CCNA Study

Author : Mehdi Najafzade

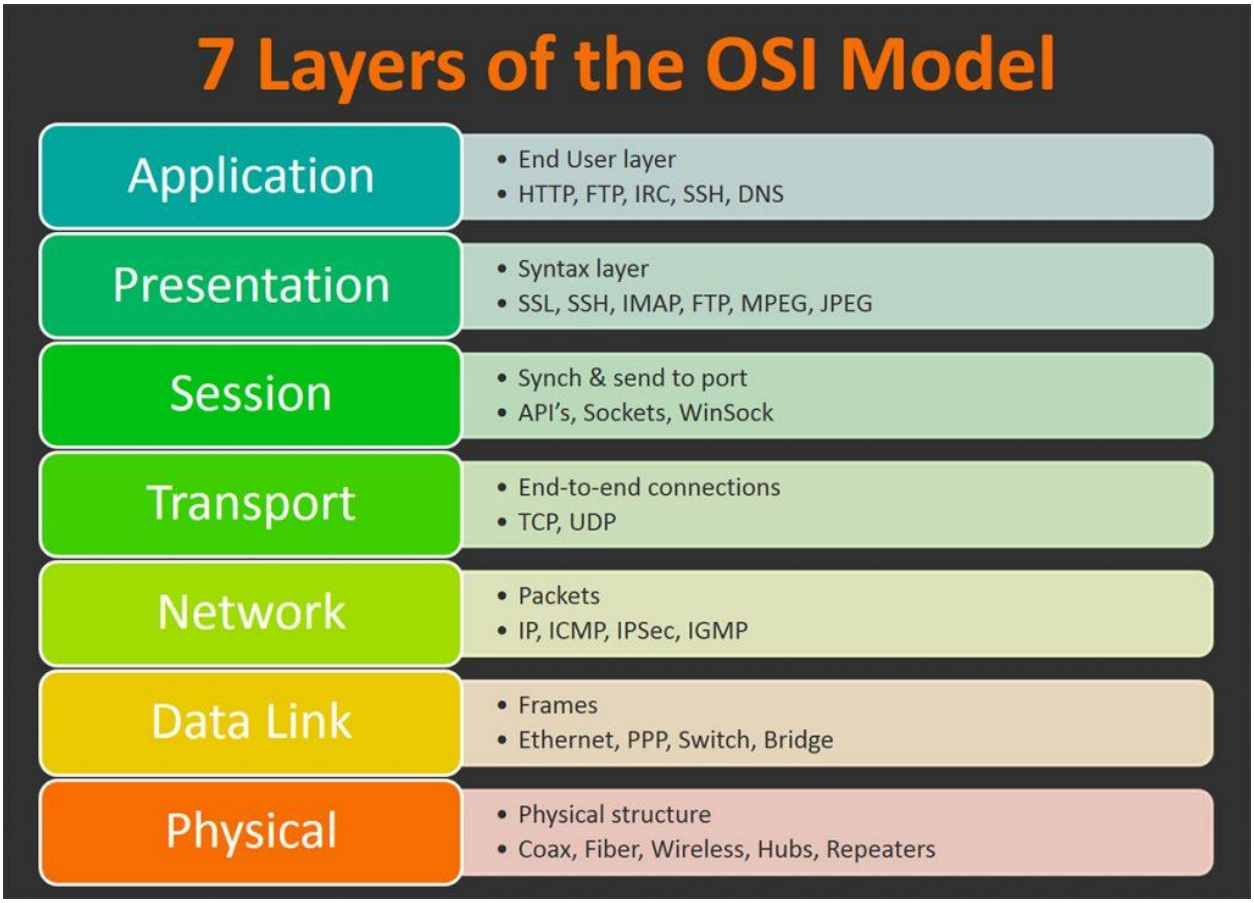
Instagram : @mehdinajafzada

Resources :

Cisco books & whole of internet :)

Volume 1

OSI & TCP/IP Models

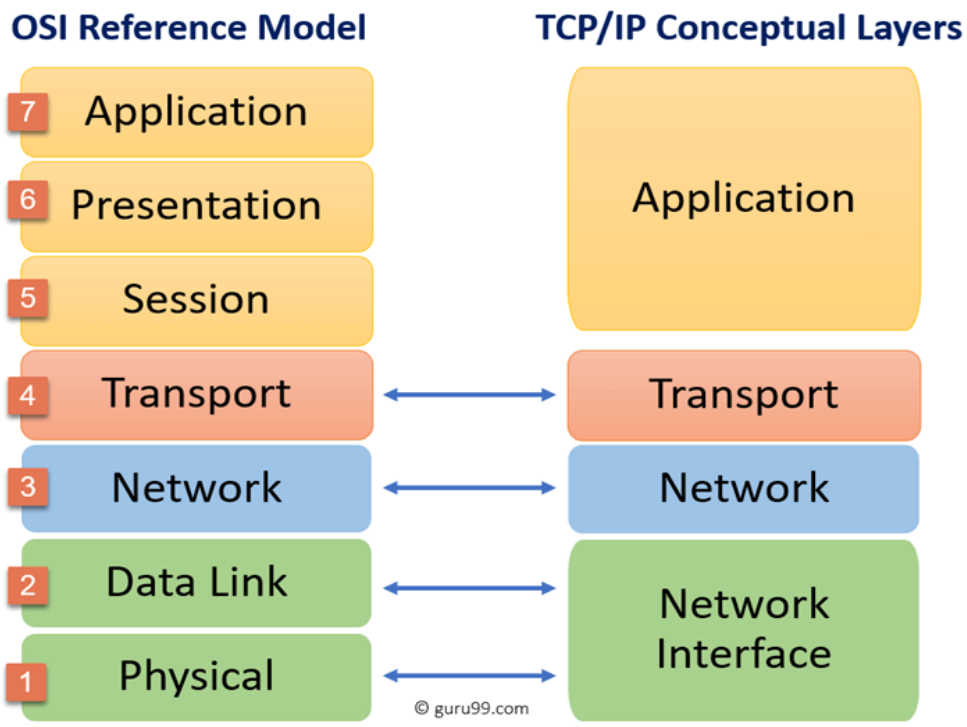


- **Physical** – defines how to move bits from one devices to another.cable, connectors, and network interface cards.
- **Data Link** – encapsulates a packet in a frame. A frame contains a header and a trailer that enable devices to communicate. A header (most commonly) contains a source and destination 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.

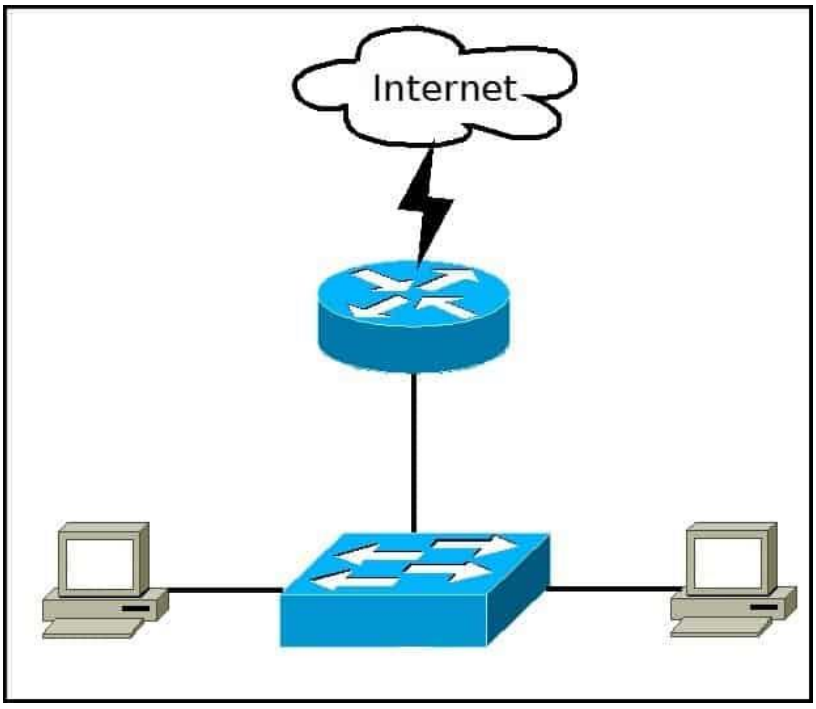
TCP/IP Model



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:



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

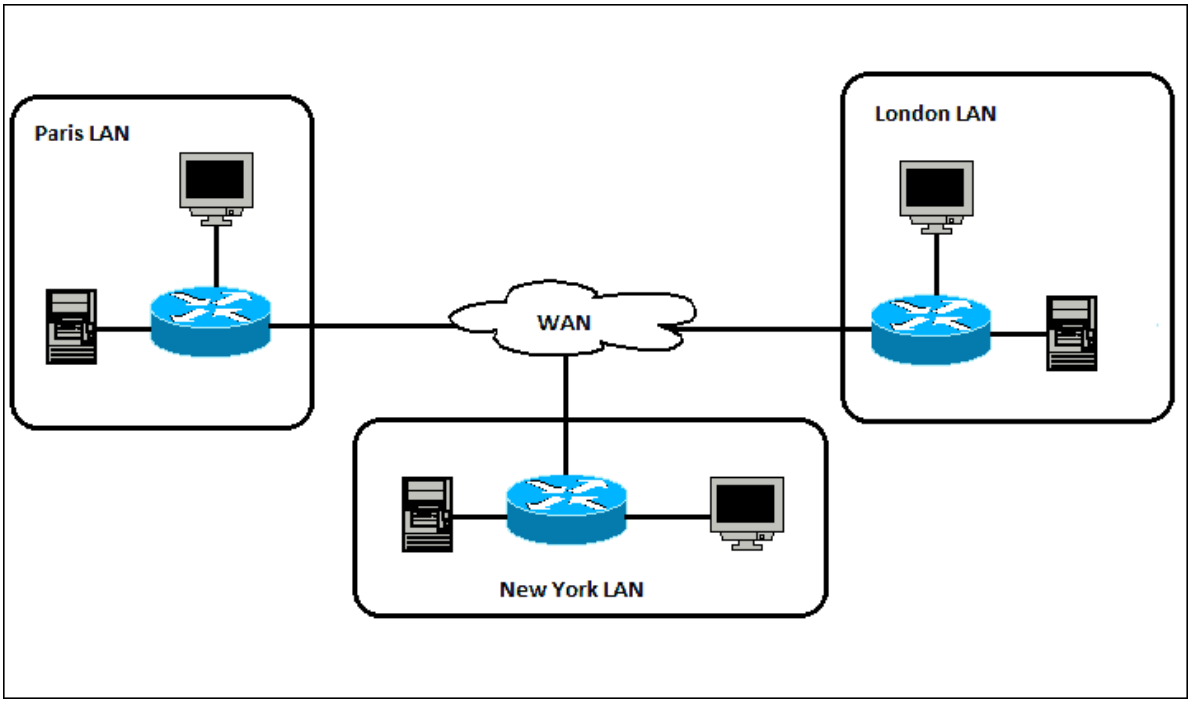
standards. WLANs use radio waves instead of wires or cables for links between devices.

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(WAN)

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 favourite email program (such as Outlook or Thunderbird):

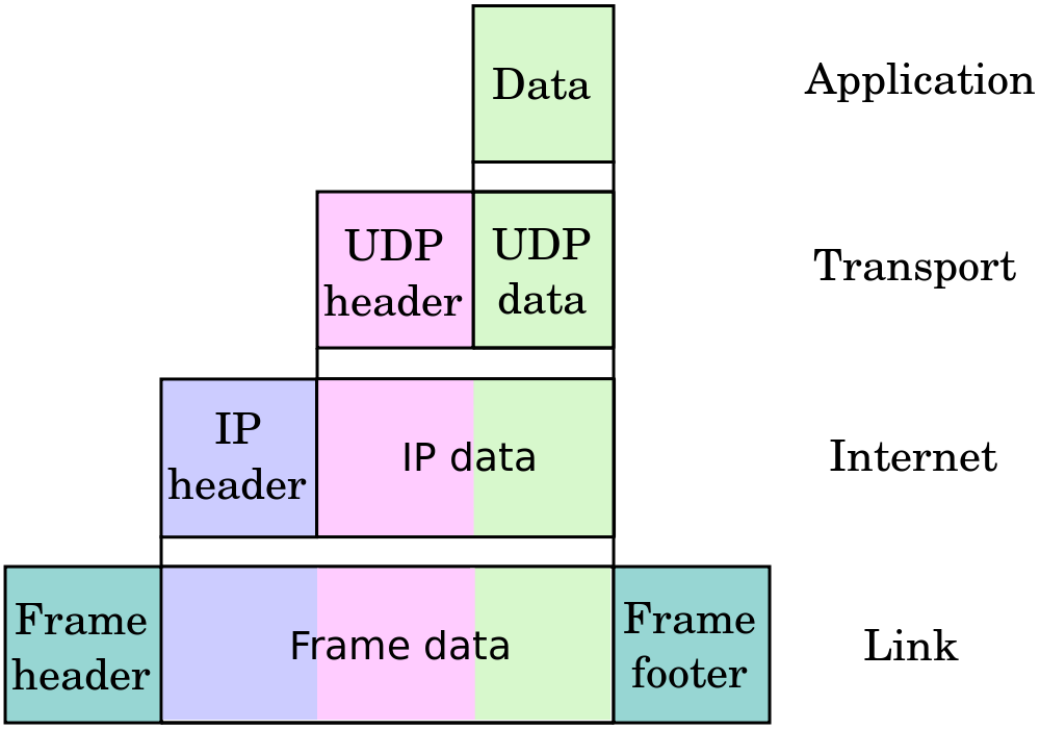
- the email is sent from the Application layer to the Transport layer.
- the Transport layer encapsulates the data and adds its own header with its own information, such as which port will be used and passes the data to the Internet layer
- the Internet layer encapsulates the received data and adds its own header, usually with information about the source and destination IP addresses. The Internet layer than passes the data to the Network Access layer
- 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

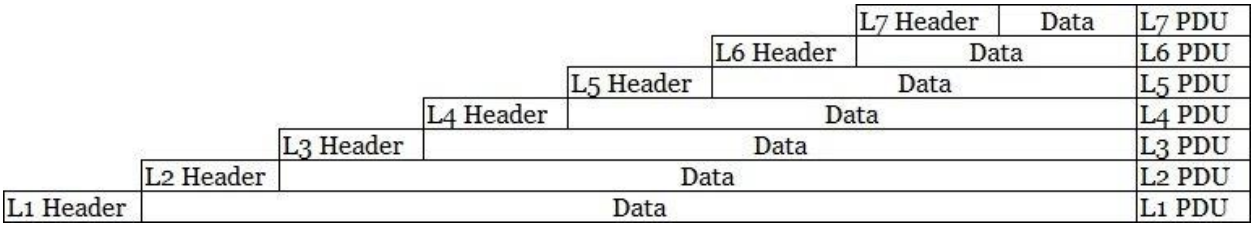
The term **decapsulation** refers to the process of removing headers and trailers as data passes from lower to upper layers. This process happens on the computer that is receiving data.

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:



Section 2

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 synchronisation.
- **SFD(Start Frame Delimeter) -** 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 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

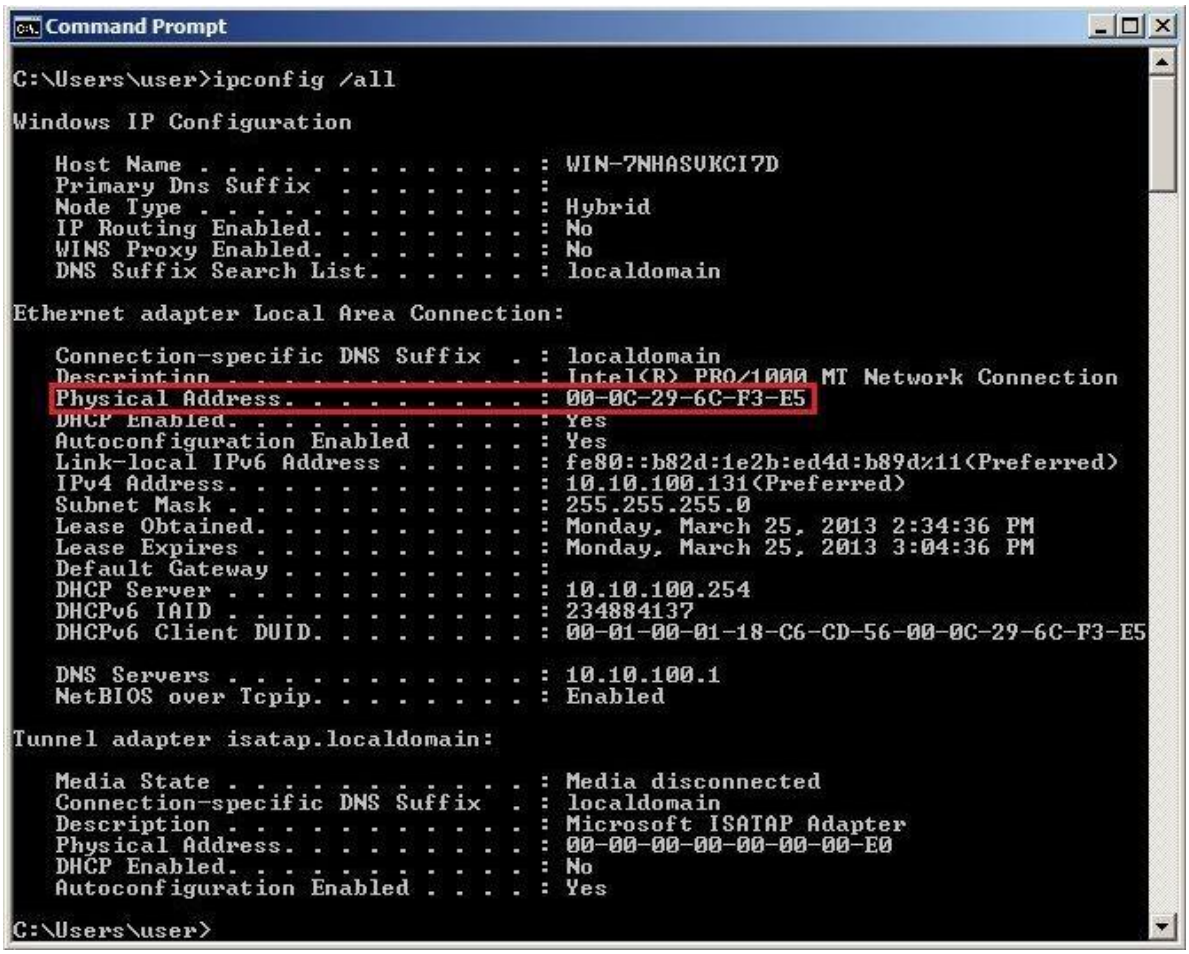
environment. It is a hardware address, which means that it is stored in the firmware of the network card.

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 globaly unique.

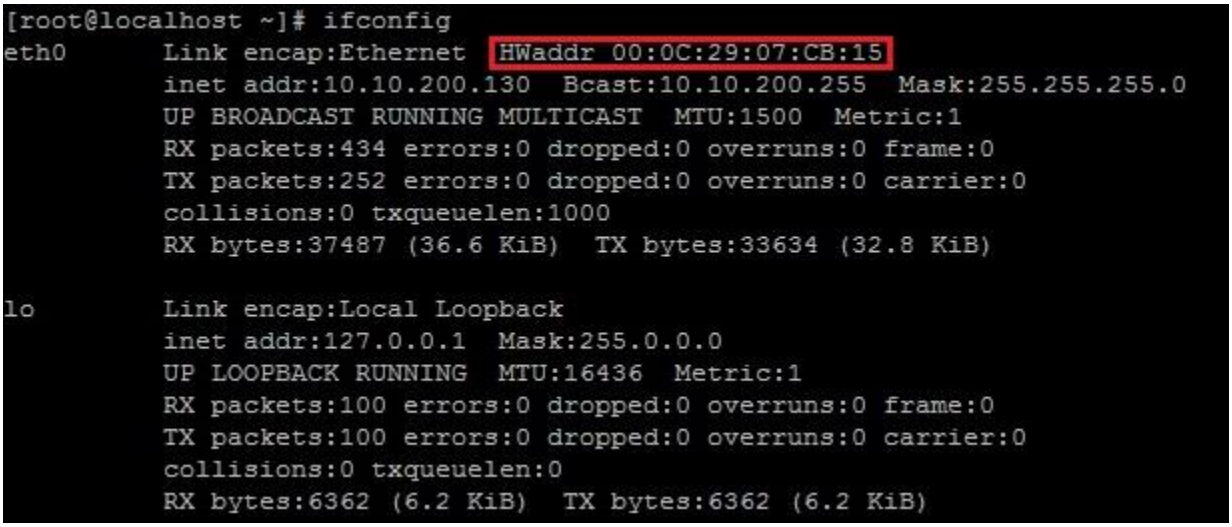
MAC addresses are usually written in the form of 12 hexadecimal digits. For example, consider the following MAC address: D8-D3-85-EB-12-E3

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

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:



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



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 seperated 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

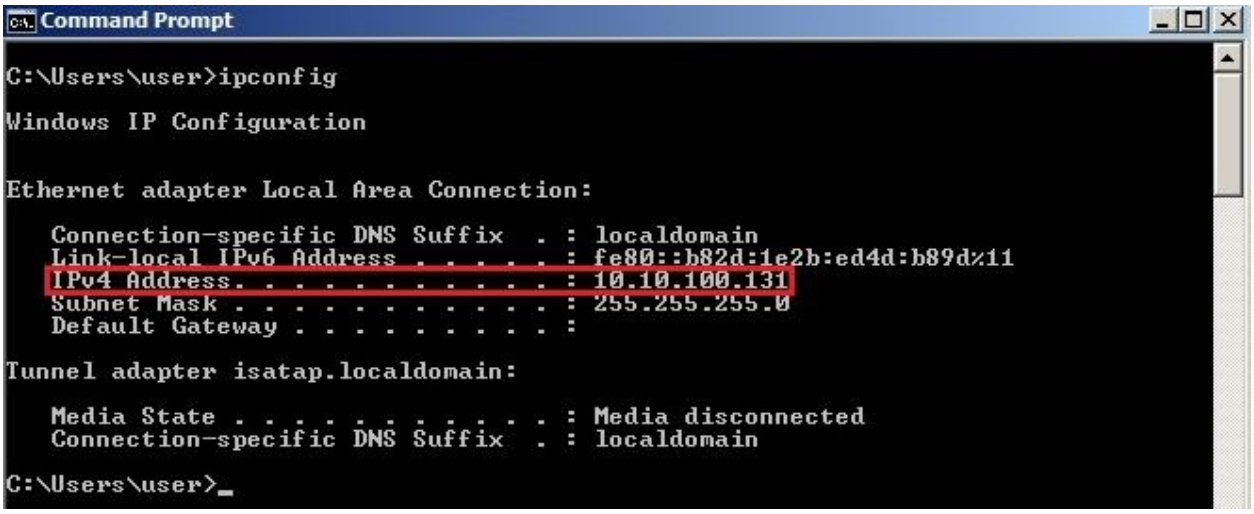
The term IP address is usually used for IPv4, which is the fourth version of the IP protocol. A newer version exists, IPv6, and uses 128-bit addressing.

Private IP addresses

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

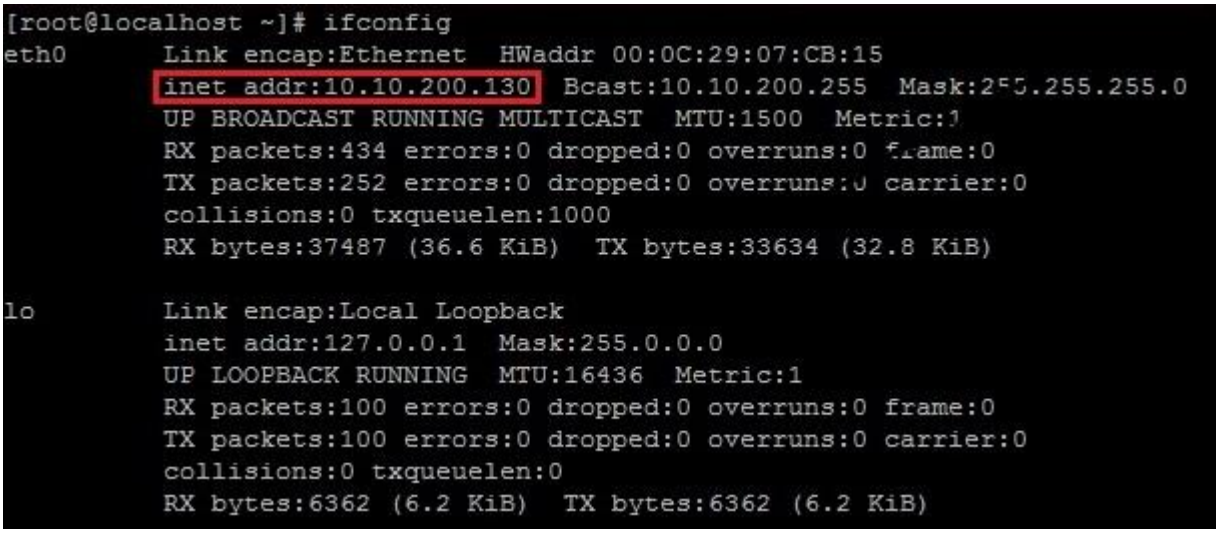
How to find out your ip address

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:



Linux users:

Enter *ifconfig*. You should see a field called inet addr:

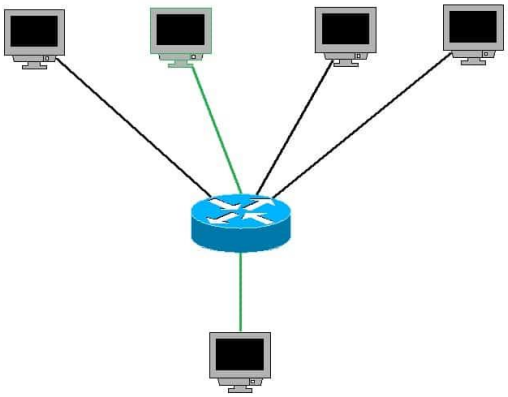


Unicast, multicast, and broadcast addresses

There are three types of Ethernet addresses:

1. **Unicast addresses**

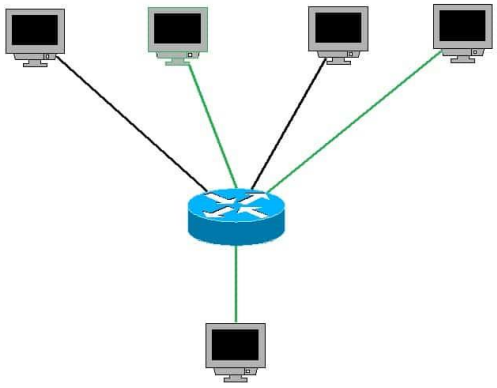
   Unicast addresses represent a single LAN interface. A unicast frame will be sent to a specific device, not to a group of devices on the LAN:



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

2. **Multicast addresses**

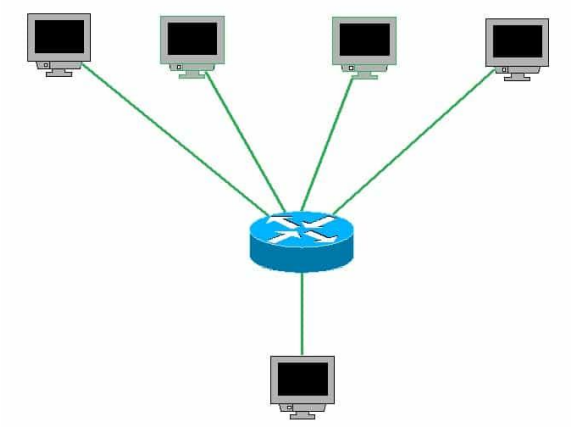
   Multicast addresses represent a group of devices in a LAN. A frame sent to a multicast address will be forwarded to a group of devices on the LAN:



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 the address used by CDP (Cisco Discovery Protocol).

3. **Broadcast addresses**

   Broadcast addresses represent all device on the LAN. Frames sent to a broadcast address will be delivered to all devices on the LAN:



   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.

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).

Routers

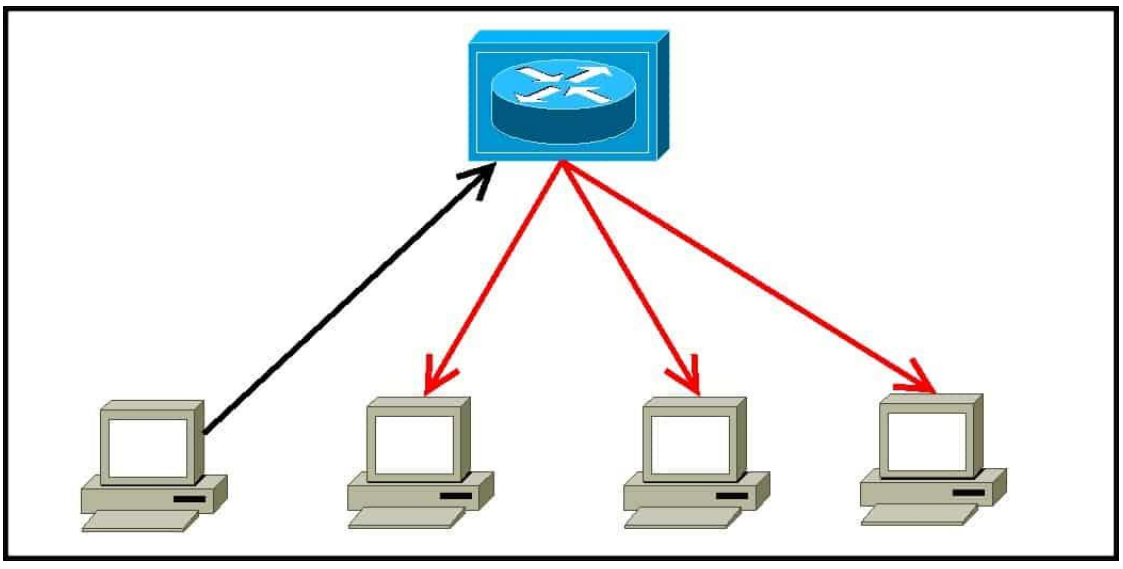
A router is a device that routes packets from one network to another. A router is most commonly an OSI Layer 3 device. Routers divide broadcast domains and have traffic filtering capabilities.The picture below shows a typical home router:

Section 3

Network hubs explained

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

As mentioned above, hubs have no way of distinguishing out which port a signal should be sent to; instead, an electrical signal is sent out each port. All nodes on the network will receive data, and the data will eventually reach the correct destination, but with a lot of unnecessary network traffic:



In the example above you can see that the hub has sent out the receiving signal out all other ports, except the incoming port. Hubs are therefore considered obsolete and switches are commonly used instead in modern LANs. Hubs have numerous disadvantages over switches, such as:

- 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 also known as multiport repeaters because that is basically what they do – repeat the electrical signal that comes in one port out all other ports (except the incoming port).

Network bridge explained

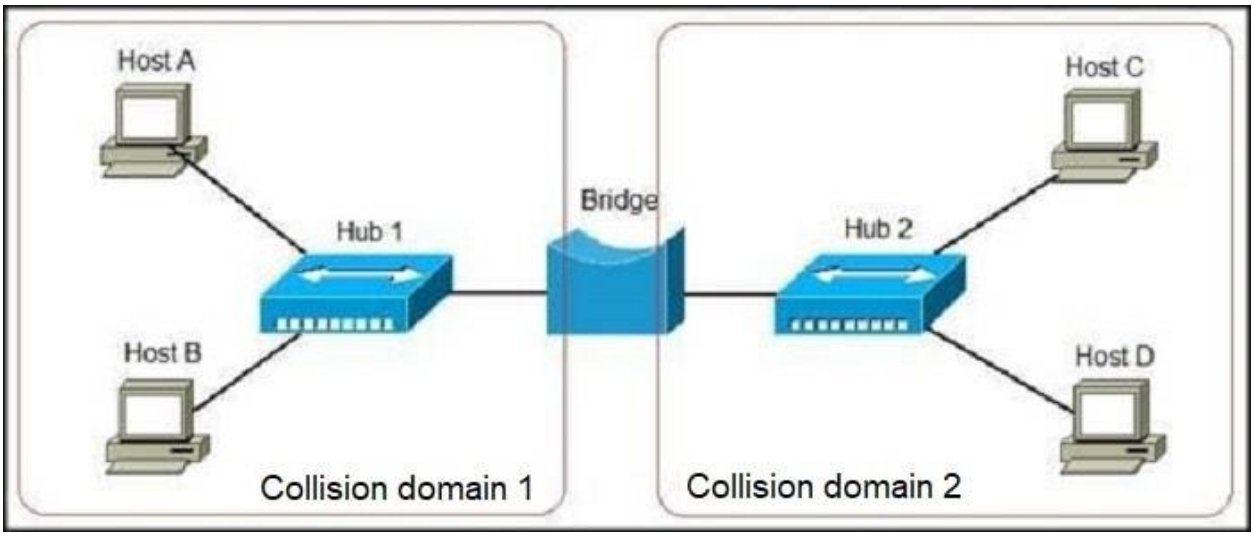
A network bridge is a device that divides a network into segments. Each segment represent a separate collision domain, so the number of collisions on the network is reduced. Also, because each collision domain has its own separate bandwidth, a bridge also improves the overall network performance.

NOTE

Unlike hubs, bridges allow multiple devices to send at the same time. This is why there are considered to be predecessors of network switches.

A bridge works at the Data link layer (Layer 2) of the OSI model, just like a switch does. It inspects incoming traffic and decide whether to forward it or filter it. Each incoming Ethernet frame is inspected for destination MAC address. If the bridge determines that the destination host is on another segment of the network, it forwards the frame to that segment.

Consider the following network:



In the example above we have a network of four computers. The network is divided into segments by a bridge. Each segment is a separate collision domain with its own bandwidth. Let's say that Host A wants to communicate with Host C. Host A will send the frame with the Host C's destination MAC address to the bridge. The bridge will inspect the frame and forward it to the segment of the network Host C is on.

Network bridges offered substantial improvements over network hubs, but are not widely used anymore in modern LANs – switches are commonly used instead. Here is why:

- most bridges have only 2 or 4 ports. A switch can have tens or even hundreds of ports

- bridges are software based, while switches are hardware-based and use chips (ASICs) when making forwarding decisions, which makes them much faster than bridges
- switches can have multiple spanning-tree instances, bridges can have only one
- switches can have multiple broadcast domains (one per VLAN)

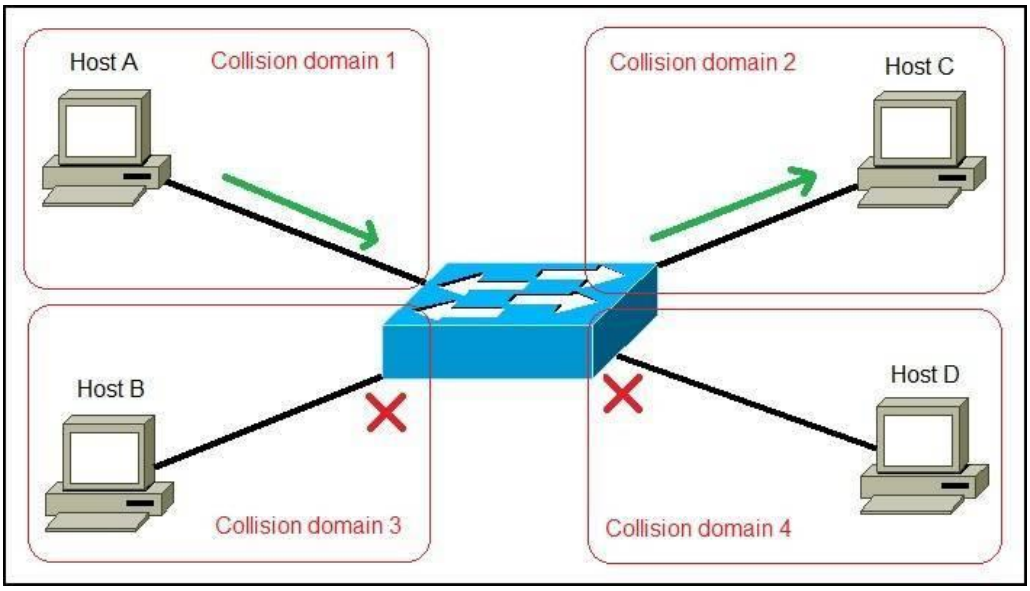
Network switch explained

Just like hubs and bridges, a switch is used to connect multiple hosts together, but it has many advantages over them. 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).

A switch manages the flow of data across a network by inspecting the incoming frame's destination MAC address and forwarding the frame only to the host for which the data was intended. Each switch has a dynamic table (called the MAC address table) that maps MAC addresses

to ports. With this information, a switch can identify which system is sitting on which port and where to send the received frame.

To better understand how a switch works, consider the following example:



As you can see from the example above, Host A is trying to communicate with Host C and sends a packet with the Host C's destination MAC address. The packet arrives at the switch, which looks at the destination MAC address. The switch then searches that MAC address in its MAC address table. If the MAC address is found, the switch then forwards the packet only out the port connected to the frame's destination. Hosts connected to other ports will not receive the frame.

Network router explained

A router is a network device that routes packets from one network to another. It is usually connected to two or more different networks. When a packet comes to a router port, the router reads the address

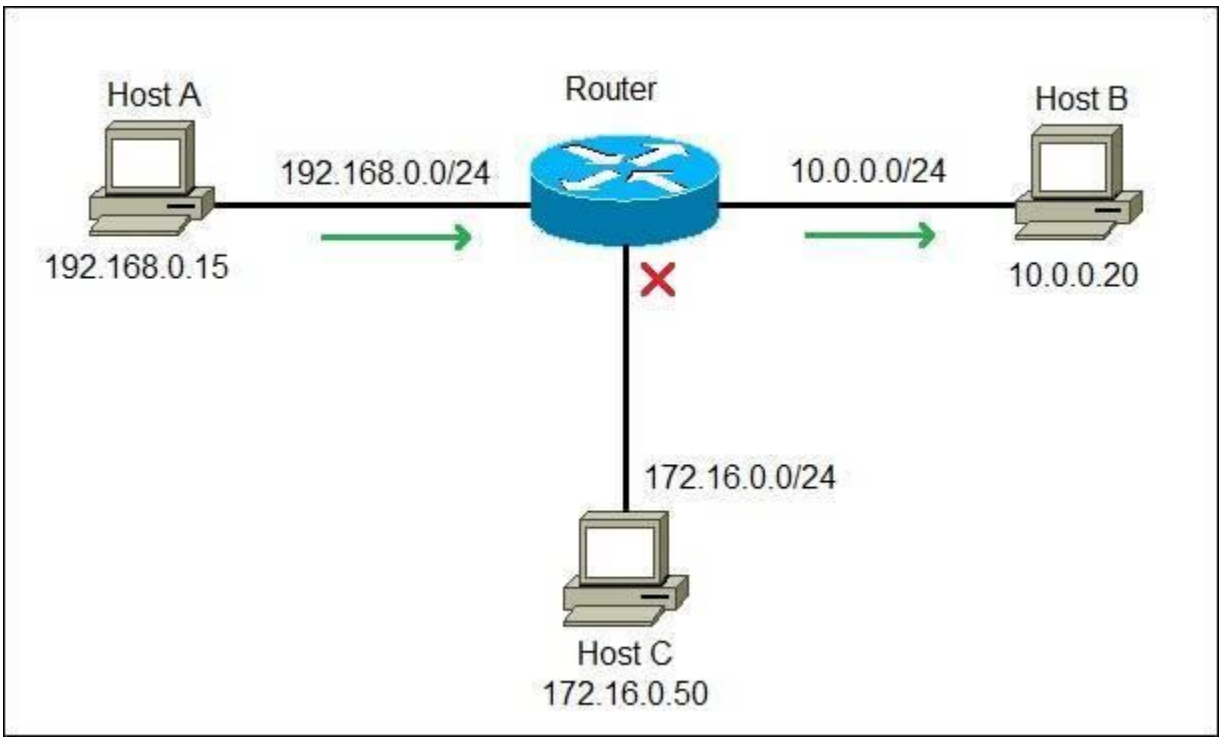
information in the packet to determine out which port the packet will be sent. For example, a router provides you with the internet access by connecting your LAN with the Internet.

NOTE

A router is most commonly an OSI Layer 3 device, since its forwarding decision is based on the information of the OSI Layer 3 – the destination IP address. Routers divide broadcast domains, provide full duplex communication, and have traffic filtering capabilities.

The picture below shows a typical home router:

If two hosts from different networks want to communicate, they will need a router in order to exchange data. Consider the following example:



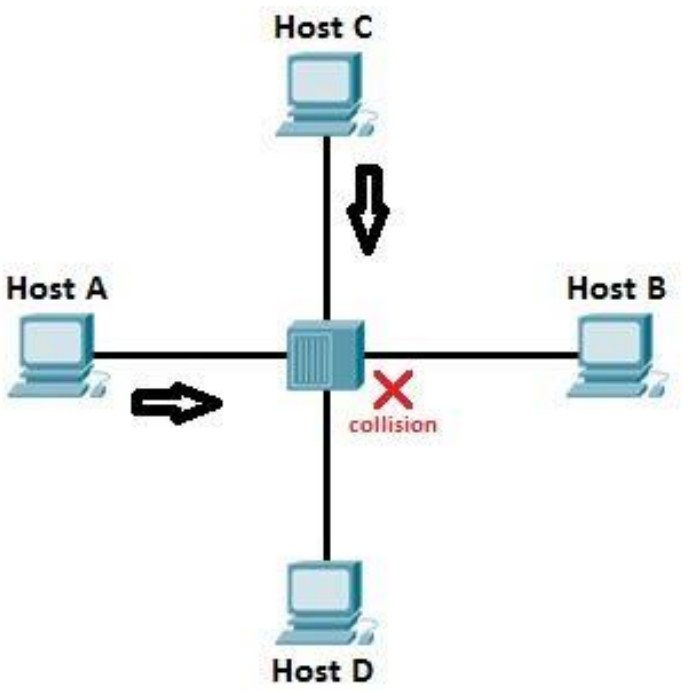
We have a network of three hosts and a router. Note that each computer is on a different network. Host A wants to communicate with Host B and sends the packet with the Host B's IP address (10.0.0.20) to the router. The router receives the packet, compares the packet's destination IP address to the entries in its routing table and finds a match. It then sends the packet out the interface associated with the network 10.0.0.0/24. Only Host B will receive and process the packet. In fact, Host C will not even be aware that the communication took place.

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 swich usually

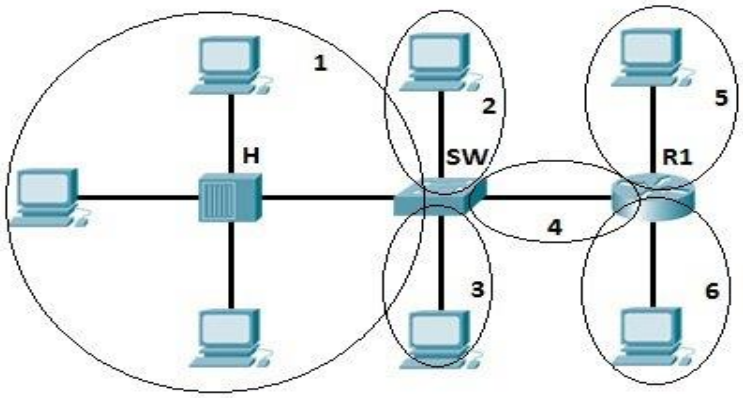
operate in a full duplex mode and there are no packet collisions in a full duplex mode.

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.

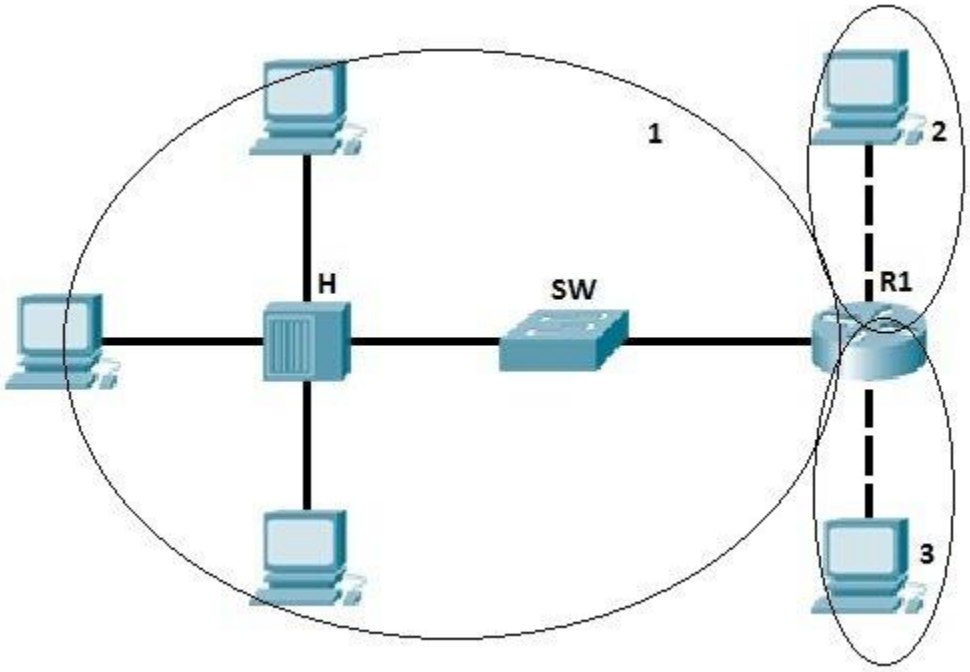
NOTE

Remember, each port on a hub is in the same collision domain. Each port on a bridge, a switch or router is in a seperate 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.

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

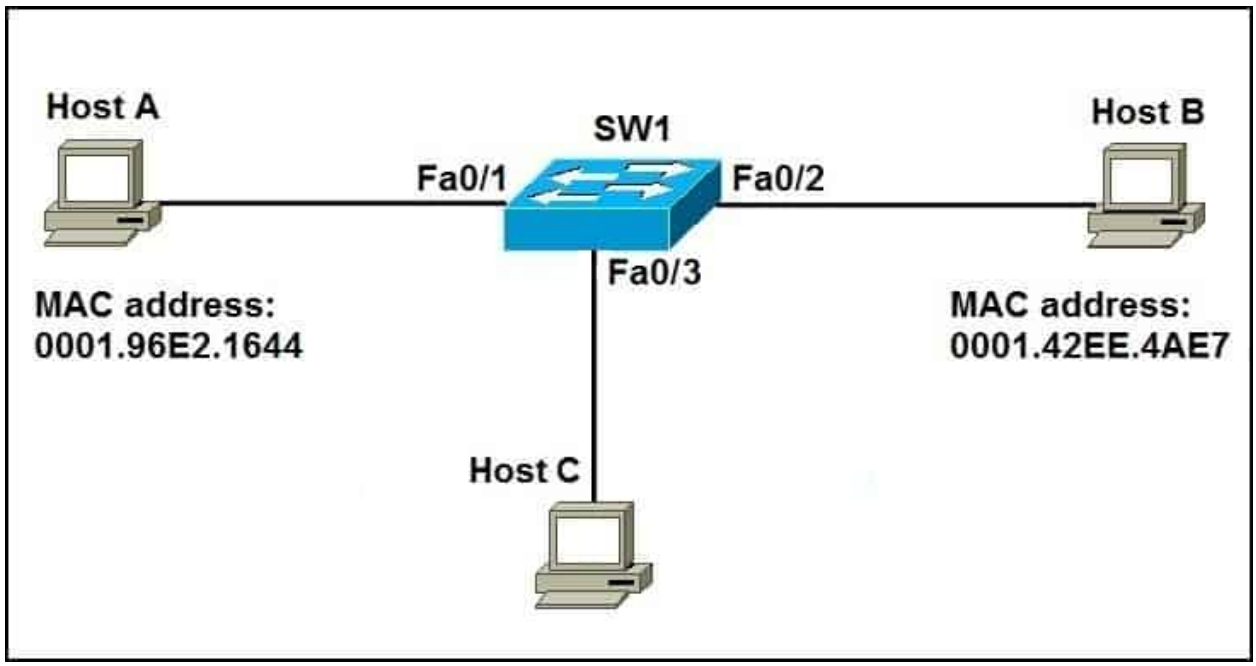
devices on the same network segment. Devices that want to communicate need to know each other MAC addresses before sending out packets.

Switches also use MAC addresses to make accurate forwarding or filtering decision. When a switch receives a frame, it associates the media access control (MAC) address of the sending device with the port on which it was received. The table that stores such associations is called a MAC address table. This table is stored in the volatile memory, so associations are erased after the switch is rebooted.

Switches usually perform these three functions in a LAN:

- **address learning** – switches learn MAC addresses by examining the source MAC address of each received frame.
- **forward/filter decisions** – switches decide whether to forward or filter a frame, based on the destination MAC address.
- **loop avoidance** – switches use **Spanning Tree Protocol (STP)** to prevent network loops while still permitting redundancy.

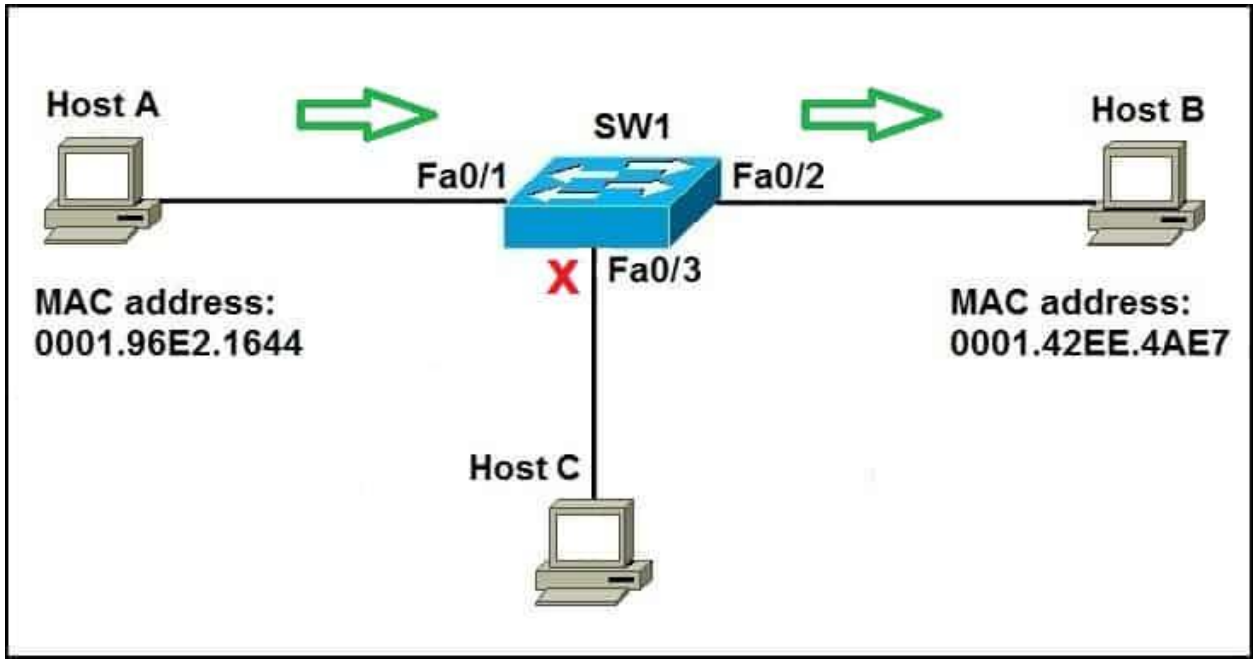
To better how a network switch works, take a look at the following example:



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). Host C receives the ARP request, but doesn't respond since the IP address listed in the request is not its own.

As mentioned above, a switch learns which MAC addresses are associated with which port by examining the source MAC address of each received frame. 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. For host A, the switch knows its MAC address because of the ARP request that included it.

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 the Fa0/2 port – the port on which host B is connected to. Other hosts on the network will not be involved in the communication:



NOTE

By default, MAC addresses stay in the switch's MAC address table for 5 minutes. So if host A and host B decide to communicate inside the next 5 minutes, a new ARP process will not be necessary.

You can display the MAC address table of the switch by using the show mac-address-table command:

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


Vlan    Mac Address       Type        Ports
```

```
----    -----------       --------    -----


   1    0003.e489.513e    DYNAMIC     Fa0/2
   1    00e0.8f13.6970    DYNAMIC     Fa0/1
```

The output is pretty much self-explanatory: all ports belong to VLAN 1 and MAC addresses associated with specific ports are listed. DYNAMIC means that the address were learned dynamically by using the source MAC address of the received frames.

Layer 2 switching

Layer 2 switching (or Data Link layer switching) is the process of using devices' MAC addresses to decide where to forward frames. 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, 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. Also. each port on a switch is a separate collision domain, so no packet collisions should occur.

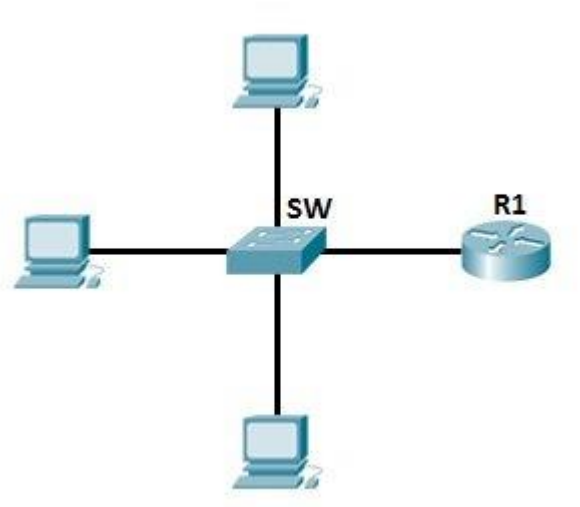
Layer 2 switches are faster than routers because they don't take up time looking at the Network layer header information. Instead, they look at

the frame's hardware addresses to decide what to do with the frame – to forward, flood, or drop it. Here are other major advantages of Layer 2 switching:

fast hardware-based bridging (using ASICs chips)

- **wire speed**
- **low latency**
- **low cost**

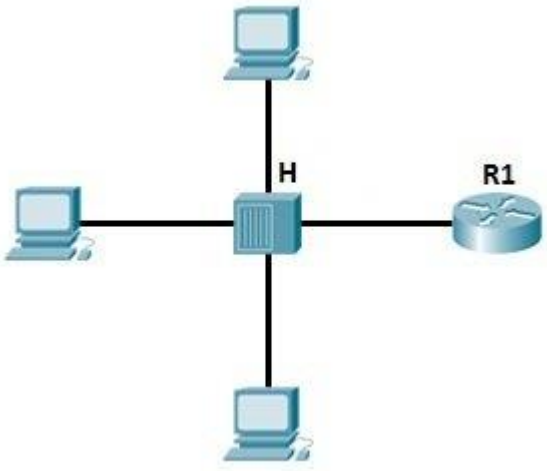
Here is an example of the typical LAN network – the switch serves as a central device that connects all devices together:



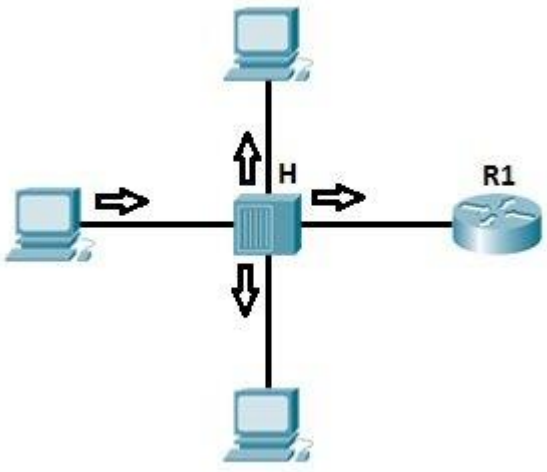
Differences between hubs and switches

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

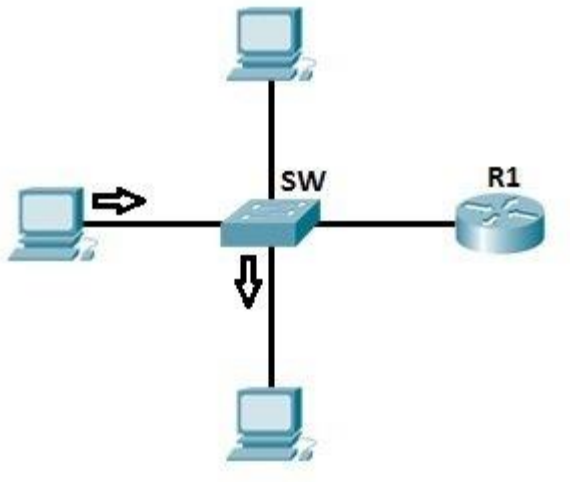
First, consider an example of a LAN in which all hosts connects to a hub:



As mentioned previously, hubs create only a single 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 frame 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 time 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.

Half duplex and Full duplex

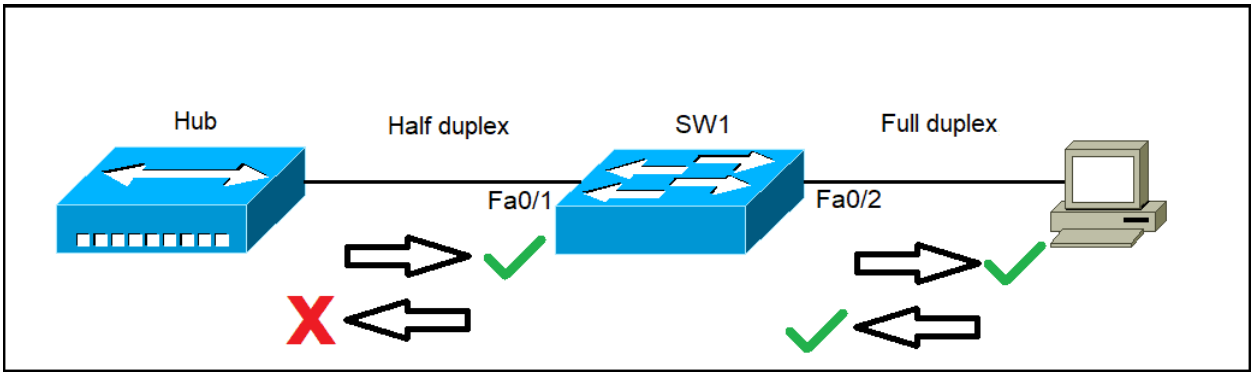
In telecommunication, a duplex communication system is a point-to-point system of two devices that can communicate with each other in both direction. These two types of duplex communication systems exist in Ethernet environments:

- **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.

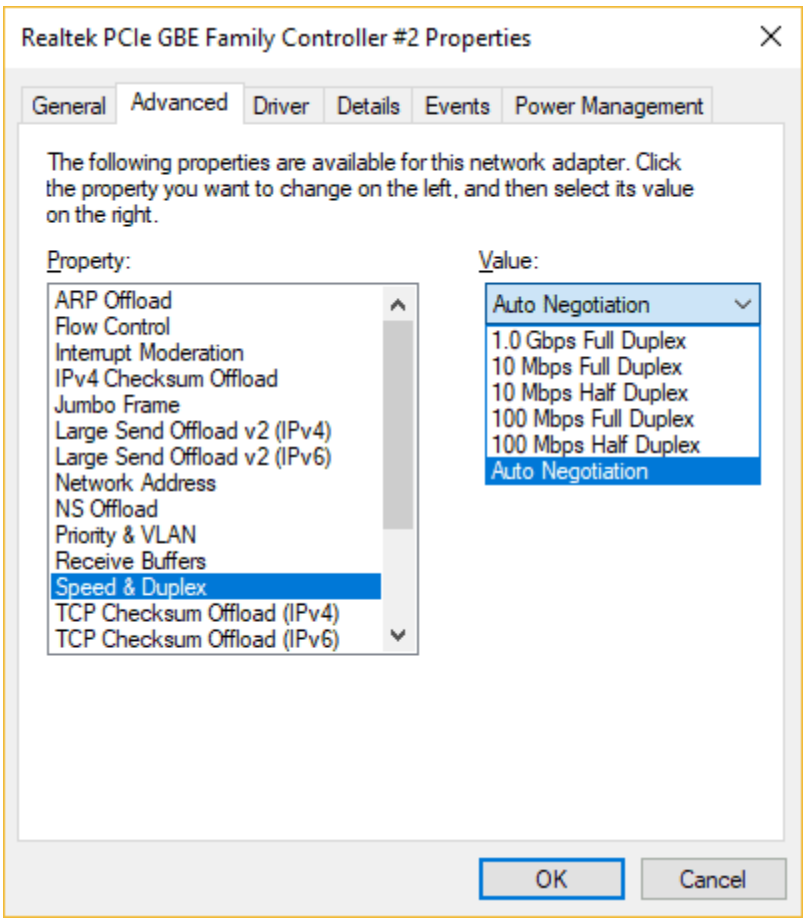
The following picture illustrates the concept:



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:


Realtek PCIe GBE Family Controller #2 Properties
General
Advanced
Driver
Details
Events
Power Management
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
OK
Cancel

Section 4

IEEE Ethernet standards

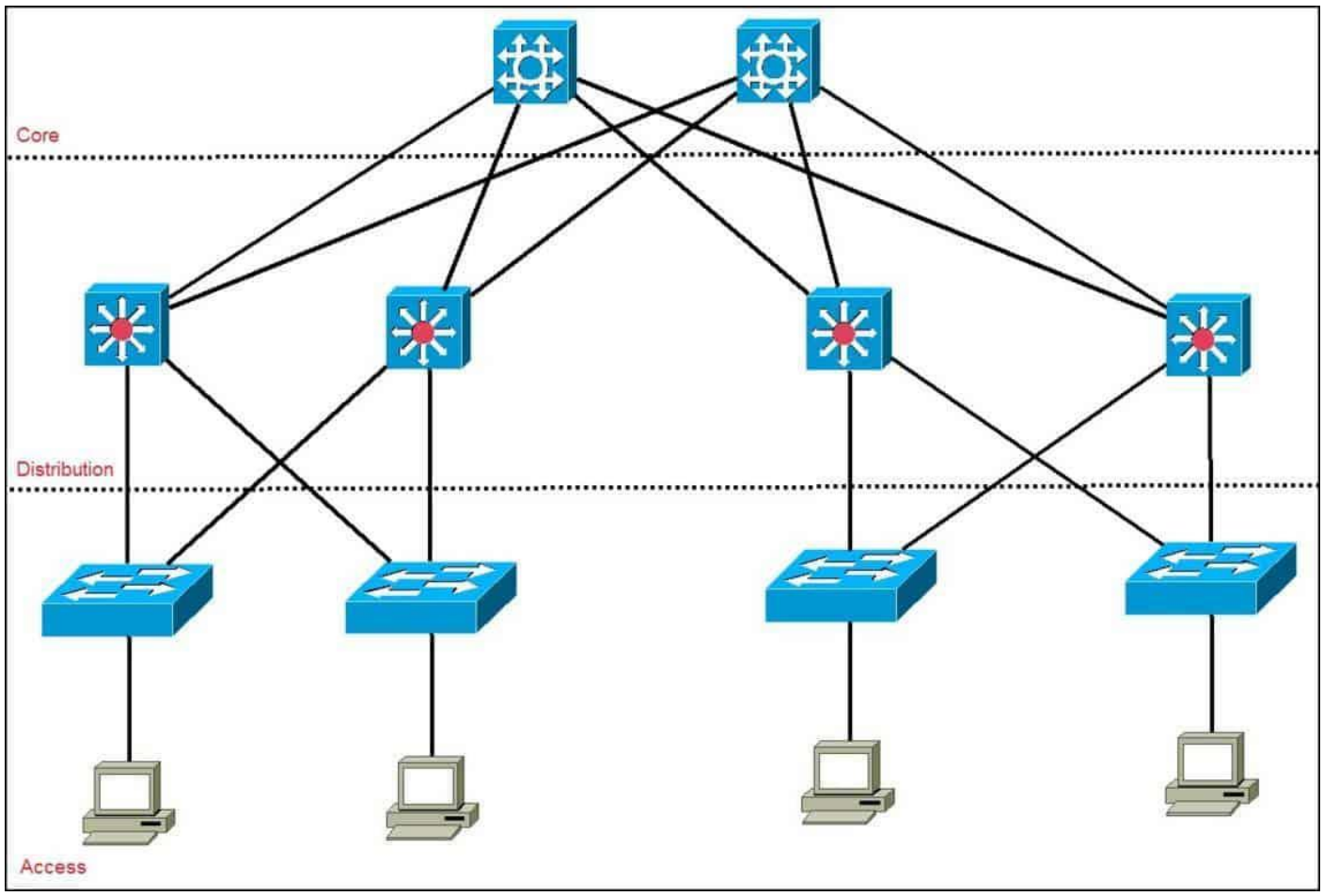
Ethernet is defined in a number of IEEE 802.3 standards. These standards define the physical and data-link layer specifications for Ethernet. The most important 802.3 standards are:

- **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.

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:

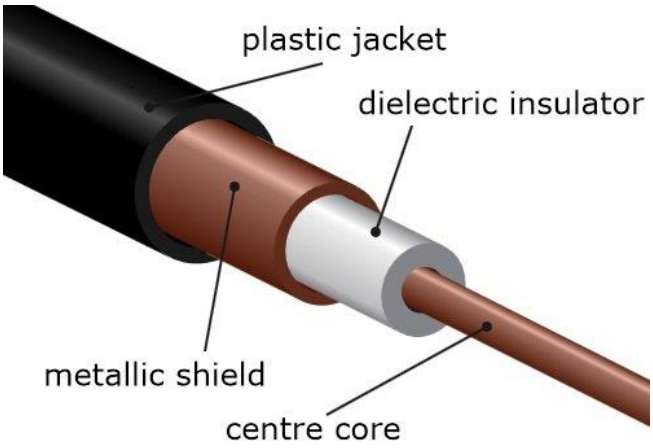
- **Access** – controls user and workgroup access to the resources on the network. This layer usually incorporates Layer 2 switches and access points that provide connectivity between workstations and servers. You can manage access control and policy, create separate collision domains, and implement 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.

Types 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 expain 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)

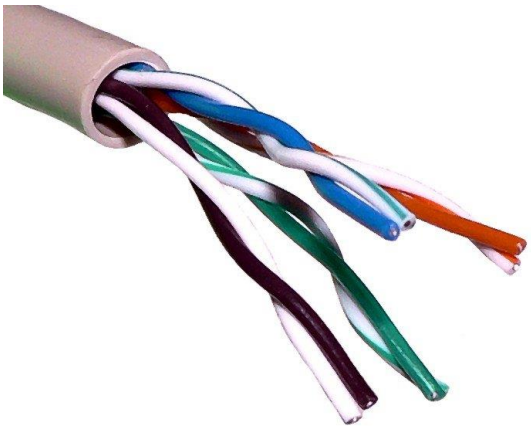


Twisted-pair cabling

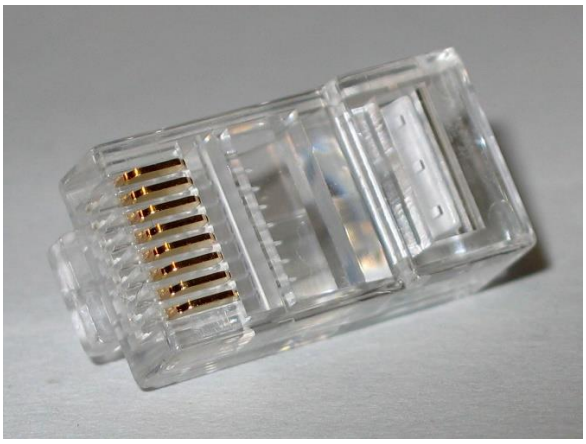
A twisted-pair cable has four pair of wires. These wires are twisted around each other to reduce crosstalk and outside interference. This type of cabling is common in current LANs.

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. Used for larger distances.
- **Multi-mode fiber (MMF)** – uses multiple rays of light to carry data. Less expensive than SMF.

Four types of connectors are commonly used:

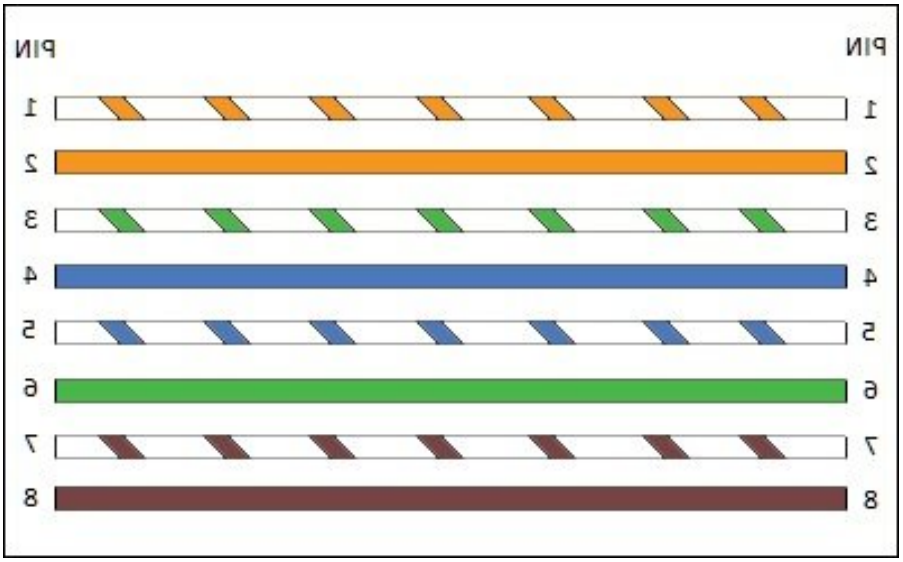
- **ST** (Straight-tip connector)
- **SC** (Subscriber connector)
- **FC** (Fiber Channel)
- **LC** (Lucent Connector)

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.):



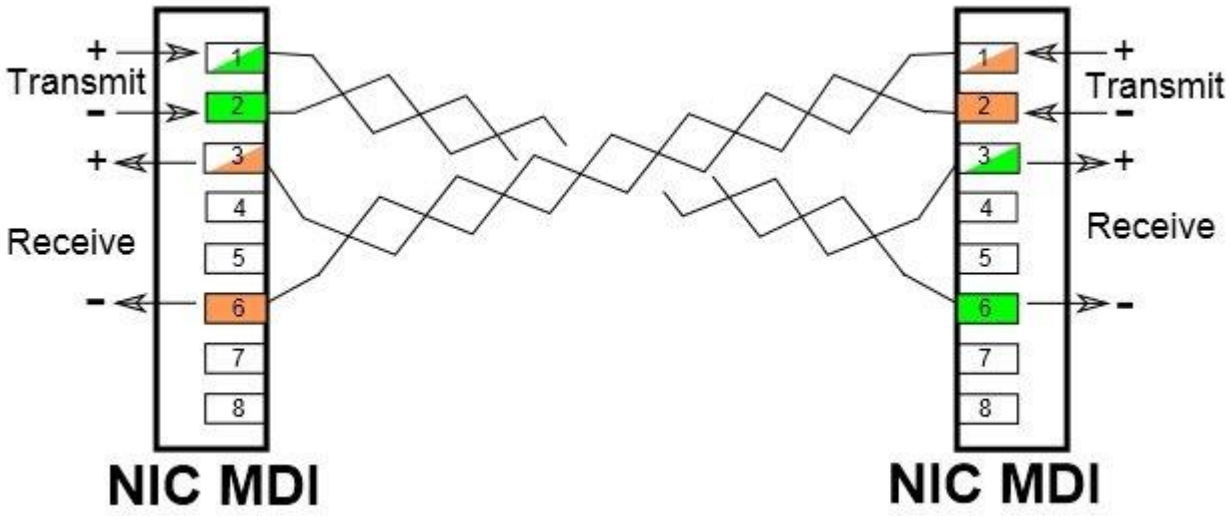
This type of cable is used to connect the following devices:

- 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.

Types of IP addresses

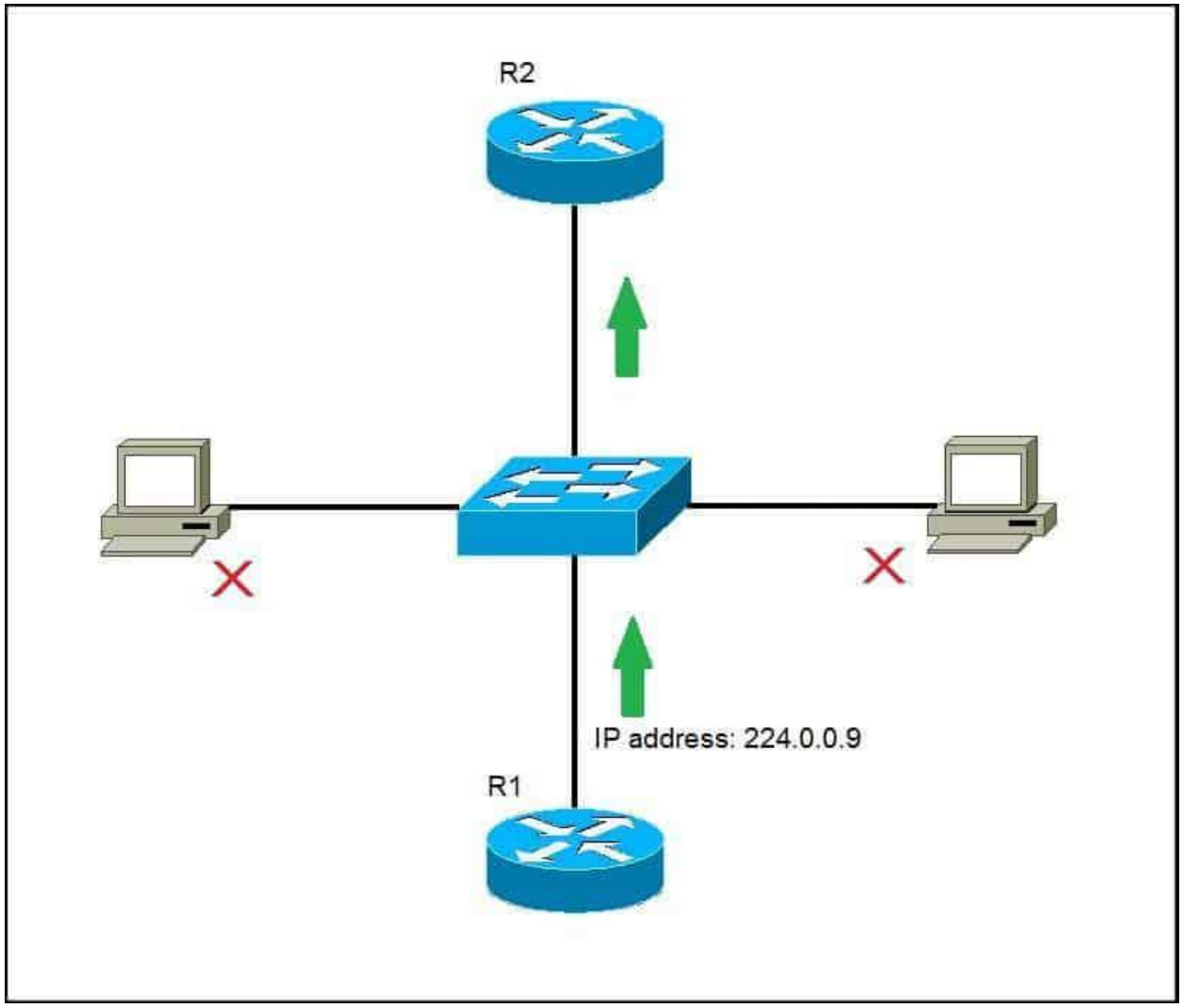
The IP addresses are divided into three different types, based on their operational characteristics:

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.

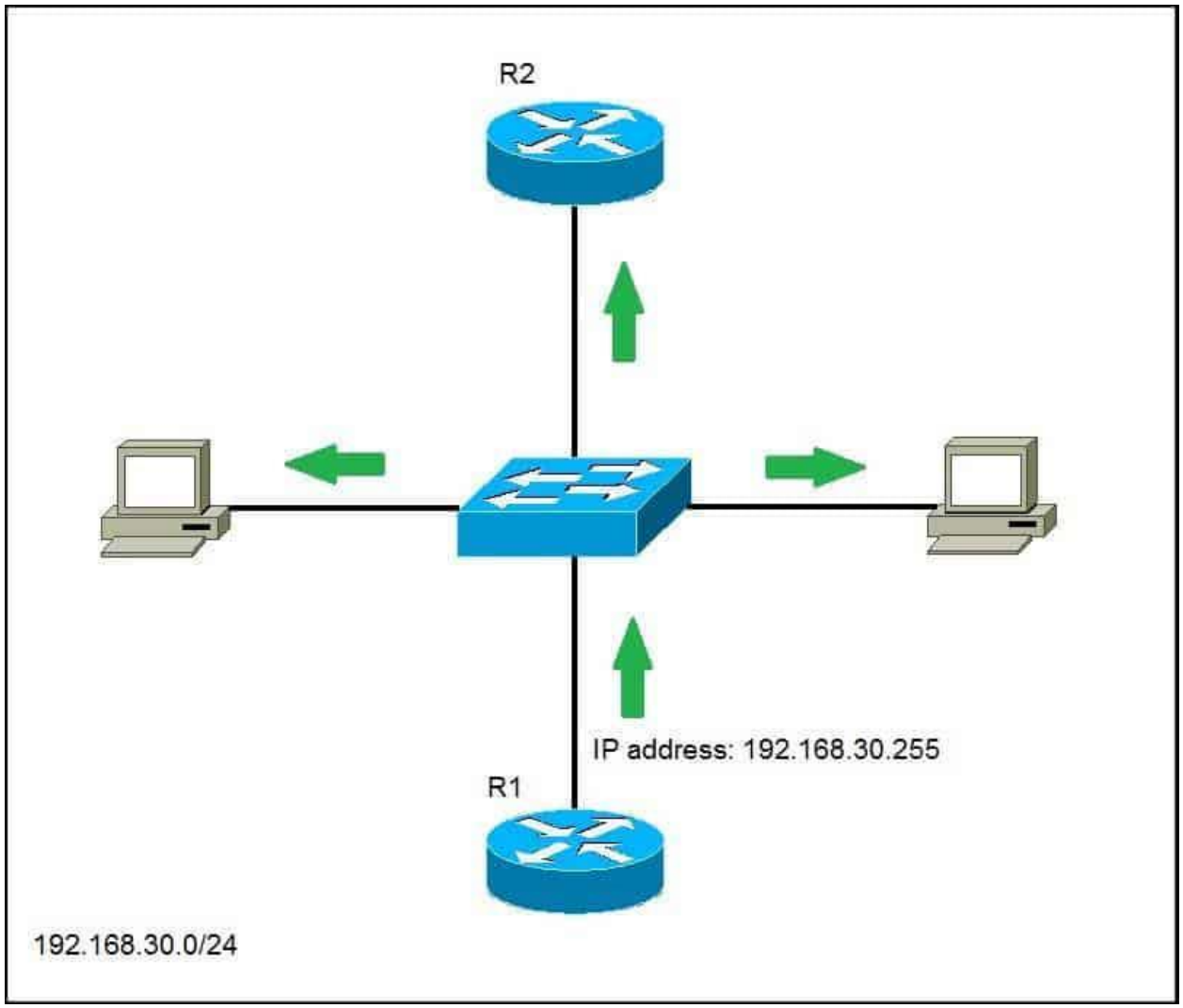
2.multicast IP addresses – used for one-to-many communication. Multicast messages are sent to IP multicast group addresses. Routers

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 192.168.30.255*. 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.

*This is because the subnet mask of 255.255.255.0 means that the last octet in the IP address represents the host bits. And 8 one’s written in decimal is 255.

Section 5

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:

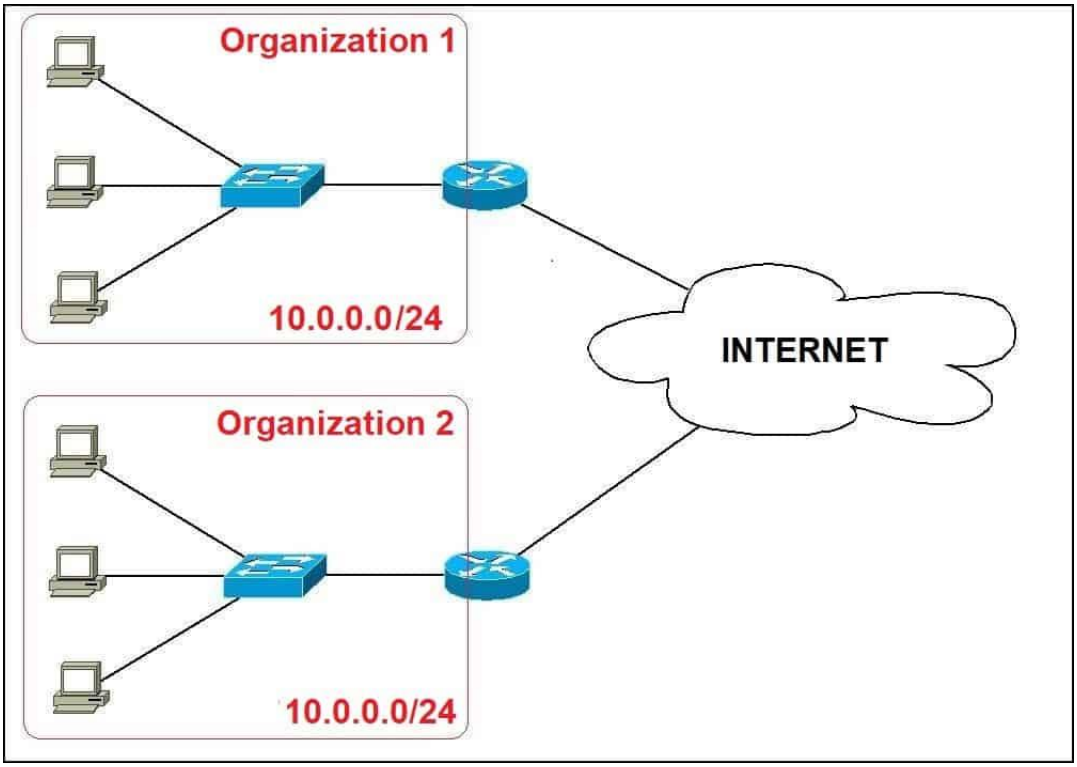
- 0.0.0.0/8 – addresses used to communicate with the local network
- 127.0.0.0/8 – loopback addresses
- 169.254.0.0/16 – link-local addresses (APIPA)

Private IP addresses explained

The original design of the Internet intended that each host on every network should have a real, routable IP address. An organization that would like to access the Internet would complete some paperwork to describe its internal network and the number of hosts on it. The organization would then receive a number of IP addresses, according to its needs. But there was one huge problem with this concept – if each host on each network in the world was provided with an unique IP address, we would have run out of IP addresses a long time ago!

Therefore, the concept of private IP addressing was developed to address the IP address exhaustion problem. The private IP addresses can be used on the private network of any organization in the world and are not globally unique.

Consider the following example:



In the example above you can see that two unrelated organizations use the same private IP network (10.0.0.0/24) inside their respective internal networks. Because private IP addresses are not globally unique, both organizations can use private IP addresses from the same range. To access the Internet, the organizations can use a technology called Network Address Translation (NAT), which we will describe in the later lessons.

There are three ranges of addresses that can be used in a private network (e.g. your home LAN or office)

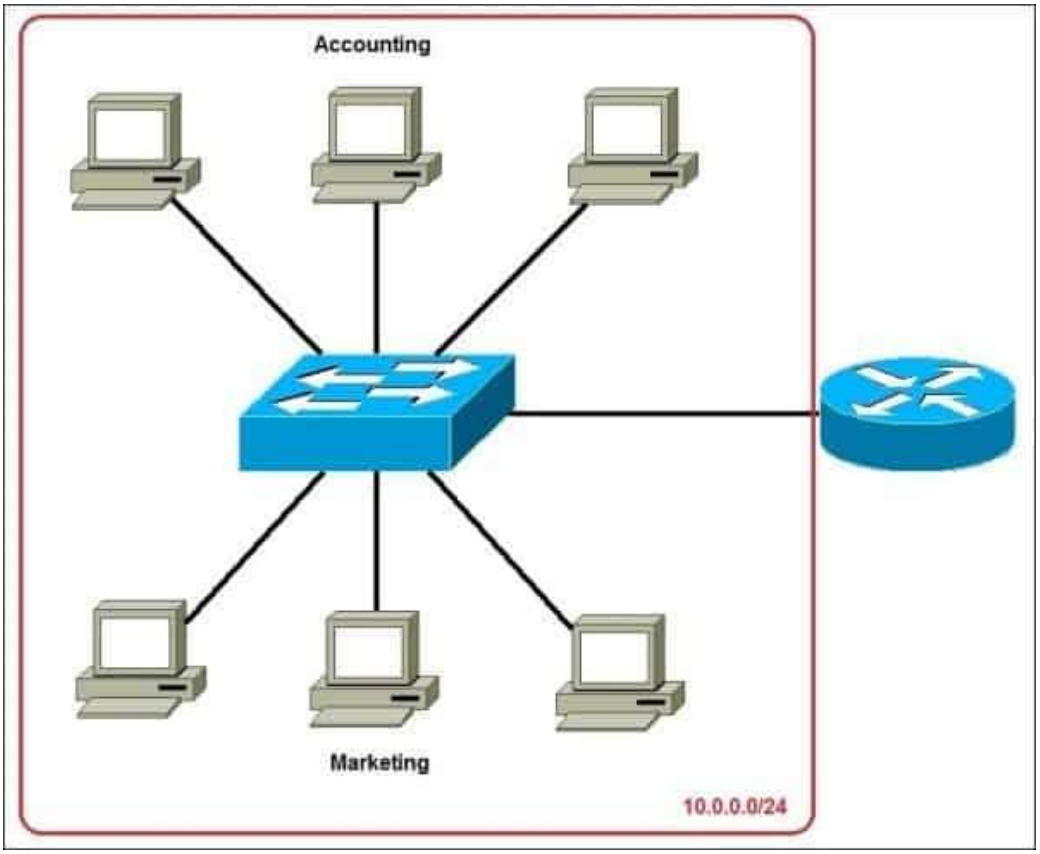
- **10.0.0.0 – 10.255.255.255**
- **172.16.0.0 – 172.31.255.255**
- **192.168.0.0 – 192.168.255.255**

Internet routers are configured to discard any packets coming from the private IP address ranges, so these addresses are not routable on the Internet.

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:

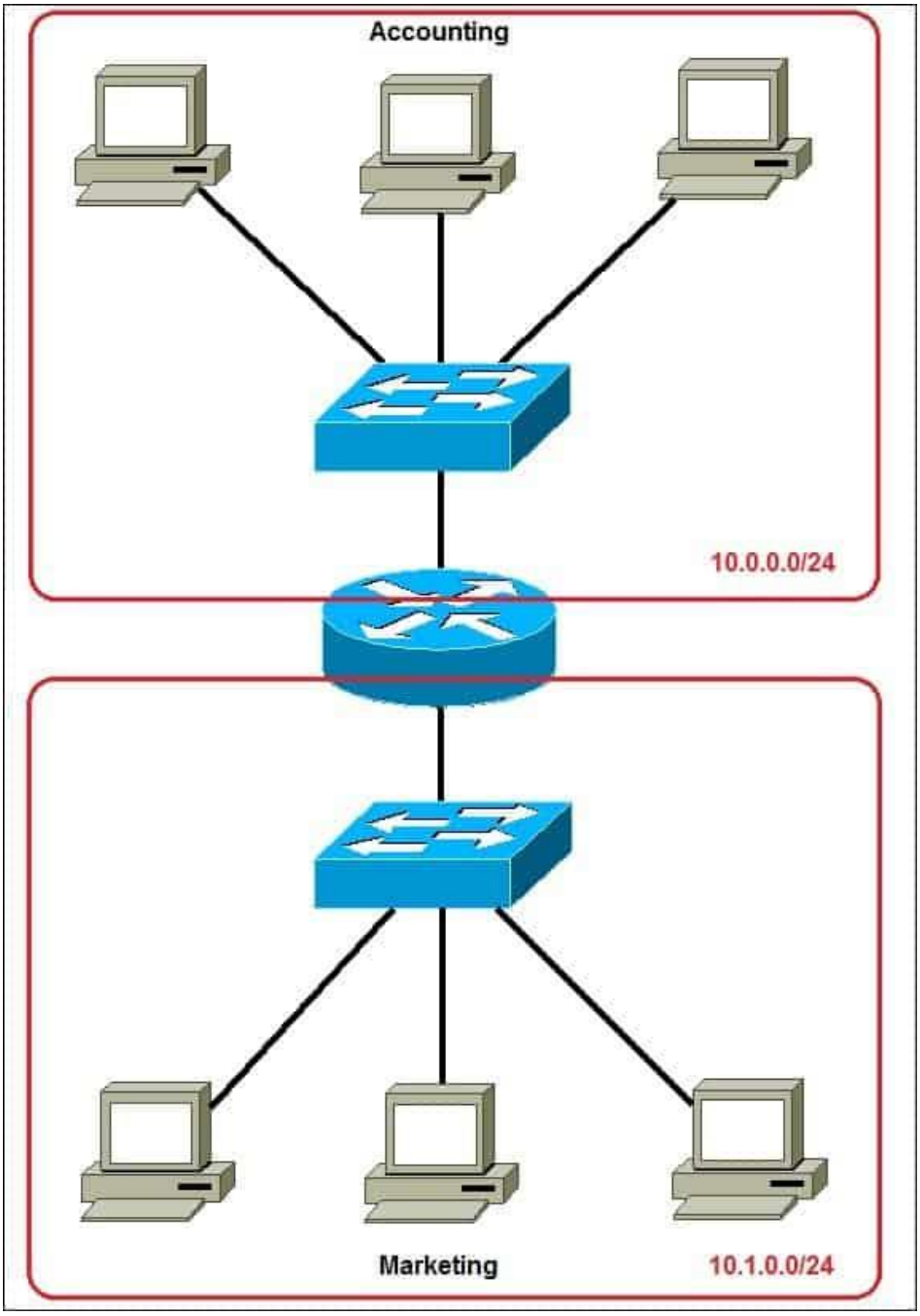


In the picture above we have one huge network: 10.0.0.0/24. All hosts on the network are in the same subnet, which has the following disadvantages:

- **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 user's workstations.
- **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 subnetted 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?

2^x = 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?

$2^y - 2$ = 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 address 192.168.0.0/24. 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 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 $2^y - 2$, 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 classsless – 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.

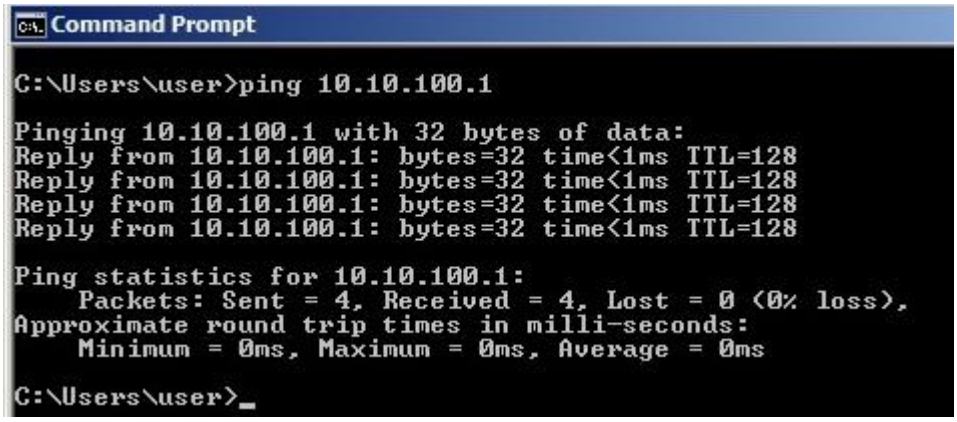
Ping explained

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 a Windows PC:



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

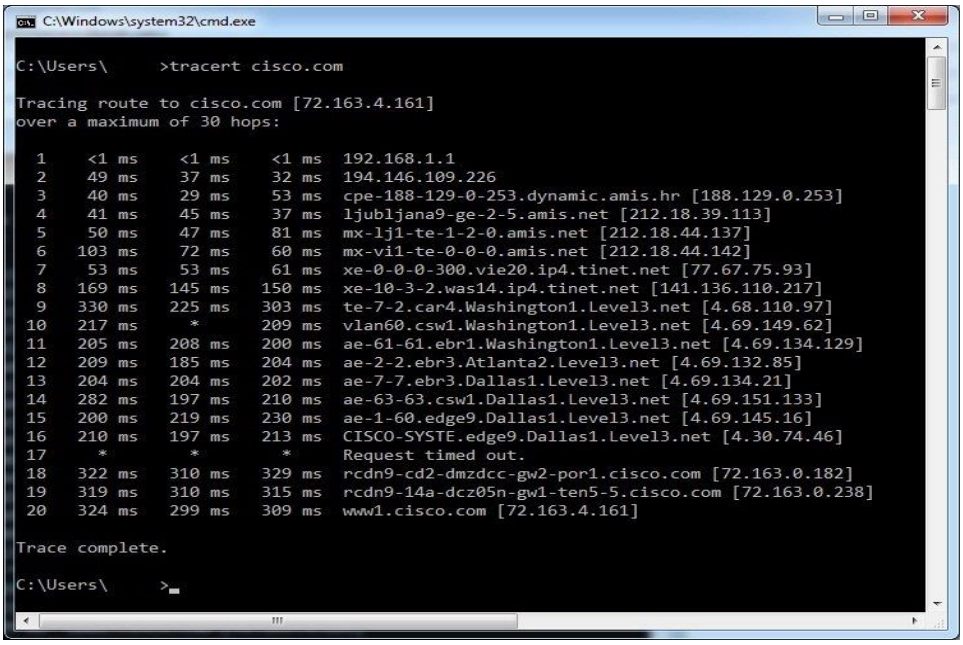
the packets. Traceroute is useful when troubleshooting network problems because it can help identify where exactly the problem is. You can figure out which router in the path to an unreachable target should be examined more closely as the probable cause of the network's failure.

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:



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 slighty 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.

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	

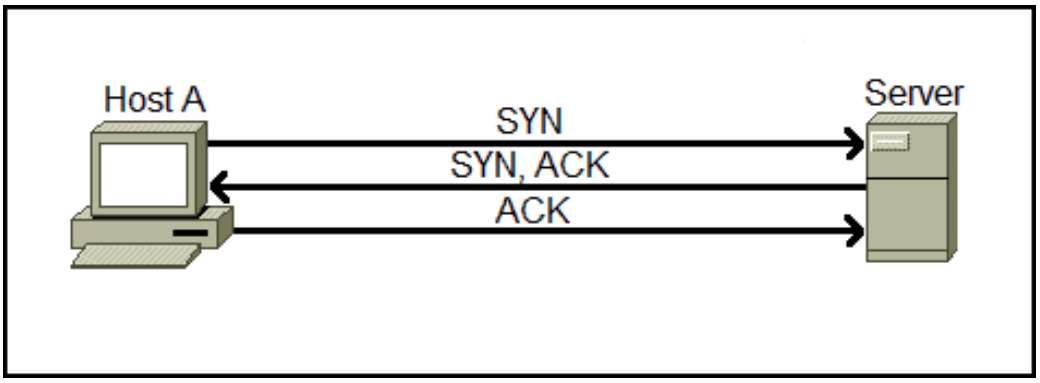
- **source port** – the port number of the application on the host sending the data.
- **destination port** – the port number of the application on the host receiving the data.
- **sequence number** – used to identify each byte of data.
- **acknowledgment number** – the next sequence number that the 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).

TCP three-way Handshake

Since TCP is a connection-oriented protocol, a connection needs to be established before two devices can communicate. TCP uses a process called three-way handshake to negotiate the sequence and acknowledgment fields and start the session. Here is a graphical representation of the process:



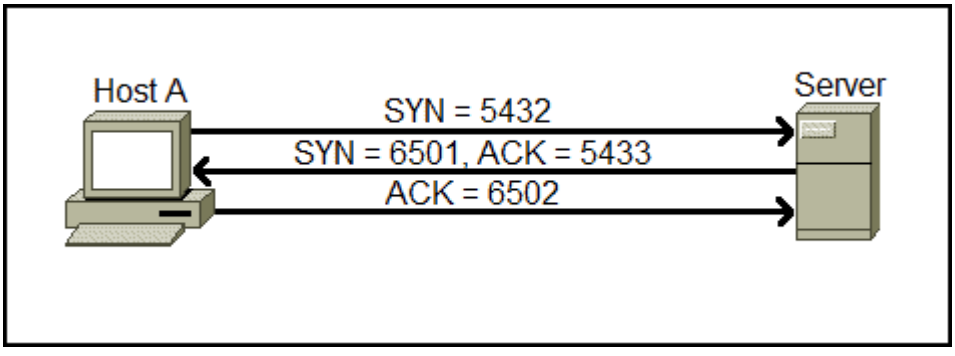
As the name implies, the three way handshake process consists of three steps:

1.Host A initiates the connection by sending the TCP SYN packet to the destination host. The packet contains the random sequence number (e.g. 5432) which marks the beginning of the sequence numbers for data that the Host A will transmit.

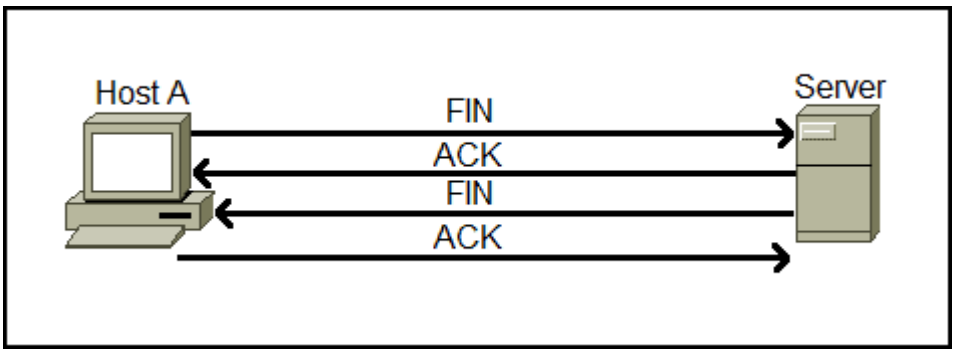
2.The Server receives the packet and responds with its own sequence number. The response also includes the acknowledgment number, which is Host A's sequence number incremented by 1 (in our case, that would be 5433).

3.Host A acknowledges the response of the Server by sending the acknowledgment number, which is the Server's sequence number incremented by 1.

Here is another picture with the numbers included:



After the data transmission process is finished, TCP will terminate the connection between two endpoints. This four-step process is illustrated below:



1. The client application that wants to close the connection sends a TCP segment with the FIN (Finished) flag set to 1.

2. The server receives the TCP segment and acknowledges it with the ACK segment.

3. Server sends its own TCP segment with the FIN flag set to 1 to the client in order to terminate the connection.

4. The client acknowledges the server's FIN segment and closes the connection.

Section 6

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.

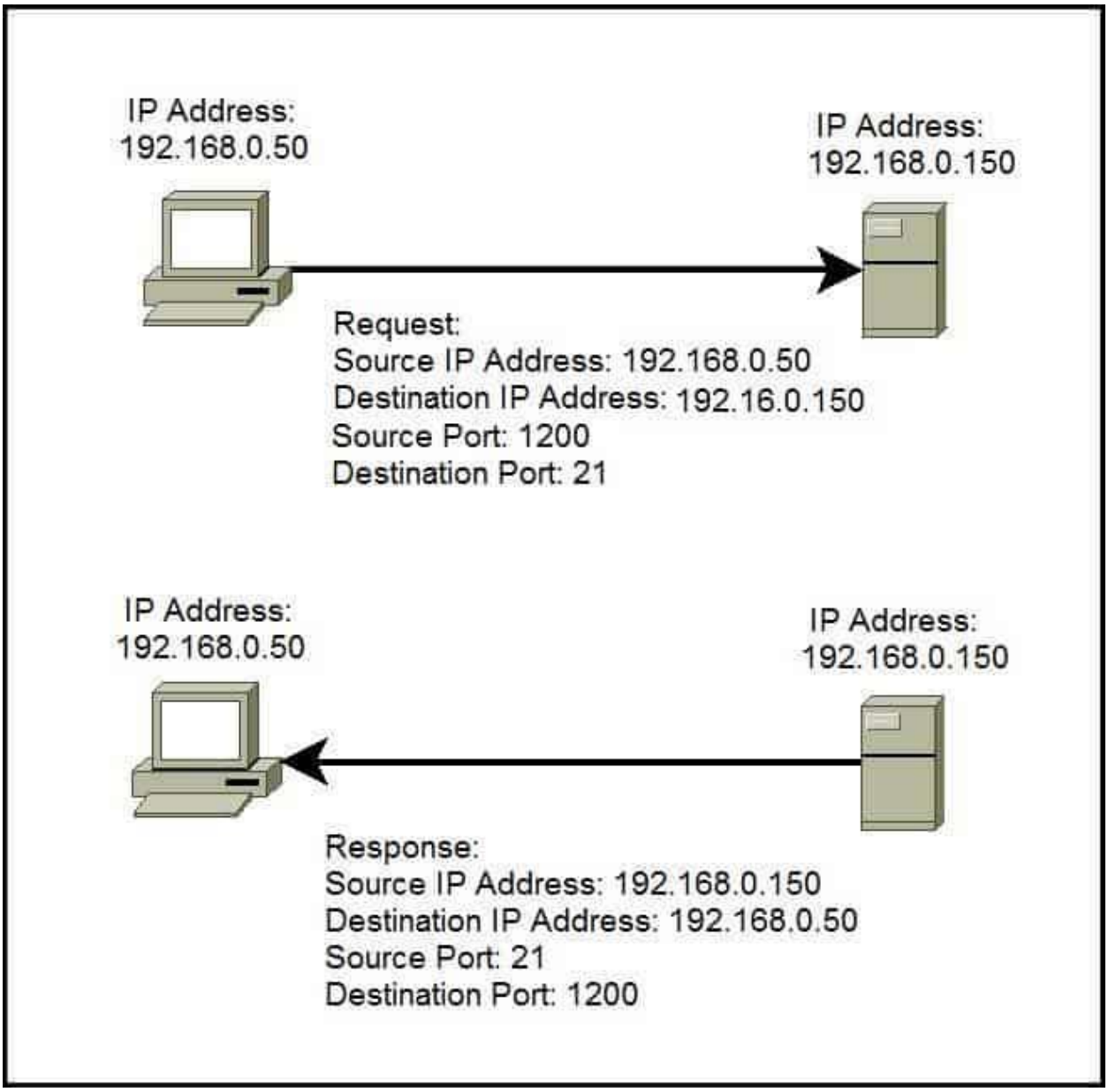
NOTE

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

Port 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	NTP	123
DNS	53	SNMP	161
HTTP	80		
POP3	110		
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.

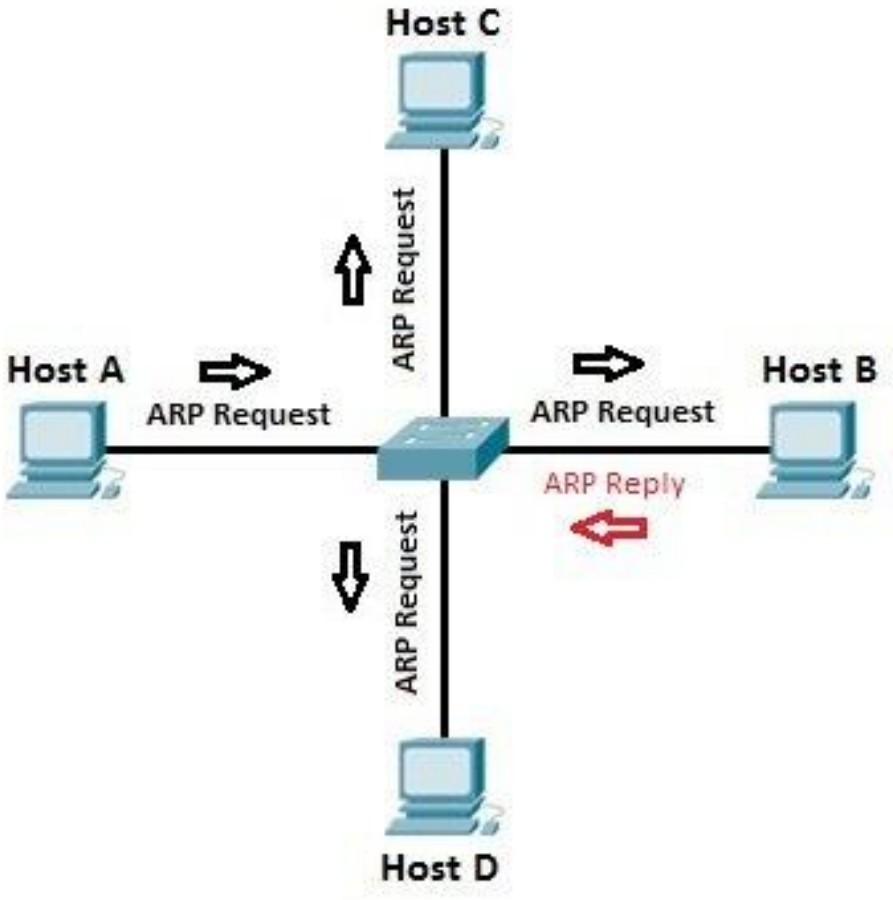
ARP (Address Resolution Protocol) explained

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).

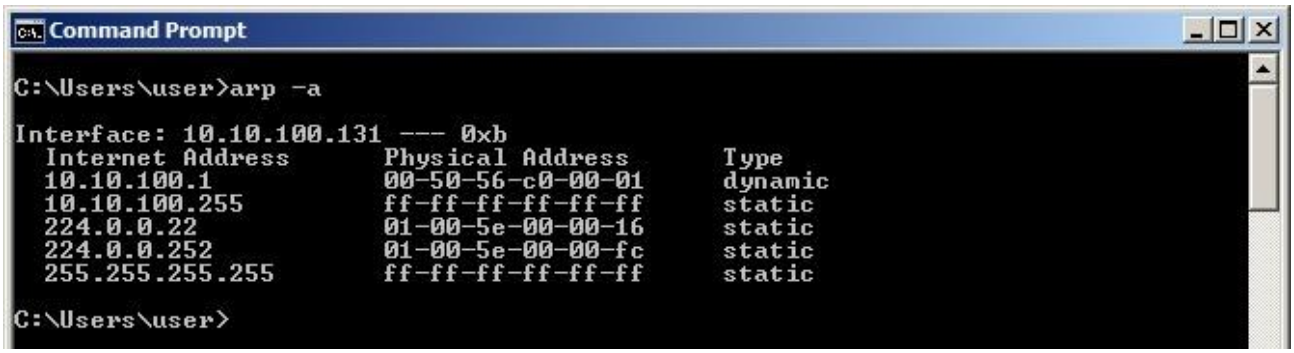
Here is the explanation otf the ARP process:



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

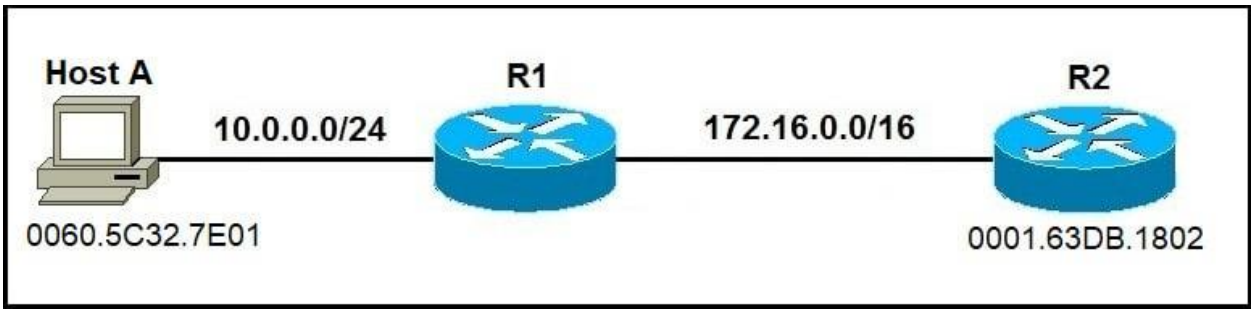
forward the frame out all interfaces (except the incoming interface). Each device on the segment will receive the packet, but because the destination IP address is host B's IP address, only host B will reply with the ARP reply packet, listing its MAC address. Host A now has enough information to send 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:



The ARP table on a Cisco router

Just like regular hosts, if a Cisco router wants to exchange frames with a host in the same subnet, it needs to know its MAC address. **The IP-to-MAC address mapping are kept in the router's ARP table**. Consider the following example:



R1 has two connected subnets – 10.0.0.0/24 and 172.16.0.0./16. Before exchanging frames with either host, R1 will need to know their MAC addresses. Here is the output of the R1's ARP table:

```
R1#show ip arp

Protocol  Address          Age (min)  Hardware Addr   Type   Interface

Internet   10.0.0.1                  -     0060.5C32.7E01   ARPA
GigabitEthernet0/0

Internet   10.0.0.10                 6     000C.85CA.AD73   ARPA
GigabitEthernet0/0

Internet   172.16.0.1                -     0060.5C32.7E02   ARPA
GigabitEthernet0/1

Internet   172.16.0.2                 10   0001.63DB.1802   ARPA
GigabitEthernet0/1
```

The ARP table contains two entries for R1's own two interfaces with the IP address of 10.0.0.1 and 172.16.0.1. The – in the age column indicates that the entry will never be timed out.

The ARP table also lists the MAC addresses of the two connected hosts. Consider the entry for Host A:

```
Protocol  Address          Age (min)  Hardware Addr   Type   Interface

Internet   10.0.0.10                 6     000C.85CA.AD73   ARPA
GigabitEthernet0/0
```

Here is a brief description of each field:

- **Protocol** – the protocol type, almost always Internet
- **Address** – the IP address associated with the MAC address, in our case the IP address of Host A
- **Age** – by default, an entry will be removed from the ARP table if it wasn't used in 240 minutes. 6 in this column means that the entry was last used 6 minutes ago. Each time an entry is used, the age will be reset back to zero.
- **Hardware** – the MAC address of the host with the corresponding IP address.
- **Type** – the type of hardware address. For Ethernet, this value will always be ARPA.
- **Interface** – the interface on R1 on which the corresponding host is connected.

Here are the steps R1 needs to take before forwarding frames to Host A:

- R1 wants to communicate with Host A. R1 checks its routing table. The subnet on which Host A resides is a directly connected subnet.
- R1 checks its ARP table to find out whether the Host A's MAC address is known. If it is not, R1 will send an ARP request to the broadcast MAC address of FF:FF:FF:FF:FF:FF.
- Host A receives the frame and sends its MAC address to R1 (ARP reply). The host also updates its own ARP table with the MAC address of the Gigabit0/0 interface on R1.
- R1 receives the reply and updates the ARP table with the MAC address of Host A.
- Since both hosts now know each other MAC addresses, the communication can occur.

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.

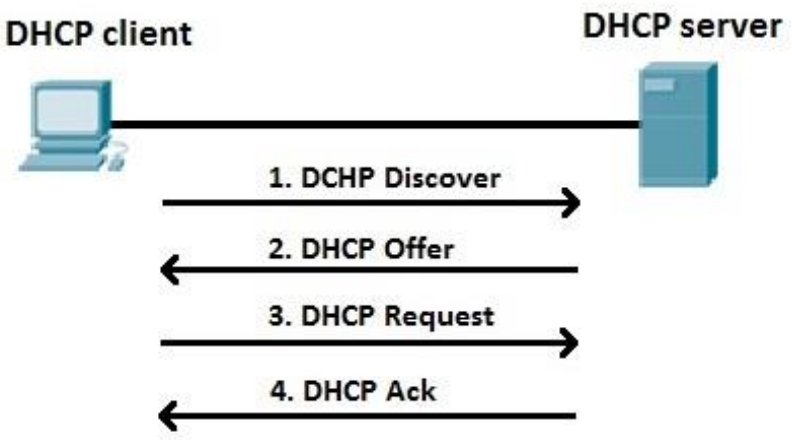
DHCP is a client-server protocol. A client is a device that is configured to use DHCP to request network parameters from a DHCP server. DHCP server maintains a pool of available IP addresses and assignes one of them to the host. A DHCP server can also provide some other parameters, such as:

- **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

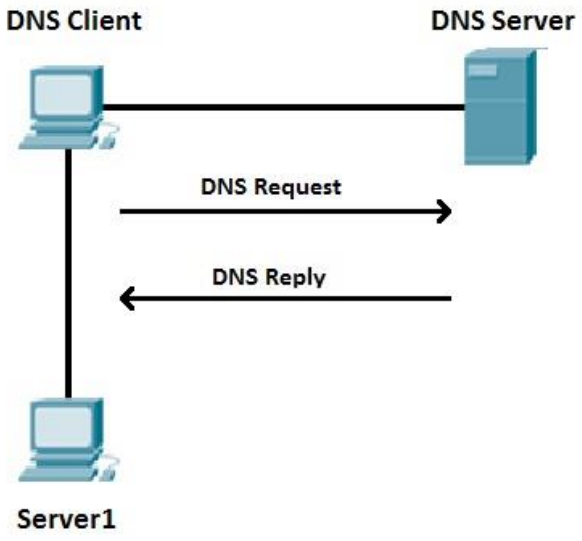
DHCP uses a well-known UDP port number 67 for the DHCP server, and the UDP port number 68 for the client.

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!**

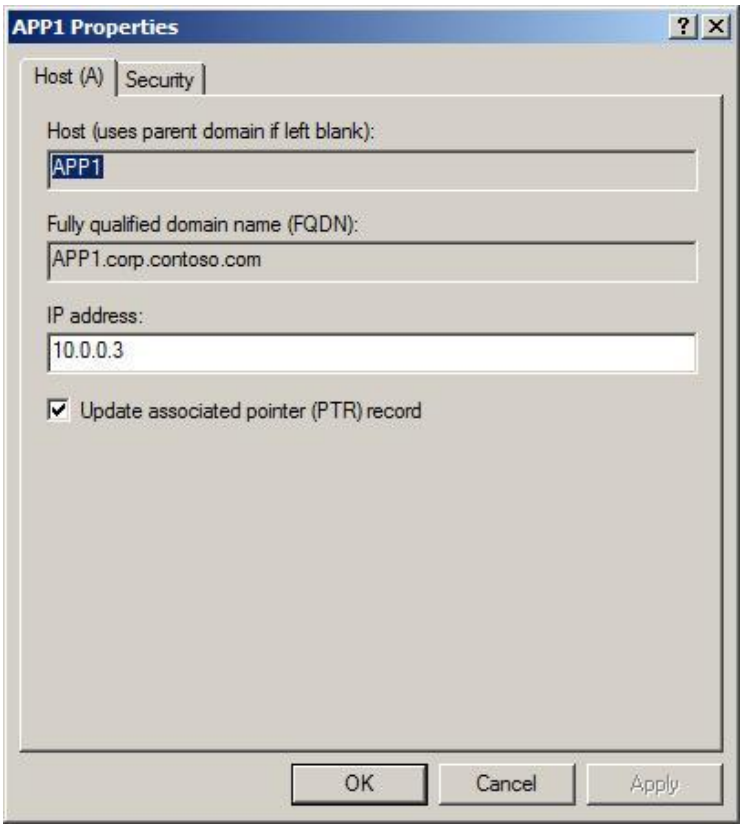
To use DNS, you must have a DNS server configured to handle the resolution process. A DNS server has a special-purpose application installed. The application maintains a table of dynamic or static hostname-to-IP address mappings. When a user request some network resource using a hostname, (e.g. by typing 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:



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.

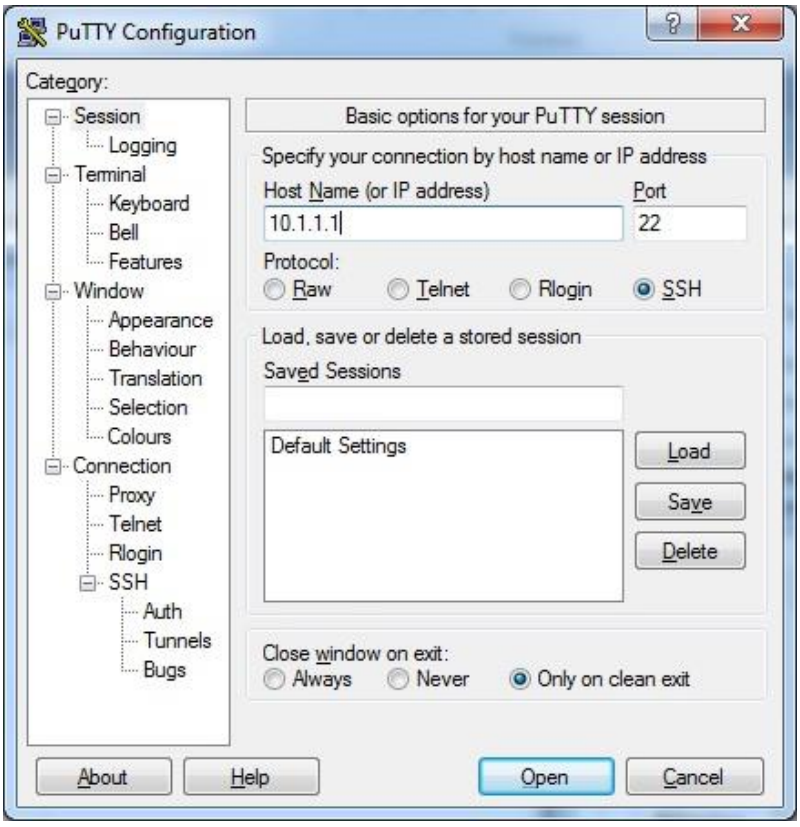
On Windows, you can start a Telnet session by typing the telnet ***IP_ADDRESS or HOSTNAME*** command:

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:



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.

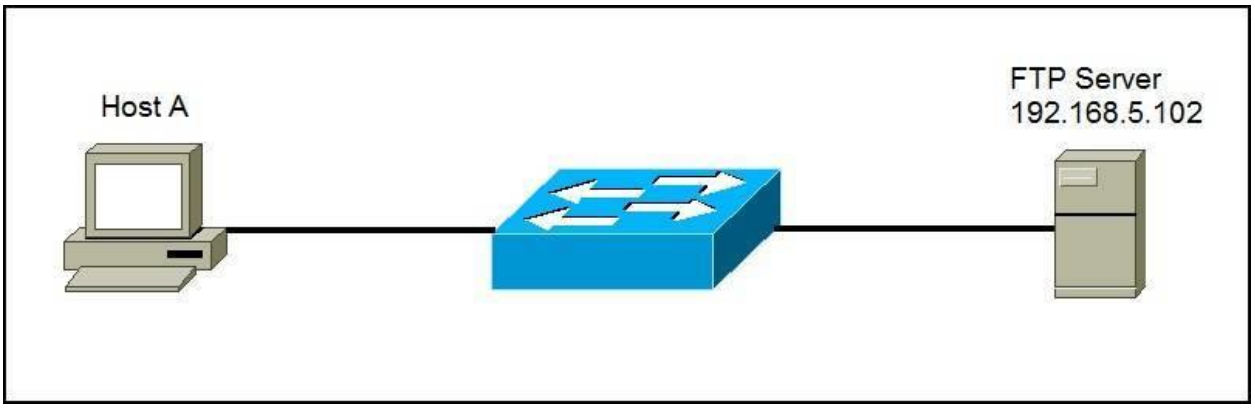
Section 7

FTP & TFTP

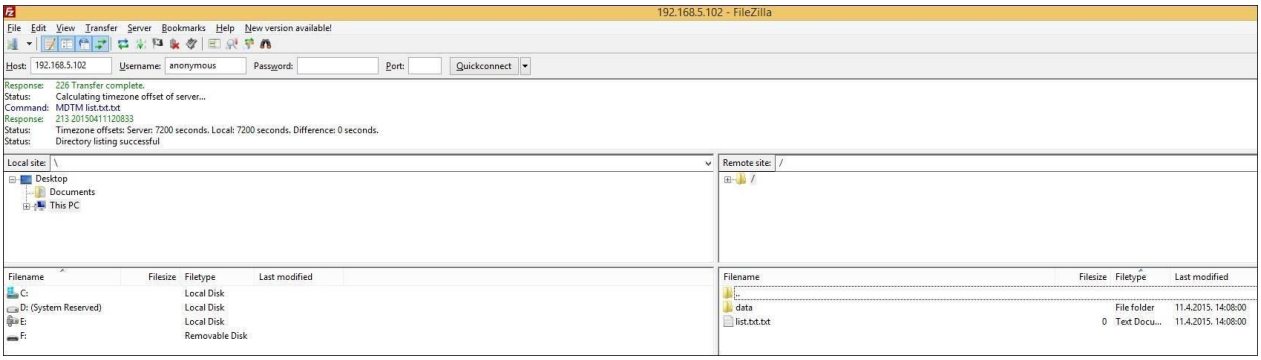
FTP (File Transfer Protocol)

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:



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

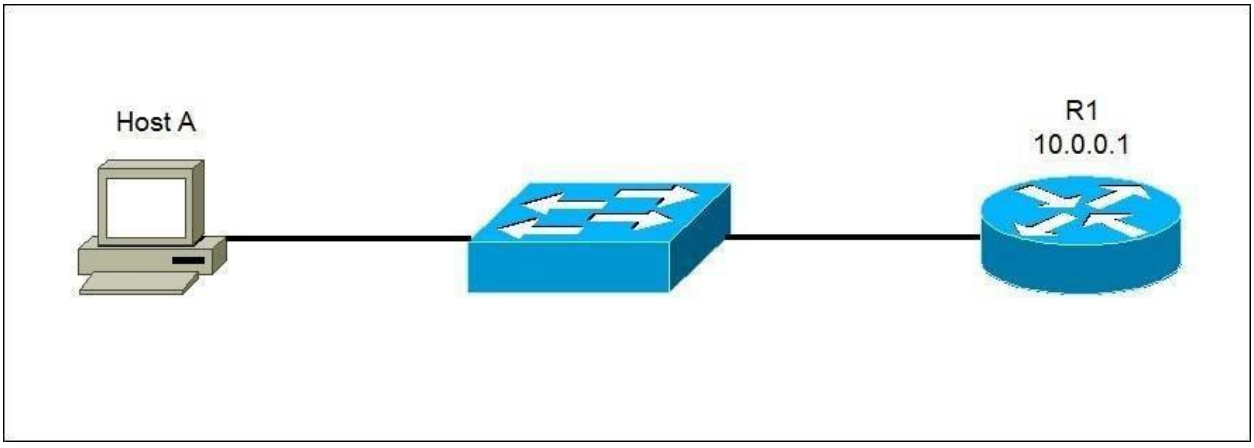
FTP uses two TCP ports: port 20 for sending data and port 21 for sending control commands. The protocol supports the use of authentication, but like Telnet, all data is sent in clear text, including usernames and passwords.

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.

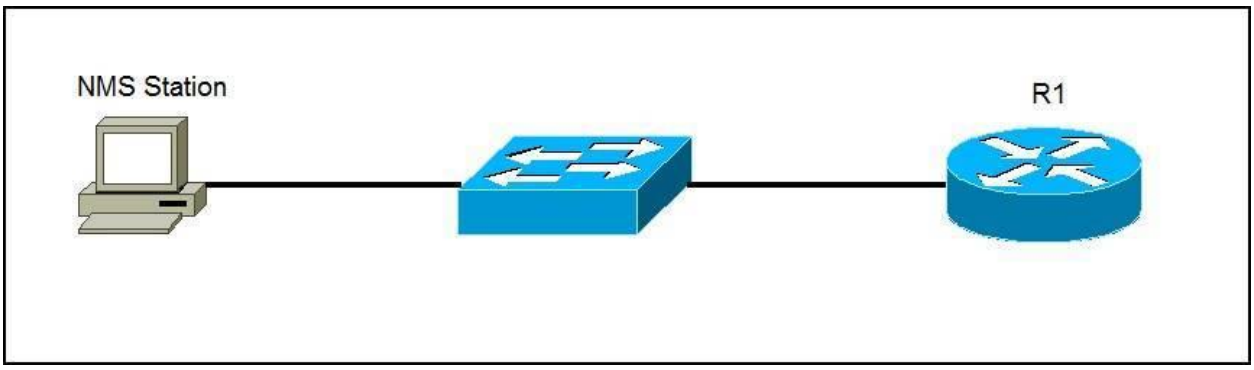
SNMP (Simple Network Management Protocol)

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.

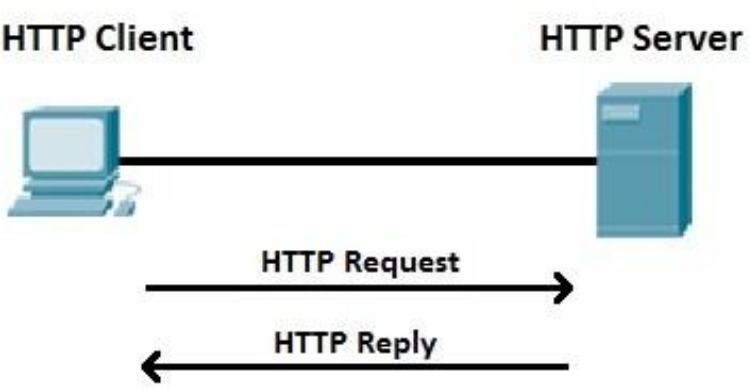
HTTP and HTTPS explained

HTTP (Hypertext Transfer Protocol)

HTTP is a 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

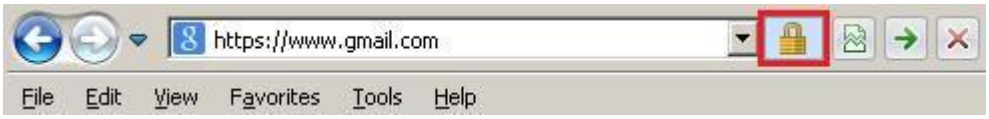
The version of HTTP most commonly used today is HTTP/1.1. A newer version, HTTP/2, is available and supported by most browser.

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 unencryped 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:



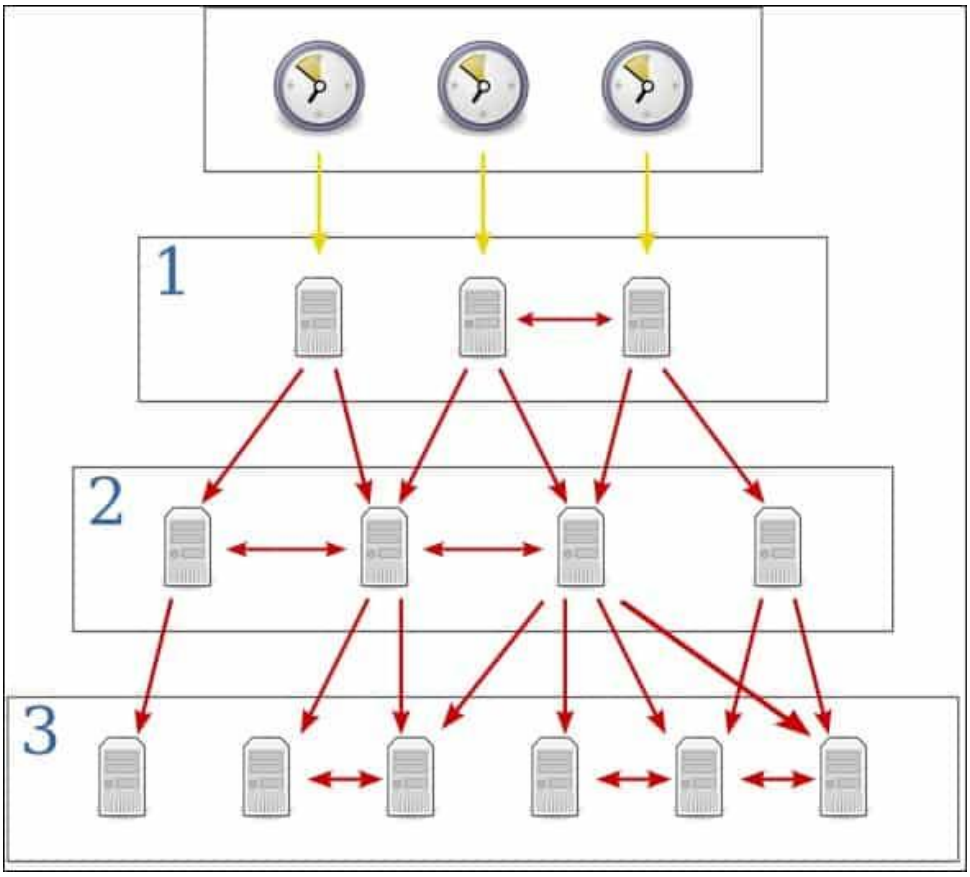
NOTE

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 and https://gmail.com:443.

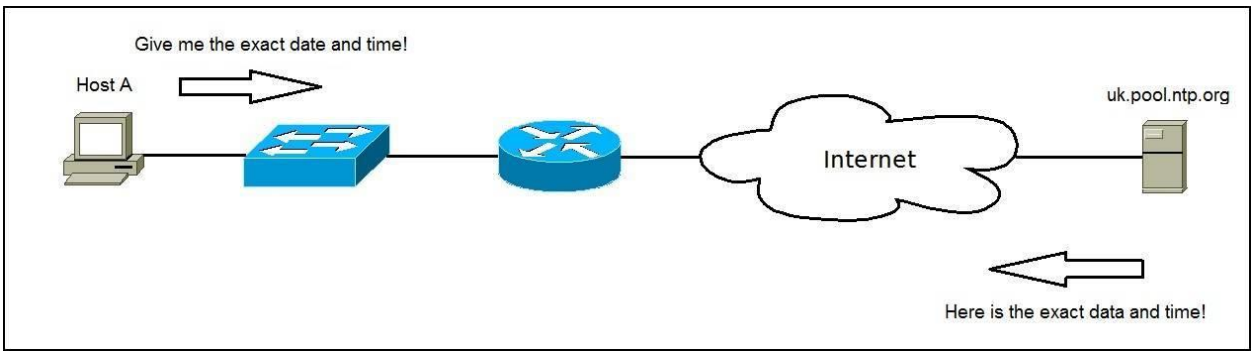
NTP (Network Time Protocol)

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:



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.

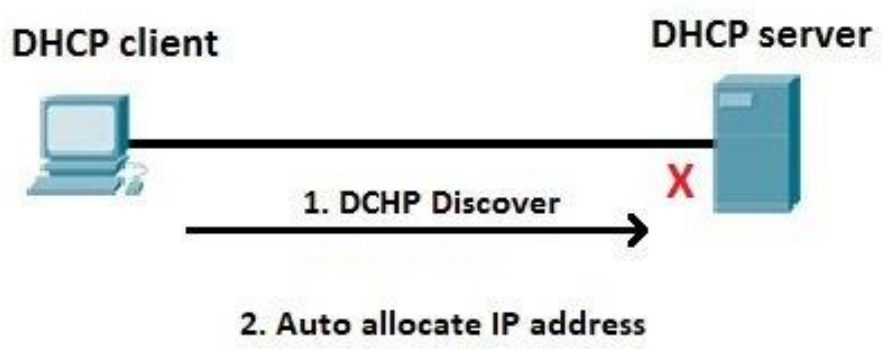
Section 8

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

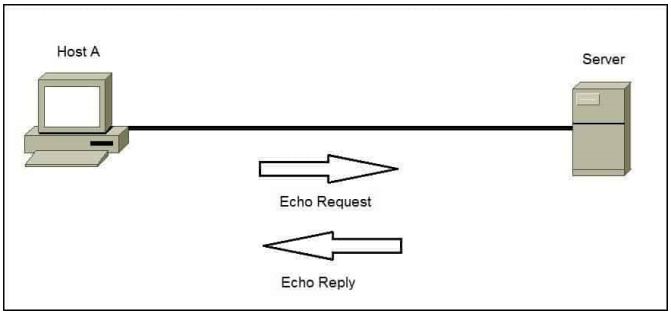
If your host is using an IP address from the APIPA range, there is usually a problem on the network. Check the network connectivity of your host and the status of the DHCP server.

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:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

```
Ping statistics for 192.168.8.11:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
```

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:

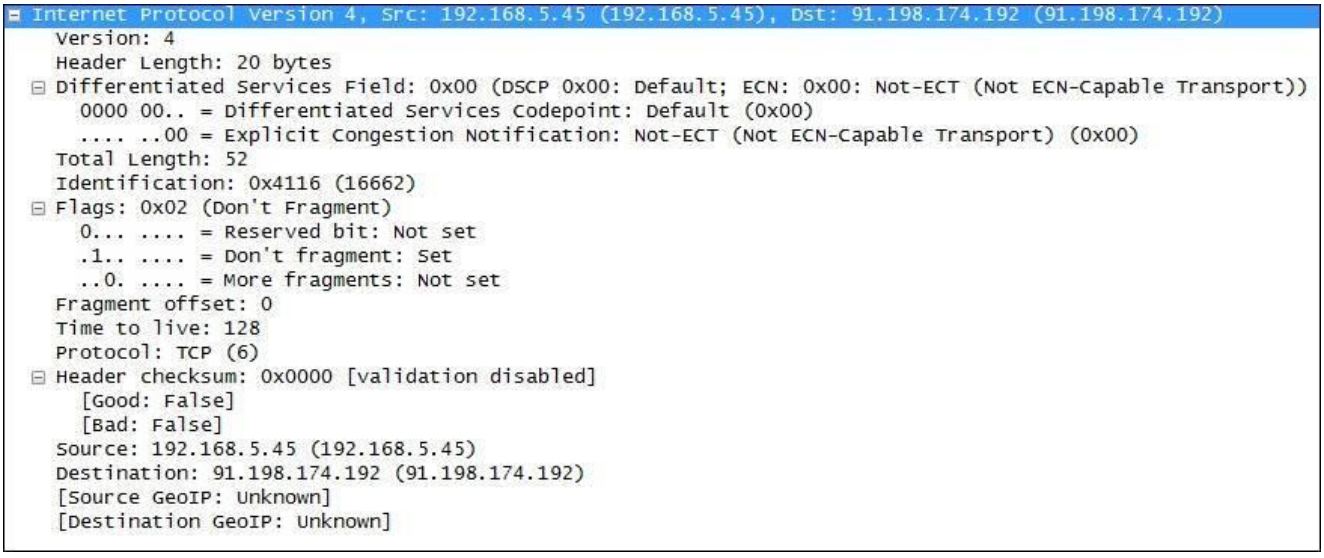
Version (4 bits)	Header length (4 bits)	Priority and Type of Service (8 bits)	Total length (16 bits)
Identification (16 bits)		Flags (3 bits)	Fragmented offset (13 bits)
Time to live (8 bits)	Protocol (8 bits)	Header checksum (16 bits)	
Source IP address (32 bits)			
Destination IP address (32 bits)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



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.

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 CatOS, a legacy version of a CLI-based operating sytem.

Below you can see how IOS looks like when a Cisco device is started for the first time, using a 3745 router as an example:

```
Cisco 3745 (R7000) processor (revision 2.0) with 249856K/12288K bytes
of memory.

Processor board ID FTX0945W0MY

R7000 CPU at 350MHz, Implementation 39, Rev 2.1, 256KB L2, 512KB L3
Cache

5 FastEthernet interfaces

DRAM configuration is 64 bits wide with parity enabled.

151K bytes of NVRAM.
```

```
Press RETURN to get started!
```

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.

NOTE

Newer Cisco devices usually include a USB console port, since serial ports are rare on modern PCs.

2. **Telnet access** – this type of access used to be a common way to access network devices. Telnet is a 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 a 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 and many submodes. We will describe the three main modes and one submode.

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

We'll describe each of the modes mentioned above in more detail in the following lectures.

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 ---
```

```
Would you like to enter the initial 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: secret

  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: cisco
```

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

Current interface summary

Interface                  IP-Address      OK? Method Status
Protocol

GigabitEthernet0/0      unassigned      YES manual administratively
down down

GigabitEthernet0/1      unassigned      YES manual administratively
down down

GigabitEthernet0/2      unassigned      YES manual administratively
down down

Vlan1                   unassigned      YES manual administratively
down down

Enter interface name used to connect to the
management network from the above interface summary:
GigabitEthernet0/2

Configuring interface GigabitEthernet0/2:
  Configure IP on this interface? [yes]: yes
    IP address for this interface: 192.168.0.1
```

```
    Subnet mask for this interface [255.255.255.0] :


The following configuration command script was created:


!
hostname R1
enable secret 5 $1$mERr$5jbOD5lHVUWxAAsNOD6eO/
enable password cisco
line vty 0 4
password cisco
!
interface Vlan1
 shutdown
 no ip address
!
interface GigabitEthernet0/0
 shutdown
 no ip address
!
interface GigabitEthernet0/1
 shutdown
 no ip address
!
interface GigabitEthernet0/2
 no shutdown
 ip address 192.168.0.1 255.255.255.0
!
```

```
end


[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.


Enter your selection [2]: 2
Building configuration...
[OK]
Use the enabled mode 'configure' command to modify this configuration.



Press RETURN to get started!
```

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.

Section 9

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.

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 submodes, 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
```

Submode commands

A global configuration mode contains many submodes. For example, if you want to configure an interface you have to enter that interface configuration mode. Each submode 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, FastEthernet, 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 submode has its own prompt. Notice how the command prompt was changed to Router(config-if) after I've entered the interface submode:

```
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#?
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#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#debug ?
  aaa            AAA Authentication, Authorization and Accounting
  custom-queue   Custom output queueing
  eigrp          EIGRP Protocol information
  frame-relay    Frame Relay
  ip             IP information
  ipv6           IPv6 information
  ntp            NTP information
  ppp            PPP (Point to Point Protocol) information
  standby        Hot Standby Router Protocol (HSRP)


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 the longer version, *show ip interface brief*:

```
Router#sh ip int b

Interface                  IP-Address      OK? Method Status
Protocol

GigabitEthernet0/0      unassigned      YES NVRAM  administratively
down down

GigabitEthernet0/1      unassigned      YES NVRAM  administratively
down down

GigabitEthernet0/2            192.168.0.1            YES  manual  up
down

Vlan1                   unassigned      YES NVRAM  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#sh ip in b

% Ambiguous command: "sh 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 submode 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.

- **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:

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!.**

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 nam**e – 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 dispay 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.

Section 10

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 submode:

```
Router(config)#interface g0/0
Router(config-if)#description WAN to London
```

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

```
!
interface GigabitEthernet0/0
 description WAN to London
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface GigabitEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
!
```

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

```
Router(config)#int g0/0

Router(config-if)#no desc
```

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 submodes. 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:

```
Router(config)#int g0/0

Router(config-if)#show interface g0/0
                     ^

% Invalid input detected at '^' marker.


Router(config-if)#
```

In the example above you can see that we're currently in the interface submode. 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:

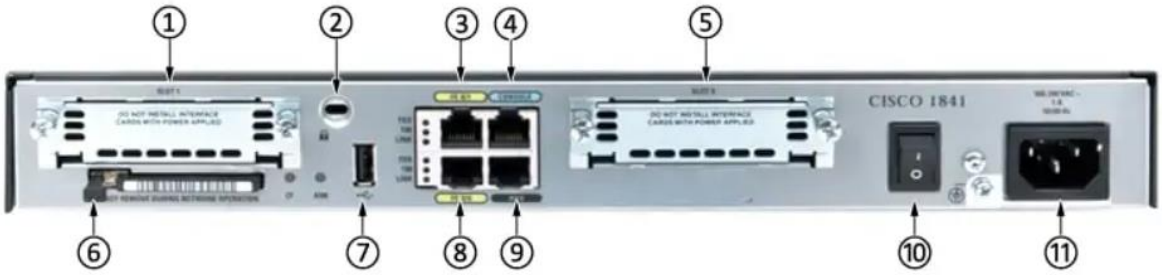
```
Router(config-if)#do show interface g0/0

GigabitEthernet0/0 is administratively down, line protocol is down
(disabled)

  Hardware is CN Gigabit Ethernet, address is 0030.a3ab.1601 (bia
0030.a3ab.1601)

  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
```

```
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, media type is RJ45
  output flow-control is unsupported, input flow-control is
unsupported
  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 watchdog, 1017 multicast, 0 pause input
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
 --More--
```

The command was now executed because of the do keyword. Notice that we're still in the interface submode 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 a switch. Let's look at the Cisco 1841 router as an example:



1. **Slot 1 Network Card expansion slot** – you can buy and install an additional interface card of various types to fit in here.
2. **Kensington Security Slot** – you can physical secure the router with a cable here to help prevent theft.
3. **Fast Ethernet port 0/1 and status indicator LED.**
4. **Console port** – you can connect directly to the router's management command line interface here via your laptop and a console cable.
5. **Slot 0 Network Card expansion slot** – another slot for additional interface cards. Note the numbering is from right to left.
6. **CompactFlash memory card slot** – the IOS operating system image lives here.
7. **USB port** – You can plug in a USB drive here to move files to and from the router.

8. **Fast Ethernet port 0/0 and status indicator LED**.
9. **Aux port** – You can connect a legacy modem here for out of band (outside the normal network path) management. Not commonly used in modern networks.
10. **On/Off switch**.
11. **Input power socket**.

To display basic information about the device interfaces in IOS, use the *show ip interface brief* command from the privileged exec mode. This is one of the most commonly used commands on Cisco devices:

```
Router#sh ip int brief

Interface                  IP-Address      OK? Method  Status
Protocol

FastEthernet0/0         192.168.0.1     YES manual administratively
down down

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:

```
Router#sh ip int brief

Interface                  IP-Address      OK? Method  Status
Protocol

FastEthernet0/0         192.168.0.1     YES manual administratively
down down
```

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 submode, where we can configure various interface options:

```
Router(config)#interface f0/0

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:

```
Router(config)#int f0/0
Router(config-if)#no shutdown


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
```

Use the show ip interface brief command to check the device's IP addresses and status of its interfaces:

```
Router#sh ip int brief
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0        192.168.0.1     YES manual up                    up
FastEthernet0/1          unassigned      YES unset  administratively
down down
Vlan1                    unassigned      YES unset  administratively
down down
```

Pipe character in IOS

IOS supports the use of the **pipe character** (represented with the | character) to filter the output of the show and more 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 few examples:

```
R1#show running-config | begin interface
interface FastEthernet0/0
 ip address 10.10.10.1 255.255.255.0
 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 FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
 --More--
```

In the picture above you can see that we've entered the show *running-config | begin interface* command (we could have abbreviated it to *show run | b int*). This command starts the output from the first occurence of the word interface.

Another example, this time with include:

```
R1#show run | include password
no service password-encryption
enable password cisco
 password cisco
 password cisco
```

As you can see from the example above, the include function displays only lines that include the word password. The include function is helpful in some situations but can also be confusing because it only includes exact matching commands with no context around them, as in this example with password cisco shown twice in the output.

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

```
R1#show run | section vty
line vty 0 4
 password cisco
 login
line vty 5 15
 password cisco
 login
```

You can see in the example above that the command displayed only the vty section of the running configuration. The section function is not

supported for all parts of the configuration, but can be very helpful for example to view all the configuration for a particular routing protocol:

```
R1#sh run | sec ospf
 ip ospf cost 100
router ospf 1
 log-adjacency-changes
 passive-interface FastEthernet0/0
 network 10.10.0.0 0.0.255.255 area 0
```

NOTE

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

Extended ping command

The ping command in Cisco IOS (and other operating systems) is used to test the accessibility of devices on a TCP/IP network. Cisco devices also support the extended ping command that allows you to perform a more advanced check of the host reachability and network connectivity. With this command, you can define the source IP address as any IP address on the router, number and size of ping packets, different timeout interval, etc.

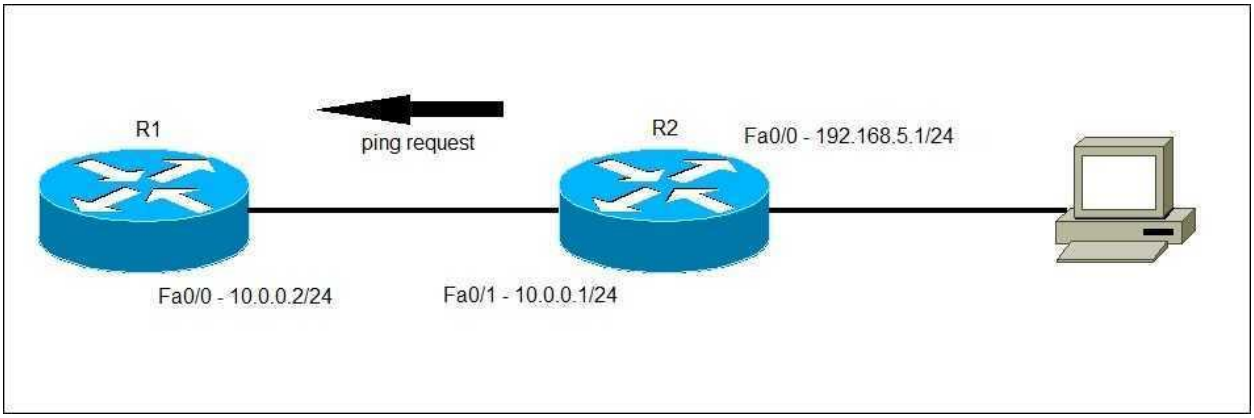
The extended ping command is invoked from the privileged exec mode by typing ping and pressing Enter. The following arguments can be modified:

- **Protocol [ip]** – specify the protocol, such as appletalk, clns, ip, novell, apollo, vines, decnet, or xns. The default is ip.

- **Target IP address** – specify the IP address or the hostname of the host to ping.
- **Repeat count** – specify the number of ping packets that will be sent to the destination address. 5 by default.
- **Datagram size** – specify the size of the ping packet (in bytes). The default is 100 bytes.
- **Timeout in seconds** – specify the timeout interval. The default is 2 seconds. The echo reply needs to be received before the timeout expires in order for ping to be successful.
- **Extended commands** – specify whether or not a series of additional commands will appear. The default is no. If you type yes additional arguments will be shown.
- **Source address or interface** – specify the interface or the IP address of the router to use as the source address for the ping packets.
- **Type of service** – specifies the Type of Service (ToS). This is the Internet service's quality selection. The default is 0.
- **Set DF bit in IP header?** – specify whether or not the Don't Fragment (DF) bit will be set on the ping packet. If yes is entered, the Don't Fragment option does not allow the packet to be fragmented. The default is no.
- **Validate reply data?** – specify whether or not to validate the reply data. The default is no.
- **Data pattern** – specify the data pattern. Data patterns are used to troubleshoot framing errors and clocking problems on serial lines. The default is [0xABCD].
- **Loose, Strict, Record, Timestamp, Verbose** – specify the IP header options.

- **Sweep range of sizes** – specify the sizes of the ping echo packets that are sent. This parameter is used to determine the minimum sizes of the MTUs configured on the nodes along the path to the destination address. The default is no.

The extended ping command is most often used to change the source IP address of the ping echo packets. Consider the following example:



By default, routers choose the IP address of the outgoing interface as the source IP address for ping echo packets. This means that R2 will use the IP address of the Fa0/1 interface (10.0.0.1) as the source IP address for the ping packets sent to 10.0.0.2 (R1). Let's try to ping R1 using the standard ping command:

```
R2#ping 10.0.0.2


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms
```

As you can see from the output, the ICMP replies were received. However, I can run the extended ping command to change the source IP

address to the IP address of the R2 Fa0/0 interface (192.168.5.1). This is done to ensure that R1 knows about the 192.168.5.1 network (in other words, that it knows where to send packets destined for the 192.168.5.0/24 network, which could indicate routing problems).

```
R2#ping
Protocol [ip]:
Target IP address: 10.0.0.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: yes
Source address or interface: 192.168.5.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.5.1
.....
Success rate is 0 percent (0/5)
```

In the output above you can see that no echo replies were received after I've changed the source IP of the ping packets. This means that R1 doesn't know how to reach the 192.168.5.0/24 network.

Types of memory on a Cisco device

Cisco devices usually have four types of memory that are being used for different purposes. These four types are:

- **ROM (Read-only memory)** – stores a bootstrap program that is used to initialize a boot process. This is a read-only type of memory, so it can't be altered.
- **RAM (Random Access Memory)** – the running configuration and routing tables of the device are stored here. This type of memory loses its content when a device is restarted.
- **Flash memory** – used to store IOS software images. Can also be used to store other files, for example backup configuration files. Retains its content even after a device is restarted.
- **NVRAM (Nonvolatile RAM)** – usually used to store a startup configuration file. This type of memory retains its content even after a device is powered down or restarted.

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.

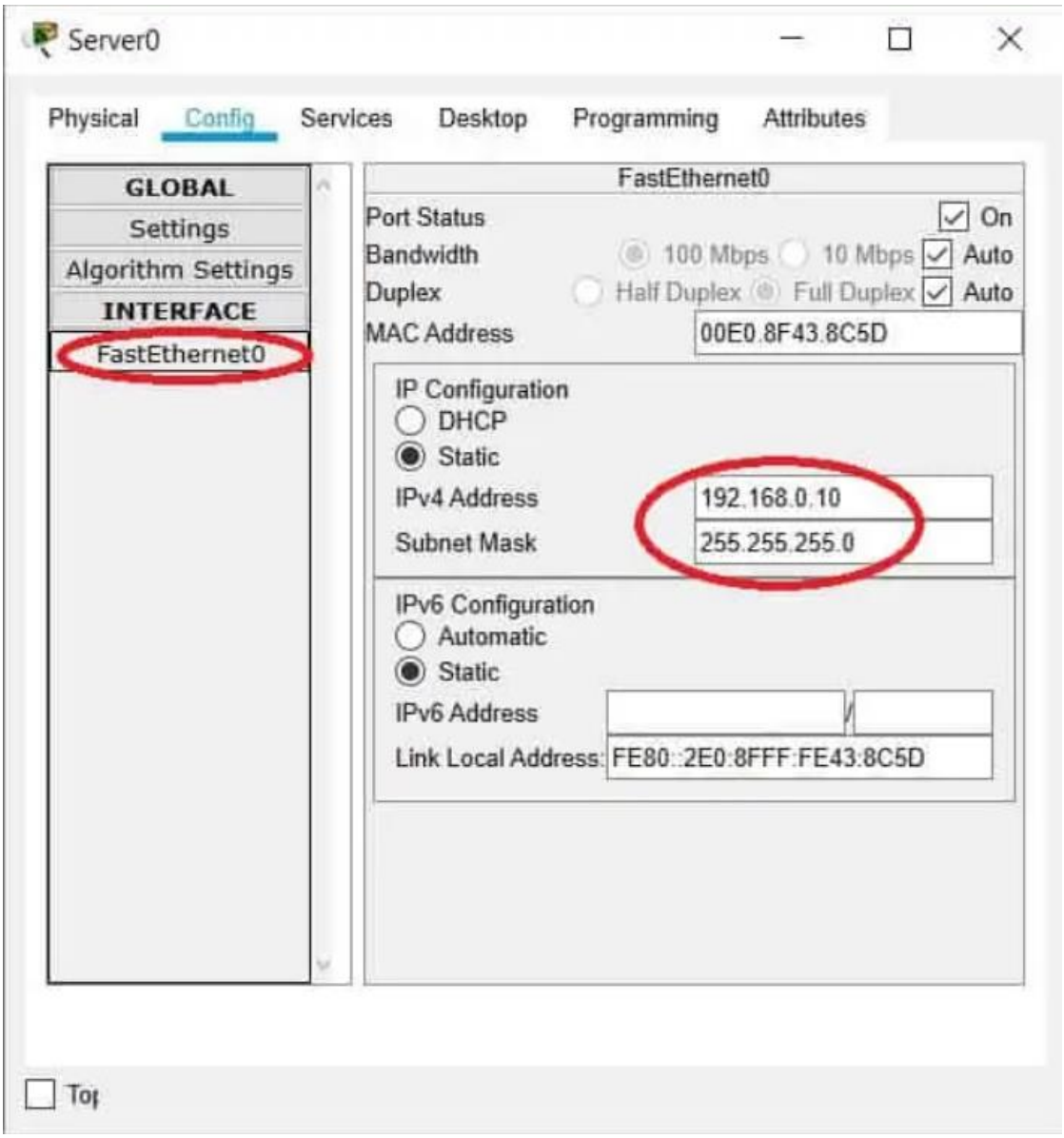
3. The bootstrap program decides which IOS image to load from the flash memory into RAM, and then loads the chosen IOS.

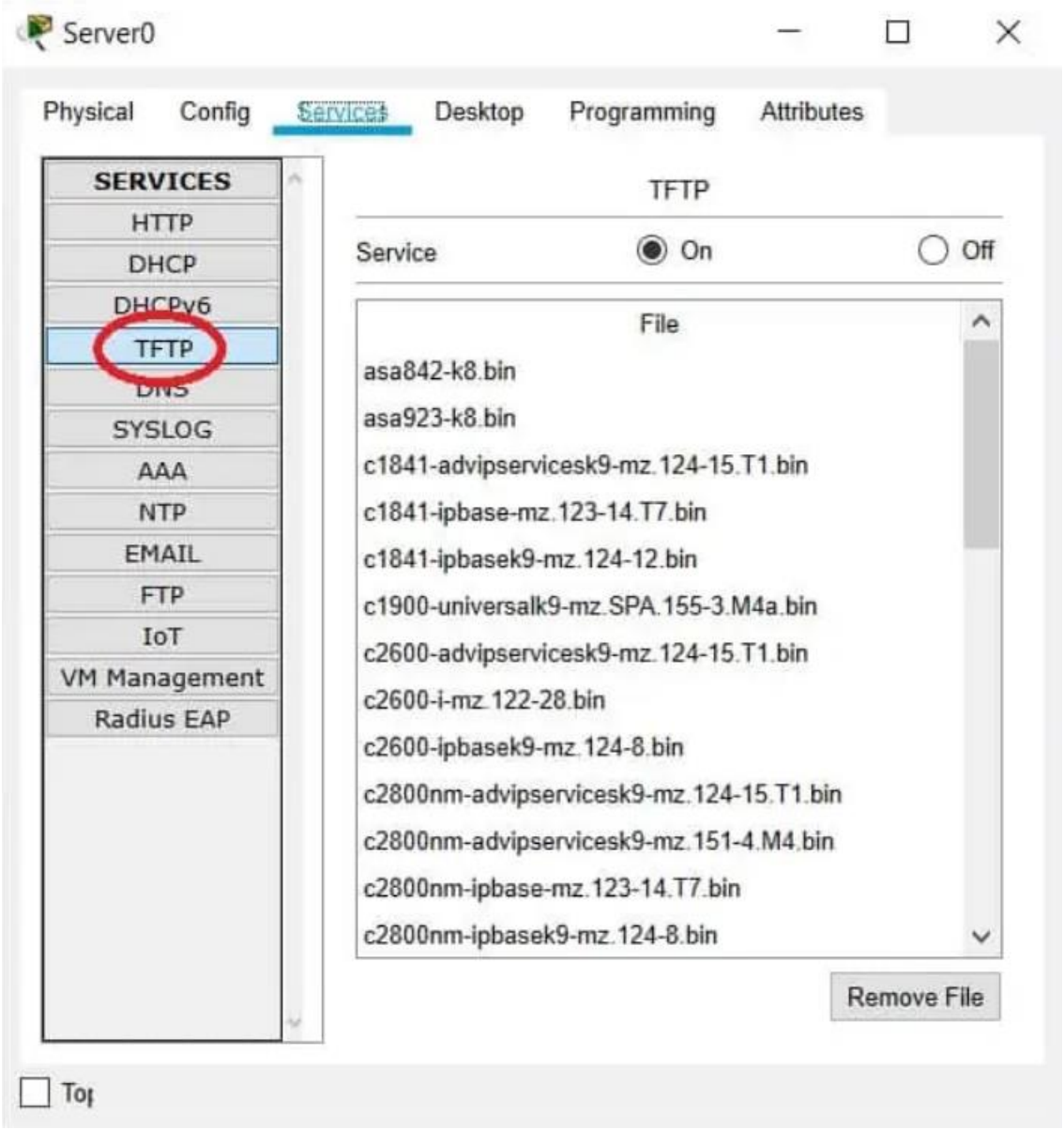
4. IOS finds the startup configuration file, usually located in NVRAM, and loads it into RAM as the running configuration.

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 command 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 Packet Tracer to do so; just add a Server to your topology, assign it an IP address and enable the TFTP service:


Server0
Physical
Config
Services
Desktop
Programming
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
FastEthernet0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
00E0.8F43.8C5D
IP Configuration
DHCP
Static
IPv4 Address
192.168.0.10
Subnet Mask
255.255.255.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address
FE80::2E0:8FFF:FE43:8C5D
Top



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

```
R1#copy running-config tftp:

Address or name of remote host []? 192.168.0.10

Destination filename [R1-confg]?


Writing running-config...!!
```

```
[OK - 561 bytes]


561 bytes copied in 0.001 secs (561000 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 running-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*:

```
R1#copy tftp: running-config
Address or name of remote host []? 192.168.0.10
Source filename []? R1-confg
Destination filename [running-config]?

Accessing tftp://192.168.0.10/R1-confg...
Loading R1-confg from 192.168.0.10: !
[OK - 561 bytes]

561 bytes copied in 0 secs
```

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

Link Layer Discovery Protocol (LLDP)

Link Layer Discovery Protocol (LLDP) is a vendor-neutral link layer protocol defined in IEEE standard 802.1AB. Just like Cisco's CDP, LLDP is used by network devices to advertise their identity, capabilities, and neighbors on a local Ethernet network. However, since LLDP is an open standard, it has one big advantage over CDP – it can be used on non-Cisco devices.

LLDP is disabled by default on Cisco devices. To enable it, use the *lldp run* command in the config mode:

```
R1(config)#lldp run
```

To display information about the LLDP neighbors, run the *show lldp neighbors* command:

```
R1#show lldp neighbors

Capability codes:

    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device

    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID           Local Intf     Hold-time  Capability      Port
ID

R2                  Gig0/0         120        R               Gig0/0



Total entries displayed: 1
```

As you can see from the output above, R1 has a single neighbor. Here is the description of the fields in the output:

- **Device ID** – the neighbor's host name
- **Local Intf** – the local device interface on which the neighbor is connected to
- **Hold-time** – the time the receiving device should hold the information sent by this device before discarding it.
- **Capability** – the type of the device. R means router, T means telephone, etc.
- **Port ID** – neighboring device's interface

To get more detail information about LLDP neighbors, run the ***show lldp neighbors detail*** command:

```
R1#show lldp neighbors detail

------------------------------------------------

Chassis id: 0002.4A13.6C01

Port id: Gig0/0

Port Description: GigabitEthernet0/0

System Name: R2

System Description:

Cisco IOS Software, C1900 Software (C1900-UNIVERSALK9-M), Version
15.1(4)M4, RELEASE SOFTWARE (fc2)

Technical Support: http://www.cisco.com/techsupport

Copyright (c) 1986-2012 by Cisco Systems, Inc.

Compiled Thurs 5-Jan-12 15:41 by pt_team

Time remaining: 90 seconds

System Capabilities: R

Enabled Capabilities: R

Management Addresses - not advertised

Auto Negotiation - supported, enabled
```

```
Physical media capabilities:
1000baseT(FD)
100baseT(FD)
Media Attachment Unit type: 10
Vlan ID: 1


Total entries displayed: 1
```

NOTE

To get information about a specific neighbor, run the *show lldp entry DEVICE_ID* command.

To display global LLDP information, run the *show lldp* command:

```
R2#show lldp
Global LLDP Information:
Status: ACTIVE
LLDP advertisements are sent every 30 seconds
LLDP hold time advertised is 120 seconds
LLDP interface reinitialisation delay is 2 seconds
```

This command displays information about whether LLDP is active on the device, the frequency of LLDP transmissions, the holdtime for packets being sent, and the delay time for LLDP to initialize on an interface.

You can also configure whether you would like your device to send or receive LLDP packets on a particular interface using the *no lldp transmit* and *no lldp receive interface mode* commands. For example, to only receive LLDP packets on the Gi0/0 interface, I would use the following command to disable the sending of LLDP packets:

```
R1(config-if)#no lldp transmit
```

Cisco Discovery Protocol (CDP) overview

CDP (Cisco Discovery Protocol) is a proprietary protocol developed by Cisco used to discovers information about the locally attached Cisco equipment. With CDP, the administrator can gather hardware and protocol information about neighboring devices, which can be helpful when troubleshooting or documenting the network.

To discover information, Cisco devices send CDP messages out each of their interfaces. These messages contain information about them, such as their hostname, network and data link addresses, the device model, IOS version, etc.

To display information about directly connected devices, we use the *show cdp neighbor* command:

```
Floor1#show cdp neighbors

Capability Codes: R - Router, T - Trans Bridge, B - Source Route
Bridge

S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID Local Intrfce Holdtme Capability Platform Port ID

Switch Gig 0/0 166 S 2960 Fas 0/1
```

As you can see from the example above, there is one directly connected device. Here is a description of each field:

- **Device ID** – the hostname of the directly connected device. In this case the hostname is Switch.
- **Local Interface** – the local interface on which the CDP messages were received (Gi0/0 in this case).

- **Holdtime** – the amount of time the local device will hold the information before discarding it if no more CDP packets are received.
- **Capability** – the capability of the directly connected device. The letter S indicates that the directly connected device is a switch. The letter R would indicate a router.
- **Platform** – the model and OS level running on the neighbor, 2960 series switch in this case.
- **Port ID** – the neighbor device's interface on which the CDP packets were sent, in this case Fa0/1.

To get even more information about the neighbors, use the show cdp neighbors detail command:

```
Floor1#show cdp neighbors detail


Device ID: Switch

Entry address(es):

Platform: cisco 2960, Capabilities: Switch

Interface:    GigabitEthernet0/0,    Port    ID    (outgoing    port):
FastEthernet0/1

Holdtime: 126



Version :

Cisco  IOS  Software,  C2960  Software  (C2960-LANBASE-M),  Version
12.2(25)FX, RELEASE SOFTWARE (fc1)

Copyright (c) 1986-2005 by Cisco Systems, Inc.

Compiled Wed 12-Oct-05 22:05 by pt_team



advertisement version: 2
```

```
Duplex: full
```

NOTE

IEEE has released a vendor-neutral link layer protocol called Link Layer Discovery Protocol (LLDP) as an alternative to CDP.

show processes command

If a Cisco device is suffering from high CPU usage, we can use the show processes command to list all running processes and determine the cause of problem. This command gives you a list of active processes, along with their corresponding process ID, priority, CPU time used, number of times invoked, and other information.

Here is an example output of this command invoked on a Cisco router:

```
R1#show processes

CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes:
0%

PID QTy PC Runtime (ms) Invoked uSecs Stacks TTY Process

1 Csp 602F3AF0 0 1627 0 2600/3000 0 Load Meter

2 Lwe 60C5BE00 4 136 29 5572/6000 0 CEF Scanner

3 Lst 602D90F8 1676 837 2002 5740/6000 0 Check heaps

4 Cwe 602D08F8 0 1 0 5568/6000 0 Chunk Manager

5 Cwe 602DF0E8 0 1 0 5592/6000 0 Pool Manager

6 Mst 60251E38 0 2 0 5560/6000 0 Timers

7 Mwe 600D4940 0 2 0 5568/6000 0 Serial Backgrou

8 Mwe 6034B718 0 1 0 2584/3000 0 OIR Handler

9 Mwe 603FA3C8 0 1 0 5612/6000 0 IPC Zone Manage
```

```
10 Mwe 603FA1A0 0 8124 0 5488/6000 0 IPC Periodic Ti

11 Mwe 603FA220 0 9 0 4884/6000 0 IPC Seat Manage

12 Lwe 60406818 124 2003 61 5300/6000 0 ARP Input

13 Mwe 60581638 0 1 0 5760/6000 0 HC Counter Time
```

The first line of the output shows the CPU utilization for the last 5 seconds, 1 minute, and 5 minutes. Here is a description of other fields in the output:

- **PID** – the Process ID.
- **Q** – the process queue priority. Possible values are: C (critical), H (high), M (medium), and L (low).
- **Ty** – scheduler test (status). Possible values are: * (currently running), E (waiting for an event), S (ready to run, voluntarily relinquished processor), rd (ready to run, wakeup conditions have occurred), we (waiting for an event), sa (sleeping until an absolute time), si (sleeping for a time interval), sp (sleeping for a time interval (alternate call), st(sleeping until a timer expires), hg (hung; the process will never execute again), xx (dead: the process has terminated, but has not yet been deleted).
- **PC** – current program counter.
- **Runtime** – CPU time the process has used.
- **Invoked** – number of times the process has been invoked.
- **microSecs** – CPU time for each process invocation.
- **Stacks** – low water mark or Total stack space available, shown in bytes.
- **TTY** – terminal that controls the process.
- **Process** – the name of the process.

Section 11

Map hostnames to IP addresses

It is possible to define static hostname-to-address mappings on a Cisco device for the purpose of name resolution. **This is usually done in environments without a DNS server**.

The mappings can be defined using the global configuration command *ip host HOSTNAME IP_ADDRESS*:

```
Floor1(config)#ip host HQ_SERVER 192.168.0.100
```

In the output above we've defined the IP address of **192.168.0.100** for the hostname **HQ_SERVER**. To display the hostname-to-address mappings, the *show hosts* command is used:

```
Floor1#show hosts
Default Domain is not set
Name/address lookup uses domain service
Name servers are 255.255.255.255

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
temp - temporary, perm - permanent
NA - Not Applicable None - Not defined

Host Port Flags Age Type Address(es)
HQ_SERVER None (perm, OK) 0 IP 192.168.0.100
```

We can ping the server using its hostname to verify that the hostnames are being resolved:

```
Floor1#ping HQ_SERVER
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.168.0.100, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms
```

You can see that HQ_SERVER responded to the ping request, which means that the name resolution was successful.

NOTE

The drawback of this method of name resolution is that we need to create static hostname-to-address mappings on each device in order to be able to resolve hostnames. If possible, use DNS instead.

Configure Cisco device as DNS client

DNS is an application layer protocol used to resolve hostnames to IP addresses. If you have a DNS server on your network, you can configure your Cisco device to use it for name resolution. Here are the steps:

- (Optional) If you've previously disabled DNS lookups on your device, re-enable it with the *ip domain-lookup* command.
- Specify the IP address of the DNS server using the *ip name-server* command. It is possible to specify up to six DNS servers.
- (Optional) Specify the domain name to append to the hostname you type in by using the *ip domain-name* command.

Here is an example configuration:

```
Floor1(config)#ip name-server 192.168.0.100
```

In the output above you can see that I've specified the IP address of my DNS server (192.168.0.100). Let's say that the DNS server contains a

record for a server called **fileshare.** I can try to ping that host using its hostname to verify that the name resolution process is indeed working:

```
Floor1#ping fileshare

Translating "fileshare"...domain server (192.168.0.100)

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.110, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/1 ms
```

As you can see from the output above, the hostname fileshare was translated to the IP address of 192.168.0.110.

no ip domain-lookup command

By default, any single word entered on an IOS device that is not recognized as a valid command is treated as a hostname to which you want to telnet. The device will try to translate that word to an IP address in a process that can last about a minute.

Consider the following example:

```
R1#writte

Translating "writte"...domain server (255.255.255.255)

% Unknown command or computer name, or unable to find computer address
```

In the output above you can see that I've mistyped the command write. The router entered the DNS resolution process which lasted about a minute. This can be annoying and this is why this feature is often turned off, especially in the lab environments.

If you don't need to have a DNS server configured for your router, you can use the *no ip domain-lookup* command to disable the DNS translation process:

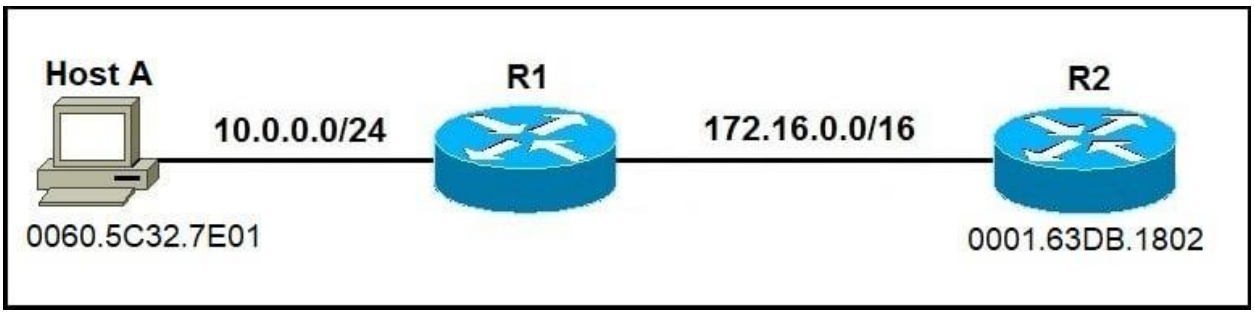
```
R1(config)#no ip domain-lookup
```

Now, if I mistype a command, the router will not perform a DNS resolution process:

```
R1#writte

Translating "writte"

% Unknown command or computer name, or unable to find computer address


R1#
```

The ARP table on a Cisco router

Just like regular hosts, if a Cisco router wants to exchange frames with a host in the same subnet, it needs to know its MAC address. The IP-to-MAC address mapping are kept in the router's ARP table. Consider the following example:



R1 has two connected subnets – 10.0.0.0/24 and 172.16.0.0./16. Before exchanging frames with either host, R1 will need to know their MAC addresses. Here is the output of the R1's ARP table:

```
R1#show ip arp

Protocol  Address          Age (min)  Hardware Addr   Type   Interface

Internet   10.0.0.1                   -    0060.5C32.7E01   ARPA
GigabitEthernet0/0

Internet   10.0.0.10                  6    000C.85CA.AD73   ARPA
GigabitEthernet0/0

Internet   172.16.0.1                 -    0060.5C32.7E02   ARPA
GigabitEthernet0/1

Internet   172.16.0.2                 10   0001.63DB.1802   ARPA
GigabitEthernet0/1
```

The ARP table contains two entries for R1's own two interfaces with the IP address of 10.0.0.1 and 172.16.0.1. The – in the age column indicates that the entry will never be timed out.

The ARP table also lists the MAC addresses of the two connected hosts. Consider the entry for Host A:

```
Protocol  Address          Age (min)  Hardware Addr   Type   Interface

Internet   10.0.0.10                  6    000C.85CA.AD73   ARPA
GigabitEthernet0/0
```

Here is a brief description of each field:

- Protocol – the protocol type, almost always Internet
- Address – the IP address associated with the MAC address, in our case the IP address of Host A
- Age – by default, an entry will be removed from the ARP table if it wasn't used in 240 minutes. 6 in this column means that the entry was last used 6 minutes ago. Each time an entry is used, the age will be reset back to zero.

- Hardware – the MAC address of the host with the corresponding IP address.
- Type – the type of hardware address. For Ethernet, this value will always be ARPA.
- Interface – the interface on R1 on which the corresponding host is connected.

Here are the steps R1 needs to take before forwarding frames to Host A:

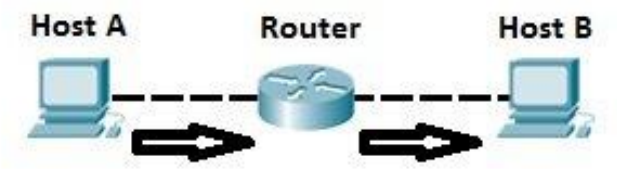
- R1 wants to communicate with Host A. R1 checks its routing table. The subnet on which Host A resides is a directly connected subnet.
- R1 checks its ARP table to find out whether the Host A's MAC address is known. If it is not, R1 will send an ARP request to the broadcast MAC address of FF:FF:FF:FF:FF:FF.
- Host A receives the frame and sends its MAC address to R1 (ARP reply). The host also updates its own ARP table with the MAC address of the Gigabit0/0 interface on R1.
- R1 receives the reply and updates the ARP table with the MAC address of Host A.
- Since both hosts now know each other MAC addresses, the communication can occur.

Section 12

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 the next hop address to which the packet should be forwarded.

Consider the following example of IP routing:



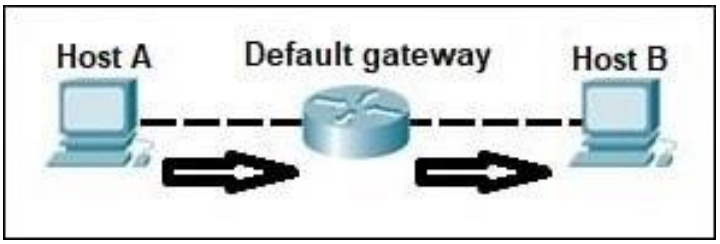
Host A wants to communicate with host B, but host B is on another network. Host A is configured to send all packets destined for remote networks to router R1. Router R1 receives the packets, examines the destination IP address and forwards the packet to the outgoing interface associated with the destination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

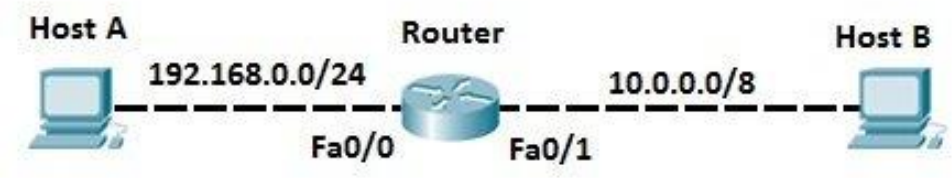
- **network destination and subnet mask** – specifies a range of IP addresses.
- **remote router** – IP address of the router used to reach that network.
- **outgoing interface** – outgoing interface the packet should go out to reach the destination network.

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


     10.0.0.0/24 is subnetted, 1 subnets

C       10.0.0.0 is directly connected, FastEthernet0/1

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

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

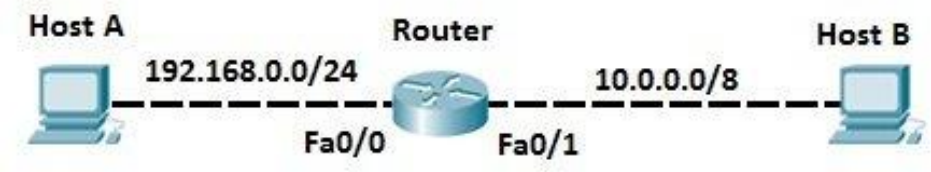
Let's explain the types of routes that can be found in a router's routing table.

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.

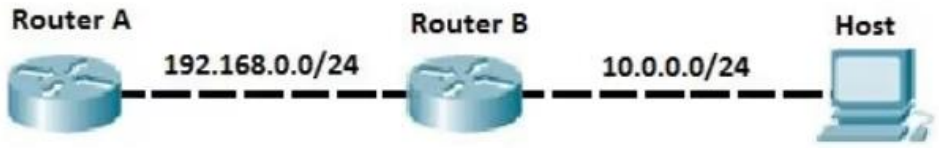
NOTE

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 10.0.0.0/24. 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.

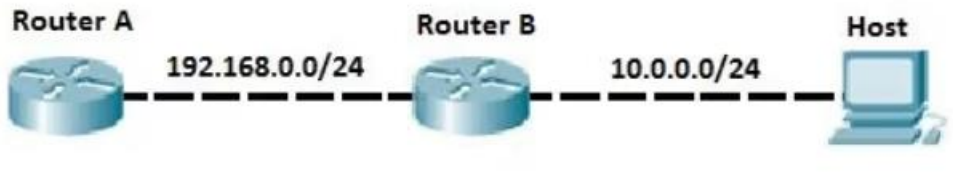
Dynamic routes

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 dynamicaly 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 procotols 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:

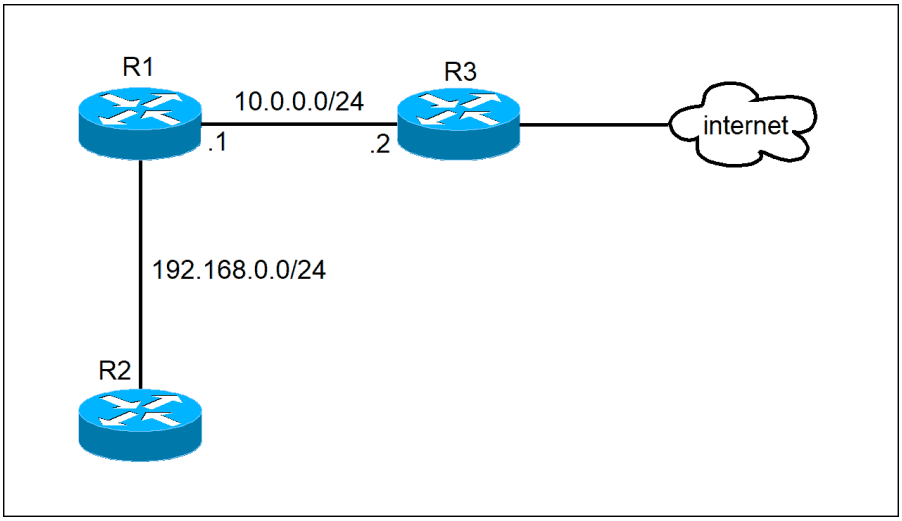
```
Router_A#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, 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.

Default static route

A default route defines where packets will be sent if no specific route for the destination network is listed in the routing table. If no default route is set, the router will discard all packets with destination addresses not found its routing table.

Consider the following example:

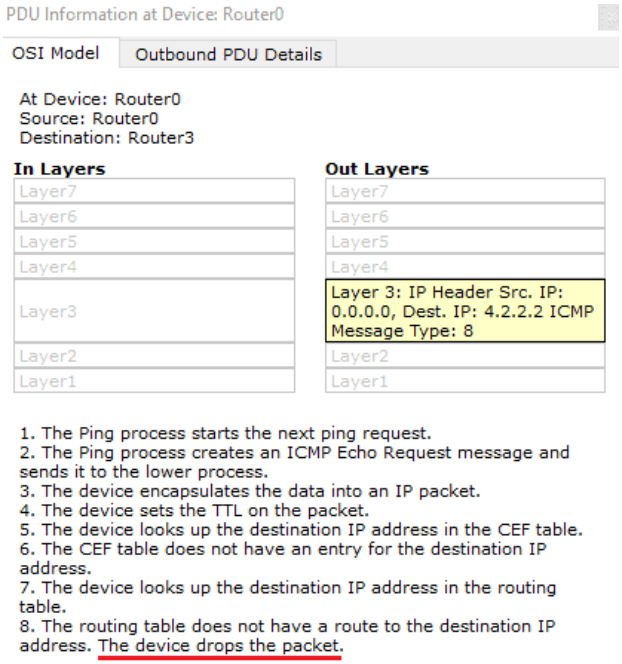


We have a network of three routers. R1 is directly connected to two subnets – 192.168.0.0/24 and 10.0.0.0/24. R3 is connected to the Internet.

Here is the routing table on R1:

```
R1#show ip route


Gateway of last resort is not set


     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, GigabitEthernet0/1
L       10.0.0.1/32 is directly connected, GigabitEthernet0/1
     192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.0.0/24 is directly connected, GigabitEthernet0/0
L       192.168.0.1/32 is directly connected, GigabitEthernet0/0
```

Notice the lack of the default gateway or default route. If R1 tries to access a public IP address (e.g. 4.2.2.2), the packets will be dropped because no route to that IP address has been found in the routing table:



To create a default static route on R1, we need to use the following command:

```
R1(config)#ip route 0.0.0.0 0.0.0.0 10.0.0.2
```

The command above instructs R1 to match all IP address and subnet masks and send the packets to 10.0.0.2 (the interface on R3 that is connected to R1). The routing table on R1 now looks like this:

```
R1#show ip route

Gateway of last resort is 10.0.0.2 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, GigabitEthernet0/1
L       10.0.0.1/32 is directly connected, GigabitEthernet0/1
     192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.0.0/24 is directly connected, GigabitEthernet0/0
```

```
L       192.168.0.1/32 is directly connected, GigabitEthernet0/0

S*   0.0.0.0/0 [1/0] via 10.0.0.2
```

Notice how the gateway of last restort is now set to 10.0.0.2. There is also a route marked with S* in the routing table, which means that the static default route we've just configured is a candidate default route (since routers can learn about multiple default routes), and * indicates that this static route is a candidate to become the default route.

Ping will now succeed:

```
R1#ping 4.2.2.2


Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 4.2.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms
```

NOTE

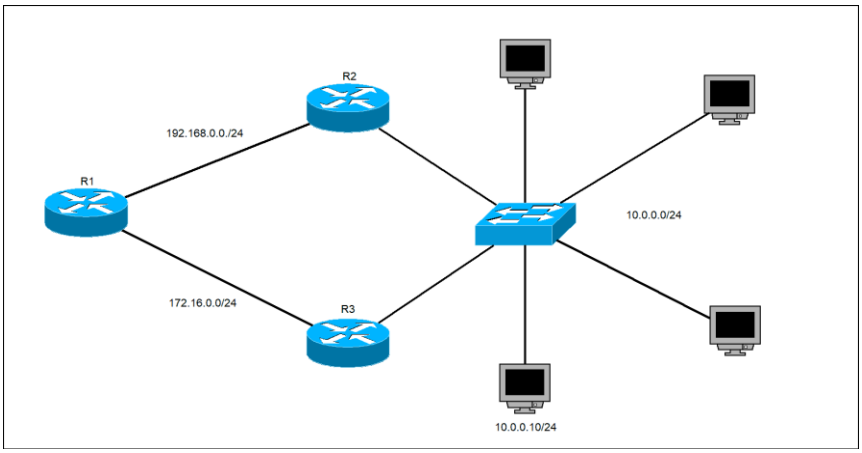
Connected routes always take precedence over static or dynamically discovered routes because they have the administrative distance value of 0 (the lowest possible value). In our case, this means that R1 will send out packets destined for 192.168.0.0/24 according to the specified connected route.

Create a static host route

In one of the previous lessons we've defined a static route for a **specific subnet** by using the subnet mask of **255.255.255.0** in the ip route command. IOS also allows you to specify a static host route for **a single**

host by specifying the **255.255.255.255 (/32)** subnet mask in the *ip route* command.

Static host routes are usually used when redundant paths exist. Consider the following example:



In the example above you can see that we have a network of three routers and a switch connected to the 10.0.0.0/24 subnet. R1 has two paths to reach that subnet – one going through R2, and the other one through R3. Let's say that we want to use the path going through R2 for all hosts, except the 10.0.0.10/24 host. For that host, we want to use the route going through R3. Here is how this can be done:

```
R1(config)#ip route 10.0.0.0 255.255.255.0 192.168.0.2
R1(config)#ip route 10.0.0.10 255.255.255.255 172.16.0.2
```

In the first command we've specified R1 to send all packets destined for the 10.0.0.0/24 network to 192.168.0.2 (the IP address of the interface on R2 connected to R1). However, for packets destined for the 10.0.0.10 host, we've instructed R1 to send all packets to 172.16.0.2 (the IP address of the interface on R3).

The two routes specified in the ip routes command above overlap (e.g. the IP address 10.0.0.10 is also included in the first command); however,

routers always use a more specific route with the longer prefix length. Since /32 is a more specific route than /24, R1 will use the route going through R3 to reach 10.0.0.10.

We can verify that packets are indeed going through desired routes by using the traceroute command on R1:

```
R1#traceroute 10.0.0.5
Type escape sequence to abort.
Tracing the route to 10.0.0.5

  1   192.168.0.2     0 msec    0 msec    0 msec
  2   10.0.0.5        0 msec    0 msec    0 msec
R1#
R1#traceroute 10.0.0.10
Type escape sequence to abort.
Tracing the route to 10.0.0.10

  1   172.16.0.2      0 msec    0 msec    0 msec
  2   10.0.0.10       0 msec    0 msec    0 msec
```

NOTE

The hosts also need to be configured with a correct default gateway – 10.0.0.10 needs to have the IP address of R3 configured as its default gateway, and other hosts on the 10.0.0.0/24 subnet need to have R2 configure as their default gateway.