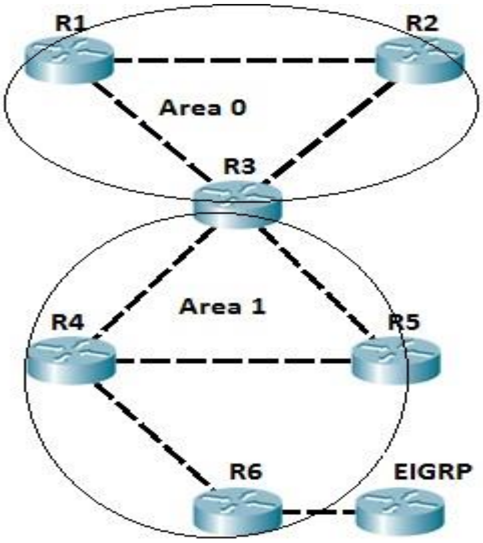
OSPF uses the concept of areas. An area is a logical grouping of contiguous networks and routers. All routers in the same area have the same topology table, but they don't know about routers in the other areas. The main benefits of creating areas is that the size of the topology and the routing table on a router is reduced, less time is required to run the SFP algorithm and routing updates are also reduced.

Each area in the OSPF network has to connect to the backbone area (area 0). All router inside an area must have the same area ID to become OSPF neighbors. A router that has interfaces in more than one area (area 0 and area 1, for example) is called Area Border Router (ABR). A router that connects an OSPF network to other routing domains (EIGRP network, for example) is called Autonomous System Border Router (ASBR).

NOTE

In OSPF, manual route summarization is possible only on ABRs and ASBRs.

To better understand the concept of areas, consider the following example.



All routers are running OSPF. Routers R1 and R2 are inside the backbone area (area 0). Router R3 is an ABR, because it has interfaces in two areas, namely area 0 and area 1. Router R4 and R5 are inside area 1. Router R6 is an ASBR, because it connects OSFP network to another routing domain (an EIGRP domain in this case). If the R1's directly connected subnet fails, router R1 sends the routing update only to R2 and R3, because all routing updates all localized inside the area.

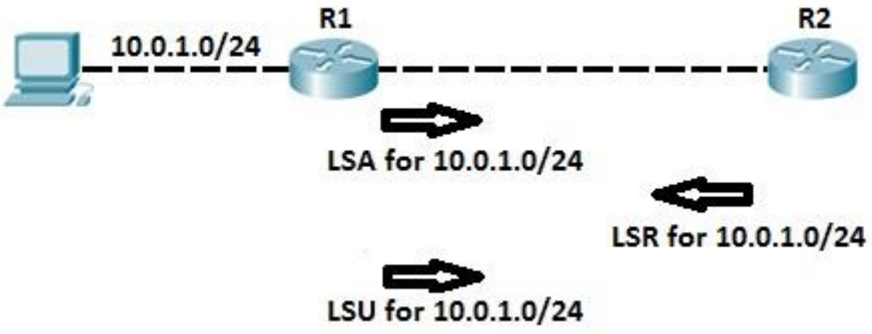
NOTE

The role of an ABR is to advertise address summaries to neighboring areas. The role of an ASBR is to connect an OSPF routing domain to another external network (e.g. Internet, EIGRP network…).

LSA, LSU and LSR

The LSAs (Link-State Advertisements) are used by OSPF routers to exchange topology information. Each LSA contains routing and topology information to describe a part of an OSPF network. When two neighbors decide to exchange routes, they send each other a list of all LSAa in their respective topology database. Each router then checks its topology database and sends a Link State Request (LSR) message requesting all LSAs not found in its topology table. Other router responds with the Link State Update (LSU) that contains all LSAs requested by the other neighbor.

The concept is explained in the following example:



After configuring OSPF on both routers, routers exchange LSAs to describe their respective topology database. Router R1 sends an LSA header for its directly connected network 10.0.1.0/24. Router R2 check its topology database and determines that it doesn't have information about that network. Router R2 then sends Link State Request message requesting further information about that network. Router R1 responds with Link State Update which contains information about subnet 10.0.1.0/24 (next hop address, cost…).

OSPF configuration

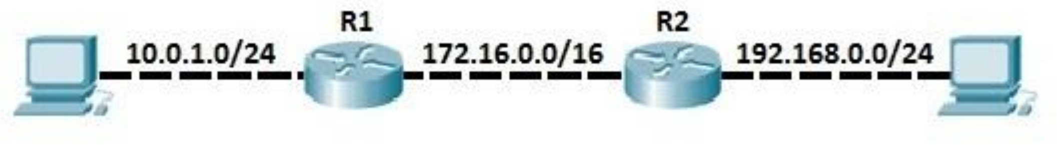
Configuring OSPF 1

OSPF basic configuration is very simple. Just like with other routing protocols covered so far (RIP, EIGRP) first you need to enable OSPF on a router. This is done by using the router ospf PROCESS-ID global configuration command. Next, you need to define on which interfaces OSPF will run and what networks will be advertised. This is done by using the network IP_ADDRESS WILDCARD_MASK AREA_ID command from the ospf configuration mode.

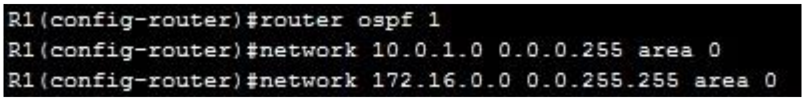
NOTE

The OSPF process number doesn’t have to be the same on all routers in order to establish a neighbor relationship, but the Area ID has to be the same on all neighboring routers in order for routers to become neighbors.

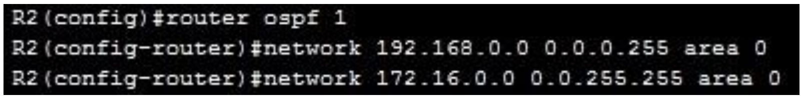
Let’s get started with some basic OSPF configuration. We will use the following network topology:



First, we need to enable OSPF on both routers. Then we need to define what network will be advertised into OSPF. This can be done by using the following sequence of commands on both routers:

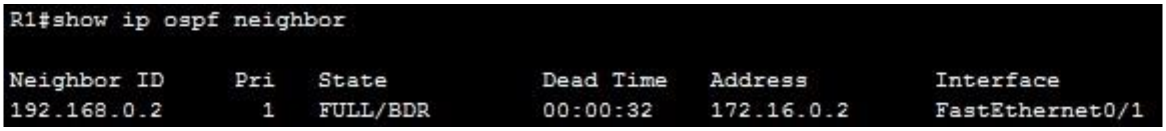


```
R1(config-router)#router ospf 1
R1(config-router)#network 10.0.1.0 0.0.0.255 area 0
R1(config-router)#network 172.16.0.0 0.0.255.255 area 0
```



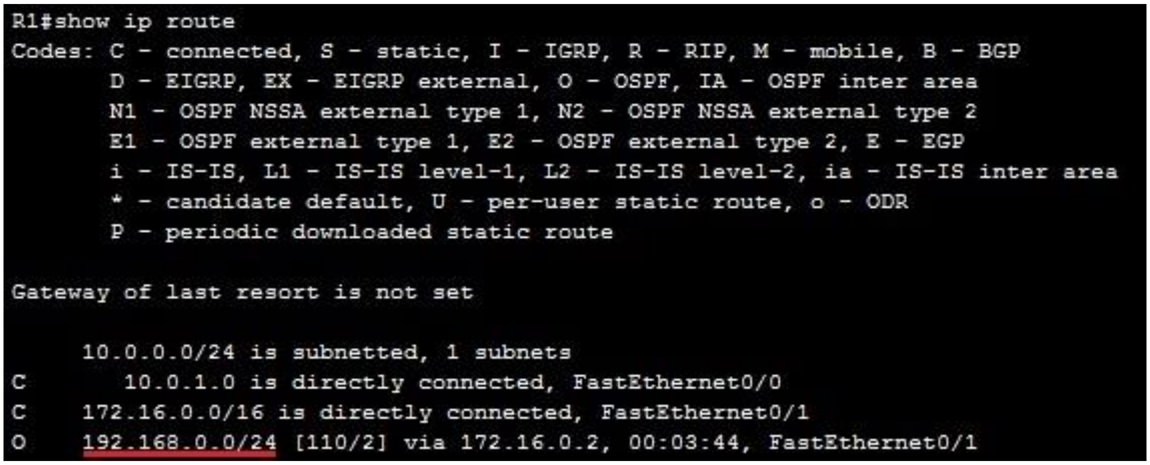
```
R2(config)#router ospf 1
R2(config-router)#network 192.168.0.0 0.0.0.255 area 0
R2(config-router)#network 172.16.0.0 0.0.255.255 area 0
```

The network commands entered on both routers include subnets directly connected to both routers. We can verify that the routers have become neighbors by typing the show ip ospf neighbors command on either router:



```
R1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.0.2       1   FULL/BDR        00:00:32    172.16.0.2      FastEthernet0/1
```

To verify if the routing updated were exchanged, we can use the show ip route command. All routes marked with the character O are OSPF routes. For example, here is the output of the command on R1:



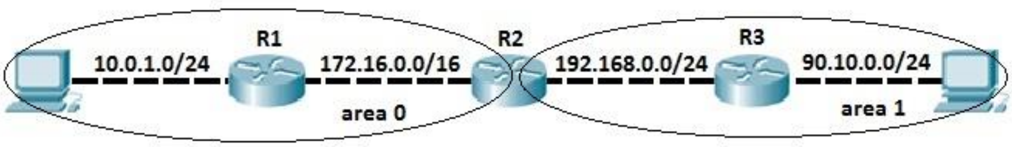
```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.1.0 is directly connected, FastEthernet0/0
C    172.16.0.0/16 is directly connected, FastEthernet0/1
O    192.168.0.0/24 [110/2] via 172.16.0.2, 00:03:44, FastEthernet0/1
```

You can see that R1 has learned about the network 192.168.0.0/24 through OSPF.

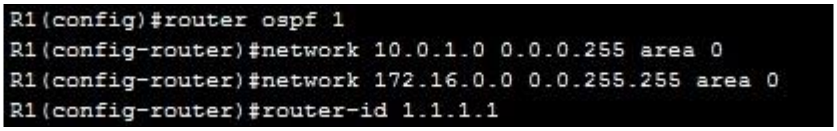
Configuring OSPF 2

Although basic OSPF configuration can be very simple, OSPF provides many extra features that can get really complex. In this example, we will configure multiarea OSPF network and some other OSPF features.

Consider the following multiarea OSPF network:



In this example we have two OSPF areas – area 0 and area 1. As you can see from the network topology depicted above, routers R1 and R3 are in the area 0 and area 1, respectively. Router 2 connects to both areas, which makes him an ABR (Area Border Router). Our goal is to advertise the subnets directly connected to R1 and R3. To do that, the following configuration on R1 will be used:



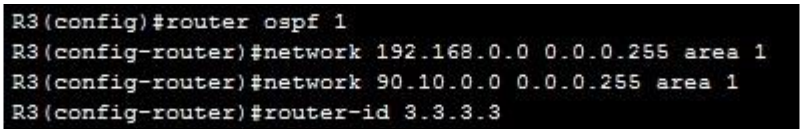
```
R1(config)#router ospf 1
R1(config-router)#network 10.0.1.0 0.0.0.255 area 0
R1(config-router)#network 172.16.0.0 0.0.255.255 area 0
R1(config-router)#router-id 1.1.1.1
```

NOTE

We have used the router-id 1.1.1.1 command to manually specify the router ID of this router. OSPF process will use that RID (router-id) when communicating with other OSPF neighbors.

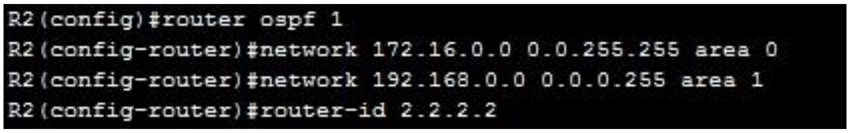
Because R1 connects only to R2, we only need to establish a neighbor relationship with R2 and advertise directly connected subnet into OSPF.

Configuration of R3 looks similar, but with one difference, namely area number. R3 is in the area 1.



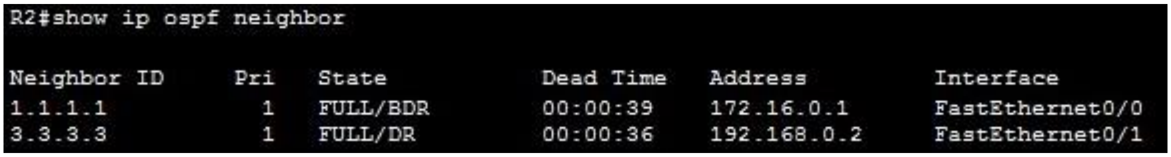
```
R3(config)#router ospf 1
R3(config-router)#network 192.168.0.0 0.0.0.255 area 1
R3(config-router)#network 90.10.0.0 0.0.0.255 area 1
R3(config-router)#router-id 3.3.3.3
```

What about R2? Well, because R2 is an ABR, we need to establish neighbor relationship with both R1 and R3. To do that, we need to specify different area ID for each neighbor relationship, 0 for R1 and 1 for R2. We can do that using the following sequence of commands:



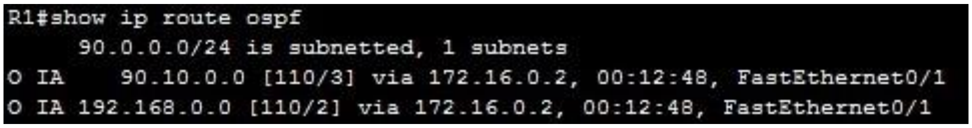
```
R2(config)#router ospf 1
R2(config-router)#network 172.16.0.0 0.0.255.255 area 0
R2(config-router)#network 192.168.0.0 0.0.0.255 area 1
R2(config-router)#router-id 2.2.2.2
```

Now R2 should have neighbor relationship with both R1 and R3. We can verify that by using the show ip ospf neighbor command:

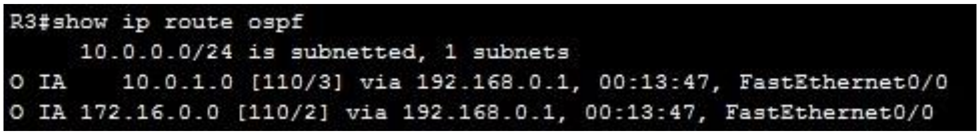


```
R2#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           1   FULL/BDR        00:00:39    172.16.0.1      FastEthernet0/0
3.3.3.3           1   FULL/DR         00:00:36    192.168.0.2     FastEthernet0/1
```

To verify if directly connected subnets are really advertised into the different area, we can use the show ip route ospf command on both R1 and R3:



```
R1#show ip route ospf
     90.0.0.0/24 is subnetted, 1 subnets
O IA    90.10.0.0 [110/3] via 172.16.0.2, 00:12:48, FastEthernet0/1
O IA 192.168.0.0 [110/2] via 172.16.0.2, 00:12:48, FastEthernet0/1
```



```
R3#show ip route ospf
     10.0.0.0/24 is subnetted, 1 subnets
O IA    10.0.1.0 [110/3] via 192.168.0.1, 00:13:47, FastEthernet0/0
O IA 172.16.0.0 [110/2] via 192.168.0.1, 00:13:47, FastEthernet0/0
```

Characters IA in front of the routes indicate that these routes reside in different areas.

NOTE

Since they reside in different areas, R1 and R3 will never establish a neighbor relationship.

Designated & Backup Designated Router

Based on the network type, OSPF router can elect one router to be a Designated Router (DR) and one router to be a Backup Designated Router (BDR). For example, on multiaccess broadcast networks (such as LANs) routers defaults to elect a DR and BDR. DR and BDR serve as the central point for exchanging OSPF routing information. Each non-DR or non-BDR router will exchange routing information only with the DR and BDR, instead of exchanging updates with every router on the network segment. DR will then distribute topology information to every other router inside the same area, which greatly reduces OSPF traffic.

To send routing information to a DR or BDR the multicast address of 224.0.0.6 is used. DR sends routing updates to the multicast address of 224.0.0.5. If DR fails, BDR takes over its role of redistributing routing information.

Every router on a network segment will establish a full neighbor relationship with the DR and BDR. Non-DR and non-BDR routers will establish a two way neighbor relationship between themselves.

NOTE

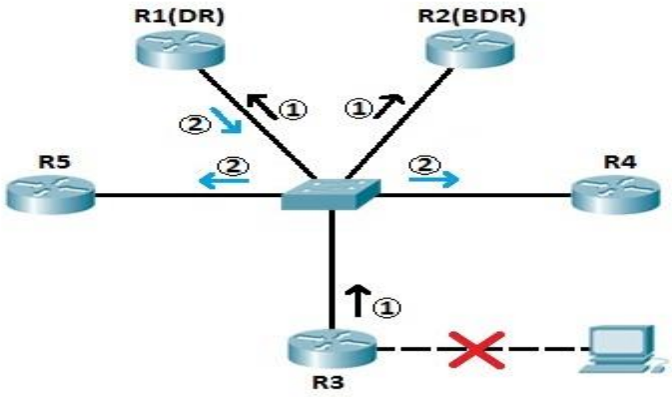
On point-to-point links, a DR and BDR are not elected since only two routers are directly connected.

On LANs, DR and BDR have to be elected. Two rules are used to elect a DR and BDR:

1. *router with the highest OSPF priority will become a DR. By default, all routers have a priority of 1.*
2. *if there is a tie, a router with the highest router ID wins the election.*

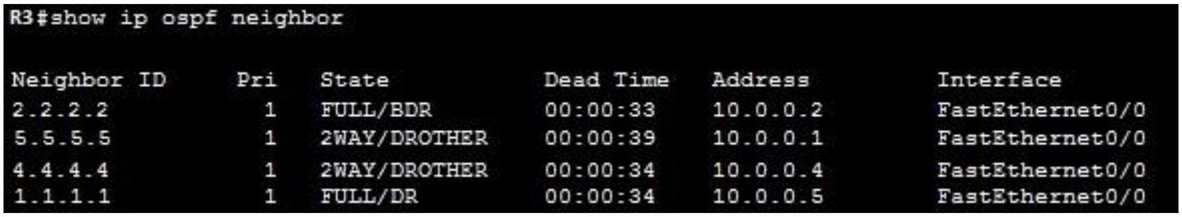
The router with the second highest OSPF priority or router ID will become a BDR.

To better understand the concept, consider the following example.



All routers depicted above are in the same area (area 0). All routers are running OSPF. Routers R1 and R2 have been elected as DR and BDR because they have the highest and the second highest router ID. If, for example, R3's directly connected subnet fails, R3 informs R1 and R2 (the DR and BDR for the segment) of *the network change (step 1). R1 then informs all other non-DR and non-BDR routers of the change in topology (step 2).*

We can verify that R1 and R2 are indeed the DR and BDR of the segment by typing the show ip ospf neighbors command on R3:



```
R3#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   FULL/BDR        00:00:33    10.0.0.2        FastEthernet0/0
5.5.5.5           1   2WAY/DROTHER    00:00:39    10.0.0.1        FastEthernet0/0
4.4.4.4           1   2WAY/DROTHER    00:00:34    10.0.0.4        FastEthernet0/0
1.1.1.1           1   FULL/DR         00:00:34    10.0.0.5        FastEthernet0/0
```

Router R3 will be in the 2WAY state with every other non-DR or non-BDR router.

NOTE

You can influence the DR and BDR election process by manually configuring the OSPF priority. This is done by using the ip ospf priority VALUE command interface command.

OSPF authentication

OSPF can be configured to authenticate every OSPF message. This is usually done to prevent a rogue router from injecting false routing information and therefore causing a Denial-of-Service attack.

Two types of authentication can be used:
1. *clear text authentication – clear text passwords are used*
2. *MD5 authentication – MD5 authentication is used. This type of authentication is more secure because the password doesn't go in clear-text over the network.*

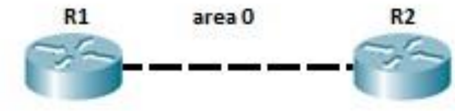
NOTE

With OSPF authentication turned on, routers must pass the authentication process before becoming OSPF neighbors.

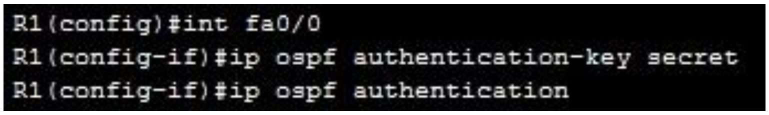
To configure clear text authentication, the following steps are required:

1. *configure the OSPF password on the interface by using the ip ospf authentication-key PASSWORD interface command*
2. *configure the interface to use OSPF clear-text authentication by using the ip ospf authentication interface command*

In the following example, we will configure OSPF clear-text authentication.

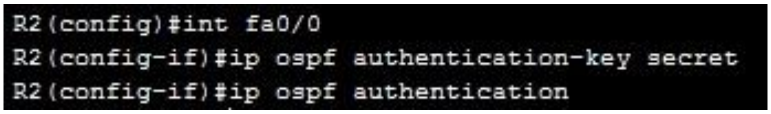


Both routers are running OSPF. On R1, we need to enter the following commands:



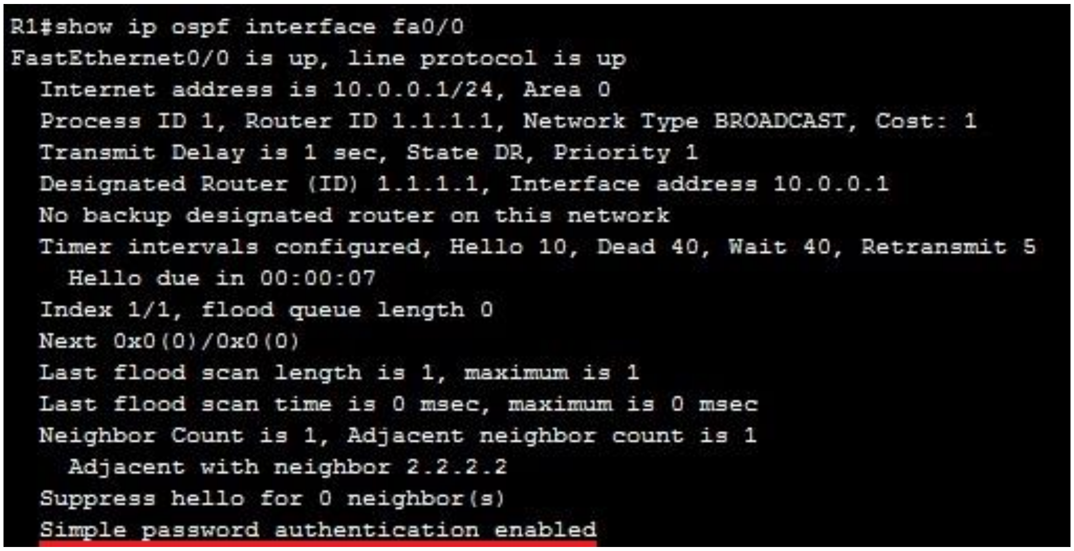
```
R1(config)#int fa0/0
R1(config-if)#ip ospf authentication-key secret
R1(config-if)#ip ospf authentication
```

The same commands have to be entered on R2:



```
R2(config)#int fa0/0
R2(config-if)#ip ospf authentication-key secret
R2(config-if)#ip ospf authentication
```

To verify that clear-text authentication is indeed turned on, we can use the show ip ospf interface INTERFACE_NUMBER/INTERFACE_TYPE command on either router:

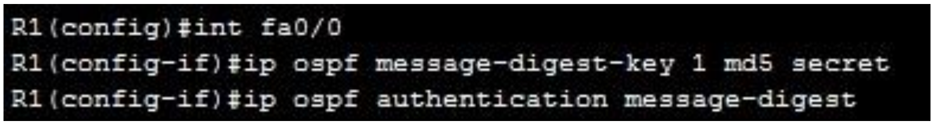


```
R1#show ip ospf interface fa0/0
FastEthernet0/0 is up, line protocol is up
  Internet address is 10.0.0.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.1.1.1, Interface address 10.0.0.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:07
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 2.2.2.2
  Suppress hello for 0 neighbor(s)
  Simple password authentication enabled
```

Configuring OSPF MD5 authentication is very similar to configuring clear-text authentication. Two commands are also used:

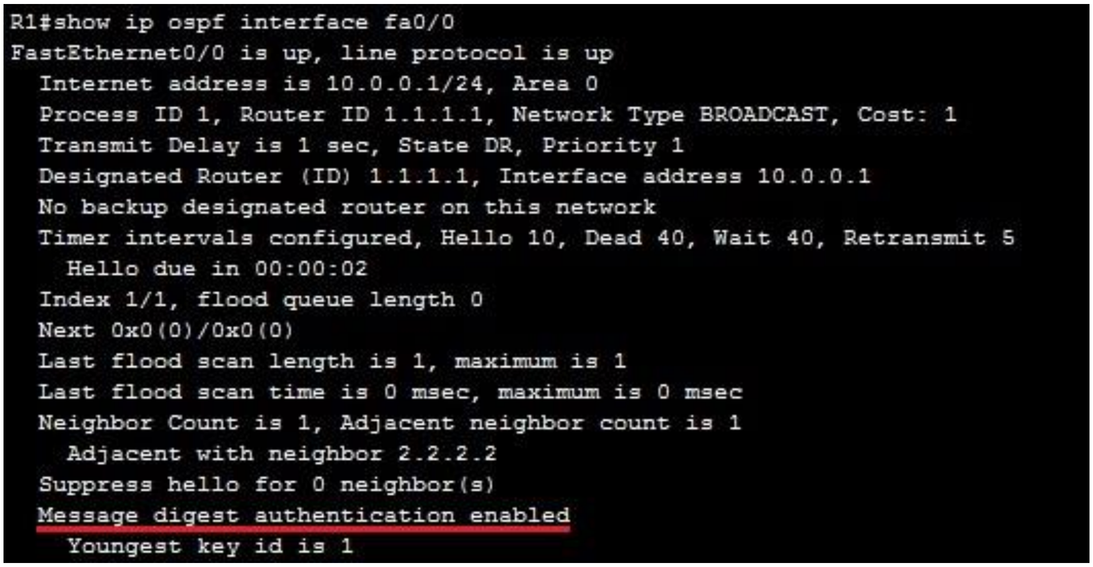
1. *First you need to configure the MD5 value on an interface by using the ip ospf message-digest-key 1 md5 VALUE interface command*
2. *Next, you need to configure the interface to use MD5 authentication by using the ip ospf authentication message-digest interface command*

Here is an example configuration on R1:



```
R1(config)#int fa0/0
R1(config-if)#ip ospf message-digest-key 1 md5 secret
R1(config-if)#ip ospf authentication message-digest
```

You can verify that R1 is using OSPF MD5 authentication by typing the show ip ospf INTERFACE/INTERFACE_TYPE command:



```
R1#show ip ospf interface fa0/0
FastEthernet0/0 is up, line protocol is up
  Internet address is 10.0.0.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.1.1.1, Interface address 10.0.0.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 2.2.2.2
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
    Youngest key id is 1
```

NOTE

OSPF authentication type can also be enabled on an area basis, instead of configuring OSPF authentication type per interface basis. This is done by using the area AREA_ID authentication [message-digest] command under the OSPF configuration mode. If you omit the message-digest keyword, a clear-text authentication will be used for that area. All interfaces inside the area will use OSPF authentication.

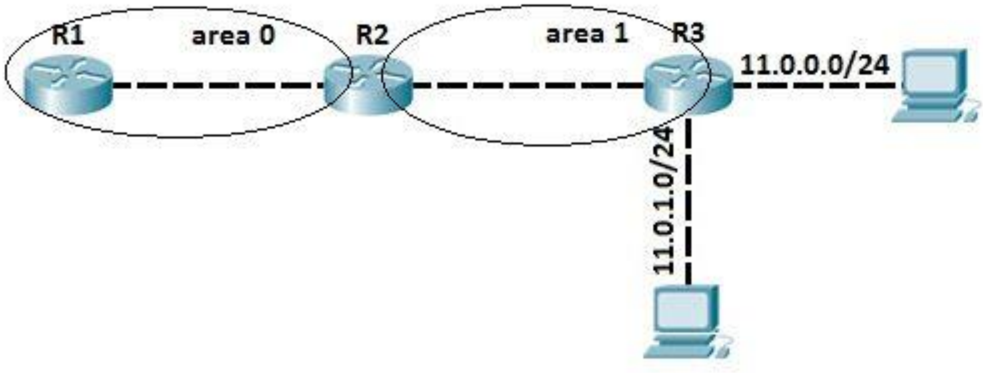
OSPF summarization

Route summarization helps reduce OSPF traffic and route computation. OSPF, unlike EIGRP, doesn't support automatic summarization. Also, unlike EIGRP, where you can summarize routes on every router in an EIGRP network, OSFP can summarize routes only on ABRs and ASBRs.

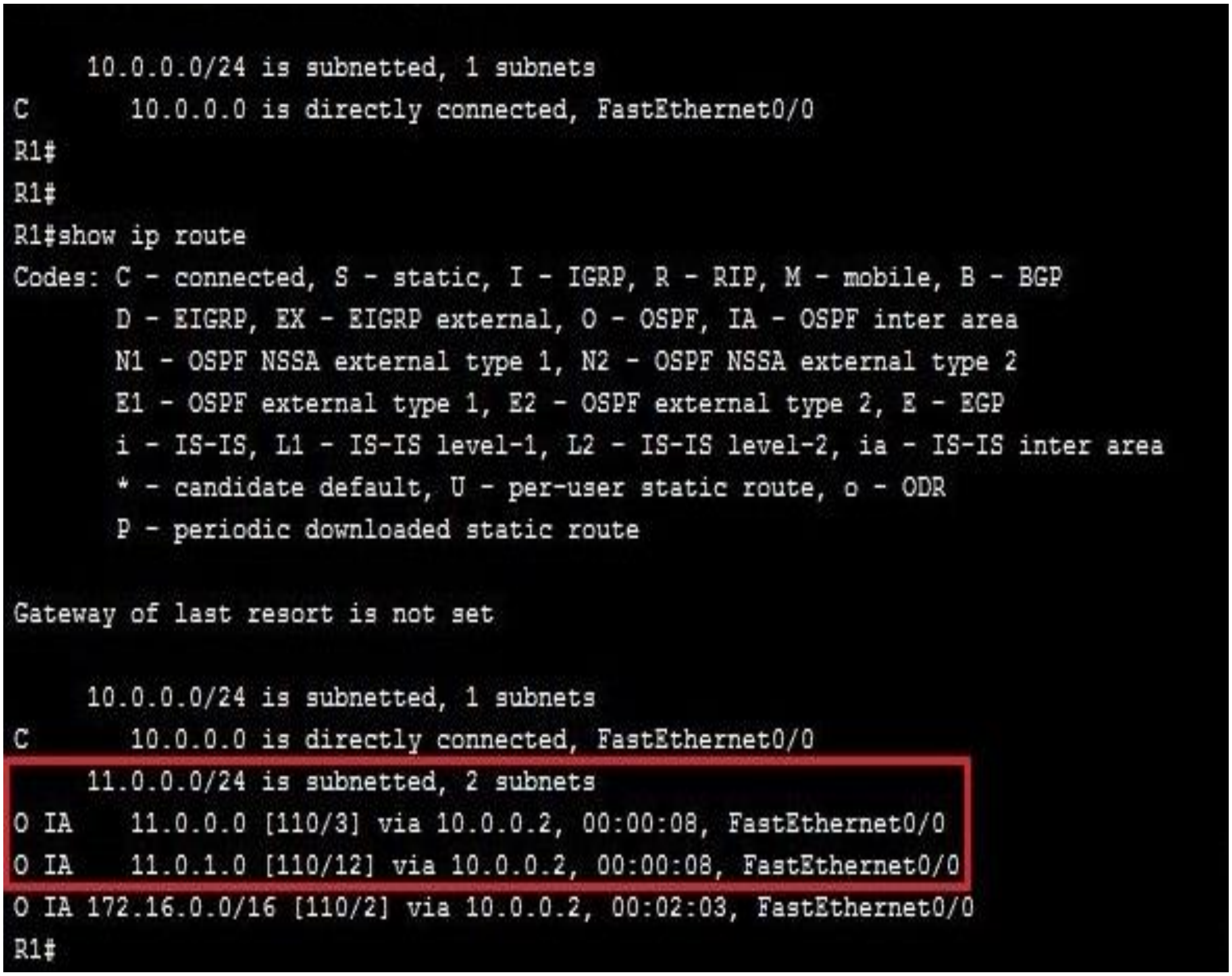
The following command is used for OSPF summarization:

```
(config-router) area AREA_ID range IP_ADDRESS MASK
```

To better understand OSPF summarization, consider the following example network:

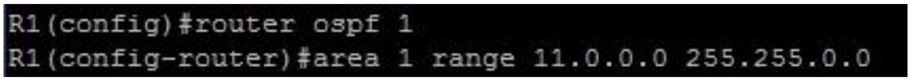


All three routers are running OSPF and exchanging routers. Before OSPF summarization is configured, the router R1 inside the backbone area has two entries for the networks 11.0.0.0/24 and 11.0.1.0/24 in its routing table.



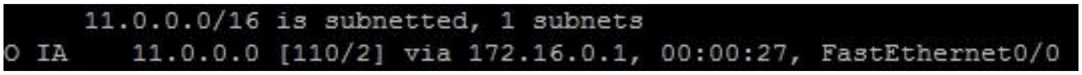
```
     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
R1#
R1#
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
     11.0.0.0/24 is subnetted, 2 subnets
O IA    11.0.0.0 [110/3] via 10.0.0.2, 00:00:08, FastEthernet0/0
O IA    11.0.1.0 [110/12] via 10.0.0.2, 00:00:08, FastEthernet0/0
O IA 172.16.0.0/16 [110/2] via 10.0.0.2, 00:02:03, FastEthernet0/0
R1#
```

We could summarize these two subnets on R2, so that R1 receive only one routing update for both subnets. To do that, the following command can be used on R2:



```
R1(config)#router ospf 1
R1(config-router)#area 1 range 11.0.0.0 255.255.0.0
```

Now, R1 has only one entry in its routing table for R3's directly connected subnets:



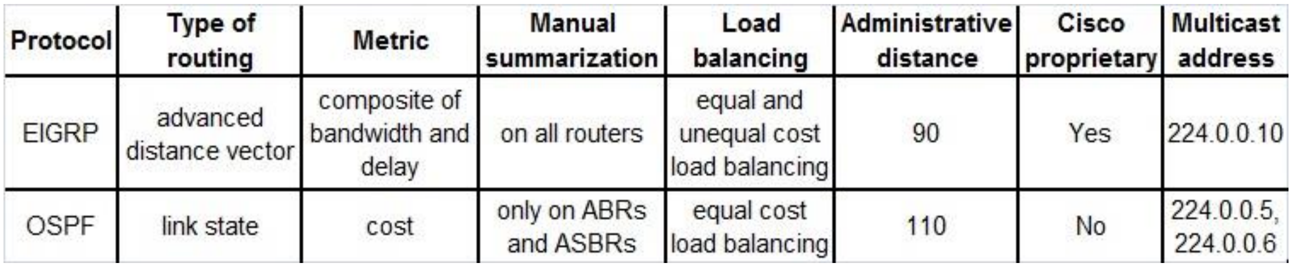
```
     11.0.0.0/16 is subnetted, 1 subnets
O IA    11.0.0.0 [110/2] via 172.16.0.1, 00:00:27, FastEthernet0/0
```

NOTE

Be careful with summarization. In this case, router R1 thinks that R2 has routes for all subnets in the range 11.0.0.0 – 11.0.255.255. When summarizing, try to be as specific as possible.

Differences between OSPF and EIGRP

The following table lists the differences between OSPF and EIGRP:



Protocol	Type of routing	Metric	Manual summarization	Load balancing	Administrative distance	Cisco proprietary	Multicast address
EIGRP	advanced distance vector	composite of bandwidth and delay	on all routers	equal and unequal cost load balancing	90	Yes	224.0.0.10
OSPF	link state	cost	only on ABRs and ASBRs	equal cost load balancing	110	No	224.0.0.5, 224.0.0.6

CHAPTER ELEVEN
LAYER 2 SWITCHING

We Will Cover These Topics in This Chapter

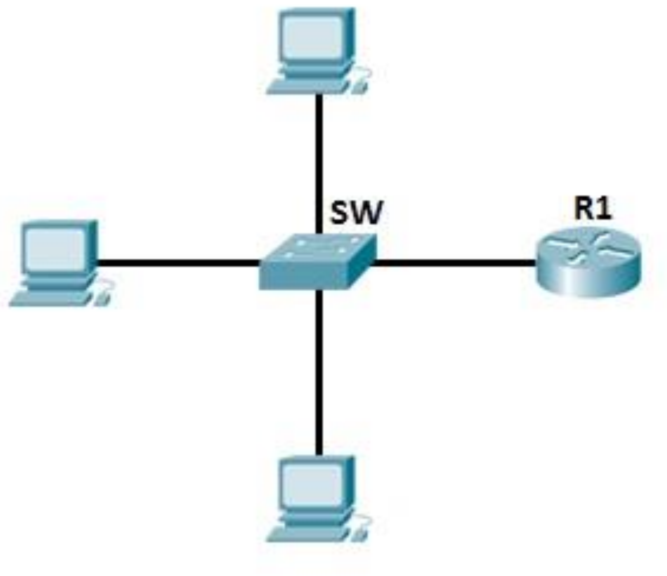
- ***Layer 2 switching***
- ***Collision & broadcast domain***
- ***CSMA/CD***

Layer 2 switching

Layer 2 switching (or Data Link layer switching) is the process of using devices' MAC addresses on a LAN to segment a network. Switches and bridges are used for Layer 2 switching. They break up one large collision domain into multiple smaller ones.

In a typical LAN, all hosts are connected to one central device. In the past, the device was usually a hub. But hubs had many disadvantages, such as not being aware of traffic that passes through them, creating one large collision domain, etc. To overcome some of the problems with hubs, the bridges were created. They were better than hubs because they created multiple collision domains, but they had limited number of ports. Finally, switches were created and are still widely used today. Switches have more ports than bridges, can inspect incoming traffic and make forwarding decisions accordingly. Each port on a switch is a separate collision domain.

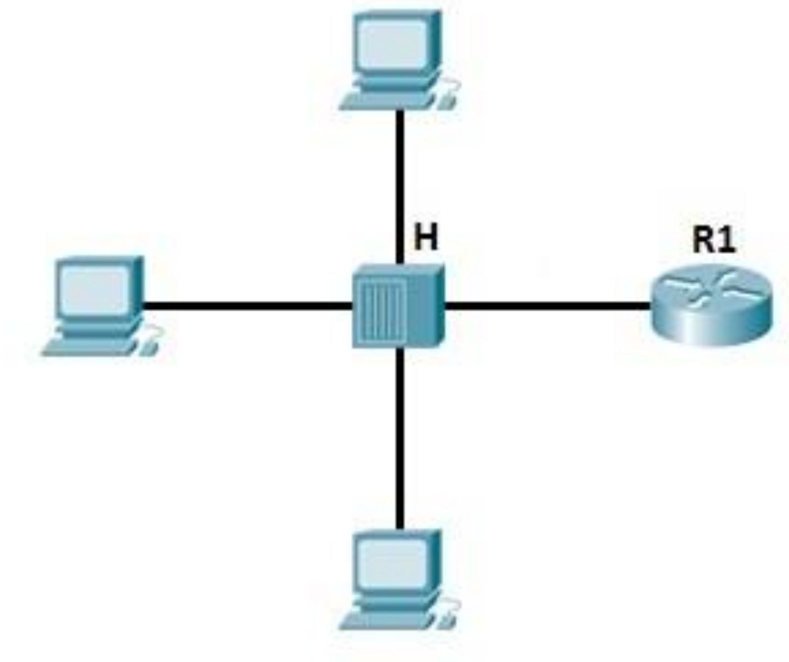
Here is an example of the typical LAN network used today – the switch serves as a central device that connects all devices together:



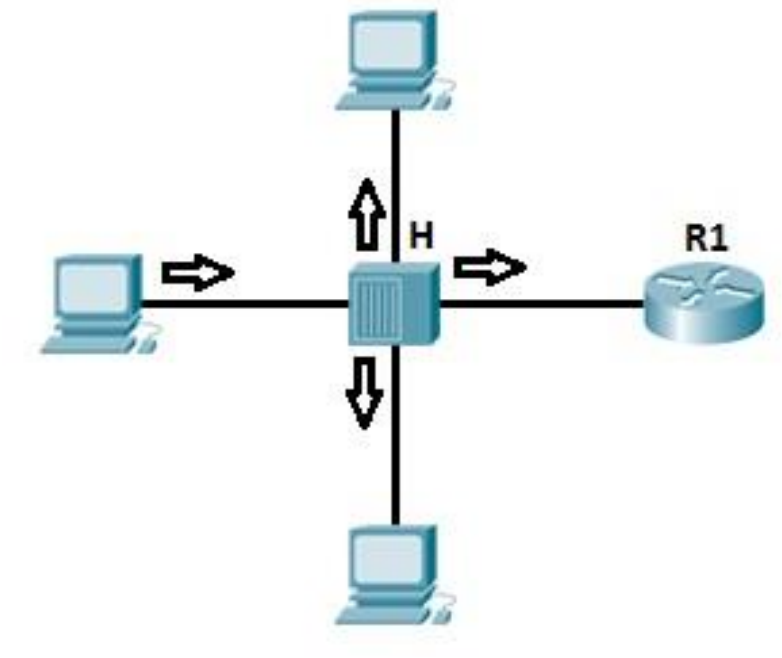
Differences between hubs and switches

To better understand the concept of packet switching based on the hardware address of a device, you need to understand how switches differ from hubs.

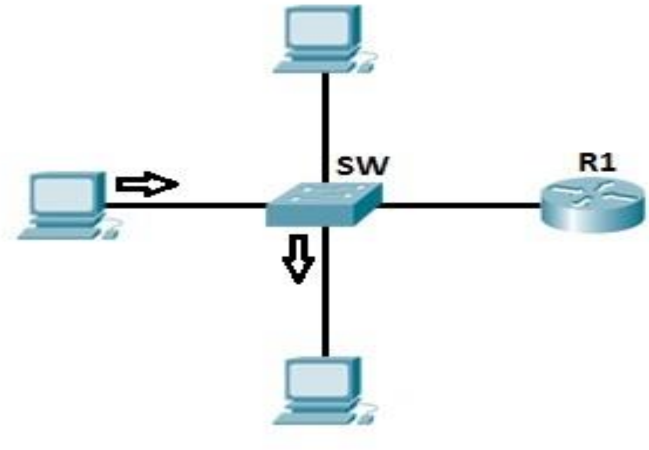
First, consider the example of a LAN, with all hosts connecting to a hub:



As mentioned previously, hubs create only one collision domain, so the chance for a collision to occur is high. The hub depicted above simply repeats the signal it receives out all ports, except the one from which the signal was received, so no packet filtering takes place. Imagine if you had 20 hosts connected to a hub, a packet would be sent to 19 hosts, instead of just one! This can also cause security problems, because an attacker can capture all traffic on the network.



Now consider the way the switches work. We have the same topology as above, only this we are using a switch instead of a hub.

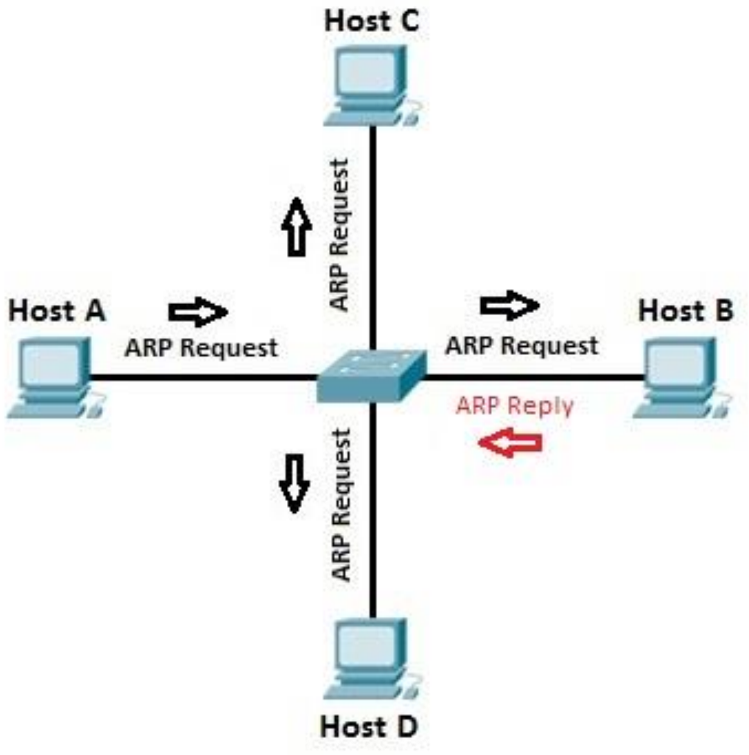


Switches increase the number of collision domains. Each port is one collision domain, which means that the chances for collisions to occur are minimal. A switch learns which device is connected to which port and forwards a frame based on the destination MAC address included in the frame. This reduces traffic on the LAN and enhances security.

How switches work

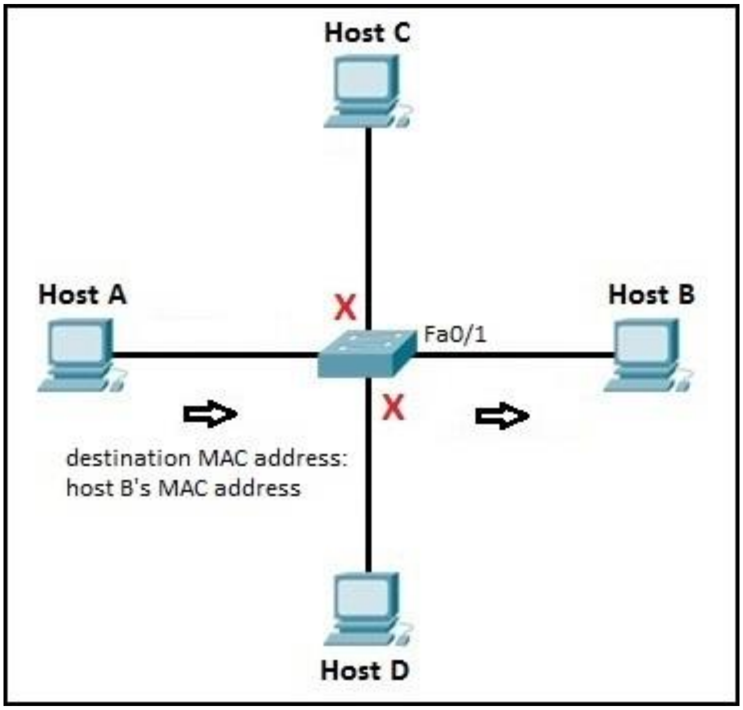
Each network card has a unique identifier called a Media Access Control (MAC) address. This address is used in LANs for communication between devices on the same network segment. Devices that want to communicate need to know each other MAC address before sending out packets. They use a process called ARP (Address Resolution Protocol) to find out the MAC address of another device. When the hardware address of the destination host is known, the sending host has all the required information to communicate with the remote host.

To better understand the concept of ARP, let's take a look at the following example:

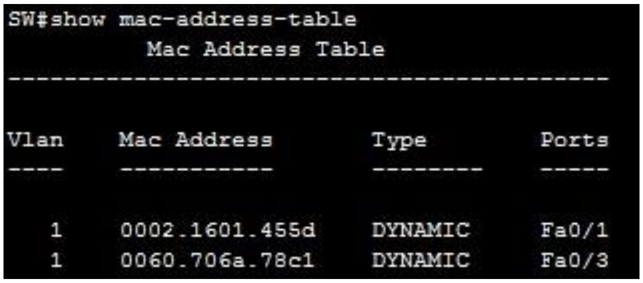


Let’s say that host A wants to communicate with host B for the first time. Host A knows the IP address of host B, but since this is the first time the two hosts communicate, the hardware (MAC) addresses are not known. Host A uses the ARP process to find out the MAC address of host B. The switch forwards the ARP request out all ports except the port the host A is connected to. Host B receives the ARP request and responds with its MAC address. Host B also learns the MAC address of host A (because host A sent its MAC address in the ARP request). The switch learns which MAC addresses are associated with which port. For example, because host B responded with the ARP reply that included its MAC address, the switch knows the MAC address of host B and stores that address in its MAC address table. The same is with host A, the switch knows the MAC address of the host A because of the ARP request.

Now, when host A sends a packet to host B, the switch looks up in its MAC address table and forwards the frame only out Fa0/1 port, the port on which host B is connected. Other hosts on the network will not be involved in the communication:



You can display the MAC address table of the switch by using the show mac-address-table command:



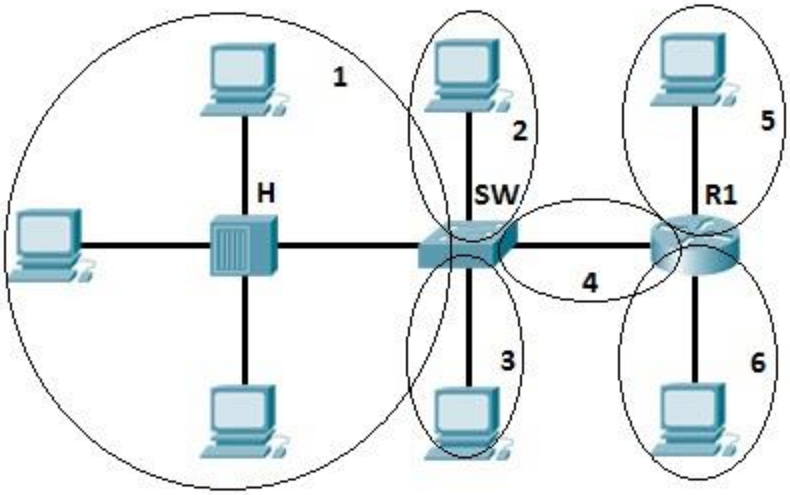
```
SW#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0002.1601.455d    DYNAMIC     Fa0/1
   1    0060.706a.78c1    DYNAMIC     Fa0/3
```

Collision & broadcast domain

Collision domain

A collision domain is, as the name implies, the part of a network where packet collisions can occur. A collision occurs when two devices send a packet at the same time on the shared network segment. The packets collide and both devices must send the packets again, which reduces network efficiency. Collisions are often in a hub environment, because each port on a hub is in the same collision domain. By contrast, each port on a bridge, a switch or a router is in a separate collision domain.

The following example illustrates collision domains:



We have 6 collision domains in the example above.

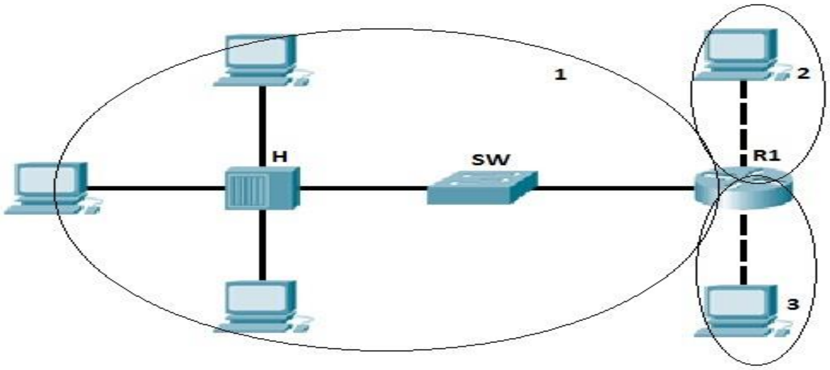
NOTE

Remember, each port on a hub is in the same collision domain. Each port on a bridge, a switch or router is in a separate collision domain.

Broadcast domain

A broadcast domain is the domain in which a broadcast is forwarded. A broadcast domain contains all devices that can reach each other at the data link layer (OSI layer 2) by using broadcast. All ports on a hub or a switch are by default in the same broadcast domain. All ports on a router are in the different broadcast domains and routers don't forward broadcasts from one broadcast domain to another.

The following example clarifies the concept:

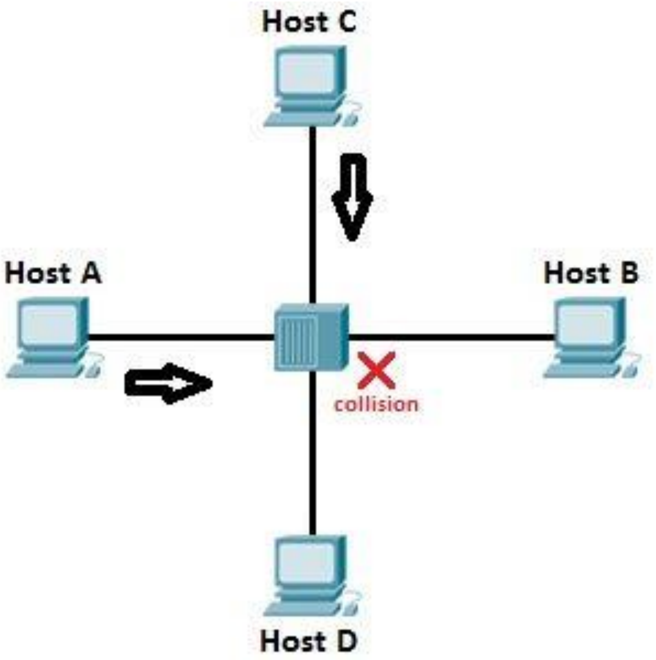


In the picture above we have three broadcast domains, since all ports on a hub or a switch are in the same broadcast domain, and all ports on a router are in a different broadcast domain.

CSMA/CD

CSMA/CD (Carrier Sense Multiple Access with Collision Detection) helps hosts to decide when to send packets on a shared network segment and how to detect collisions if they occur. For example, in a hub network, two devices can send packets at the same time. This can cause a collision. CSMA/CD enables devices to "sense" the wire to ensure that no other device is currently transmitting packets. But, if two devices "sense" that the wire is clear and send packets at the same time, a collision can occur. If the collision occur, packets have to be resend after a random period of time.

Consider the following example:



In the topology above we have a hub network. Host A is trying to communicate with host B. Host A "senses" the wire and decides to send packets. But, in the same time, host C sends its packets to host D and the collision occurs. The sending devices (host A and host C) detect the collision and resend the packet after a random period of time.

NOTE

Since switches are now commonly used in networks instead of hubs, CSMA/CD is not really used anymore. Each port on a switch usually operate in a full duplex mode and there are no packet collisions in a full duplex mode.

CHAPTER 12
WHAT IS A VLAN?

We Will Cover These Topics in This Chapter

- ***What is a VLAN***
- ***Configuring VLANs***
- ***Configuring access & trunk ports***
- ***Frame tagging***
- ***IEEE 802***
- ***Inter-Switch Link (ISL)***
- ***What is VTP***
- ***VTP modes***
- ***VTP configuration***
- ***What is STP***
- ***How STP works***

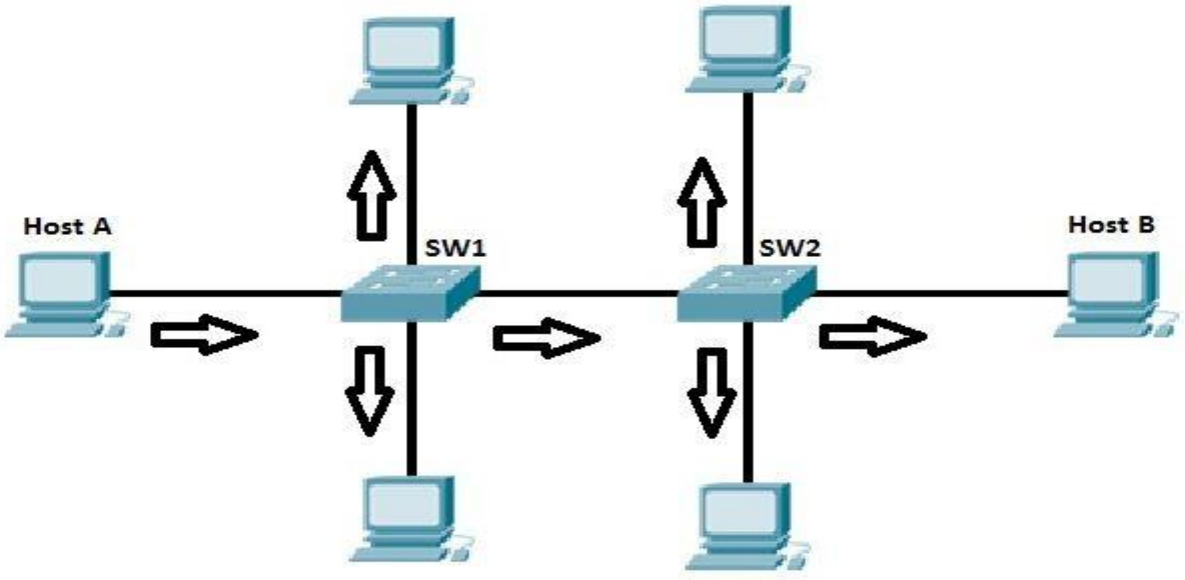
What is a VLAN?

VLANs (Virtual LANs) are logical grouping of devices in the same broadcast domain. VLANs are usually configured on switches by placing some interfaces into one broadcast domain and some interfaces into another. VLANs can be spread across multiple switches, with each VLAN being treated as its own subnet or broadcast domain. This means that frames broadcasted onto the network will be switched only between the ports within the same VLAN.

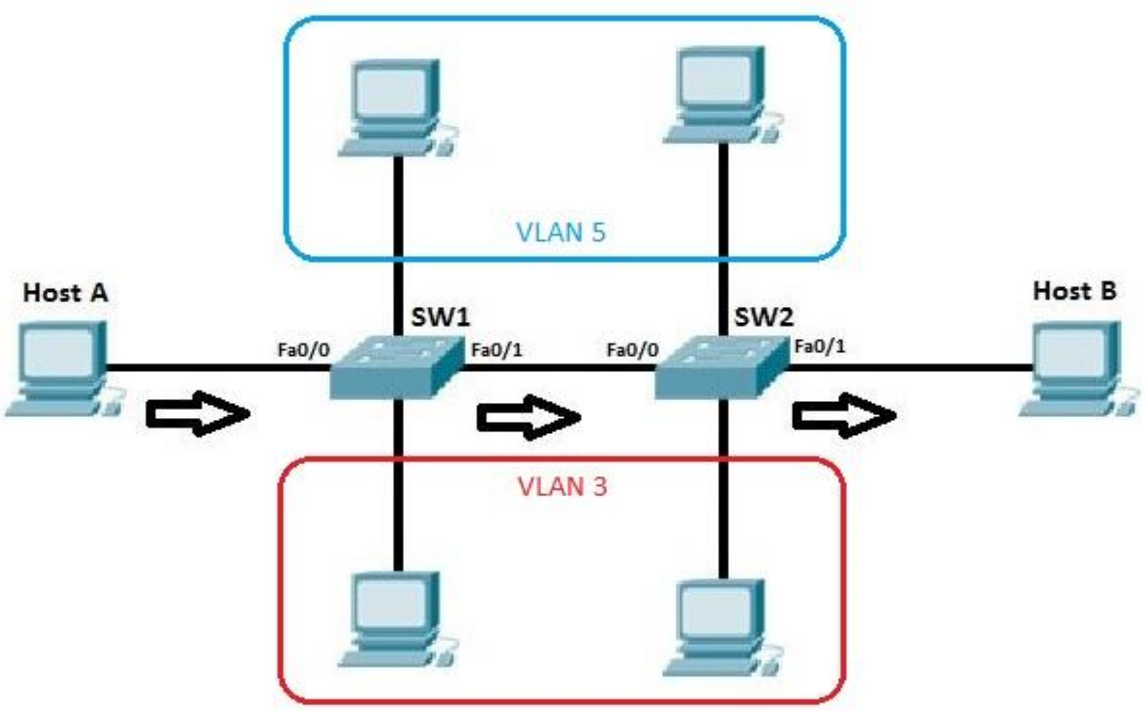
A VLAN acts like a physical LAN, but it allows hosts to be grouped together in the same broadcast domain even if they are not connected to the same switch. Here are the main reasons why you should use VLANs in your network:

- *VLANs increase the number of broadcast domains while decreasing their size.*
- *VLANs reduce security risks by reducing the number of hosts that receive copies of frames that the switches flood.*
- *you can keep hosts that hold sensitive data on a separate VLAN to improve security.*
- *you can create more flexible network designs that group users by department instead of by physical location.*
- *network changes are achieved with ease by just configuring a port into the appropriate VLAN.*

The following topology shows a network with all hosts inside the same VLAN:



Without VLANs, a broadcast sent from host A would reach all devices on the network. By placing interfaces Fa0/0 and Fa0/1 on both switches into a separate VLAN, a broadcast from host A would reach only host B, since each VLAN is a separate broadcast domain and only host B is inside the same VLAN as *host A. Hosts in VLAN 3 and VLAN 5 will not even be aware that the communication took place. This is shown in the picture below:*



NOTE

To reach hosts in another VLAN, a router is needed.

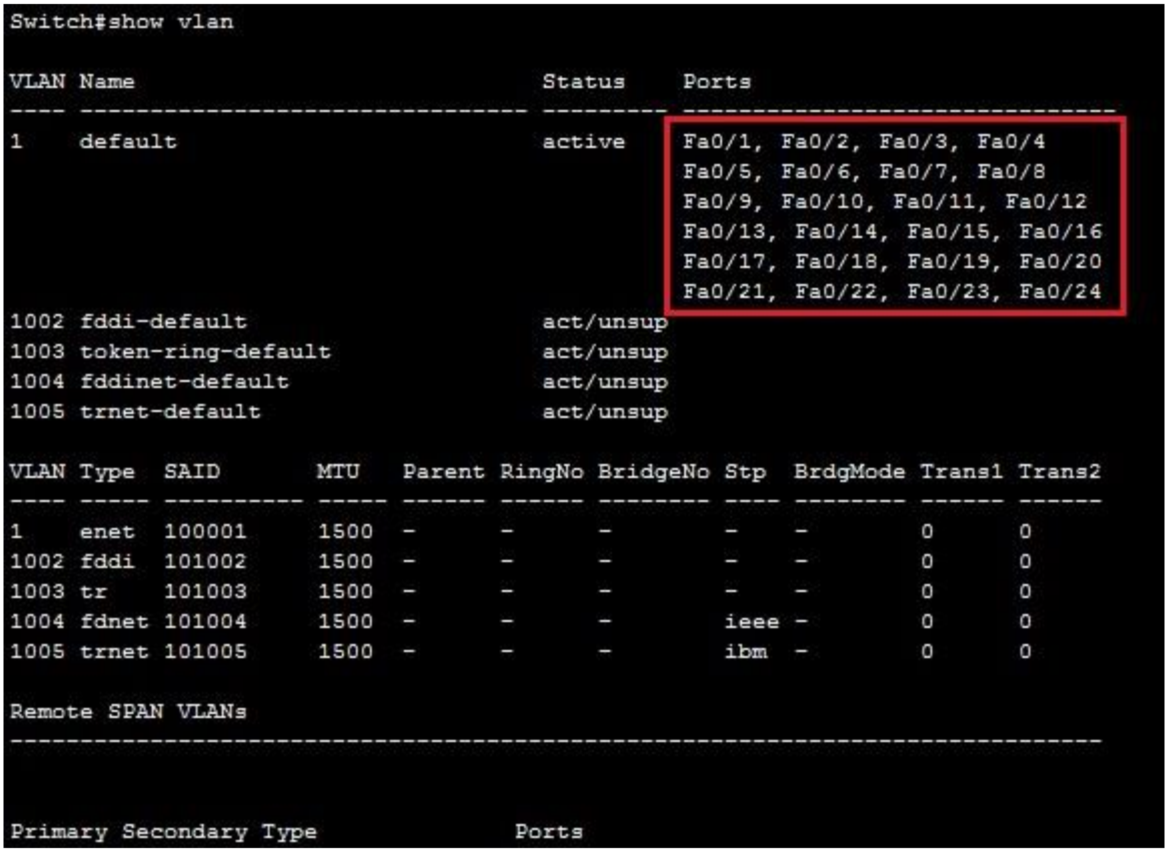
Access & trunk ports

Each port on a switch can be configured as either an access or a trunk port. An access port is a port that can be assigned to a single VLAN. This type of interface is configured on switch ports that are connected to devices with a normal network card, for example a host on a network. A trunk interface is an interface that is connected to another switch. This type of interface can carry traffic of multiple VLANs.

In the example network pictured above, the link between SW1 and SW2 would be configured as a trunk interface. All other switch ports connect to end user devices, so they would need to be configured as access ports.

Configuring VLANs

By default, all ports on a switch are in the VLAN 1. We can verify that by typing the show vlan command from the IOS enable mode of a switch:



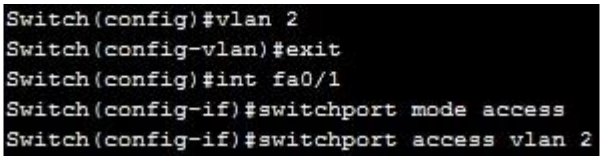
```
Switch#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

Remote SPAN VLANs
------------------------------------------------------------------------------


Primary Secondary Type              Ports
```

In the picture above, you can see that all of the 24 ports of the switch are in the same VLAN, namely VLAN 1.

Two steps are required to create a VLAN and assign a switch port to the VLAN:

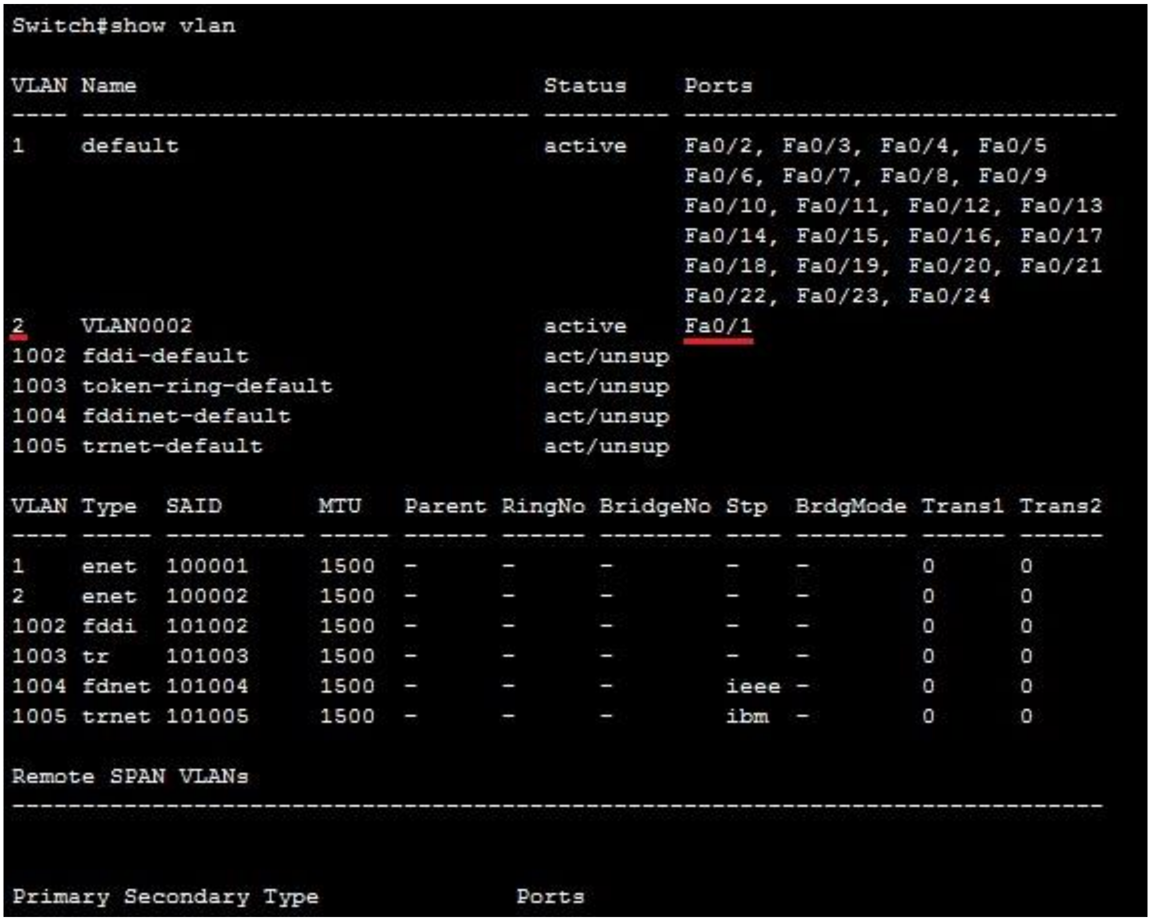
1. *create a vlan using the vlan NUMBER global mode command*
2. *assigning a port to the VLAN by using two interface subcommands. The first command is the switchport mode access command. This command specifies that the interface is an access interface. The second command is the switchport access vlan NUMBER command. This command assigns the interface to a VLAN.*

Here is an example of assigning the VLAN 2 to the interface:



```
Switch(config)#vlan 2
Switch(config-vlan)#exit
Switch(config)#int fa0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
```

The first command (vlan 2) created the VLAN 2. We've then entered the Fa0/1 sub interface mode and configured the interface as an access interface that belongs to VLAN 2. To verify this, we can again use the show vlan command:



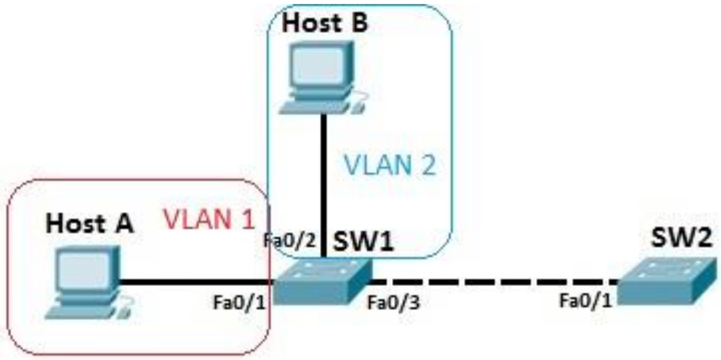
```
Switch#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                Fa0/22, Fa0/23, Fa0/24
2    VLAN0002                         active    Fa0/1
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
2    enet  100002     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

Remote SPAN VLANs
------------------------------------------------------------------------------


Primary Secondary Type              Ports
```

Configuring access & trunk ports

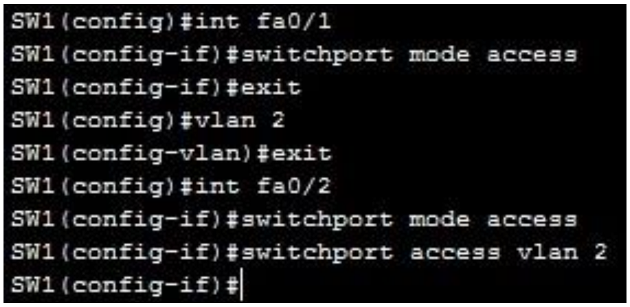
To configure an interface to be an access interface, the switchport mode access interface command is used. This type of interface can be assigned only to a single VLAN.

To configure a trunk interface, the switchport mode trunk interface command is used. This type of interface can carry traffic of multiple VLANs.

An example will help you understand the concept.



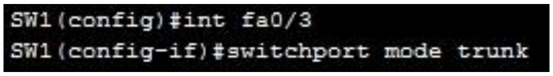
Host A and host B are in different VLANs, VLAN 1 and VLAN 2. These ports need to be configured as access ports and assigned to their respective VLANs by using the following sequence of commands:



```
SW1(config)#int fa0/1
SW1(config-if)#switchport mode access
SW1(config-if)#exit
SW1(config)#vlan 2
SW1(config-vlan)#exit
SW1(config)#int fa0/2
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 2
SW1(config-if)#
```

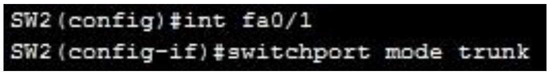
Because the link between SW1 and SW2 needs to carry traffic of multiple VLANs, it needs to be configured as a trunk interface. This is done by using the following commands on both SW1 and SW2:

On SW1:



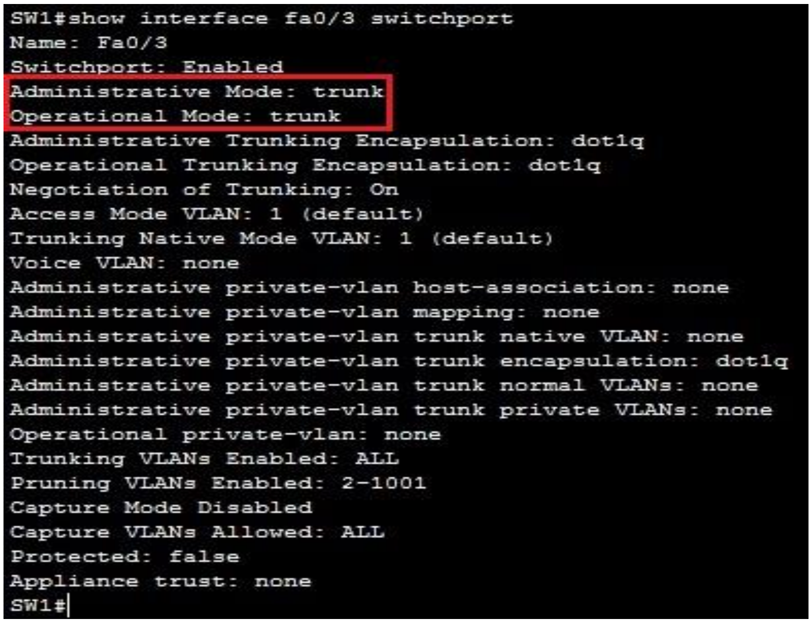
```
SW1(config)#int fa0/3
SW1(config-if)#switchport mode trunk
```

On SW2:



```
SW2(config)#int fa0/1
SW2(config-if)#switchport mode trunk
```

Now the link between SW1 and SW2 can carry traffic from both the VLAN1 and VLAN2. You can verify that an interface is indeed a trunk interface by using the show interface Fa0/3 switchport command on SW1:



```
SW1#show interface fa0/3 switchport
Name: Fa0/3
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
Appliance trust: none
SW1#
```

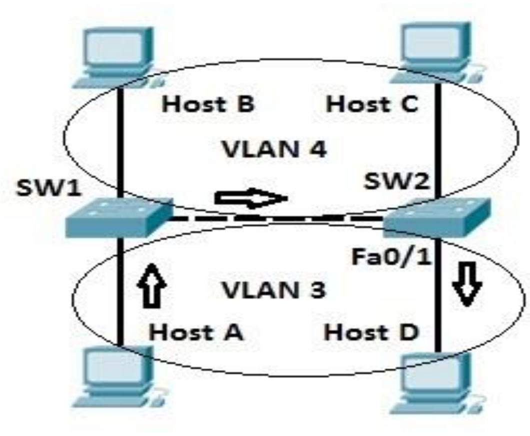
NOTE

VLAN 1 doesn’t have to be created, it exists by default. Also, by default, all ports are in the VLAN 1, so Fa0/1 doesn’t need the switchport access vlan 1 command.

Frame tagging

To identify the VLAN a packet is belonging to, switches use tagging to assign a numerical value to each frame in a network with multiple VLANs. This is done to ensure that switches know out which ports to forward frames.

For example, consider the following network topology.



There are two VLANs in the topology pictured above, namely VLAN 3 and VLAN 4. Host A sends a broadcast packet to switch SW1. Switch SW1 receives the packet, tags the packet with the VLAN ID of 3 and sends it to SW2. SW2 receives the packet, looks up at the VLAN ID, and forwards the packet only out the port Fa0/1, since only that port is in VLAN 3. Host B and host C will not receive the packet because they are in different VLAN than host A.

NOTE

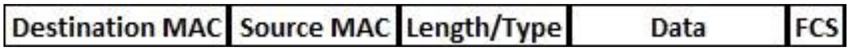
When forwarding a tagged frame to a host device a switch will remove the VLAN tag, since host devices don’t understand tagging and would drop the packet.

IEEE 802.1Q

IEEE 802.1Q is one of the VLAN tagging protocols supported by Cisco switches. This standard was created by the Institute of Electrical and Electronics Engineers (IEEE), so it an open standard and can be used on non-Cisco switches.

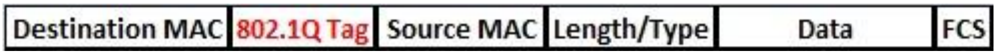
To identify to which VLAN a frame belongs to, a field is inserted into the frame’s header.

Original frame:



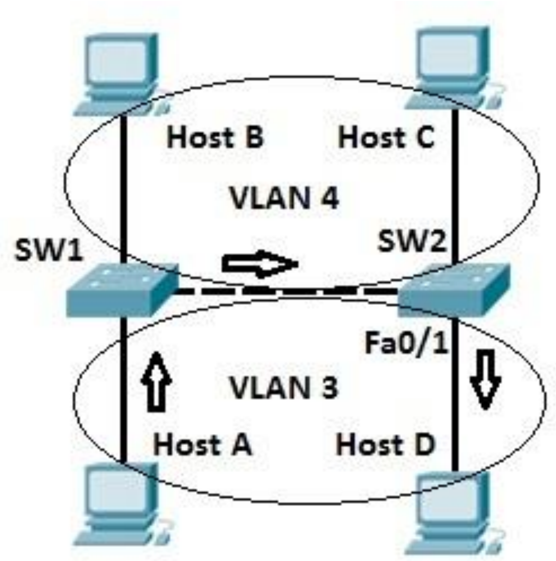
Destination MAC	Source MAC	Length/Type	Data	FCS

802.1Q frame:



Destination MAC	802.1Q Tag	Source MAC	Length/Type	Data	FCS

An example will attempt to clarify the concept. Let's say that we have a network of 2 switches and 4 hosts. Hosts A and host D are in VLAN 2, while hosts B and C are in VLAN 3.



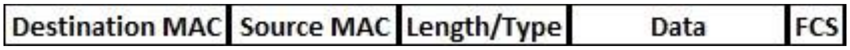
On the segment between two switches, a process called VLAN trunking is used. Let's say that host A sends a broadcast frame. SW1 "tags" the frame by inserting the VLAN ID in the header of the frame before sending the frame to SW2. SW2 receives the frame and knows that the frame belongs to VLAN 3, so it sends the frame only to host D, since that host is in VLAN 3.

Inter-Switch Link (ISL)

Another VLAN tagging protocol is Inter-Switch Link (ISL). This protocol is Cisco proprietary, which means that, unlike 802.1Q, it can be used only between Cisco switches. It is considered to be deprecated, and newer Cisco switches don't even support it.

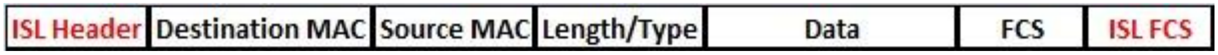
ISL works by encapsulating a frame in an ISL header and trailer. The encapsulated frame remains unchanged. The VLAN ID is included in the ISL header.

Original frame:



Destination MAC	Source MAC	Length/Type	Data	FCS

ISL frame:



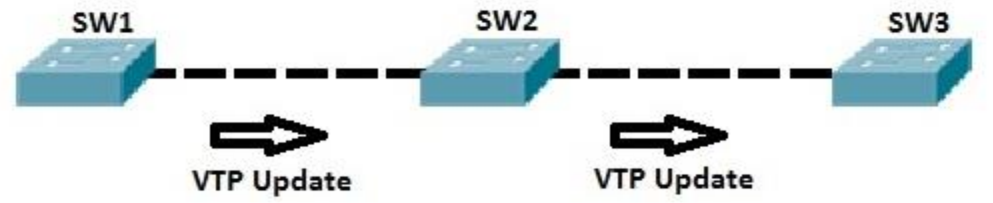
ISL Header	Destination MAC	Source MAC	Length/Type	Data	FCS	ISL FCS

What is VTP?

VTP (VLAN Trunking Protocol) is a Cisco proprietary protocol used by Cisco switches to exchange VLAN information. With VTP, you can synchronize VLAN information (like VLAN ID or VLAN name) with switches inside the same VTP domain.

To better understand the true value of VTP, consider an example network with 100 switches. Without VTP, if you want to create a VLAN on each switch, you would have to manually enter VLAN configuration commands on each switch! VTP enables you to create the VLAN only on a single switch. That switch can then propagate information about that VLAN to each switch on a network and cause other switches to create that VLAN too. Likewise, if you want to delete a VLAN, you only need to delete it on one switch, and the change is automatically propagated to every other switch inside the same VTP domain.

The following network topology explains the concept more thoroughly.



On SW1, we have created a new VLAN. SW1 sends a VTP update to SW2, which in turn sends its VTP update to SW3. Now all three switches have the same VLAN created.

NOTE

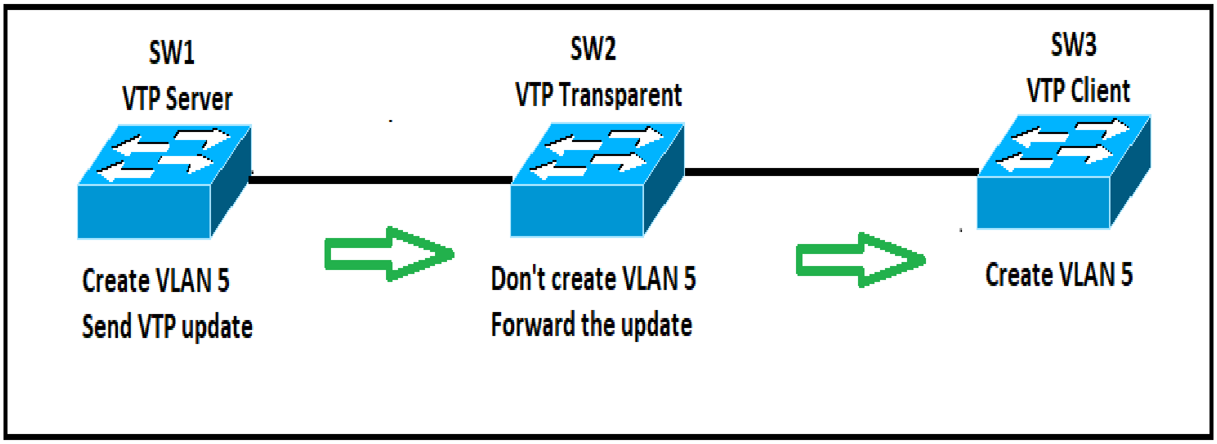
VTP does not advertise information about which switch ports are assigned to which VLAN.

VTP modes

Each switch can use one of three different VTP modes:

- *VTP client mode – a switch using this mode can't change its VLAN configuration. That means that a VTP client switch can't create or delete VLANs. Received VTP updates are processed and forwarded.*
- *VTP server mode – a switch using this mode can create and delete VLANs. A VTP server switch will propagate VLAN changes. This is the default mode for Cisco switches.*
- *VTP transparent mode – a switch using this mode doesn't share its VLAN database, but it forwards received VTP advertisements. You can create and delete VLANs on a VTP transparent switch, but the changes are not sent to other switches.*

Consider the following example:



We have a simple network of three switches. SW1 is configured as VTP server. After the VLAN 5 is created on SW1, this switch will notify the connected switch (SW2) about the created VLAN. SW2 will receive the update but, since it uses the VTP transparent mode, it will not create this VLAN in its configuration. However, it will forward the VTP update to SW3. Since SW3 is configured as VTP client, it will process the update and create VLAN 5.

VTP configuration

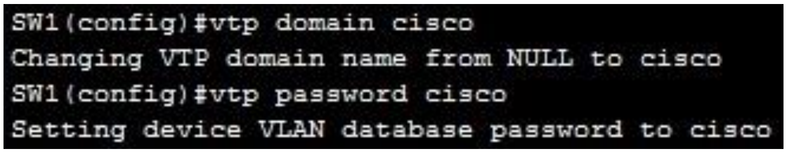
In a typical network, some switches are configured as VTP servers and other switches are configured as VTP clients. A VLAN created on a VTP server switch is automatically advertised to all switches inside the same VTP domain. A VTP domain is simply the collection of switches with the same VTP domain name configured.

To exchange VTP messages, five requirements must be met:

1. a switch has to be configured as either a VTP server or VTP client
2. the VTP domain name has to be the same on both switches
3. if present, the VTP domain password has to be the same
4. VTP versions have to match
5. the link between the switches has to be a trunk link

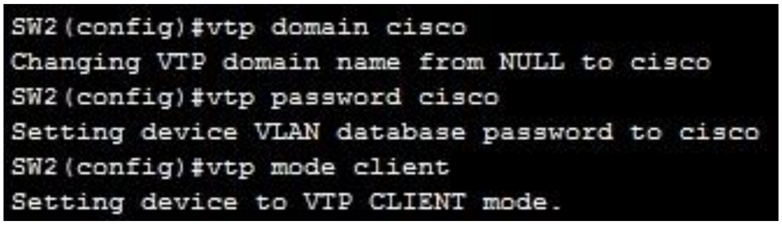
Consider the following simple example. Switches SW1 and SW2 are connected via trunk link. We will configure SW1 to serve as a VTP server and SW2 to serve as a VTP client.

First, we configure SW1:



```
SW1(config)#vtp domain cisco
Changing VTP domain name from NULL to cisco
SW1(config)#vtp password cisco
Setting device VLAN database password to cisco
```

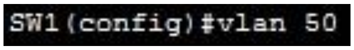
Next, we configure SW2:



```
SW2(config)#vtp domain cisco
Changing VTP domain name from NULL to cisco
SW2(config)#vtp password cisco
Setting device VLAN database password to cisco
SW2(config)#vtp mode client
Setting device to VTP CLIENT mode.
```

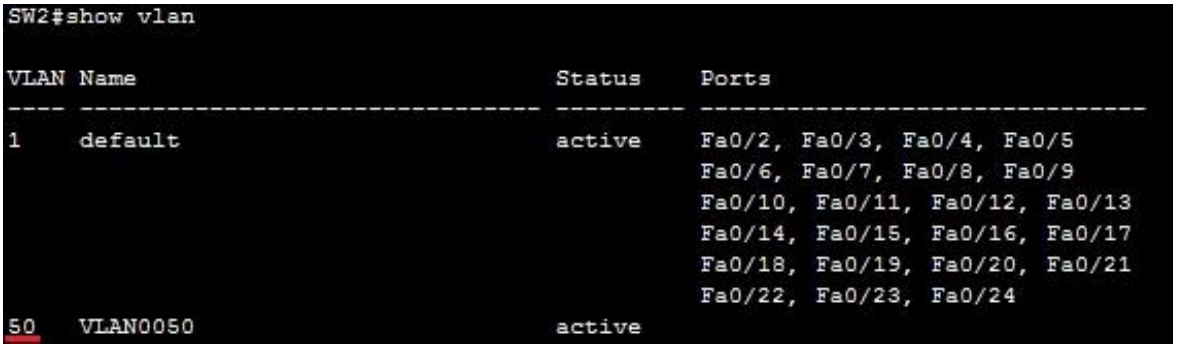
Now, we will create VLAN 50 on SW1. The information about this VLAN will automatically be propagated to SW2. SW2 should also create that VLAN.

On SW1, we will create the new VLAN:



```
SW1(config)#vlan 50
```

VTP forces SW2 to create the same VLAN:



```
SW2#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                Fa0/22, Fa0/23, Fa0/24
50   VLAN0050                         active
```

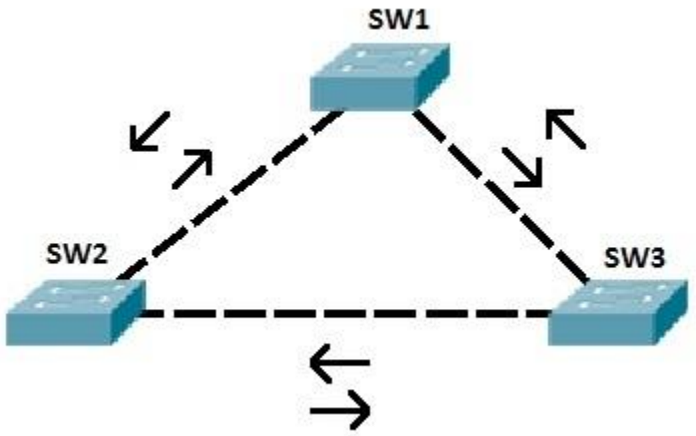
NOTE

Because the VTP server mode is the default VTP mode, we didn't need to specify the VTP mode on SW1.

What is STP?

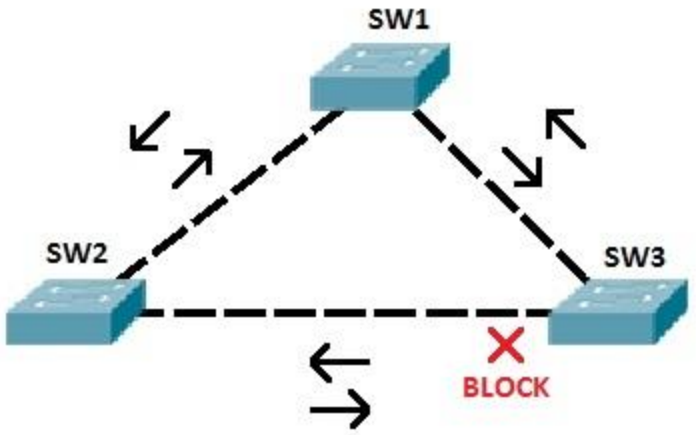
Spanning Tree Protocol (STP) is a network protocol designed to prevent layer 2 loops. It is standardized as IEEE 802.D protocol. STP blocks some ports on switches with redundant links to prevent broadcast storms and ensure loop-free topology. With STP in place, you can have redundant links between switches to provide redundancy.

To better understand the importance of STP and how to use STP to prevent broadcast storms on a network with redundant links, consider the following example.



SW2 sends a broadcast frame to SW1 and SW3. Both switches receive the frame and forward the frame out every port, except the port the frame was received on. SW1 sends the frame to SW3. SW3 receives the frame, and sends the frame back to SW2. SW2 then again forwards the frame to SW1! The same thing also happens in the opposite direction. Without STP in place, these frames would loop forever. STP prevents loops by placing one of the switch ports into blocking state.

So, our topology above could look like this:



In the topology above, STP has placed one port on SW3 into the blocking state. That way, if SW3 receives a broadcast frame from SW1, it will not forward it out the port connected to SW2.

NOTE

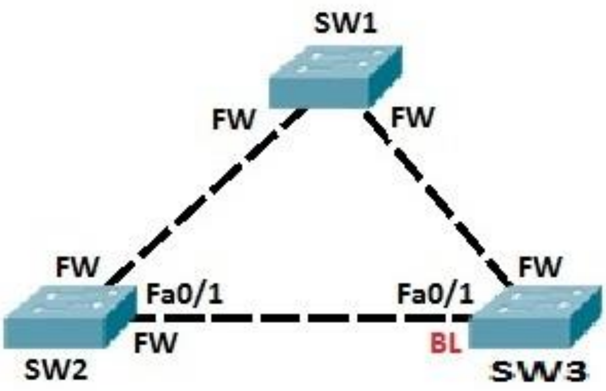
STP enables layer 2 redundancy. In the example above, if the link between SW3 and SW1 fails, STP would converge and unblock the port on SW3.

How STP works

STP uses the Spanning-Tree Algorithm (SPA) to create a topology database of the network. To prevent loops, SPA places some interfaces into forwarding state and some interfaces into blocking state. How does STP decides in which state to port will be placed? A couple of criteria exist:

1. all switches in a network elect a root bridge (switch). All working interfaces on the root bridge are placed in forwarding state. The switch with the lowest switch ID will become the root bridge.
2. all other switches, called "non root bridges", determine the best path to get to the root bridge. The port used to reach the root bridge (root port) is placed in forwarding state.
3. on the shared Ethernet segments, the switch with the lowest cost to reach the root bridge is placed into forwarding state.
4. all other interfaces are placed in blocking state and will not forward frames.

An example will help you understand the concept.



Let's say that SW1 advertised the lowest switch ID and is elected as the root bridge. All ports on SW1 are placed into forwarding state. SW2 and SW3 choose ports with the lowest cost to reach the root bridge to be the root ports. These ports are also placed into forwarding state. On the shared Ethernet segment between SW2 and SW3, port Fa0/1 on SW2 has the lowest cost to reach the root bridge. This port is placed into forwarding state. To prevent loops, port Fa0/1 on SW3 is placed into blocking state.

NOTE

A switch with the lowest switch ID will become the root bridge. A switch ID consists of two components: the switch's priority (by default 32,768 on Cisco switches) and the switch's MAC address.

BPDU

BPDUs (Bridge Protocol Data Units) are used by switches to share information with each other and learn the topology of the network. With BPDUs, the loops in the network are detected. BPDUs are compared and used to elect a root switch. Hello BPDUs are the most common messages. They list the switch ID of the sender and the root bridge ID.

CHAPTER 12
WHAT IS A VLAN?

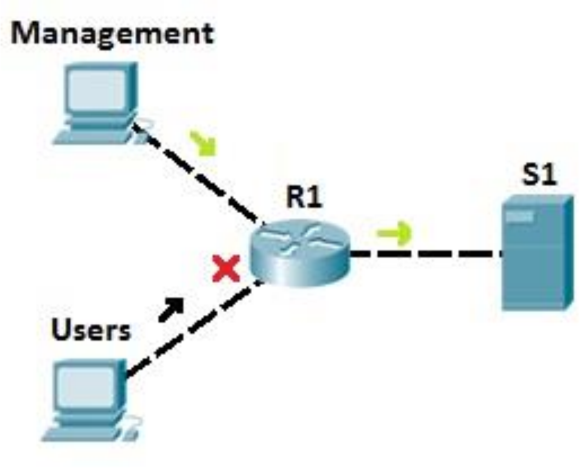
We Will Cover These Topics in This Chapter

- *What are ACLs*
- *Types of ACLs*
- *Configuring standard ACLs*
- *Configuring extended ACLs*

What are ACLs?

ACLs are a set of rules used most commonly to filter network traffic. They are used on network devices with packet filtering compatibilities (e.g. routers or firewalls). ACLs are applied on the interface basis to packets leaving or entering an interface.

For example, on how ACLs are used, consider the following network topology:



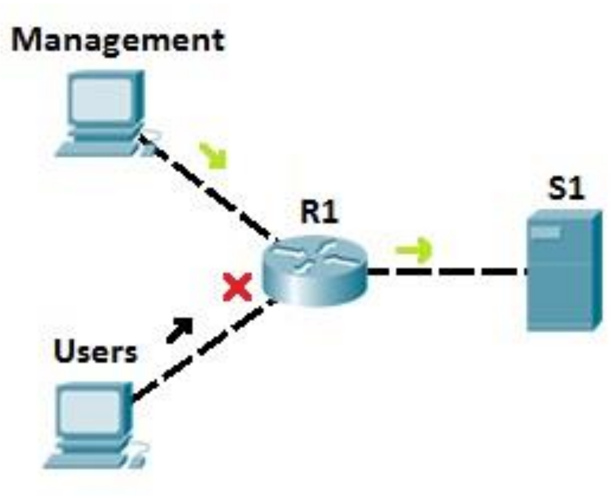
Let's say that server S1 holds some important documents that need to be available only to the company's management. We could configure an access list on R1 to enable access to S1 only to users from the management network. All other traffic going to S1 will be blocked. This way, we can ensure that only authorized user can access the sensitive files on S1.

Types of ACLs

There are two types of access lists:

1. standard access lists – with standard access lists, you can filter only on the source IP address of a packet. These types of access list are not as powerful as extended access lists, but they are less processor intensive for the router.

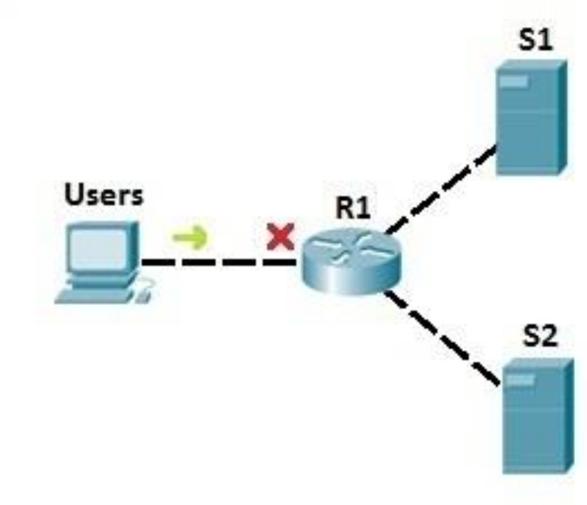
The following example describes the way in which standard access lists can be used.



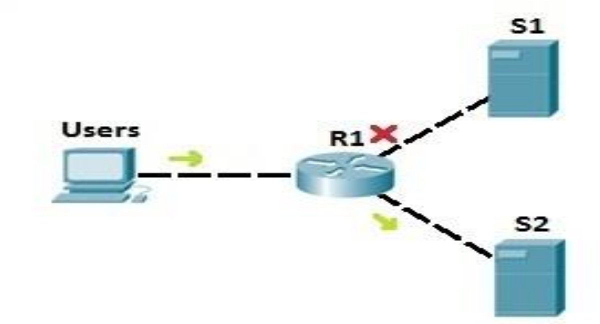
Let's say that server S1 holds some important documents that need to be available only to company's management. We could configure an access list on R1 to enable access to S1 only to users from the management network. All other traffic going to S1 will be blocked. This way, we can ensure that only authorized user can access sensitive files on S1.

2. extended access lists – with extended access lists, you can be more precise in your filtering. You can evaluate source and destination IP addresses, type of layer 3 protocol, source and destination port, etc. Extended access lists are more complex to configure and consume more CPU time than the standard access lists, but they allow a much more granular level of control.

To demonstrate the usefulness of extended ACLs, we will use the following example.



In the example network above, we have used the standard access list to prevent all users to access server S1. But, with that configuration, we have also disable access to S2! To be more specific, we can use extended access lists. Let's say that we need to prevent users from accessing server S1. We could place an extended access list on R1 to prevent users only from accessing S1 *(we would use an access list to filter the traffic according to the destination IP address). That way, no other traffic is forbidden, and users can still access the other server, S2:*



Configuring standard ACLs

To create an standard access list on a Cisco router, the following command is used from the router's global configuration mode:

```
R1(config)# access-list ACL_NUMBER permit|deny IP_ADDRESS WILDCARD_MASK
```

NOTE

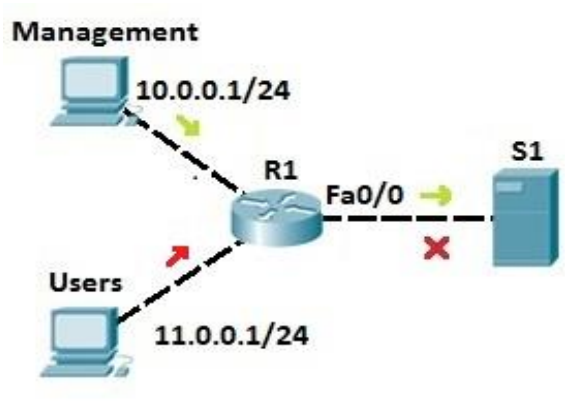
ACL number for the standard ACLs has to be between 1–99 and 1300–1999.

You can also use the host keyword to specify the host you want to permit or deny:

```
R1(config)# access-list ACL_NUMBER permit|deny host IP_ADDRESS
```

Once the access list is created, it needs to be applied to an interface. You do that by using the ip access-group ACL_NUMBER in|out interface subcommand. in and out keywords specify in which direction you are activating the ACL. in means that ACL is applied to the traffic coming into the interface, while the out keyword means that the ACL is applied to the traffic leaving the interface.

Consider the following network topology:



We want to allow traffic from the management LAN to the server S1. First, we need to write an ACL to permit traffic from LAN 10.0.0.0/24 to S1. We can use the following command on R1:

```
R1(config)#access-list 1 permit 10.0.0.0 0.0.0.255
```

The command above permits traffic from all IP addresses that begin with 10.0.0. We could also target the specific host by using the host keyword:

```
R1(config)#access-list 1 permit host 10.0.0.1
```

The command above permits traffic only from the host with the IP address of 10.0.0.1.

Next, we will deny traffic from the Users LAN (11.0.0.0/24):

```
R1(config)#access-list 1 deny 11.0.0.0 0.0.0.255
```

Next, we need to apply the access list to an interface. It is recommended to place the standard access lists as close to the destination as possible. In our case, this is the Fa0/0 interface on R1. Since we want to evaluate all packets trying to exit out Fa0/0, we will specify the outbound direction with the out keyword:

```
R1(config-if)#ip access-group 1 out
```

NOTE
At the end of each ACL there is an implicit deny all statement. This means that all traffic not specified in earlier ACL statements will be forbidden, so the second ACL statement (access-list 1 deny 11.0.0.0 0.0.0.255) wasn’t even necessary.

Configuring extended ACLs

To be more precise when matching a certain network traffic, extended access lists are used. With extended access lists, you can match more information, such as:

- *source and destination IP address*
- *type of TCP/IP protocol (TCP, UDP, IP…)*
- *source and destination port numbers*

Two steps are required to configure extended access lists:

1. configure extended access lists using the following command:

```
(config) access list NUMBER permit|deny IP_PROTOCOL SOURCE_ADDRESS WILDCARD_MASK
[PROTOCOL_INFORMATION] DESTINATION_ADDRESS WILDCARD_MASK PROTOCOL_INFORMATION
```

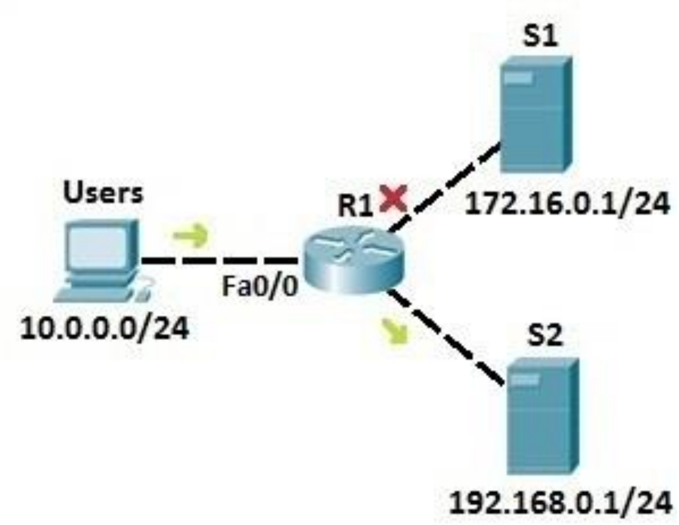
2. apply an access list to an interface using the following command:

```
(config) ip access-group ACL_NUMBER in | out
```

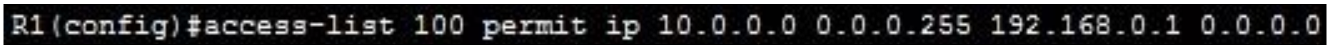
NOTE

Extended access lists numbers are in ranges from 100 to 199 and from 2000 to 2699. You should always place extended ACLs as close to the source as possible.

To better understand the usefulness of extended access lists, consider the following example.

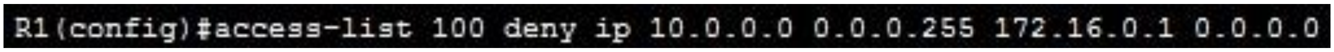


We want Users from the network 10.0.0.0/24 to be able to access the server S2 (IP address 192.168.0.1) and prevent them from accessing server S1 (IP address 172.16.0.1/24). First, we need to configure an access list to permit Users the access to server S2:



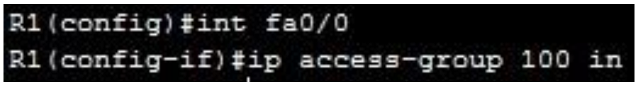
```
R1(config)#access-list 100 permit ip 10.0.0.0 0.0.0.255 192.168.0.1 0.0.0.0
```

Next, we need to deny Users the right to access S1 by using the deny statement:



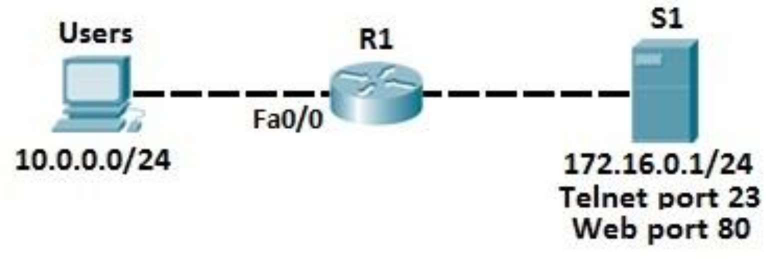
```
R1(config)#access-list 100 deny ip 10.0.0.0 0.0.0.255 172.16.0.1 0.0.0.0
```

Finally, we need to apply the access list to the interface on R1:



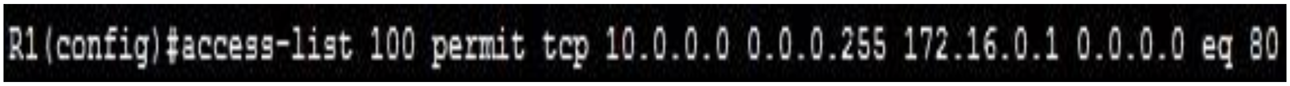
```
R1(config)#int fa0/0
R1(config-if)#ip access-group 100 in
```

Here is another example of using extended access lists. In this example we will use extended ACLs to filter traffic by the port used.



Again, we have the Users network (10.0.0.0/24). On the right side, we have a server that serves as a web server, listening on port 80. We need to permit Users to access web sites on S1, but we also need to deny other type of access, for example the Telnet access.

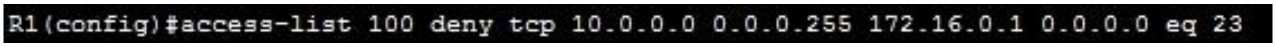
First, we need to allow traffic from Users network to the web server port of 80. We can do that by using the following command:



```
R1(config)#access-list 100 permit tcp 10.0.0.0 0.0.0.255 172.16.0.1 0.0.0.0 eq 80
```

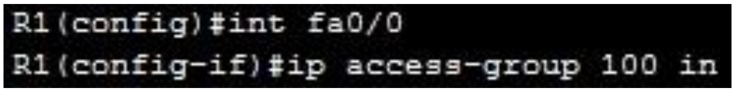
By using the TCP keyword, we can filter packets by source and destination ports. In the example above, we have permited traffic originating from the 10.0.0.0 network to the host 172.16.0.1 on the port 80. The last part of the statement, eq 80, specifies the destination port of 80.

Now we need to disable telnet traffic from the network 10.0.0.0 to 172.16.0.1. To do that, we need to create a deny statement:



```
R1(config)#access-list 100 deny tcp 10.0.0.0 0.0.0.255 172.16.0.1 0.0.0.0 eq 23
```

Next, we need to apply our access list to the interface:



```
R1(config)#int fa0/0
R1(config-if)#ip access-group 100 in
```

NOTE

Since at the end of each access list there is an explicit deny all statement, the second ACL statement wasn’t really necessary. After applying an access list, every traffic not explicitly permitted will be denied.

CHAPTER 14
WHAT IS NAT?

We Will Cover These Topics in This Chapter

- *What is NAT*
- *Static NAT*
- *Dynamic NAT*
- *Port Address Translation (PAT) configuration*

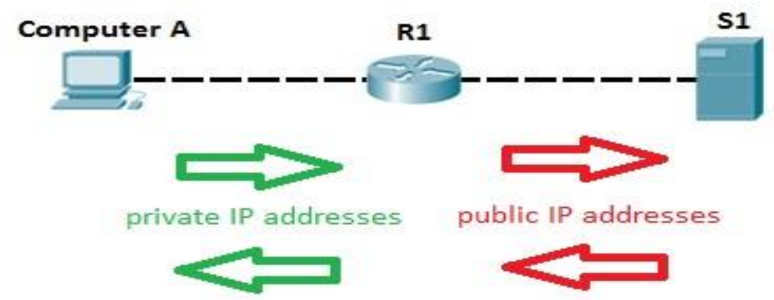
What is NAT?

NAT (Network Address Translation) is a process of changing the source and destination IP addresses and ports. Address translation reduces the need for IPv4 public addresses and hides private network address ranges. The process is usually done by routers or firewalls.

There are three types of address translation:

1. Static NAT – translates one private IP address to a public one. The public IP address is always the same.
2. Dynamic NAT – private IP addresses are mapped to the pool of public IP addresses.
3. Port Address Translation (PAT) – one public IP address is used for all internal devices, but a different port is assigned to each private IP address. Also known as NAT Overload.

An example will help you understand the concept.



Computer A request a web page from an Internet server. Because Computer A uses private IP addressing, the source address of the request has to be changed by the router because private IP addresses are not routable on the Internet. Router R1 receives the request, changes the source IP address to its public IP address and sends the packet to server S1. Server S1 receives the packet and replies to router R1. Router R1 receives the packet, changes the destination IP addresses to the private IP address of Computer A and sends the packet to Computer A.

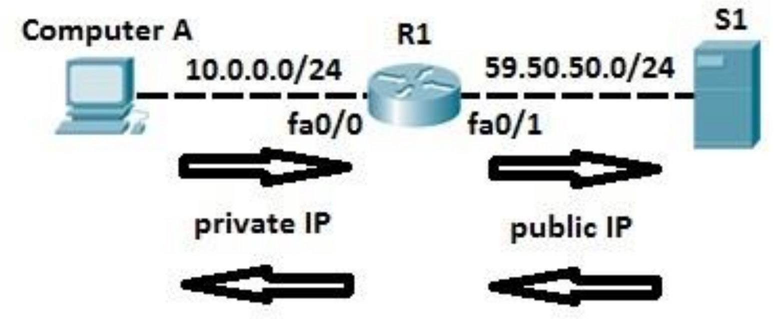
Static NAT

With static NAT, routers or firewalls translate one private IP address to a single public IP address. Each private IP address is mapped to a single public IP address. Static NAT is not often used because it requires one public IP address for each private IP address.

To configure static NAT, three steps are required:

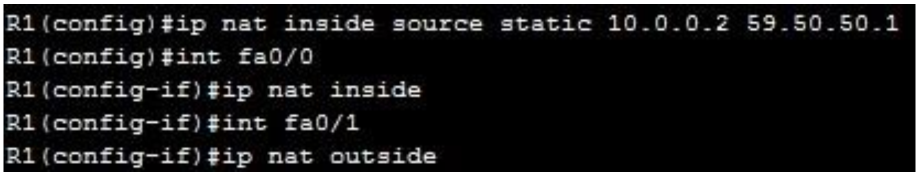
1. configure private/public IP address mapping by using the ip nat inside source static PRIVATE_IP PUBLIC_IP command
2. configure the router's inside interface using the ip nat inside command
3. configure the router's outside interface using the ip nat outside command

Here is an example.



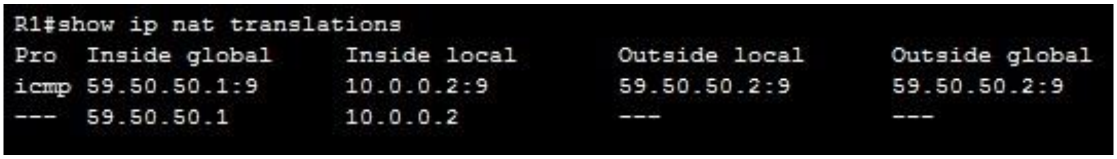
Computer A requests a web resource from S1. Computer A uses its private IP address when sending the request to router R1. Router R1 receives the request, changes the private IP address to the public one and sends the request to S1. S1 responds to R1. R1 receives the response, looks up in its NAT table and changes the destination IP address to the private IP address of Computer A.

In the example above, we need to configure static NAT. To do that, the following commands are required on R1:



```
R1(config)#ip nat inside source static 10.0.0.2 59.50.50.1
R1(config)#int fa0/0
R1(config-if)#ip nat inside
R1(config-if)#int fa0/1
R1(config-if)#ip nat outside
```

Using the commands above, we have configured a static mapping between Computer A's private IP address of 10.0.0.2 and router's R1 public IP address of 59.50.50.1. To check NAT, you can use the show ip nat translations command:



```
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 59.50.50.1:9      10.0.0.2:9         59.50.50.2:9       59.50.50.2:9
---  59.50.50.1        10.0.0.2           ---                ---
```

Dynamic NAT

With dynamic NAT, you specify two sets of addresses on your Cisco router:

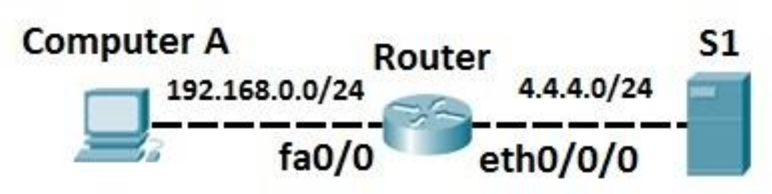
1. inside addresses that will be translated.
2. a pool of global addresses.

Unlike with static NAT, where you had to manually define a static mapping between a private and a public address, with dynamic NAT the mapping of a local address to a global address happens dynamically. This means that the router dynamically picks an address from the global address pool that is not currently assigned. It can be any address from the pool of global addresses. The dynamic entry stays in the NAT translations table as long as the traffic is exchanged. The entry times out after a period of inactivity and the global IP address can be used for new translations.

To configure dynamic NAT, the following steps are required:

1. configure the router's inside interface using the ip nat inside command
2. configure the router's outside interface using the ip nat outside command
3. configure an ACL that has a list of the inside source addresses that will be translated
4. configure the pool of global IP addresses using the ip nat pool NAME FIRST_IP_ADDRESS LAST_IP_ADDRESS netmask SUBNET_MASK command
5. enable dynamic NAT with the ip nat inside source list ACL_NUMBER pool NAME global configuration command

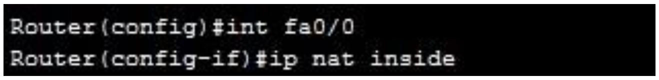
Here is an example.



Computer A requests a web resource from S1. Computer A uses its private IP address when sending the request to router R1. Router R1 receives the request, changes the private IP address to one of the available global addresses in the pool and sends the request to S1. S1 responds to R1. R1 receives the response, looks up in its NAT table and changes the destination IP address to the private IP address of Computer A.

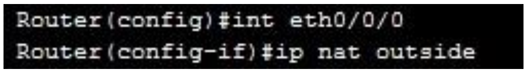
In the example above we need to configure dynamic NAT. To do that, the following commands are required on R1:

1. To configure the router's inside interface:



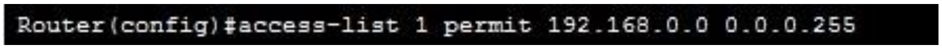
```
Router(config)#int fa0/0
Router(config-if)#ip nat inside
```

2. To configure the router's outside interface:



```
Router(config)#int eth0/0/0
Router(config-if)#ip nat outside
```

3. To configure an ACL that has a list of the inside source addresses that will be translated:

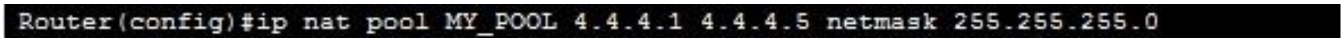


```
Router(config)#access-list 1 permit 192.168.0.0 0.0.0.255
```

NOTE

The access list configured above matches all hosts from the 192.168.0.0/24 subnet.

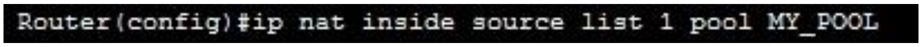
4. To configure the pool of global IP addresses:



```
Router(config)#ip nat pool MY_POOL 4.4.4.1 4.4.4.5 netmask 255.255.255.0
```

The pool configured above consists of 5 addresses: 4.4.4.1, 4.4.4.2, 4.4.4.3, 4.4.4.4, and 4.4.4.5.

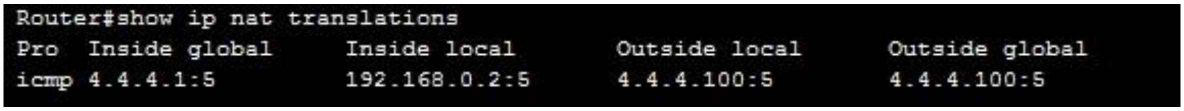
5. To enable dynamic NAT:



```
Router(config)#ip nat inside source list 1 pool MY_POOL
```

The command above instructs the router to translate all addresses specified in the access list 1 to the pool of global addresses called MY_POOL.

You can list all NAT translations using the show ip nat translations command:



```
Router#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 4.4.4.1:5         192.168.0.2:5      4.4.4.100:5        4.4.4.100:5
```

In the example above, you can see that the private IP address of Computer A (192.168.0.2) has been translated to the first available global address (4.4.4.1).

NOTE

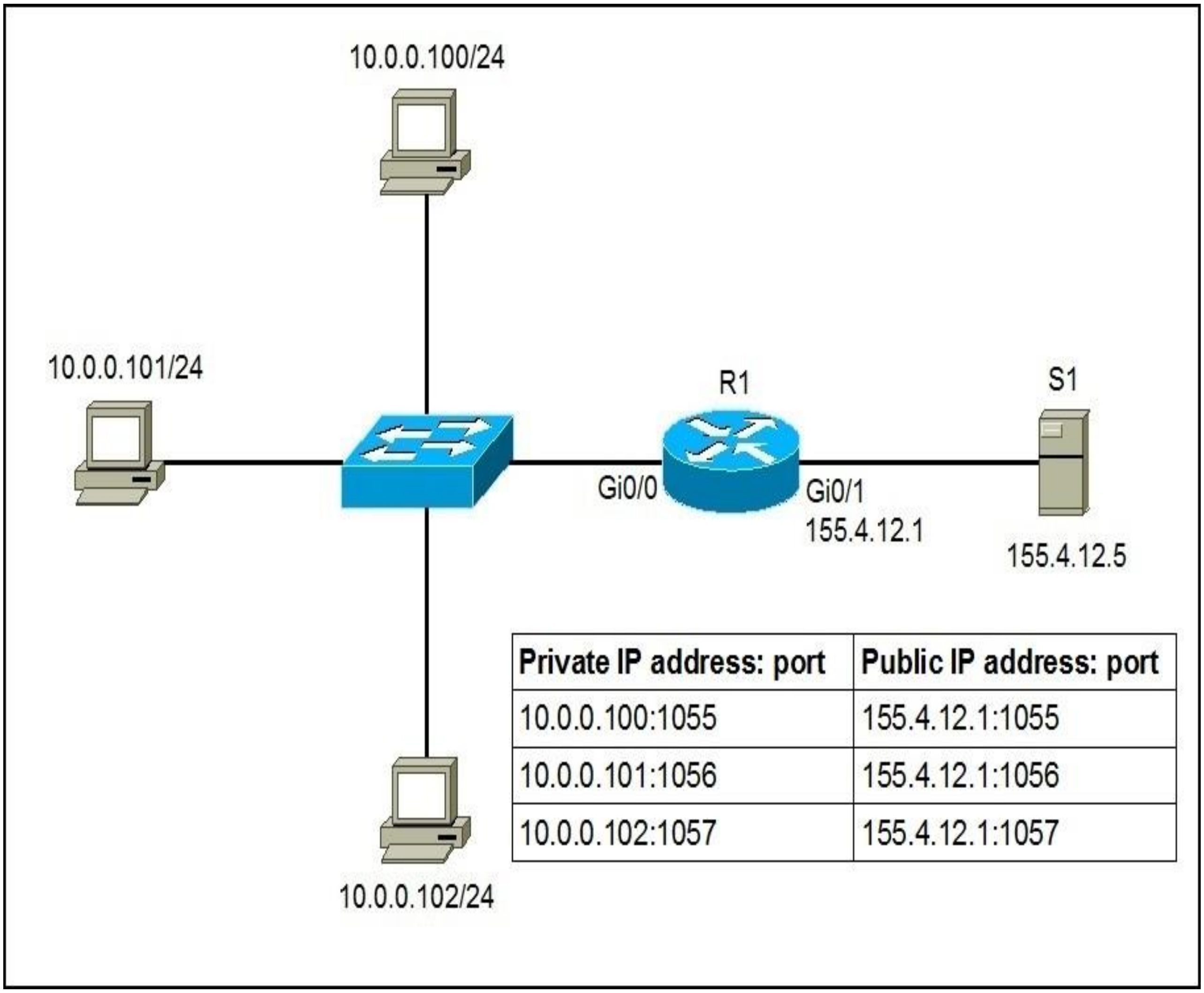
*You can remove all NAT translations from the table by using the clear ip nat translation * command.*

Port Address Translation (PAT) configuration

With Port Address Translation (PAT), a single public IP address is used for all internal private IP addresses, but a different port is assigned to each private IP address. This type of NAT is also known as NAT Overload and is the typical form of NAT used in today's networks. It is even supported by most consumer-grade routers.

PAT allows you to support many hosts with only few public IP addresses. It works by creating dynamic NAT mapping, in which a global (public) IP address and a unique port number are selected. The router keeps a NAT table entry for every unique combination of the private IP address and port, with translation to the global address and a unique port number.

We will use the following example network to explain the benefits of using PAT:



Private IP address: port	Public IP address: port
10.0.0.100:1055	155.4.12.1:1055
10.0.0.101:1056	155.4.12.1:1056
10.0.0.102:1057	155.4.12.1:1057

As you can see in the picture above, PAT uses unique source port numbers on the inside global (public) IP address to distinguish between translations. For example, if the host with the IP address of 10.0.0.101 wants to access the server S1 on the Internet, the host's private IP address will be translated by R1 to 155.4.12.1:1056 and the request will be sent to S1. S1 will respond to 155.4.12.1:1056. R1 will receive that response, look up in its NAT translation table, and forward the request to the host.

To configure PAT, the following commands are required:

- *configure the router's inside interface using the ip nat inside command.*
- *configure the router's outside interface using the ip nat outside command.*
- *configure an access list that includes a list of the inside source addresses that should be translated.*
- *enable PAT with the ip nat inside source list ACL_NUMBER interface TYPE overload global configuration command.*

Here is how we would configure PAT for the network picture above.

First, we will define the outside and inside interfaces on R1:

```
R1(config)#int Gi0/0

R1(config-if)#ip nat inside

R1(config-if)#int Gi0/1

R1(config-if)#ip nat outside
```

Next, we will define an access list that will include all private IP addresses we would like to translate:

```
R1(config-if)#access-list 1 permit 10.0.0.0 0.0.0.255
```

The access list defined above includes all IP addresses from the 10.0.0.0 – 10.0.0.255 range.

Now we need to enable NAT and refer to the ACL created in the previous step and to the interface whose IP address will be used for translations:

```
R1(config)#ip nat inside source list 1 interface Gi0/1 overload
```

To verify the NAT translations, we can use the show ip nat translations command after hosts request a web resource from S1:

```
R1#show ip nat translations

Pro Inside global Inside local Outside local Outside global

tcp 155.4.12.1:1024 10.0.0.100:1025 155.4.12.5:80 155.4.12.5:80

tcp 155.4.12.1:1025 10.0.0.101:1025 155.4.12.5:80 155.4.12.5:80

tcp 155.4.12.1:1026 10.0.0.102:1025 155.4.12.5:80 155.4.12.5:80
```

Notice that the same IP address (155.4.12.1) has been used to translate three private IP addresses (10.0.0.100, 10.0.0.101, and 10.0.0.102). The port number of the public IP address is unique for each connection. So when S1 responds to 155.4.12.1:1026, R1 look into its NAT translations table and forward the response to 10.0.0.102:1025

CHAPTER 15
WHAT IS IPV6?

We Will Cover These Topics in This Chapter

- ***What is IPv6***
- ***IPv6 address format***
- ***Types of IPv6 addresses***
- ***IPv6 unicast addresses***
- ***IPv6 unique local addresses***
- ***IPv6 link-local addresses***
- ***IPv6 address prefixes***
- ***IPv6 interface identifier***
- ***IPv6 transition options***
- ***IPv6 routing protocols***
- ***How to configure IPv6***
- ***RIPng***
- ***Differences between IPv4 and IPv6***

What is IPv6?

IPv6 is the newest version of the IP protocol. IPv6 was developed to overcome many deficiencies of IPv4, most notably the problem of IPv4 address exhaustion. Unlike IPv4, which has only about 4.3 billion (2 raised to power 32) available addresses, IPv6 allows for 3.4 × 10 raised to power 38 addresses.

IPv6 features

Here is a list of the most important features of IPv6:

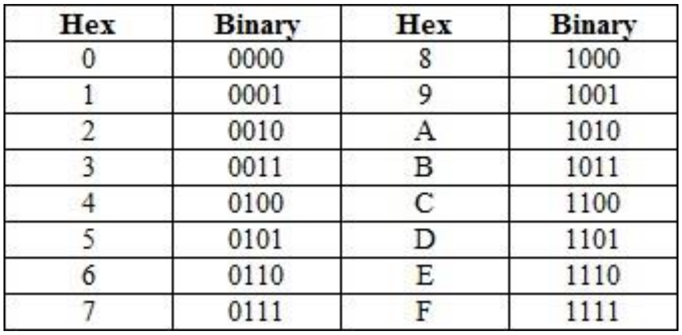
- *Large address space: IPv6 uses 128-bit addresses, which means that for each person on the Earth there are 48,000,000,000,000,000,000,000,000,000 addresses!*
- *Enhanced security: IPSec (Internet Protocol Security) is built into IPv6 as part of the protocol . This means that two devices can dynamically create a secure tunnel without user intervention.*
- *Header improvements: the packed header used in IPv6 is simpler than the one used in IPv4. The IPv6 header is not protected by a checksum so routers do not need to calculate a checksum for every packet.*
- *No need for NAT: since every device has a globally unique IPv6 address, there is no need for NAT.*
- *Stateless address autoconfiguration: IPv6 devices can automatically configure themselves with an IPv6 address.*

IPv6 address format

Unlike IPv4, which uses a dotted-decimal format with each byte ranges from 0 to 255, IPv6 uses eight groups of four hexadecimal digits separated by colons. For example, this is a valid IPv6 address:

```
2340:0023: AABA:0A01:0055:5054: 9ABC:ABB0
```

If you don't know how to convert hexadecimal number to binary, here is a table that will help you do the conversion:



Hex	Binary	Hex	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

IPv6 address shortening

The IPv6 address given above looks daunting, right? Well, there are two conventions that can help you shorten what must be typed for an IP address:

1. a leading zero can be omitted

For example, the address listed above (2340:0023:AABA:0A01:0055:5054:9ABC:ABB0) can be shortened to2340:23:AABA:A01:55:5054:9ABC:ABB0

2. successive fields of zeroes can be represented as two colons (::)

For example, 2340:0000:0000:0000:0455:0000:AAAB:1121 can be written as 2340::0455:0000:AAAB:1121

NOTE
You can shorten an address this way only for one such occurrence. The reason is obvious – if you had more than occurrence of double colon you wouldn't know how many sets of zeroes were being omitted from each part.

Here is a couple of more examples that can help you grasp the concept of IPv6 address shortening:

Long version: 1454:0045:0000:0000:4140:0141:0055:ABBB
Shortened version: 1454:45::4140:141:55:ABBB

Long version: 0000:0000:0001:AAAA:BBBC:A222:BBBA:0001
Shortened version: ::1:AAAA:BBBC:A222:BBBA:1

Types of IPv6 addresses
Three categories of IPv6 addresses exist:

- *Unicast – represents a single interface. Packets addressed to a unicast address are delivered to a single interface.*
- *Anycast – identifies one or more interfaces. For example, servers that support the same function can use the same unicast IP address. Packets sent to that IP address are forwarded to the nearest server. Anycast addresses are used for load-balancing. Known as "one-to-nearest" address.*
- *Multicast – represent a dynamic group of hosts. Packets sent to this address are delivered to many interfaces. Multicast addresses in IPv6 have a similar purpose as their counterparts in IPv4.*

NOTE

IPv6 doesn't use the broadcast method. It has been replaced with anycast and multicast addresses.

IPv6 unicast addresses

Unicast addresses represent a single interface. Packets addressed to a unicast address will be delivered to a specific network interface.

There are three types of IPv6 unicast addresses:

- *global unicast – similar to IPv4 public IP addresses. These addresses are assigned by the IANA and used on public networks. They have a prefix of 2000::/3, (all the addresses that begin with binary 001).*
- *unique local – similar to IPv4 private addresses. They are used in private networks and aren't routable on the Internet. These addresses have a prefix of FD00::/8.*
- *link local – these addresses are used for sending packets over the local subnet. Routers do not forward packets with this addresses to other subnets. IPv6 requires a link-local address to be assigned to every network interface on which the IPv6 protocol is enabled. These addresses have a prefix of FE80::/10.*

Let's describe each of the IPv6 unicast address type in more detail.

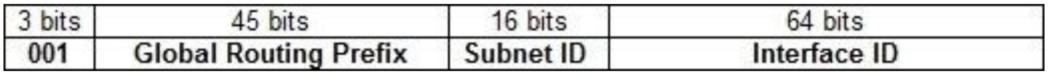
IPv6 global unicast addresses

IPv6 global addresses are similar to IPv4 public addresses. As the name implies, they are routable on the internet. Currently IANA has assigned only 2000::/3 addresses to the global pool.

A global IPv6 address consists of two parts:

- *subnet ID – 64 bits long. Contains the site prefix (obtained from a Regional Internet Registry) and the subnet ID (subnets within the site).*
- *interface ID – 64 bits long. typically composed of a part of the MAC address of the interface.*

Here is a graphical representation of the two parts of an global IPv6 address:



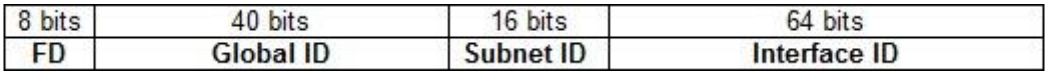
3 bits	45 bits	16 bits	64 bits
001	**Global Routing Prefix**	**Subnet ID**	**Interface ID**

IPv6 unique local addresses

Unique local IPv6 addresses have the similar function as IPv4 private addresses. They are not allocated by an address registry and are not meant to be routed outside their domain. Unique local IPv6 addresses begin withFD00::/8.

A unique local IPv6 address is constructed by appending a randomly generated 40-bit hexadecimal string to the FD00::/8 prefix. The subnet field and interface ID are created in the same way as with global IPv6 addresses.

A graphical representation of an unique local IPv6 address:



8 bits	40 bits	16 bits	64 bits
FD	**Global ID**	**Subnet ID**	**Interface ID**

NOTE

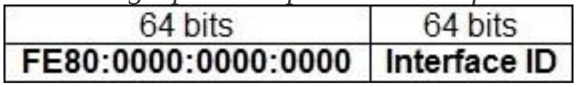
The original IPv6 RFCs defined a private address class called site local. This class has been deprecated and replaced with unique local addresses.

IPv6 link-local addresses

Link-local IPv6 addresses have a smaller scope as to how far they can travel: only within a network segment that a host is connected to. Routers will not forward packets destined to a link-local address to other links. A link-local IPv6 address must be assigned to every network interface on which the IPv6 protocol is enabled. A host can automatically derive its own link local IP address or the address can be manually configured.

Link-local addresses have a prefix of FE80::/10. They are mostly used for auto-address configuration and neighbor discovery.

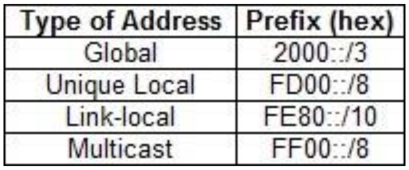
Here is a graphical representation of a link local IPv6 address:



64 bits	64 bits
FE80:0000:0000:0000	**Interface ID**

IPv6 address prefixes

Here is a summary of the most common address prefixes in IPv6:



Type of Address	Prefix (hex)
Global	2000::/3
Unique Local	FD00::/8
Link-local	FE80::/10
Multicast	FF00::/8

IPv6 interface identifier

The second part of an IPv6 unicast or anycast address is typically a 64-bit interface identifier used to identify a host's network interface. A 64-bit interface ID is created by inserting the hex value of FFFE in the middle of the MAC address of the network card. Also, the 7th Bit in the first byte is flipped to a binary 1 (if the 7th bit is set to 0 it means that the MAC address is a burned-in MAC address). When this is done, the interface ID is commonly called the modified extended unique identifier 64 (EUI-64).

For example, if the MAC address of a network card is 00:BB:CC:DD:11:22 the interface ID would be 02BBCCFFFEDD1122.

Why is that so?
Well, first we need to flip the seventh bit from 0 to 1. MAC addresses are in hex format. The binary format of the MAC address looks like this:

```
hex 00BBCCDD1122

binary 0000 0000 1011 1011 1100 1100 1101 1101 0001 0001 0010 0010
```

We need to flip the seventh bit:

```
binary 0000 0010 1011 1011 1100 1100 1101 1101 0001 0001 0010 0010
```

Now we have this address in hex:

```
hex 02BBCCDD1122
```

Next we need to insert FFFD in the middle of the address listed above:

```
hex 02BBCCFFFEDD1122
```

So, the interface ID is now 02BB:CCFF:FEDD:1122.

Another example, this time with the MAC address of 00000C432A35.

1. Convert to binary and flip the seventh bit to one:

binary: 0000 0010 0000 0000 0000 1100 0100 0011 0010 1010 0011 0101

2. Convert back to hex:

hex: 02000C432A35

3. Insert FFFT in the middle:

IPv6 transition options

IPv4 and IPv6 networks are not interoperable and the number of devices that use IPv4 number is still large. Some of these devices do not support IPv6 at all, so the migration process is necessary since IPv4 and IPv6 will likely coexist for some time.

Many transition mechanisms have been proposes.

1. IPv4/IPv6 Dual Stacks – each device on the network is configured with both an IPv4 and IPv6 address. When two devices want to communicate, they first agree on which IP version to use.
2. NAT64 – creates mapping between two address types. e.g. by mapping multiple IPv6 addresses to one IPv4 address.
3. Tunneling – Pv4 packets are tunneled over IPv6 infrastructure or vice versa.

IPv6 routing protocols

Like IPv4, IPv6 also supports routing protocols that enable routers to exchange information about connected networks. IPv6 routing protocols can be internal (RIPng, EIGRP for IPv6…) and external (BGP).

As with IPv4, IPv6 routing protocols can be distance vector and link-state. An example of a distance vector protocol is RIPng with hop count as the metric. An example of a link-state routing protocol is OSPF with cost as the metric.

IPv6 supports the following routing protocols:

- *RIPng (RIP New Generation)*
- *OSPFv3*
- *EIGRP for IPv6*
- *IS-IS for IPv6*

How to configure IPv6

Cisco routers do not have IPv6 routing enabled by default. To configure IPv6 on a Cisco routers, you need to do two things:

1. *enable IPv6 routing on a Cisco router using the ipv6 unicast-routing global configuration command. This command globally enables IPv6 and must be the first command executed on the router.*
2. *configure the IPv6 global unicast address on an interface using the ipv6 address address/prefix-length [eui-64]command. If you omit omit the eui-64 parameter, you will need to configure the entire address manually. After you enter this command, the link local address will be automatically derived.*

Here is an IPv6 configuration example:

```
R1(config)#ipv6 unicast-routing

R1(config)#int Gi0/0

R1(config-if)#ipv6 address 2001:0BB9:AABB:1234::/64 eui-64
```

We can verify that the IPv6 address has been configured by using the show ipv6 interface Gi0/0 command:

```
R1#show ipv6 interface Gi0/0

GigabitEthernet0/0 is up, line protocol is up

  IPv6 is enabled, link-local address is FE80::201:42FF:FE65:3E01

  No Virtual link-local address(es):

  Global unicast address(es):

    2001:BB9:AABB:1234:201:42FF:FE65:3E01, subnet is 2001:BB9:AABB:1234::/64 [EUI]

  Joined group address(es):

    FF02::1

    FF02::2

    FF02::1:FF65:3E01

  MTU is 1500 bytes

  ....
```

From the output above we can verify two things:

1. the link local IPv6 address has been automatically configured. Link local IP addresses begin with FE80::/10 and the interface ID is used for the rest of the address. Because the MAC address of the interface is 00:01:42:65:3E01, the calculated address is FE80::201:42FF:FE65:3E01.
2. the global IPv6 address has been created using the modified EUI-64 method. Remember that IPv6 global addresses begin with 2000::/3. So in our case, the IPv6 global address is 2001:BB9:AABB:1234:201:42FF:FE65:3E01.

We will also create an IPv6 address on another router. This time we will enter the whole address:

```
R2(config-if)#ipv6 address 2001:0BB9:AABB:1234:1111:2222:3333:4444/64
```

Notice that the IPv6 address is in the same subnet as the one configured on R1 (2001:0BB9:AABB:1234/64). We can test the connectivity between the devices using ping for IPv6:

```
R1#ping ipv6 2001:0BB9:AABB:1234:1111:2222:3333:4444
```

```
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2001:0BB9:AABB:1234:1111:2222:3333:4444, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

RIPng

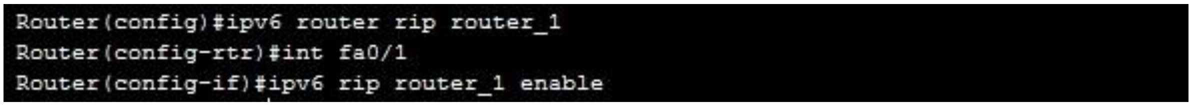
RIPng is an extension of RIP developed for support of IPv6. Here are some of its features:

- *just like RIP for IPv4, it uses hop count as the metric*
- *sends updates every 30 seconds*
- *RIPng messages use the UDP port 521 and the multicast address of FF02::9*

The configuration of RIPng is requires at least two steps:

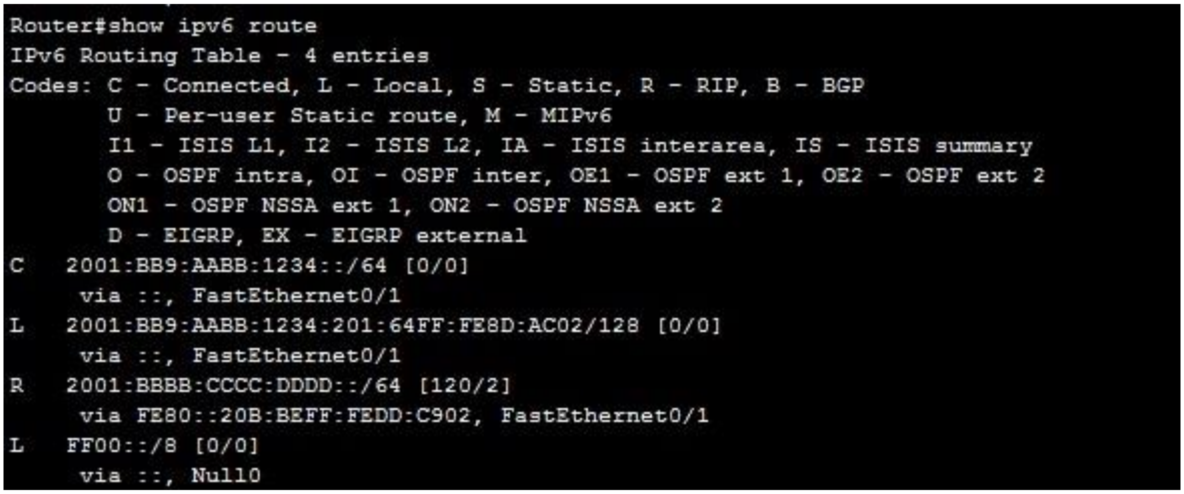
1. enable RIPng using the global configuration command ipv6 router rip tag. The tag is used to differentiate between multiple RIP processes. It does not have to be the same on all routers in order to exchange routing information..
2. enable the routing protocol on the interface using the ipv6 rip tag enable. The tag has to match the one used in the ipv6 router rip tag command.

Here is an example:



```
Router(config)#ipv6 router rip router_1
Router(config-rtr)#int fa0/1
Router(config-if)#ipv6 rip router_1 enable
```

We have done a similar configuration on the second router. To verify that routers are indeed exchanging route information using RIPng we can use the show ipv6 route command:

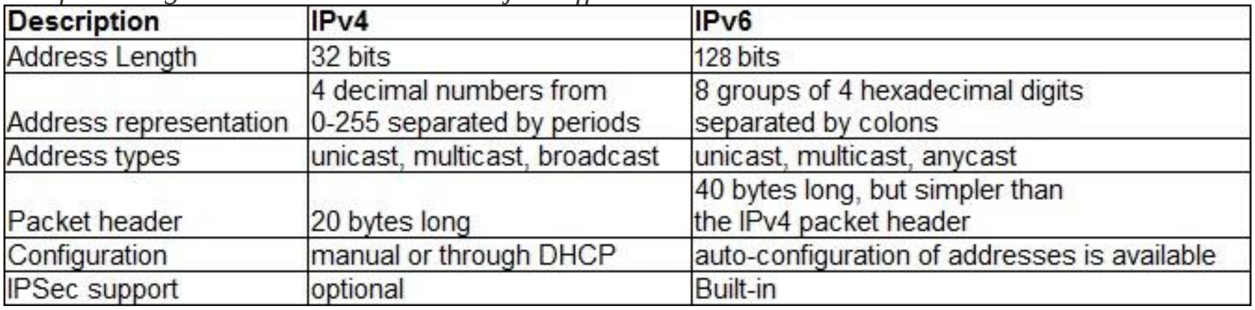


```
Router#show ipv6 route
IPv6 Routing Table - 4 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route, M - MIPv6
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
C   2001:BB9:AABB:1234::/64 [0/0]
     via ::, FastEthernet0/1
L   2001:BB9:AABB:1234:201:64FF:FE8D:AC02/128 [0/0]
     via ::, FastEthernet0/1
R   2001:BBBB:CCCC:DDDD::/64 [120/2]
     via FE80::20B:BEFF:FEDD:C902, FastEthernet0/1
L   FF00::/8 [0/0]
     via ::, Null0
```

In the picture above, we can see that the router has received a route to the network 2001:BBBB:CCCC:DDDD::/64

Differences between IPv4 and IPv6

The following table summarizes the major differences between IPv4 and IPv6:



Description	**IPv4**	**IPv6**
Address Length	32 bits	128 bits
Address representation	4 decimal numbers from 0-255 separated by periods	8 groups of 4 hexadecimal digits separated by colons
Address types	unicast, multicast, broadcast	unicast, multicast, anycast
Packet header	20 bytes long	40 bytes long, but simpler than the IPv4 packet header
Configuration	manual or through DHCP	auto-configuration of addresses is available
IPSec support	optional	Built-in

Source:

www.Study-ccna.com

www.google.com

www.ccna-totoiaral.com

CCNA Routing and Switching Study Guide – 2013

CCENT/CCNA ICND 100 – 105

CCNA introduction

Table of Contents