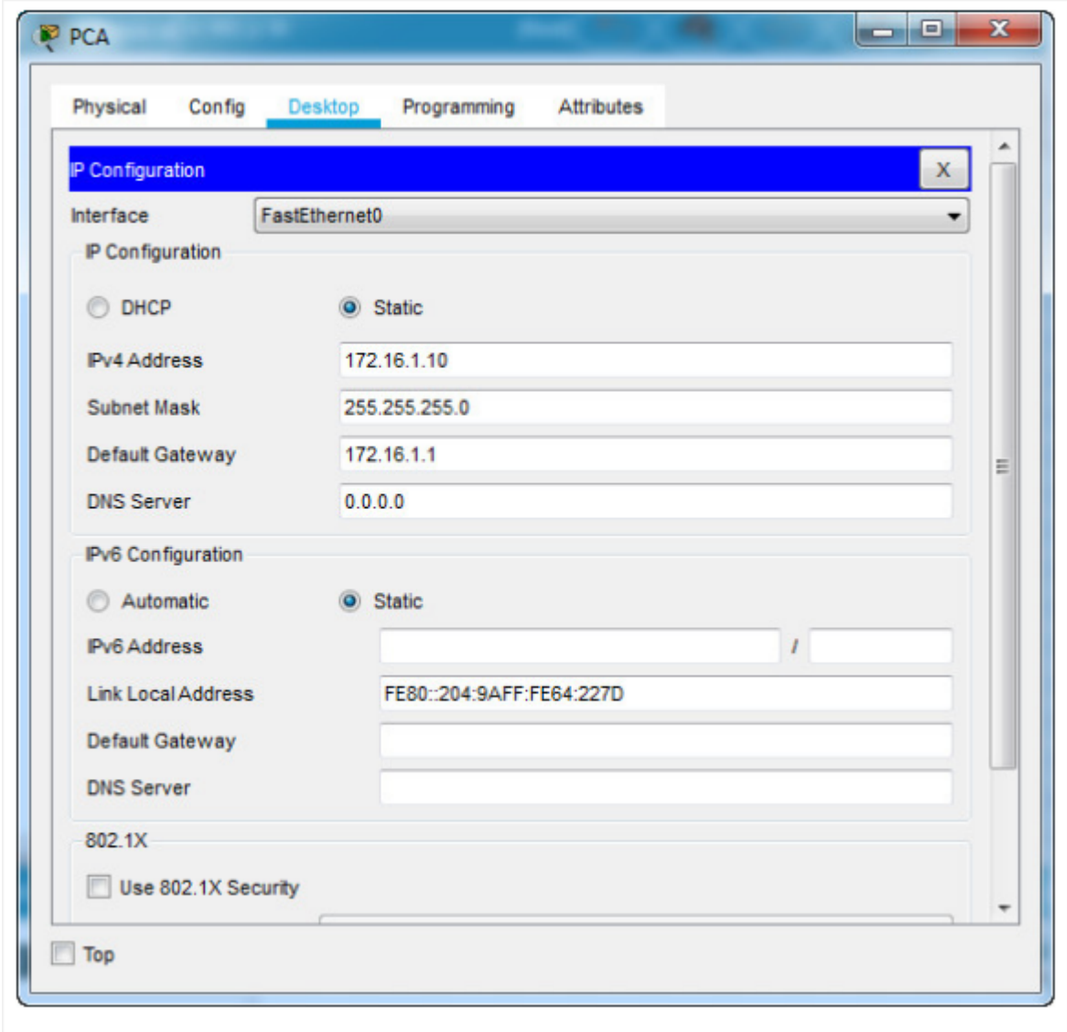
16.4.6 Packet Tracer – Configure Secure Passwords and SSH

Router RTA

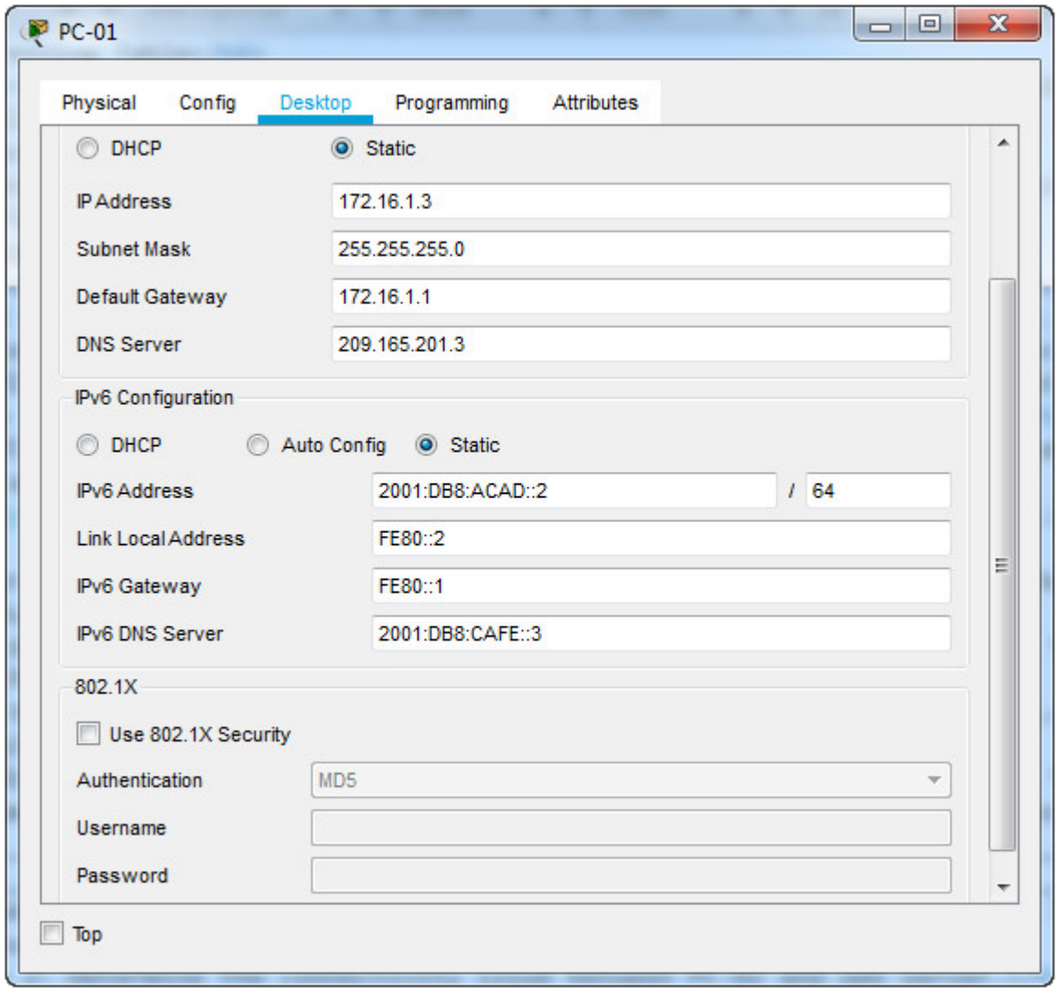
```
enable
config terminal
hostname RTA
interface g0/0
ip address 172.16.1.1 255.255.255.0
no shutdown
exit

service password-encryption
security password min-length 10

enable secret itexamanswers
no ip domain-lookup
ip domain-name CCNA.com
username jony secret itexamanswers.net
crypto key generate rsa
1024

login block-for 180 attempts 4 within 120

line vty 0 4
transport input ssh
login local
exec-timeout 6
end

copy running-config startup-config
```

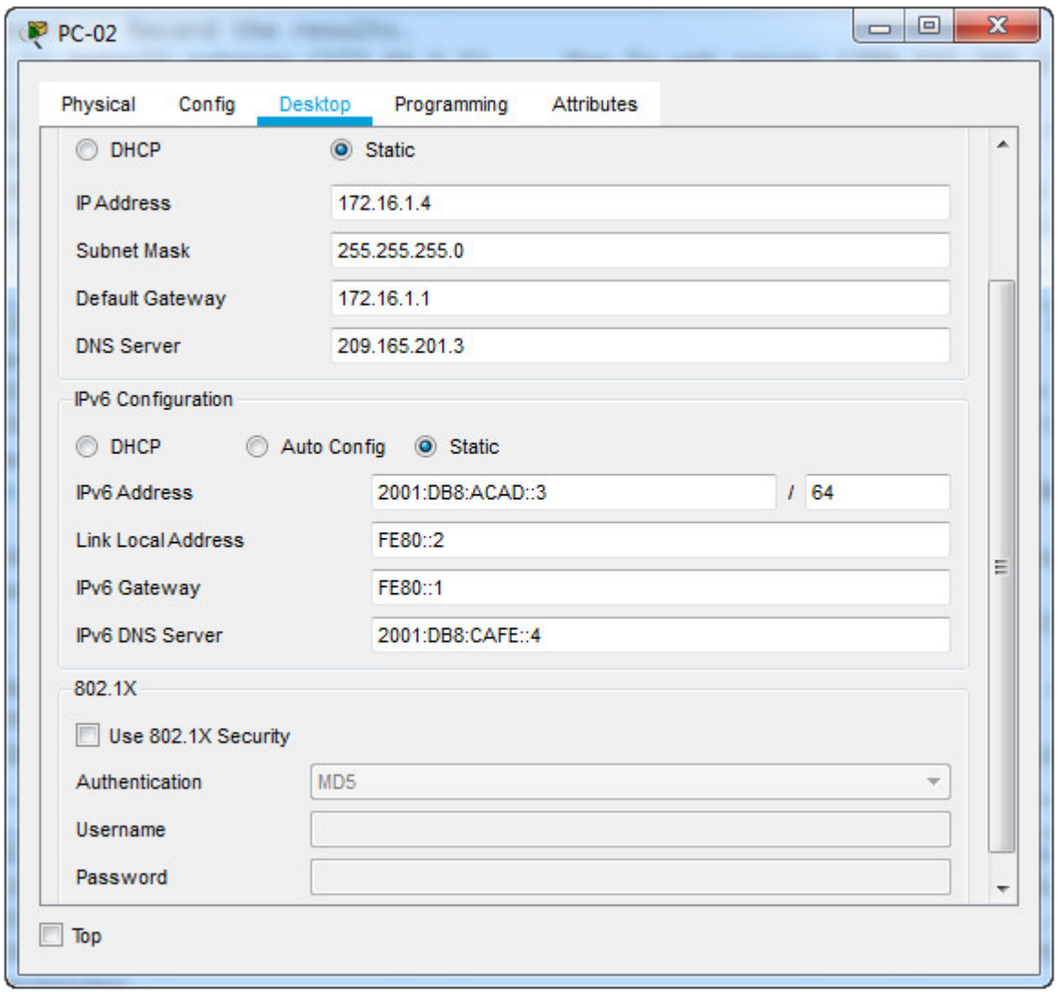
Switch SW1

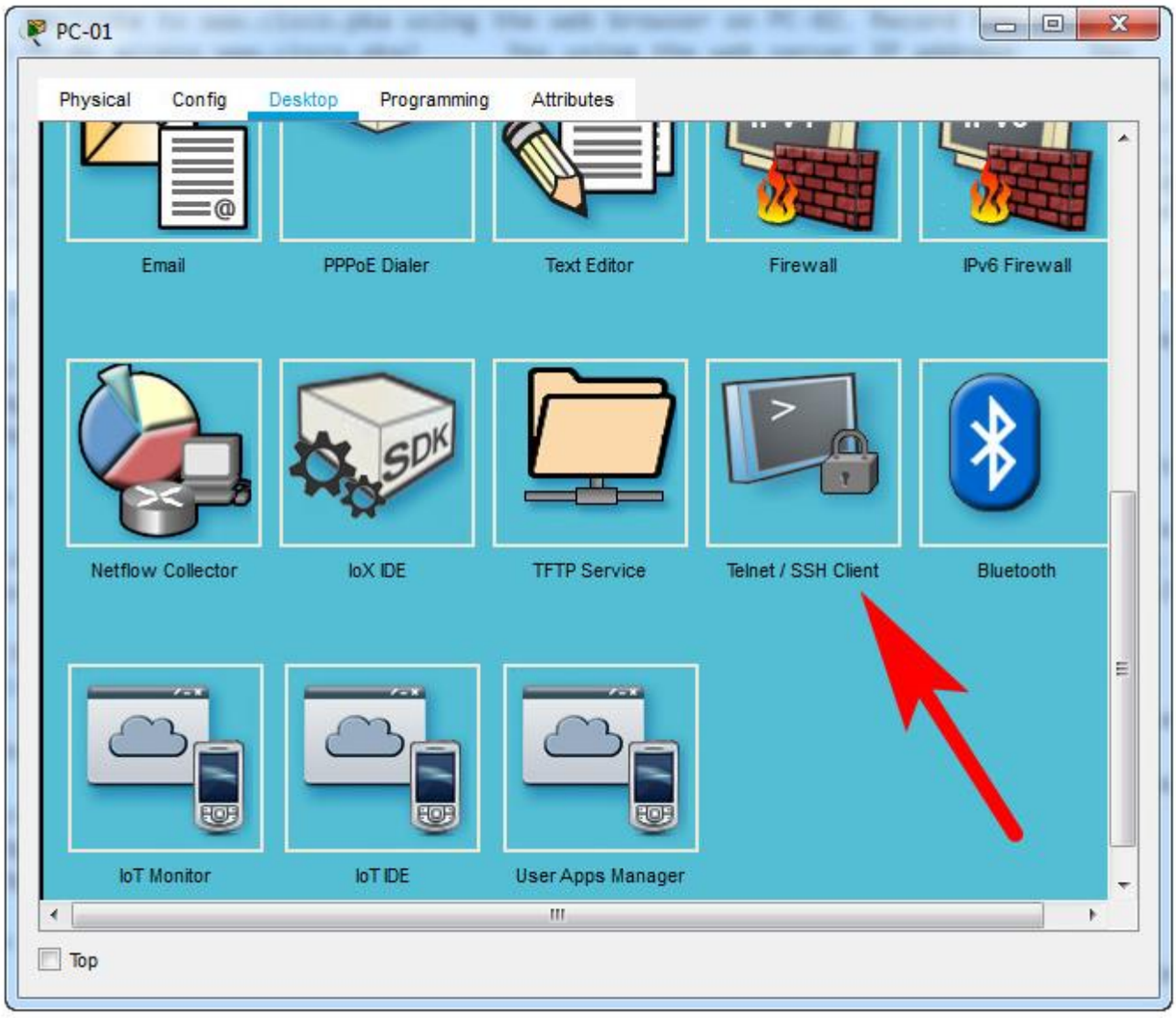
```
enable
config terminal
hostname SW1

interface vlan 1
ip address 172.16.1.2 255.255.255.0
no shutdown
exit
ip default-gateway 172.16.1.1
interface range F0/2-24, G0/2
shutdown
exit
service password-encryption
enable secret class
no ip domain-lookup
ip domain-name CCNA.com
crypto key generate rsa
1024
username admin_switch secret p@ssword
line vty 0 15
transport input ssh
login local
exec-timeout 6
end
copy running-config startup-config
```

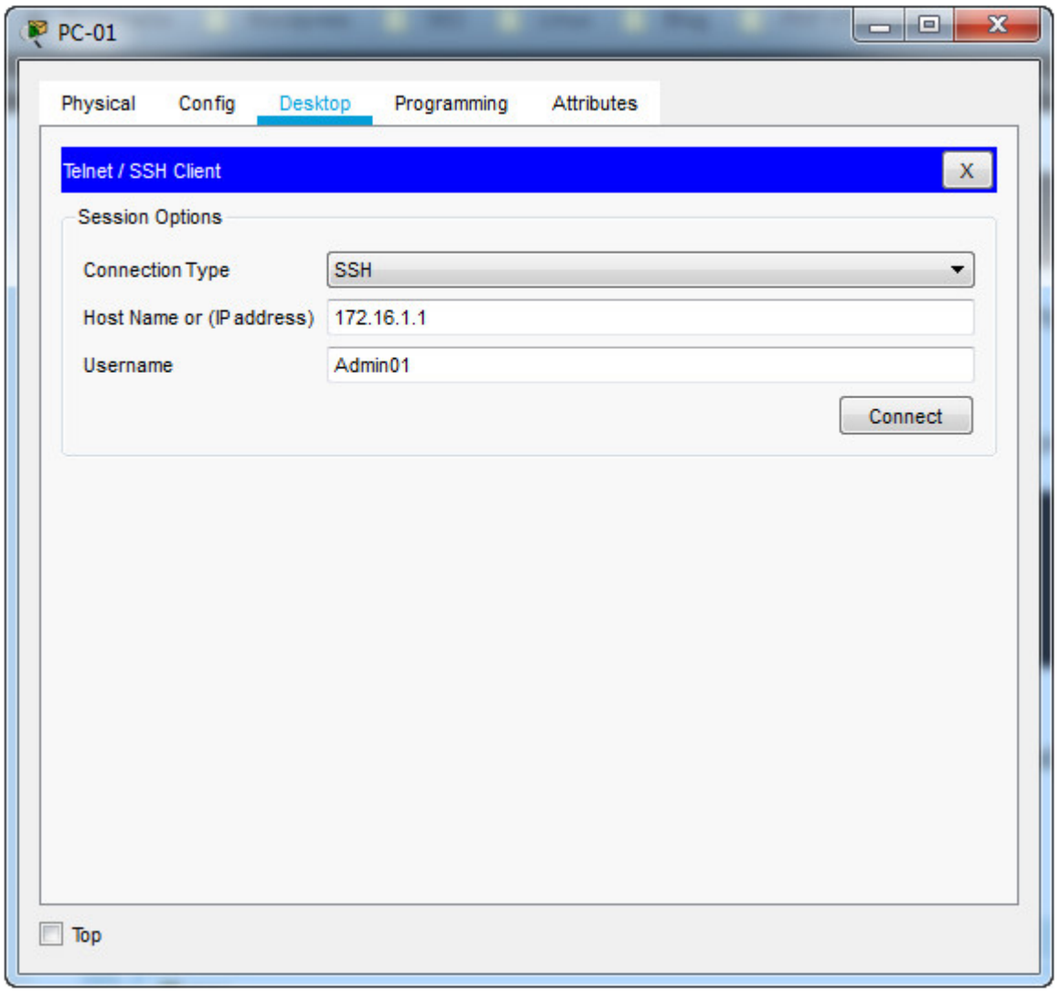


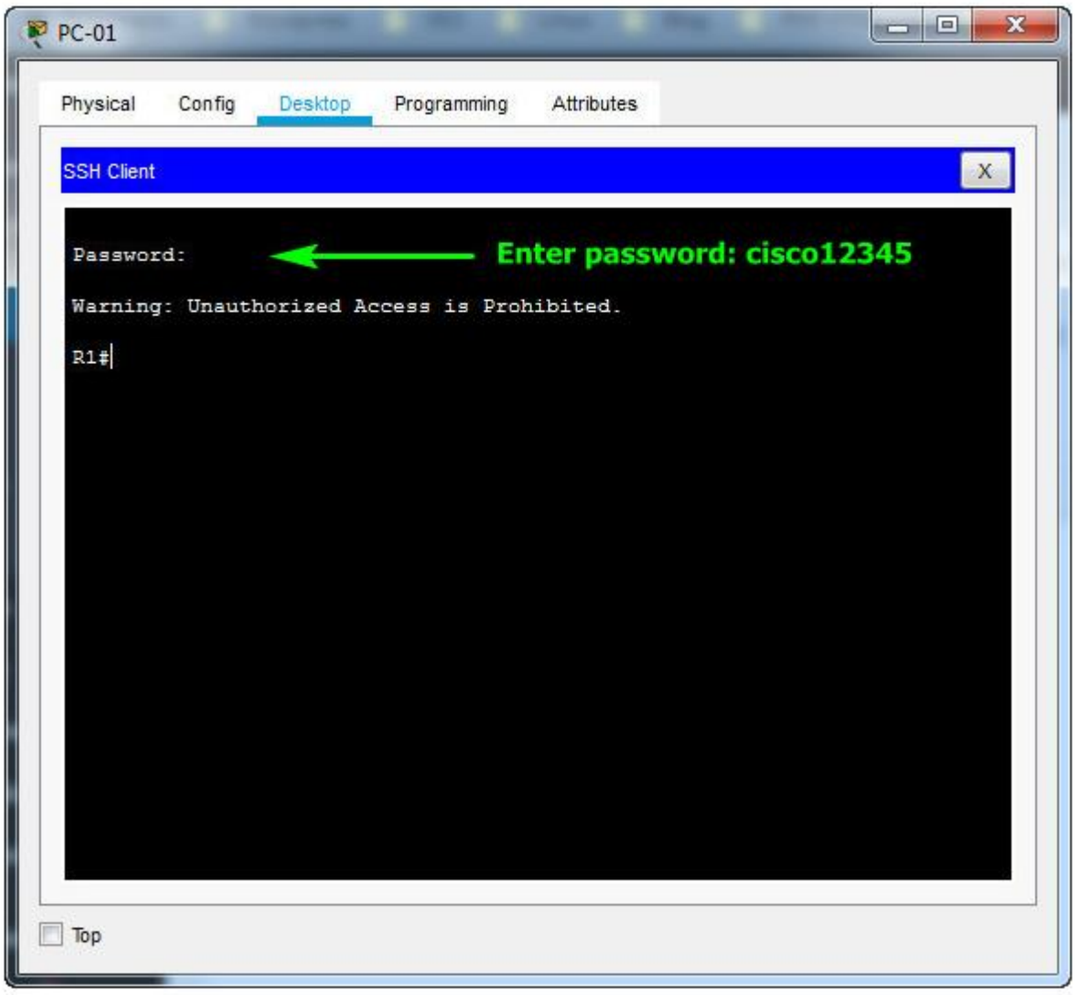
17.7.6 Packet Tracer – Troubleshoot Connectivity Issues






PC-01
Physical
Config
Desktop
Programming
Attributes
Email
PPPoE Dialer
Text Editor
Firewall
IPv6 Firewall
Netflow Collector
IoX IDE
TFTP Service
Telnet / SSH Client
Bluetooth
IoT Monitor
IoT IDE
User Apps Manager
Top


PC-01
Physical
Config
Desktop
Programming
Attributes
Telnet / SSH Client
X
Session Options
Connection Type
SSH
Host Name or (IP address)
172.16.1.1
Username
Admin01
Connect
Top



Enter the command below to change the IP address from 172.16.3.1 to 172.16.2.1

```
R1#enable
R1#configure terminal
R1(config)#interface g0/1
R1(config-if)#ip address 172.16.2.1 255.255.255.0
R1(config-if)#no shutdown
```

17.8.3 Packet Tracer – Troubleshooting Challenge

R1 Configuration

Login R1 with passwords:

– Console password: **Ciscoconpa55**

– Enable password: **Ciscoenpa55**

```
interface GigabitEthernet0/1
ip address 172.16.1.126 255.255.255.192
username Admin1 secret Admin1pa55
line vty 0 4
transport input ssh
```

S1 Configuration

No Change

S2 Configuration

Login S2 with passwords:

– Console password: **Ciscoconpa55**

– Enable password: **Ciscoenpa55**

```
interface Vlan1
ip address 172.16.1.125 255.255.255.192
```

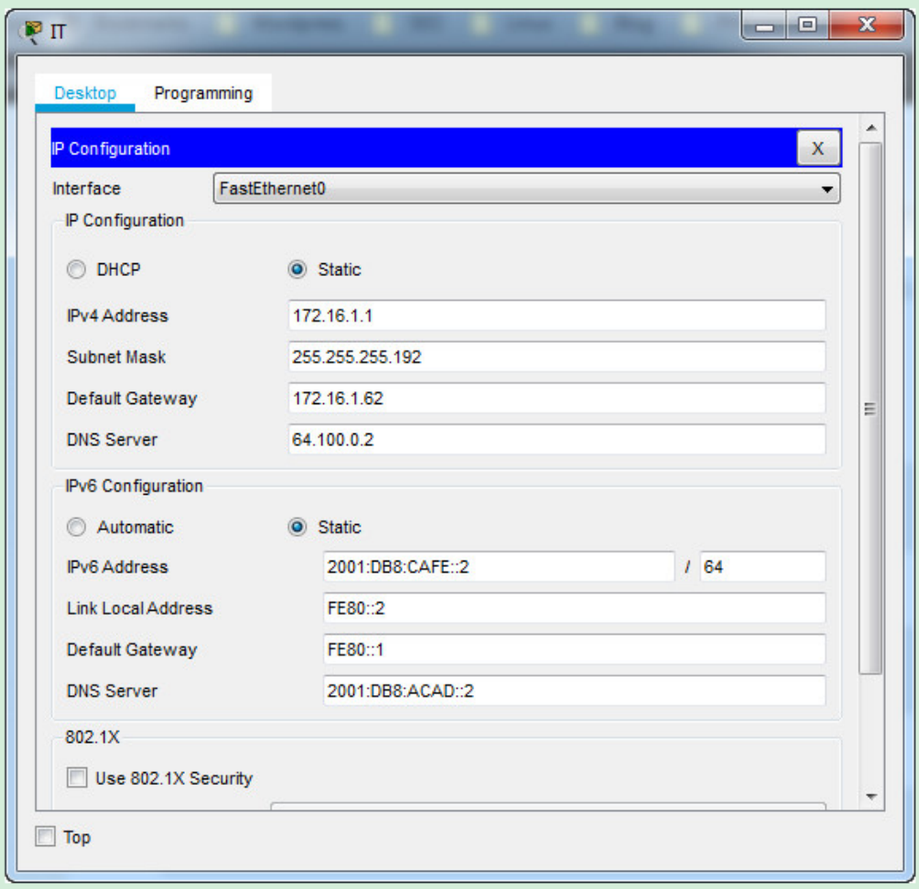
S3 Configuration

No Change

IT PC Configuration

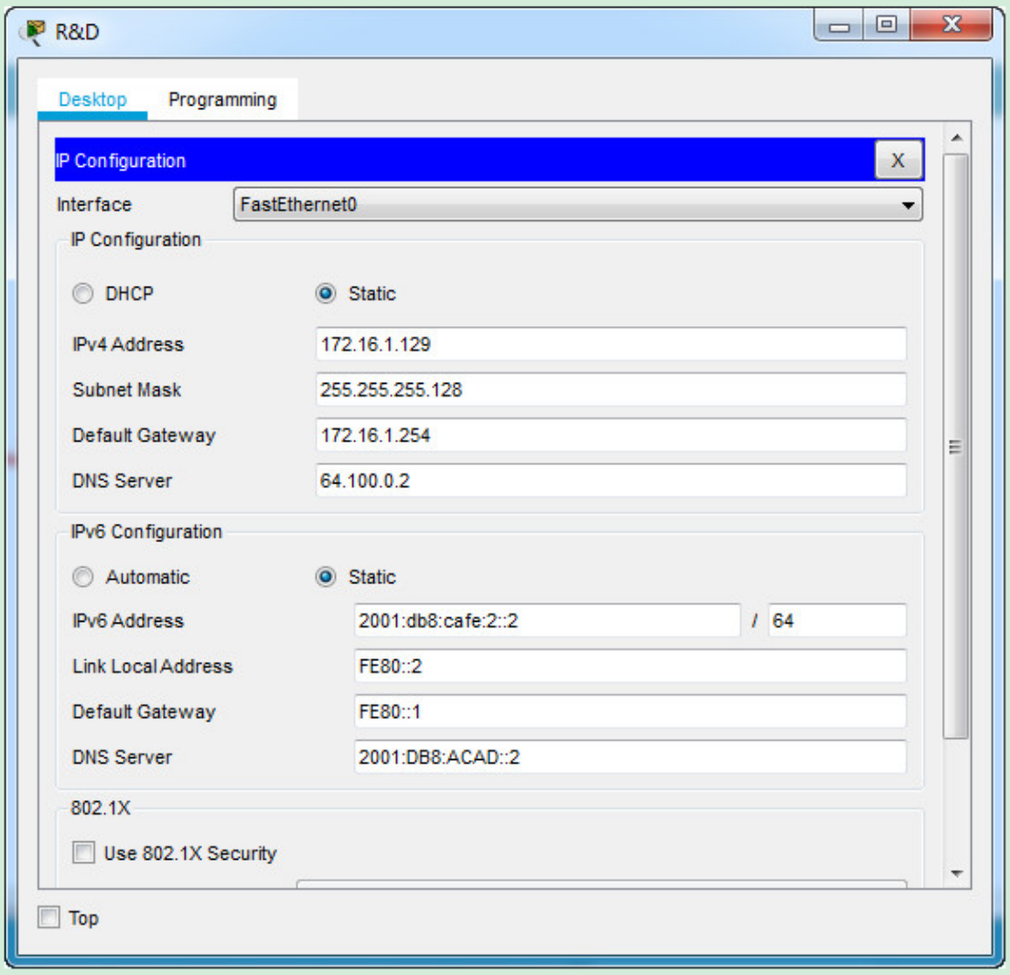
Incorrect IPv4 address ~~change to~~> **172.16.1.1**

Incorrect default gateway ~~change to~~> **172.16.1.62**



R&D PC Configuration

Incorrect IPv6 address ~~change to~~> **2001:db8:cafe:2::2/64**



11.1.1

Network and Host Portions

An IPv4 address is a 32-bit hierarchical address that is made up of a network portion and a host portion. When determining the network portion versus the host portion, you must look at the 32-bit stream, as shown in the figure.

The diagram shows the breakdown of an IPv4 address into the network and host portions. The IPv4 address is 192.168.10.10. Underneath, the address is converted into 11000000 10101000 00001010 00001010. A dashed line shows the separation between the network and host portions. This occurs after the third octet and the 24th bit.

IPv4 Address

11000000 10101000 0000101000001010 192 . 168 . 10 . 10

Network PortionHost PortionIPv4 Address

The bits within the network portion of the address must be identical for all devices that reside in the same network. The bits within the host portion of the address must be unique to identify a specific host within a network. If two hosts have the same bit-pattern in the specified network portion of the 32-bit stream, those two hosts will reside in the same network.

But how do hosts know which portion of the 32-bits identifies the network and which identifies the host? That is the role of the subnet mask.

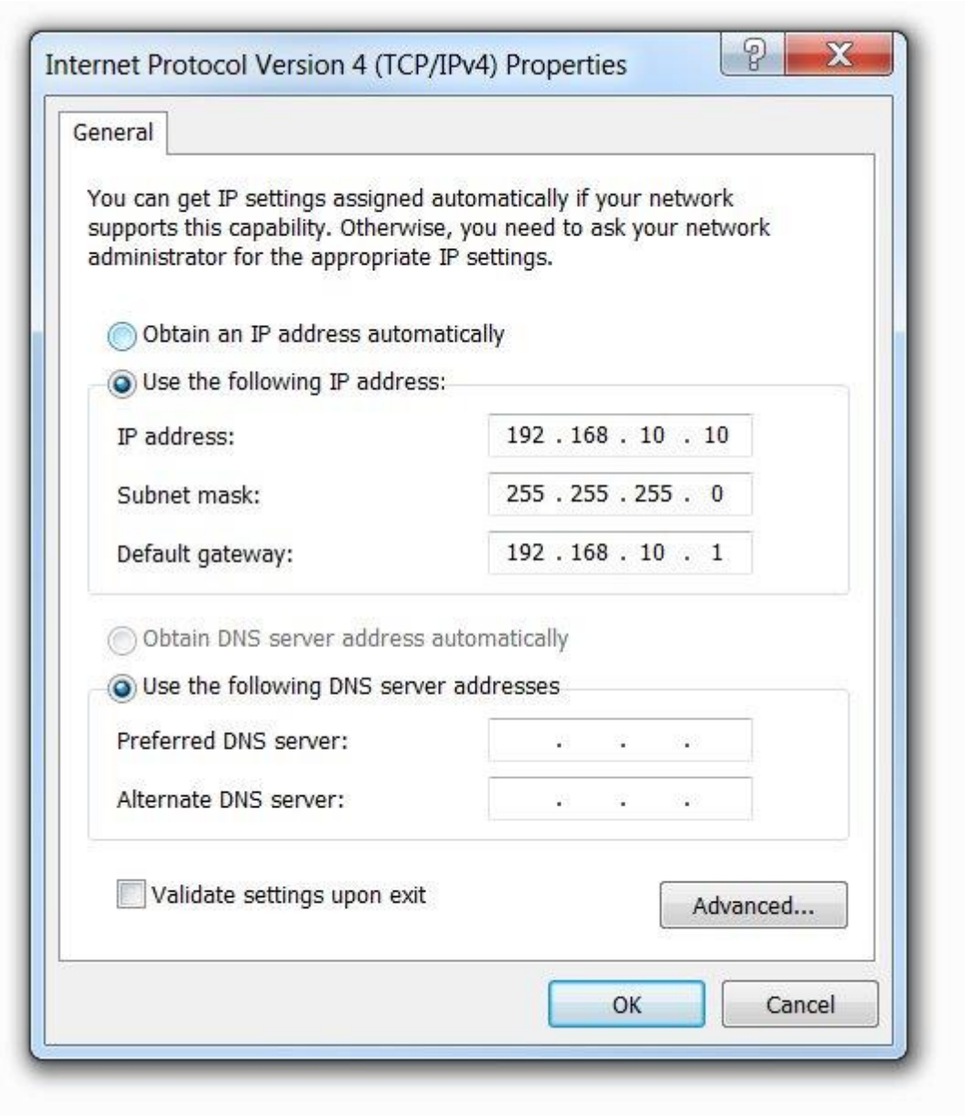
11.1.2

The Subnet Mask

As shown in the figure, assigning an IPv4 address to a host requires the following:

- **IPv4 address** - This is the unique IPv4 address of the host.
- **Subnet mask**- This is used to identify the network/host portion of the IPv4 address.

IPv4 Configuration on a Windows Computer



Note: A default gateway IPv4 address is required to reach remote networks and DNS server IPv4 addresses are required to translate domain names to IPv4 addresses.

The IPv4 subnet mask is used to differentiate the network portion from the host portion of an IPv4 address. When an IPv4 address is assigned to a device, the subnet mask is used to determine the network address of the device. The network address represents all the devices on the same network.

The next figure displays the 32-bit subnet mask in dotted decimal and binary formats.

subnet mask of 255.255.255.0 on top with the binary representation of 11111111 111111111 11111111 0000000 underneath; a dashed line is drawn after the third octet and the 24th bit

Subnet Mask

255 . 255 . 255 . 11111111 000000000 11111111 11111111

Subnet Mask

Notice how the subnet mask is a consecutive sequence of 1 bits followed by a consecutive sequence of 0 bits.

To identify the network and host portions of an IPv4 address, the subnet mask is compared to the IPv4 address bit for bit, from left to right as shown in the figure.

The figure shows an IPv4 address, written in both dotted-decimal and binary, with the subnet mask below, also written in dotted-decimal and binary, used to show the division between the network portion and host portion of the address. The IPv4 address is 192.168.10.10 which is converted to 11000000 10101000 00001010 00001010. The subnet mask is 255.255.255.0 which is converted to 11111111 11111111 11111111 00000000. A dashed line shows the separation between the network and host portions. This occurs after the third octet and 24th bit.

Associating an IPv4 Address with its Subnet Mask

255 . 255 . 255 . 0 11000000 10101000 0000101000001010 192 . 168 . 10 . 10 11111111 0000000011111111 11111111

Network PortionHost PortionIPv4 AddressSubnet Mask

Note that the subnet mask does not actually contain the network or host portion of an IPv4 address, it just tells the computer where to look for the part of the IPv4 address that is the network portion and which part is the host portion.

The actual process used to identify the network portion and host portion is called ANDing.

11.1.3

The Prefix Length

Expressing network addresses and host addresses with the dotted decimal subnet mask address can become cumbersome. Fortunately, there is an alternative method of identifying a subnet mask, a method called the prefix length.

The prefix length is the number of bits set to 1 in the subnet mask. It is written in “slash notation”, which is noted by a forward slash (/) followed by the number of bits set to 1. Therefore, count the number of bits in the subnet mask and prepend it with a slash.

Refer to the table for examples. The first column lists various subnet masks that can be used with a host address. The second column displays the converted 32-bit binary address. The last column displays the resulting prefix length.

Comparing the Subnet Mask and Prefix Length

Subnet Mask32-bit AddressPrefix Length255.0.0.011111111.00000000.00000000.00000000/8255.255.0.011111111.11111111.00000000.00000000/1625.255.255.011111111.11111111.11111111.00000000/24255.255.255.12811111111.11111111.11111111.10000000/25255.255.255.19211111111.11111111.11111111.11000000/26255.255.255.22411111111.11111111.11111111.11100000/27255.255.255.24011111111.11111111.11111111.11110000/28255.255.255.24811111111.11111111.11111111.11111000/29255.255.255.25211111111.11111111.11111111.11111100/30

Subnet Mask	32-bit Address	Prefix Length
255.0.0.0	11111111.00000000.00000000.00000000	/8
255.255.0.0	11111111.11111111.00000000.00000000	/16
255.255.255.0	11111111.11111111.11111111.00000000	/24
255.255.255.128	11111111.11111111.11111111.10000000	/25
255.255.255.192	11111111.11111111.11111111.11000000	/26
255.255.255.224	11111111.11111111.11111111.11100000	/27
255.255.255.240	11111111.11111111.11111111.11110000	/28
255.255.255.248	11111111.11111111.11111111.11111000	/29
255.255.255.252	11111111.11111111.11111111.11111100	/30

Note: A network address is also referred to as a prefix or network prefix. Therefore, the prefix length is the number of 1 bits in the subnet mask.

When representing an IPv4 address using a prefix length, the IPv4 address is written followed by the prefix length with no spaces. For example, 192.168.10.10 255.255.255.0 would be written as 192.168.10.10/24. Using various types of prefix lengths will be discussed later. For now, the focus will be on the /24 (i.e. 255.255.255.0) prefix

11.1.4

Determining the Network: Logical AND

A logical AND is one of three Boolean operations used in Boolean or digital logic. The other two are OR and NOT. The AND operation is used in determining the network address.

Logical AND is the comparison of two bits that produce the results shown below. Note how only a 1 AND 1 produces a 1. Any other combination results in a 0.

- 1 AND 1 = 1
- 0 AND 1 = 0
- 1 AND 0 = 0
- 0 AND 0 = 0

Note: In digital logic, 1 represents True and 0 represents False. When using an AND operation, both input values must be True (1) for the result to be True (1).

To identify the network address of an IPv4 host, the IPv4 address is logically ANDed, bit by bit, with the subnet mask. ANDing between the address and the subnet mask yields the network address.

To illustrate how AND is used to discover a network address, consider a host with IPv4 address 192.168.10.10 and subnet mask of 255.255.255.0, as shown in the figure:

- **IPv4 host address (192.168.10.10)** - The IPv4 address of the host in dotted decimal and binary formats.
- **Subnet mask (255.255.255.0)** - The subnet mask of the host in dotted decimal and binary formats.
- **Network address (192.168.10.0)** - The logical AND operation between the IPv4 address and subnet mask results in an IPv4 network address shown in dotted decimal and binary formats.

The diagram shows the ANDing process between an IPv4 host address and a subnet mask resulting in the IPv4 network address of the host. The IPv4 host address is 192.168.10.10. Below that, the address is converted into 11000000 10101000 00001010 00001010. Below that, the subnet mask of 255.255.255.0 is written. Below that, the subnet mask is converted to 11111111 11111111 11111111 00000000. A line is drawn underneath the binary equivalent of the subnet mask. Below the line is the dotted-decimal and binary equivalent of the IPv4 network address as determined by the ANDing process. A blue shaded box shows the first bit of the IPv4 host address, a 1, compared to the first bit of the subnet mask, also a 1, resulting in a 1 as the first bit value in the IPv4 network address. The IPv4 network address is 192.168.10.0 with a binary equivalent of 11000000 101001000 00001010 00000000.

0000 10101100 00001010 10000000 1010192168...10100000 00001111 11111111 11111111 11110000 00001100 00001010 10000000 1010255255...2550192168...100

IPv4 host
addressSubnet MaskIPv4
network
address**ANDEquals**

Using the first sequence of bits as an example, notice the AND operation is performed on the 1-bit of the host address with the 1-bit of the subnet mask. This results in a 1 bit for the network address. 1 AND 1 = 1.

The AND operation between an IPv4 host address and subnet mask results in the IPv4 network address for this host. In this example, the AND operation between the host address of 192.168.10.10 and the subnet mask 255.255.255.0 (/24), results in the IPv4 network address of 192.168.10.0/24. This is an important IPv4 operation, as it tells the host what network it belongs to.

11.1.5

Video - Network, Host and Broadcast Addresses

Click Play to view a demonstration of how the network, host, and broadcast addresses are determined for a given IPv4 address and subnet mask.

Play Video

Network, Host, and Broadcast Addresses

Within each network are three types of IP addresses:

- Network address
- Host addresses
- Broadcast address

Using the topology in the figure, these three types of addresses will be examined.

The diagram is network topology with four hosts connected to a switch which is connected to a router. The router interface has an IP address of 192.168.10.1/24 and the hosts have the following IP addreses: 192.168.10.10/24, 192.168.10.55/24, 192.168.10.101/24, and 192.168.10.12/24. The fourth octet of the router interface and the hosts is shown in a different color. A circle encompasses the router interface, switch, and all the hosts within which the network address of 192.168.10.0/24 is written, also with the fourth octet shown in a different color.

192.168.10.1/24192.168.10.10/24192.168.10.55/24192.168.10.101/24192.168.10.12/24192.168.10.0/24

Network Address

Network address

A network address is an address that represents a specific network. A device belongs to this network if it meets three criteria:

- It has the same subnet mask as the network address.
- It has the same network bits as the network address, as indicated by the subnet mask.
- It is located on the same broadcast domain as other hosts with the same network address.

A host determines its network address by performing an AND operation between its IPv4 address and its subnet mask.

As shown in the table, the network address has all 0 bits in the host portion, as determined by the subnet mask. In this example, the network address is 192.168.10.0/24. A network address cannot be assigned to a device.

Network, Host, and Broadcast Addresses

Table caption

	Network Portion	Host Portion	Host Bits
Subnet mask **255.255.255.**0 or **/24**	255 255 255 11111111 11111111 11111111	0 00000000	
Network address **192.168.10.**0 or **/24**	192 168 10 11000000 10101000 00001010	0 00000000	All 0s
First address **192.168.10.**1 or **/24**	192 168 10 11000000 10101000 00001010	1 00000001	All 0s and a 1
Last address **192.168.10.**254 or **/24**	192 168 10 11000000 10101000 00001010	254 11111110	All 1s and a 0
Broadcast address **192.168.10.**255 or **/24**	192 168 10 11000000 10101000 00001010	255 11111111	All 1s

Host addresses

Host addresses are addresses that can be assigned to a device such as a host computer, laptop, smart phone, web camera, printer, router, etc. The host portion of the address is the bits indicated by 0 bits in the subnet mask. Host addresses can have any combination of bits in the host portion except for all 0 bits (this would be a network address) or all 1 bits (this would be a broadcast address).

All devices within the same network, must have the same subnet mask and the same network bits. Only the host bits will differ and must be unique.

Notice that in the table, there is a first and last host address:

- **First host address** - This first host within a network has all 0 bits with the last (right-most) bit as a 1 bit. In this example it is 192.168.10.1/24.
- **Last host address** - This last host within a network has all 1 bits with the last (right-most) bit as a 0 bit. In this example it is 192.168.10.254/24.

Any addresses between and including, 192.168.10.1/24 through 192.168.10.254/24 can be assigned to a device on the network.

Broadcast address

A broadcast address is an address that is used when it is required to reach all devices on the IPv4 network. As shown in the table, the network broadcast address has all 1 bits in the host portion, as determined by the subnet mask. In this example, the network address is 192.168.10.255/24. A broadcast address cannot be assigned to a device.

11.2.1

Unicast

In the previous topic you learned about the structure of an IPv4 address; each has a network portion and a host portion. There are different ways to send a packet from a source device, and these different transmissions affect the destination IPv4 addresses.

Unicast transmission refers to one device sending a message to one other device in one-to-one communications.

A unicast packet has a destination IP address that is a unicast address which goes to a single recipient. A source IP address can only be a unicast address, because the packet can only originate from a single source. This is regardless of whether the destination IP address is a unicast, broadcast or multicast.

Play the animation to see an example of unicast transmission.

This animation consists of three hosts and a printer connected to a switch and router. The animation illustrates the host with IP address 172.16.4.1 sending a unicast packet to IP address 172.16.4.253. When the switch receives the frame, it forwards it out to the printer with IP address 172.16.4.253.

Destination: 172.16.4.253/24

Source: 172.16.4.1/24

Note: In this course, all communication between devices is unicast unless otherwise noted.

IPv4 unicast host addresses are in the address range of 1.0.0.1 to 223.255.255.255. However, within this range are many addresses that are reserved for special purposes. These special purpose addresses will be discussed later in this module.

11.2.2

Broadcast

Broadcast transmission refers to a device sending a message to all the devices on a network in one-to-all communications.

A broadcast packet has a destination IP address with all ones (1s) in the host portion, or 32 one (1) bits.

Note: IPv4 uses broadcast packets. However, there are no broadcast packets with IPv6.

A broadcast packet must be processed by all devices in the same broadcast domain. A broadcast domain identifies all hosts on the same network segment. A broadcast may be directed or limited. A directed broadcast is sent to all hosts on a specific network. For example, a host on the 172.16.4.0/24 network sends a packet to 172.16.4.255. A limited broadcast is sent to 255.255.255.255. By default, routers do not forward broadcasts.

Play the animation to see an example of a limited broadcast transmission.

This animation consists of three hosts and a printer connected to a switch and router. The animation illustrates the host with IP address 172.16.4.1 sending a broadcast packet. When the switch receives the broadcast packet, it forwards it out all ports to the other hosts, printer, and router.

Source: 172.16.4.1/24

Destination: 255.255.255.255

Limited Broadcast

Broadcast packets use resources on the network and make every receiving host on the network process the packet. Therefore, broadcast traffic should be limited so that it does not adversely affect the performance of the network or devices. Because routers separate broadcast domains, subdividing networks can improve network performance by eliminating excessive broadcast traffic.

IP Directed Broadcasts

In addition to the 255.255.255.255 broadcast address, there is a broadcast IPv4 address for each network. Called a directed broadcast, this address uses the highest address in the network, which is the address where all the host bits are 1s. For example, the directed broadcast address for 192.168.1.0/24 is 192.168.1.255. This address allows communication to all the hosts in that network. To send data to all the hosts in a network, a host can send a single packet that is addressed to the broadcast address of the network.

A device that is not directly connected to the destination network forwards an IP directed broadcast in the same way it would forward unicast IP packets destined to a host on that network. When a directed broadcast packet reaches a router that is directly connected to the destination network, that packet is broadcast on the destination network.

Note: Because of security concerns and prior abuse from malicious users, directed broadcasts are turned off by default starting with Cisco IOS Release 12.0 with the global configuration command **no ip directed-broadcasts**.

11.2.3

Multicast

Multicast transmission reduces traffic by allowing a host to send a single packet to a selected set of hosts that subscribe to a multicast group.

A multicast packet is a packet with a destination IP address that is a multicast address. IPv4 has reserved the 224.0.0.0 to 239.255.255.255 addresses as a multicast range.

Hosts that receive particular multicast packets are called multicast clients. The multicast clients use services requested by a client program to subscribe to the multicast group.

Each multicast group is represented by a single IPv4 multicast destination address. When an IPv4 host subscribes to a multicast group, the host processes packets addressed to this multicast address, and packets addressed to its uniquely allocated unicast address.

Routing protocols such as OSPF use multicast transmissions. For example, routers enabled with OSPF communicate with each other using the reserved OSPF multicast address 224.0.0.5. Only devices enabled with OSPF will process these packets with 224.0.0.5 as the destination IPv4 address. All other devices will ignore these packets.

The animation demonstrates clients accepting multicast packets.

This animation consists of three hosts and a printer connected to a switch and router. The animation illustrates the host with IP address 172.16.4.1 sending a multicast packet to the multicast group IP address 224.10.10.5. When the switch receives the multicast packet, it forwards it out all ports to the other hosts, printer, and router. However, only two hosts which are members of the multicast group address, will process the packets. All other hosts drop the packet.

Source: 172.16.4.1/24

Destination: 224.10.10.5

11.2.4

Activity - Unicast, Broadcast, or Multicast

Instructions:

Click Start to see a destination IP address. Next, click the host or hosts which will receive a packet based on the address type (unicast, broadcast, or multicast). Click **Check** to verify your answer. Click **New Problem** again to get a new problem.

11.3.1

Public and Private IPv4 Addresses

Just as there are different ways to transmit an IPv4 packet, there are also different types of IPv4 addresses. Some IPv4 addresses cannot be used to go out to the internet, and others are specifically allocated for routing to the internet. Some are used to verify a connection and others are self-assigned. As a network administrator, you will eventually become very familiar with the types of IPv4 addresses, but for now, you should at least know what they are and when to use them.

Public IPv4 addresses are addresses which are globally routed between internet service provider (ISP) routers. However, not all available IPv4 addresses can be used on the internet. There are blocks of addresses called private addresses that are used by most organizations to assign IPv4 addresses to internal hosts.

In the mid-1990s, with the introduction of the World Wide Web (WWW), private IPv4 addresses were introduced because of the depletion of IPv4 address space. Private IPv4 addresses are not unique and can be used internally within any network.

Note: The long-term solution to IPv4 address depletion was IPv6.

The Private Address Blocks

Network Address and PrefixRFC 1918 Private Address Range10.0.0.0/810.0.0.0 - 10.255.255.255172.16.0.0/12172.16.0.0 - 172.31.255.255192.168.0.0/16192.168.0.0 - 192.168.255.255	
Network Address and Prefix	**RFC 1918 Private Address Range**
10.0.0.0/8	10.0.0.0 - 10.255.255.255
172.16.0.0/12	172.16.0.0 - 172.31.255.255
192.168.0.0/16	192.168.0.0 - 192.168.255.255

Note: Private addresses are defined in RFC 1918 and sometimes referred to as RFC 1918 address space.

11.3.2

Routing to the Internet

Most internal networks, from large enterprises to home networks, use private IPv4 addresses for addressing all internal devices (intranet) including hosts and routers. However, private addresses are not globally routable.

In the figure, customer networks 1, 2, and 3 are sending packets outside their internal networks. These packets have a source IPv4 address that is a private address and a destination IPv4 address that is public (globally routable). Packets with a private address must be filtered (discarded) or translated to a public address before forwarding the packet to an ISP.

The diagram is a network topology with three networks, each connected to a different ISP router. The ISP routers are performing NAT between each network and the Internet.

Private IPv4 Addresses and Network Address Translation (NAT)

10.0.0.0/8172.16.0.0/16192.168.0.0/24ISP1ISP2ISP3

This packet has a source IPv4 address that is a private address. I will translate it to a public IPv4 address using NATNetwork 1Network 2Network 3Internet

Before the ISP can forward this packet, it must translate the source IPv4 address, which is a private address, to a public IPv4 address using Network Address Translation (NAT). NAT is used to translate between private IPv4 and public IPv4 addresses. This is usually done on the router that connects the internal network to the ISP network. Private IPv4 addresses in the organization's intranet will be translated to public IPv4 addresses before routing to the internet.

Note: Although, a device with a private IPv4 address is not directly accessible from another device across the internet, the IETF does not consider private IPv4 addresses or NAT as effective security measures.

Organizations that have resources available to the internet, such as a web server, will also have devices that have public IPv4 addresses. As shown in the figure, this part of the network is known as the DMZ (demilitarized zone). The router in the figure not only performs routing, it also performs NAT and acts as a firewall for security.

The diagram is a network topology showing a router in the center with three connections; one to the company Intranet, one to a DMZ, and one to the Internet. On the left is the Intranet with devices using private IPv4 addresses. At the top, is the DMZ with two servers using public IPv4 addresses. On the right is the Internet cloud. The router is acting as a firewall and performing NAT.

Private IPv4
addressesRouter to the Internet**Internet**Public IPv4
addresses**DMZIntranet**

Note: Private IPv4 addresses are commonly used for educational purposes instead of using a public IPv4 address that most likely belongs to an organization.

11.3.3

Activity - Pass or Block IPv4 Addresses

Instructions:

Decide to Pass or Block each IP address depending on whether it is Public (the Internet) or Private (small local network). Click Start to begin and click on either Pass or Block.

Start

Reset

Block

Pass

11.3.4

Special Use IPv4 Addresses

There are certain addresses, such as the network address and broadcast address, that cannot be assigned to hosts. There are also special addresses that can be assigned to hosts, but with restrictions on how those hosts can interact within the network.

Loopback addresses

Loopback addresses (127.0.0.0 /8 or 127.0.0.1 to 127.255.255.254) are more commonly identified as only 127.0.0.1, these are special addresses used by a host to direct traffic to itself. For example, it can be used on a host to test if the TCP/IP configuration is operational, as shown in the figure. Notice how the 127.0.0.1 loopback address replies to the **ping** command. Also note how any address within this block will loop back to the local host, which is shown with the second **ping** in the figure.

Pinging the Loopback Interface

```
C:\Users\NetAcad> ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:

    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\NetAcad> ping 127.1.1.1

Pinging 127.1.1.1 with 32 bytes of data:

Reply from 127.1.1.1: bytes=32 time<1ms TTL=128

Reply from 127.1.1.1: bytes=32 time<1ms TTL=128

Reply from 127.1.1.1: bytes=32 time<1ms TTL=128

Reply from 127.1.1.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.1.1.1:

    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\NetAcad>
```

Link-Local addresses

Link-local addresses (169.254.0.0 /16 or 169.254.0.1 to 169.254.255.254) are more commonly known as the Automatic Private IP Addressing (APIPA) addresses or self-assigned addresses. They are used by a Windows DHCP client to self-configure in the event that there are no DHCP servers available. Link-local addresses can be used in a peer-to-peer connection but are not commonly used for this purpose.

Legacy Classful Addressing

In 1981, IPv4 addresses were assigned using classful addressing as defined in RFC 790 (https://tools.ietf.org/html/rfc790), Assigned Numbers. Customers were allocated a network address based on one of three classes, A, B, or C. The RFC divided the unicast ranges into specific classes as follows:

- **Class A (0.0.0.0/8 to 127.0.0.0/8)** - Designed to support extremely large networks with more than 16 million host addresses. Class A used a fixed /8 prefix with the first octet to indicate the network address and the remaining three octets for host addresses (more than 16 million host addresses per network).
- **Class B (128.0.0.0 /16 - 191.255.0.0 /16)** - Designed to support the needs of moderate to large size networks with up to approximately 65,000 host addresses. Class B used a fixed /16 prefix with the two high-order octets to indicate the network address and the remaining two octets for host addresses (more than 65,000 host addresses per network).
- **Class C (192.0.0.0 /24 - 223.255.255.0 /24)** - Designed to support small networks with a maximum of 254 hosts. Class C used a fixed /24 prefix with the first three octets to indicate the network and the remaining octet for the host addresses (only 254 host addresses per network).

Note: There is also a Class D multicast block consisting of 224.0.0.0 to 239.0.0.0 and a Class E experimental address block consisting of 240.0.0.0 - 255.0.0.0.

At the time, with a limited number of computers using the internet, classful addressing was an effective means to allocate addresses. As shown in the figure, Class A and B networks have a very large number of host addresses and Class C has very few. Class A networks accounted for 50% of the IPv4 networks. This caused most of the available IPv4 addresses to go unused.

The diagram is a pie chart showing the percentage of Class A, B, C, D, & E IPv4 addressing with the total number of networks and hosts per class A, B, and C networks. Percentages are: class A = 50%, class B = 25%, class C = 12.5%, and class D and E = 12.5%. For the total number of networks and total number of hosts per network: class A = 128 networks with 16,777,214 total hosts per network; class B = 16,384 networks with 65,534 total hosts per network; and class C = 2,097,152 networks with 254 total hosts per network.

In the mid-1990s, with the introduction of the World Wide Web (WWW), classful addressing was deprecated to more efficiently allocate the limited IPv4 address space. Classful address allocation was replaced with classless addressing, which is used today. Classless addressing ignores the rules of classes (A, B, C). Public IPv4 network addresses (network addresses and subnet masks) are allocated based on the number of addresses that can be justified.

11.3.6

Assignment of IP Addresses

Public IPv4 addresses are addresses which are globally routed over the internet. Public IPv4 addresses must be unique.

Both IPv4 and IPv6 addresses are managed by the Internet Assigned Numbers Authority (IANA). The IANA manages and allocates blocks of IP addresses to the Regional Internet Registries (RIRs). The five RIRs are shown in the figure.

RIRs are responsible for allocating IP addresses to ISPs who provide IPv4 address blocks to organizations and smaller ISPs. Organizations can also get their addresses directly from an RIR (subject to the policies of that RIR).

This figure shows the geographic locations of the Reginal Internet Registries (RIR). The regions governed by each RIR are as follows: AfriNIC (African Network Information Center) – serving the Africa Region, APNIC (Asia Pacific Network Information Centre) – serving the Asia/Pacific Region, ARIN (American Registry for Internet Numbers) – serving the North America Region, LACNIC (Regional Latin-American and Caribbean IP Address Registy) – serving Latin America and some Caribbean Islands, and RIPE NCC (Reseaux IP Europeens Network Coordination Centre) – serving Europe, the Middle East, and Central Asia.

Regional Internet Registries

- **AfriNIC** (African Network Information Centre) - Africa Region
- **APNIC** (Asia Pacific Network Information Centre) - Asia/Pacific Region
- **ARIN** (American Registry for Internet Numbers) - North America Region
- **LACNIC** (Regional Latin-American and Caribbean IP Address Registry) - Latin America and some Caribbean Islands
- **RIPE NCC** (Réseaux IP Européens Network Coordination Centre) - Europe, the Middle East, and Central Asia

11.4.1

Broadcast Domains and Segmentation

Have you ever received an email that was addressed to every person at your work or school? This was a broadcast email. Hopefully, it contained information that each of you needed to know. But often a broadcast is not really pertinent to everyone in the mailing list. Sometimes, only a segment of the population needs to read that information.

In an Ethernet LAN, devices use broadcasts and the Address Resolution Protocol (ARP) to locate other devices.. ARP sends Layer 2 broadcasts to a known IPv4 address on the local network to discover the associated MAC address. Devices on Ethernet LANs also locate other devices using services. A host typically acquires its IPv4 address configuration using the Dynamic Host Configuration Protocol (DHCP) which sends broadcasts on the local network to locate a DHCP server.

Switches propagate broadcasts out all interfaces except the interface on which it was received. For example, if a switch in the figure were to receive a broadcast, it would forward it to the other switches and other users connected in the network.

A router, R1, is connected to a switch via interface G0/0. The switch has connections to three other switches. The broadcast domain consists of the four switches and the router interface to which they are connected. A connection from the router to the Internet is not within the broadcast domain.

Routers Segment Broadcast Domains

R1G0/0

InternetBroadcast Domain

Routers do not propagate broadcasts. When a router receives a broadcast, it does not forward it out other interfaces. For instance, when R1 receives a broadcast on its Gigabit Ethernet 0/0 interface, it does not forward out another interface.

Therefore, each router interface connects to a broadcast domain and broadcasts are only propagated within that specific broadcast domain.

11.4.2

Problems with Large Broadcast Domains

A large broadcast domain is a network that connects many hosts. A problem with a large broadcast domain is that these hosts can generate excessive broadcasts and negatively affect the network. In the figure, LAN 1 connects 400 users that could generate an excess amount of broadcast traffic. This results in slow network operations due to the significant amount of traffic it can cause, and slow device operations because a device must accept and process each broadcast packet.

A router, R1, is connected to a switch via interface G0/0. The switch has connections to three other switches. The broadcast domain consists of the four switches and the router interface to which they are connected. This is

identified as LAN1 with an address of 172.16.0.0/16. A connection from the router to the Internet is not within the broadcast domain.

A Large Broadcast Domain

R1G0/0LAN 1: 172.16.0.0/16

Internet(400 users)

The solution is to reduce the size of the network to create smaller broadcast domains in a process called subnetting. These smaller network spaces are called subnets.

In the figure, the 400 users in LAN 1 with network address 172.16.0.0 /16 have been divided into two subnets of 200 users each: 172.16.0.0 /24 and 172.16.1.0 /24. Broadcasts are only propagated within the smaller broadcast domains. Therefore, a broadcast in LAN 1 would not propagate to LAN 2.

A router, R1, is connected to two LANs which represent two different broadcast domains. Connected on the left via G0/0 is a switch supporting 200 users in LAN 1 with a network address of 172.16.0.0/24. Connected on the right via G0/1 is a switch supporting 200 users in LAN 2 with a network address of 172.16.1.0/24.

Communicating Between Networks

LAN 1: 172.16.0.0/24LAN 2: 172.16.1.0/24G0/1R1G0/0

Internet(200 users)(200 users)

Notice how the prefix length has changed from a single /16 network to two /24 networks. This is the basis of subnetting: using host bits to create additional subnets.

Note: The terms subnet and network are often used interchangeably. Most networks are a subnet of some larger address block.

11.4.3

Reasons for Segmenting Networks

Subnetting reduces overall network traffic and improves network performance. It also enables an administrator to implement security policies such as which subnets are allowed or not allowed to communicate together. Another reason is that it reduces the number of devices affected by abnormal broadcast traffic due to misconfigurations, hardware/software problems, or malicious intent.

There are various ways of using subnets to help manage network devices.

Click each image for an illustration of how network administrators can group devices and services into subnets.
Location

Group or Function

Device Type

Subnetting by Location

The diagram shows a five floor building with a switch on each floor. Each switch is on a different LAN/subnet with a different network address, all connected to the same router, R1, via a different gigabit Ethernet interface. The following subnets are shown from the first to the fifth floor: LAN 1 has a network address of 10.0.1.0/24 and is connected to G0/0; LAN 2 has a network address of 10.0.2.0/24 and is connected to G0/1; LAN 3 has a network address of 10.0.3.0/24 and is connected to G0/2; LAN 4 has a network address of 10.0.4.0/24 and is connected to G0/3; and LAN 5 has a network address of 10.0.5.0/24 and is connected to G0/4. R1 also has a connection to the Internet.

R1G0/0G0/1G0/2G0/3G0/4

LAN 5: 10.0.5.0 /24 (Fifth floor)**LAN 4: 10.0.4.0 /24** (Fourth floor)**LAN 3: 10.0.3.0 /24** (Third floor)**LAN 2: 10.0.2.0 /24** (Second floor)**LAN 1: 10.0.1.0 /24**(First floor)Internet

Network administrators can create subnets using any other division that makes sense for the network. Notice in each figure, the subnets use longer prefix lengths to identify networks.

Understanding how to subnet networks is a fundamental skill that all network administrators must develop. Various methods have been created to help understand this process. Although a little overwhelming at first, pay close attention to the detail and, with practice, subnetting will become easier.

Subnet on an Octet Boundary

In the previous topic you learned several good reasons for segmenting a network. You also learned that segmenting a network is called subnetting. Subnetting is a critical skill to have when administering an IPv4 network. It is a bit daunting at first, but it gets much easier with practice.

IPv4 subnets are created by using one or more of the host bits as network bits. This is done by extending the subnet mask to borrow some of the bits from the host portion of the address to create additional network bits. The more host bits that are borrowed, the more subnets that can be defined. The more bits that are borrowed to increase the number of subnets reduces the number of hosts per subnet.

Networks are most easily subnetted at the octet boundary of /8, /16, and /24. The table identifies these prefix lengths. Notice that using longer prefix lengths decreases the number of hosts per subnet.

Subnet Masks on Octet Boundaries

Prefix LengthSubnet MaskSubnet Mask in Binary (n = network, h = host)# of hosts/8255.0.0.0nnnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh 11111111.00000000.00000000.0000000016,777,214/16255.255.0.0nnnnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh 11111111.11111111.00000000.0000000065,534/24255.255.255.0nnnnnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh 11111111.11111111.11111111.00000000254

Prefix Length	Subnet Mask	Subnet Mask in Binary (n = network, h = host)	# of hosts
/8	**255**.0.0.0	**nnnnnnnn**.hhhhhhhh.hhhhhhhh.hhhhhhhh **11111111**.00000000.00000000.00000000	16,777,214
/16	**255.255**.0.0	**nnnnnnnn.nnnnnnnn**.hhhhhhhh.hhhhhhhh **11111111.11111111**.00000000.00000000	65,534
/24	**255.255.255**.0	**nnnnnnnn.nnnnnnnn.nnnnnnnn**.hhhhhhhh **11111111.11111111.11111111**.00000000	254

To understand how subnetting on the octet boundary can be useful, consider the following example. Assume an enterprise has chosen the private address 10.0.0.0/8 as its internal network address. That network address can connect 16,777,214 hosts in one broadcast domain. Obviously, having more than 16 million hosts on a single subnet is not ideal.

The enterprise could further subnet the 10.0.0.0/8 address at the octet boundary of /16 as shown in the table. This would provide the enterprise the ability to define up to 256 subnets (i.e., 10.0.0.0/16 - 10.255.0.0/16) with each subnet capable of connecting 65,534 hosts. Notice how the first two octets identify the network portion of the address whereas the last two octets are for host IP addresses.

Subnetting Network 10.0.0.0/8 using a /16

Subnet Address (256 Possible Subnets)Host Range (65,534 possible hosts per subnet)Broadcast10.0.0.0/1610.0.0.1 - 10.0.255.25410.0.255.25510.1.0.0/1610.1.0.1 - 10.1.255.25410.1.255.25510.2.0.0/1610.2.0.1 - 10.2.255.25410.2.255.25510.3.0.0/1610.3.0.1 - 10.3.255.25410.3.255.25510.4.0.0/1610.4.0.1 - 10.4.255.25410.4.255.25510.5.0.0/1610.5.0.1 - 10.5.255.25410.5.255.25510.6.0.0/1610.6.0.1 - 10.6.255.25410.6.255.25510.7.0.0/1610.7.0.1 - 10.7.255.25410.7.255.255.....10.255.0.0/1610.255.0.1 - 10.255.255.25410.255.255.255

Subnet Address (256 Possible Subnets)	Host Range (65,534 possible hosts per subnet)	Broadcast
10.0.0.0**/16**	**10.0**.0.1 - **10.0**.255.254	**10.0**.255.255
10.1.0.0**/16**	**10.1**.0.1 - **10.1**.255.254	**10.1**.255.255

Subnet Address (256 Possible Subnets)Host Range (65,534 possible hosts per subnet)Broadcast10.0.0.0/1610.0.0.1 - 10.0.255.25410.0.255.25510.1.0.0/1610.1.0.1 - 10.1.255.25410.1.255.25510.2.0.0/1610.2.0.1 - 10.2.255.25410.2.255.25510.3.0.0/1610.3.0.1 - 10.3.255.25410.3.255.25510.4.0.0/1610.4.0.1 - 10.4.255.25410.4.255.25510.5.0.0/1610.5.0.1 - 10.5.255.25410.5.255.25510.6.0.0/1610.6.0.1 - 10.6.255.25410.6.255.25510.7.0.0/1610.7.0.1 - 10.7.255.25410.7.255.255.....10.255.0.0/1610.255.0.1 - 10.255.255.25410.255.255.255

Subnet Address (256 Possible Subnets)	**Host Range (65,534 possible hosts per subnet)**	**Broadcast**
10.2.0.0**/16**	**10.2**.0.1 - **10.2**.255.254	**10.2**.255.255
10.3.0.0**/16**	**10.3**.0.1 - **10.3**.255.254	**10.3**.255.255
10.4.0.0**/16**	**10.4**.0.1 - **10.4**.255.254	**10.4**.255.255
10.5.0.0**/16**	**10.5**.0.1 - **10.5**.255.254	**10.5**.255.255
10.6.0.0**/16**	**10.6**.0.1 - **10.6**.255.254	**10.6**.255.255
10.7.0.0**/16**	**10.7**.0.1 - **10.7**.255.254	**10.7**.255.255
...	...	...
10.255.0.0**/16**	**10.255**.0.1 - **10.255**.255.254	**10.255**.255.255

Alternatively, the enterprise could choose to subnet the 10.0.0.0/8 network at the /24 octet boundary, as shown in the table. This would enable the enterprise to define 65,536 subnets each capable of connecting 254 hosts. The /24 boundary is very popular in subnetting because it accommodates a reasonable number of hosts and conveniently subnets at the octet boundary.

Subnetting Network 10.0.0.0/8 using a /24 Prefix

Subnet Address (65,536 Possible Subnets)Host Range (254 possible hosts per subnet)Broadcast10.0.0.0/2410.0.0.1 - 10.0.0.25410.0.0.25510.0.1.0/2410.0.1.1 - 10.0.1.25410.0.1.25510.0.2.0/2410.0.2.1 - 10.0.2.25410.0.2.255.....10.0.255.0/2410.0.255.1 - 10.0.255.25410.0.255.25510.1.0.0/2410.1.0.1 - 10.1.0.25410.1.0.25510.1.1.0/2410.1.1.1 - 10.1.1.25410.1.1.25510.1.2.0/2410.1.2.1 - 10.1.2.25410.1.2.255.....10.100.0.0/2410.100.0.1 - 10.100.0.25410.100.0.255.....10.255.255.0/2410.255.255.1 - 10.2255.255.25410.255.255.255

Subnet Address (65,536 Possible Subnets)	**Host Range (254 possible hosts per subnet)**	**Broadcast**
10.0.0.0**/24**	**10.0.0**.1 - **10.0.0**.254	**10.0.0**.255
10.0.1.0**/24**	**10.0.1**.1 - **10.0.1**.254	**10.0.1**.255
10.0.2.0**/24**	**10.0.2**.1 - **10.0.2**.254	**10.0.2**.255
...	...	...
10.0.255.0**/24**	**10.0.255**.1 - **10.0.255**.254	**10.0.255**.255
10.1.0.0**/24**	**10.1.0**.1 - **10.1.0**.254	**10.1.0**.255
10.1.1.0**/24**	**10.1.1**.1 - **10.1.1**.254	**10.1.1**.255
10.1.2.0**/24**	**10.1.2**.1 - **10.1.2**.254	**10.1.2**.255

Subnet Address (65,536 Possible Subnets)Host Range (254 possible hosts per subnet)Broadcast10.0.0.0/2410.0.0.1 - 10.0.0.25410.0.0.25510.0.1.0/2410.0.1.1 - 10.0.1.25410.0.1.25510.0.2.0/2410.0.2.1 - 10.0.2.25410.0.2.255.....10.0.255.0/2410.0.255.1 - 10.0.255.25410.0.255.25510.1.0.0/2410.1.0.1 - 10.1.0.25410.1.0.25510.1.1.0/2410.1.1.1 - 10.1.1.25410.1.1.25510.1.2.0/2410.1.2.1 - 10.1.2.25410.1.2.255.....10.100.0.0/2410.100.0.1 - 10.100.0.25410.100.0.255.....10.255.255.0/2410.255.255.1 - 10.2255.255.25410.255.255.255

Subnet Address (65,536 Possible Subnets)	**Host Range (254 possible hosts per subnet)**	**Broadcast**
...	...	...
10.100.0.0**/24**	**10.100.0**.1 - **10.100.0**.254	**10.100.0**.255
...	...	...
10.255.255.0**/24**	**10.255.255**.1 - **10.2255.255**.254	**10.255.255**.255

11.5.2

Subnet within an Octet Boundary

The examples shown thus far borrowed host bits from the common /8, /16, and /24 network prefixes. However, subnets can borrow bits from any host bit position to create other masks.

For instance, a /24 network address is commonly subnetted using longer prefix lengths by borrowing bits from the fourth octet. This provides the administrator with additional flexibility when assigning network addresses to a smaller number of end devices.

Refer to the table to see six ways to subnet a /24 network.

Subnet a /24 Network

Prefix LengthSubnet MaskSubnet Mask in Binary (n = network, h = host)# of subnets# of hosts/25255.255.255.128nnnnnnnn.nnnnnnnn.nnnnnnnn.nhhhhhhh 11111111.11111111.11111111.100000002126/26255.255.255.192nnnnnnnn.nnnnnnnn.nnnnnnnn.nnhhhhhh 11111111.11111111.11111111.11000000462/27255.255.255.224nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnhhhhh 11111111.11111111.11111111.11100000830/28255.255.255.240nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnhhhh 11111111.11111111.11111111.111100001614/29255.255.255.248nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnhhh 11111111.11111111.11111111.11111000326/30255.255.255.252nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnnhh 11111111.11111111.11111111.11111100642

Prefix Length	**Subnet Mask**	**Subnet Mask in Binary (n = network, h = host)**	**# of subnets**	**# of hosts**
/25	255.255.255.128	nnnnnnnn.nnnnnnnn.nnnnnnnn.**n**hhhhhhh 11111111.11111111.11111111.**1**0000000	**2**	126
/26	255.255.255.192	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nn**hhhhhh 11111111.11111111.11111111.**11**000000	**4**	62
/27	255.255.255.224	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnn**hhhhh 11111111.11111111.11111111.**111**00000	**8**	30
/28	255.255.255.240	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnnn**hhhh 11111111.11111111.11111111.**1111**0000	**16**	14
/29	255.255.255.248	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnnnn**hhh 11111111.11111111.11111111.**11111**000	**32**	6
/30	255.255.255.252	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnnnnn**hh 11111111.11111111.11111111.**111111**00	**64**	2

For each bit borrowed in the fourth octet, the number of subnetworks available is doubled, while reducing the number of host addresses per subnet:

- **/25 row** - Borrowing 1 bit from the fourth octet creates 2 subnets supporting 126 hosts each.
- **/26 row** - Borrowing 2 bits creates 4 subnets supporting 62 hosts each.
- **/27 row** - Borrowing 3 bits creates 8 subnets supporting 30 hosts each.
- **/28 row** - Borrowing 4 bits creates 16 subnets supporting 14 hosts each.
- **/29 row** - Borrowing 5 bits creates 32 subnets supporting 6 hosts each.
- **/30 row** - Borrowing 6 bits creates 64 subnets supporting 2 hosts each.

11.6.1

Create Subnets with a Slash 16 prefix

Some subnetting is easier than other subnetting. This topic explains how to create subnets that each have the same number of hosts.

In a situation requiring a larger number of subnets, an IPv4 network is required that has more hosts bits available to borrow. For example, the network address 172.16.0.0 has a default mask of 255.255.0.0, or /16. This address has 16 bits in the network portion and 16 bits in the host portion. The 16 bits in the host portion are available to borrow for creating subnets. The table highlights all the possible scenarios for subnetting a /16 prefix.

Subnet a /16 Network

Prefix LengthSubnet MaskNetwork Address (n = network, h = host)# of subnets# of hosts/17255.255.128.0nnnnnnnn.nnnnnnnn.nhhhhhhh.hhhhhhhh
11111111.11111111.10000000.00000000232766/18255.255.192.0nnnnnnnn.nnnnnnnn.nnhhhhhh.hhhhhhhh
11111111.11111111.11000000.00000000416382/19255.255.224.0nnnnnnnn.nnnnnnnn.nnnhhhhh.hhhhhhhh
11111111.11111111.11100000.0000000088190/20255.255.240.0nnnnnnnn.nnnnnnnn.nnnnhhhh.hhhhhhhh
11111111.11111111.11110000.00000000164094/21255.255.248.0nnnnnnnn.nnnnnnnn.nnnnnhhh.hhhhhhhh
11111111.11111111.11111000.00000000322046/22255.255.252.0nnnnnnnn.nnnnnnnn.nnnnnnhh.hhhhhhhh
11111111.11111111.11111100.00000000641022/23255.255.254.0nnnnnnnn.nnnnnnnn.nnnnnnnh.hhhhhhhh
11111111.11111111.11111110.00000000128510/24255.255.255.0nnnnnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh
11111111.11111111.11111111.00000000256254/25255.255.255.128nnnnnnnn.nnnnnnnn.nnnnnnnn.nhhhhhhh
11111111.11111111.11111111.10000000512126/26255.255.255.192nnnnnnnn.nnnnnnnn.nnnnnnnn.nnhhhhhh
11111111.11111111.11111111.11000000102462/27255.255.255.224nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnhhhhh
11111111.11111111.11111111.11100000204830/28255.255.255.240nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnhhhh
11111111.11111111.11111111.11110000409614/29255.255.255.248nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnhhh
11111111.11111111.11111111.1111100081926/30255.255.255.252nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnnhh
11111111.11111111.11111111.11111100163842

Prefix Length	Subnet Mask	Network Address (n = network, h = host)	# of subnets	# of hosts
/17	255.255.128.0	nnnnnnnn.nnnnnnnn.**n**hhhhhhh.hhhhhhhh 11111111.11111111.**1**0000000.00000000	**2**	32766
/18	255.255.192.0	nnnnnnnn.nnnnnnnn.**nn**hhhhhh.hhhhhhhh 11111111.11111111.**11**000000.00000000	**4**	16382
/19	255.255.224.0	nnnnnnnn.nnnnnnnn.**nnn**hhhhh.hhhhhhhh 11111111.11111111.**111**00000.00000000	**8**	8190
/20	255.255.240.0	nnnnnnnn.nnnnnnnn.**nnnn**hhhh.hhhhhhhh 11111111.11111111.**1111**0000.00000000	**16**	4094
/21	255.255.248.0	nnnnnnnn.nnnnnnnn.**nnnnn**hhh.hhhhhhhh 11111111.11111111.**11111**000.00000000	**32**	2046
/22	255.255.252.0	nnnnnnnn.nnnnnnnn.**nnnnnn**hh.hhhhhhhh 11111111.11111111.**111111**00.00000000	**64**	1022
/23	255.255.254.0	nnnnnnnn.nnnnnnnn.**nnnnnnn**h.hhhhhhhh 11111111.11111111.**1111111**0.00000000	**128**	510
/24	255.255.255.0	nnnnnnnn.nnnnnnnn.**nnnnnnnn**.hhhhhhhh 11111111.11111111.**11111111**.00000000	**256**	254
/25	255.255.255.128	nnnnnnnn.nnnnnnnn.**nnnnnnnn.n**hhhhhhh 11111111.11111111.**11111111.1**0000000	**512**	126

Prefix LengthSubnet MaskNetwork Address (n = network, h = host)# of subnets# of hosts/17255.255.128.0nnnnnnnn.nnnnnnnn.nhhhhhhh.hhhhhhhh
11111111.11111111.10000000.00000000232766/18255.255.192.0nnnnnnnn.nnnnnnnn.nnhhhhhh.hhhhhhhh
11111111.11111111.11000000.00000000416382/19255.255.224.0nnnnnnnn.nnnnnnnn.nnnhhhhh.hhhhhhhh
11111111.11111111.11100000.0000000088190/20255.255.240.0nnnnnnnn.nnnnnnnn.nnnnhhhh.hhhhhhhh
11111111.11111111.11110000.00000000164094/21255.255.248.0nnnnnnnn.nnnnnnnn.nnnnnhhh.hhhhhhhh
11111111.11111111.11111000.00000000322046/22255.255.252.0nnnnnnnn.nnnnnnnn.nnnnnnhh.hhhhhhhh
11111111.11111111.11111100.00000000641022/23255.255.254.0nnnnnnnn.nnnnnnnn.nnnnnnnh.hhhhhhhh
11111111.11111111.11111110.00000000128510/24255.255.255.0nnnnnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh
11111111.11111111.11111111.00000000256254/25255.255.255.128nnnnnnnn.nnnnnnnn.nnnnnnnn.nhhhhhhh
11111111.11111111.11111111.10000000512126/26255.255.255.192nnnnnnnn.nnnnnnnn.nnnnnnnn.nnhhhhhh
11111111.11111111.11111111.11000000102462/27255.255.255.224nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnhhhhh
11111111.11111111.11111111.11100000204830/28255.255.255.240nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnhhhh
11111111.11111111.11111111.11110000409614/29255.255.255.248nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnhhh
11111111.11111111.11111111.11111100081926/30255.255.255.252nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnnhh
11111111.11111111.11111111.11111100163842

Prefix Length	Subnet Mask	Network Address (n = network, h = host)	# of subnets	# of hosts
/26	255.255.255.192	nnnnnnnn.nnnnnnnn.**nnnnnnnn.nn**hhhhhh 11111111.11111111.**11111111.11**000000	**1024**	62
/27	255.255.255.224	nnnnnnnn.nnnnnnnn.**nnnnnnnn.nnn**hhhhh 11111111.11111111.**11111111.111**00000	**2048**	30
/28	255.255.255.240	nnnnnnnn.nnnnnnnn.**nnnnnnnn.nnnn**hhhh 11111111.11111111.**11111111.1111**0000	**4096**	14
/29	255.255.255.248	nnnnnnnn.nnnnnnnn.**nnnnnnnn.nnnnn**hhh 11111111.11111111.**11111111.11111**000	**8192**	6
/30	255.255.255.252	nnnnnnnn.nnnnnnnn.**nnnnnnnn.nnnnnn**hh 11111111.11111111.**11111111.111111**00	**16384**	2

Although you do not need to memorize this table, you still need a good understanding of how each value in the table is generated. Do not let the size of the table intimidate you. The reason it is big is that it has 8 additional bits that can be borrowed, and, therefore, the numbers of subnets and hosts are simply larger.

11.6.2

Create 100 Subnets with a Slash 16 prefix

Consider a large enterprise that requires at least 100 subnets and has chosen the private address 172.16.0.0/16 as its internal network address.

When borrowing bits from a /16 address, start borrowing bits in the third octet, going from left to right. Borrow a single bit at a time until the number of bits necessary to create 100 subnets is reached.

The figure displays the number of subnets that can be created when borrowing bits from the third octet and the fourth octet. Notice there are now up to 14 host bits that can be borrowed.

The graphic shows how to compute the number of subnets created when borrowing bits from the third and fourth octets of an IPv4 network address. The formula to determine the number of subnets created is 2 to the power of the number of bits borrowed. The graphic shows an address of 172.16.0.0. Underneath, are the letters nnnnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh. It starts by borrowing the first h bit in the third octet which results in 2 to the power of 1 = 2 subnets. When the first two h bits in the third octet are borrowed, the formula is 2 to the power of 2 = 4. This continues until the first 14 h bits are borrowed from the third and fourth octets resulting in 2 to the power of 14 = 16384. The last two h bits in the fourth octet remain the same.

Number of Subnets Created

nnnnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh172.016.0.

Borrowing **1** bit:2^**1** = **2**Borrowing **2** bit:2^**2** = **4**Borrowing **3** bit:2^**3** = **8**Borrowing **4** bit:2^**4** = **16**Borrowing **5** bit:2^**5** = **32**Borrowing **6** bit:2^**6** = **64**
Borrowing **7** bit:2^**7** = **128**Borrowing **8** bit:2^**8** = **256**Borrowing **9** bit:2^**9** = **512**Borrowing **10** bit:2^**10** = **1024**Borrowing **11** bit:2^**11** = **2048**Borrowi
ng **12** bit:2^**12** = **4096**Borrowing **13** bit:2^**13** = **8192**Borrowing **14** bit:2^**14** = **16384**

To satisfy the requirement of 100 subnets for the enterprise, 7 bits (i.e., 2^7 = 128 subnets) would need to be borrowed (for a total of 128 subnets), as shown in the figure.

The graphic shows the decimal and bit representation of a network address, and below it a subnet mask, when seven bits are borrowed in the third octet to create subnets. The first two octets are shown in decimal and the last two octets are shown in binary. The network address is 172.16.0000 0000.0000 0000. The subnet mask is 255.255.1111 1110.0000 0000.

172.16.0.0/23 Network

172. 16. 0000 000 0. 0000 0000255. 255. 1111 111 0. 0000 0000

Recall that the subnet mask must change to reflect the borrowed bits. In this example, when 7 bits are borrowed, the mask is extended 7 bits into the third octet. In decimal, the mask is represented as 255.255.254.0, or a /23 prefix, because the third octet is 11111110 in binary and the fourth octet is 00000000 in binary.

The figure displays the resulting subnets from 172.16.0.0 /23 up to 172.16.254.0 /23.

The graphic shows the subnets created when using a /23 subnet mask with the address 172.16.0.0. First, it shows the decimal and bit representation of the network address, and below it the subnet mask. The first two octets are shown in decimal and the last two octets are shown in binary. The network address is 172.16.0000 0000.0000 0000. The subnet mask is 255.255.1111 1110.0000 0000. The first two octets and the first seven bits in the third octet are shaded gray and the last bit in the third octet and the entire fourth octet are shaded purple. Below, the text reads: borrowing 7 bits creates 128 subnets. Below that, it shows the first three subnets and the last subnet created. Again, the first two octets are shown in decimal and the last two octets are shown in binary. The first subnet is 172.16.0000 0000.0000 0000 or 172.16.0.0/23. The second subnet is 172.16.0000 0010.0000 0000 or 172.16.2.0/23. The third subnet is 172.16.0000 0100.0000 0000 or 172.16.4.0/23. The text ..to.. is used to show that this process continues until you reach the last subnet created which is 172.16.1111 1110.0000 0000 or 172.16.254.0/23.

Resulting /23 Subnets

172.16.0.0/23172.16.2.0/23172.16.4.0/23172.16.00000000.00000000255.255.11111110.00000000172.16.254.0/23172.16.00000000.00000000172.16.00000010.00000000172.16.00000100.00000000172.16.11111110.00000000

Borrowing 7 bits creates 128 subnets. . to . .

After borrowing 7 bits for the subnet, there is one host bit remaining in the third octet, and 8 host bits remaining in the fourth octet, for a total of 9 bits that were not borrowed. 2^9 results in 512 total host addresses. The first address is reserved for the network address and the last address is reserved for the broadcast address, so subtracting for these two addresses (2^9 - 2) equals 510 available host addresses for each /23 subnet.

As shown in the figure, the first host address for the first subnet is 172.16.0.1, and the last host address is 172.16.1.254.

The graphic shows the address range for the 172.16.0.0/23 subnet. The first two octets are shown in decimal and the last two octets are shown in binary, then the address is shown in its dotted decimal format. The network address is 172.16.0000 0000.0000 0000 = 172.16.0.0/23. The first host address is 172.16.0000 0000.0000 0001 = 172.16.0.1/23. The last host address is 172.16.0000 0001.1111 1110 = 172.16.255.254/23 (change to 172.16.1.254 when fixed). The broadcast address is 172.16.0000 0001.1111 1111 = 172.16.255.255/23 (change to 172.16.1.255 when fixed).

Address Range for 172.16.0.0/23 Subnet

172.16.00000000.00000000172.16.00000000.00000001172.16.00000001.11111110172.16.00000001.11111111= 172.16.0.0/23= 172.16.0.1/23= 172.16.1.254/23= 172.16.1.255/23

Network AddressFirst Host AddressLast Host AddressBroadcast Address

11.6.3

Create 1000 Subnets with a Slash 8 prefix

Some organizations, such as small service providers or large enterprises, may need even more subnets. For example, take a small ISP that requires 1000 subnets for its clients. Each client will need plenty of space in the host portion to create its own subnets.

The ISP has a network address 10.0.0.0 255.0.0.0 or 10.0.0.0/8. This means there are 8 bits in the network portion and 24 host bits available to borrow toward subnetting. Therefore, the small ISP will subnet the 10.0.0.0/8 network.

To create subnets, you must borrow bits from the host portion of the IPv4 address of the existing internetwork. Starting from the left to the right with the first available host bit, borrow a single bit at a time until you reach the

number of bits necessary to create 1000 subnets. As shown in the figure, you need to borrow 10 bits to create 1024 subnets (2^{10} = 1024). This includes 8 bits in the second octet and 2 additional bits from the third octet.

The graphic shows how to compute the number of subnets created when borrowing bits from the second and third octets of an IPv4 network address. The formula to determine the number of subnets created is 2 to the power of the number of bits borrowed. The graphic shows an address of 10.0.0.0. Underneath, are the letters nnnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh. It starts by borrowing the first h bit in the second octet which results in 2 to the power of 1 = 2 subnets. When the first two h bits in the second octet are borrowed, the formula is 2 to the power of 2 = 4. This continues until the first 10 h bits are borrowed from the second and third octets resulting in 2 to the power of 10 = 1024.

Number of Subnets Created

nnnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh10.00.0.2^1 = 22^2 = 42^3 = 82^4 = 162^5 = 322^6 = 642^7 = 1282^8 = 2562^9 = 5122^10 = 1024

Borrowing **1** bit:Borrowing **2** bit:Borrowing **3** bit:Borrowing **4** bit:Borrowing **5** bit:Borrowing **6** bit:Borrowing **7** bit:Borrowing **8** bit:Borrowing **9** bit:Borrowing **10** bit:

This figure displays the network address and the resulting subnet mask, which converts to 255.255.192.0 or 10.0.0.0/18.

The graphic shows the decimal and bit representation of a network address, and below it a subnet mask, when 10 bits are borrowed in the second and third octets to create subnets. The first octet is shown in decimal and the last three octets are shown in binary. The network address is 10.1111 1111.1100 0000.0000 0000 (should be 10.0000 0000.0000 0000.0000 0000 when fixed). The subnet mask is 255.255.1111 1110.0000 0000.

10.0.0.0/18 Network

10.00000000.00000000.00000000255.11111111.11000000.00000000

This figure displays the subnets resulting from borrowing 10 bits, creating subnets from 10.0.0.0/18 to 10.255.192.0/18.

The graphic shows the subnets created when using a /18 subnet mask with the address 10.0.0.0. First, it shows the decimal and bit representation of the network address, and below it the subnet mask. The first octet is shown in decimal and the last three octets are shown in binary. The network address is 10.0000 0000.0000 0000.0000 0000. The subnet mask is 255.1111 1111.1100 0000.0000 0000. The first octet and the next 10 bits are shaded gray and the remaining bits are shaded purple. Below, the text reads: borrowing 10 bits creates 1024 subnets. Below that, it shows the first five subnets and the last subnet created. Again, the first octet is shown in decimal and the last three octets are shown in binary. The first subnet is 10.0000 0000.0000 0000.0000 0000 or 10.0.0.0/18. The second subnet is 10.0000 0000.0100 0000.0000 0000 or 10.0.64.0/18. The third subnet is 10.0000 0000.1000 0000.0000 0000 or 10.0.128.0/18. The fourth subnet is 10.0000 0000.1100 0000.0000 0000 or 10.0.192.0/18. The fifth subnet is 10.0000 0001.0000 0000.0000 0000 or 10.1.0.0/18. The text ..to.. is used to show that this process continues until you reach the last subnet created which is 10.1111 1111.1100 0000.0000 0000 or 10.255.192.0/18.

Resulting /18 Subnets

10.0.0.0/1810.0.64.0/1810.0.128.0/18255.11111111.11000000.0000000010.00000000.00000000.0000000010.00000000.00000000.0000000010.00000000.01000000.0000000010.00000000.10000000.0000000010.0.192.0/1810.1.0.0/1810.00000000.11000000.0000000010.00000001.00000000.0000000010.11111111.11000000.0000000010.255.192.0/18

Borrowing 10 bits creates 1024 subnets. . to . .

Borrowing 10 bits to create the subnets, leaves 14 host bits for each subnet. Subtracting two hosts per subnet (one for the network address and one for the broadcast address) equates to 2^{14} - 2 = 16382 hosts per subnet. This means that each of the 1000 subnets can support up to 16,382 hosts.

This figure displays the specifics of the first subnet.

The graphic shows the address range for the 10.0.0.0/18 subnet. The first octet is shown in decimal and the last three octets are shown in binary, then the address is shown in its dotted decimal format. The network address is 10.0000 0000.0000 0000.0000 0000 = 10.0.0.0/18. The first host address is 10.0000 0000.0000 0000.0000 0001 = 10.0.0.1/18. The last host address is 10.0000 0000.0011 1111.1111 1110 = 10.0.63.254/18. The broadcast address is 10.0000 0000.0011 1111. 1111 1111 = 10.0.63.255/18.

Address Range for 10.0.0.0/18 Subnet

10.00000000.00000000.0000000110.00000000.00000000.0000000010.00000000.00111111.1111111110.00000000.00111111.11111110= 10.0.0.0/18= 10.0.0.1/18= 10.0.63.254/18= 10.0.63.255/18

Network AddressFirst Host AddressLast Host AddressBroadcast Address

11.6.4

11.7.1

Subnet Private versus Public IPv4 Address Space

While it is nice to quickly segment a network into subnets, your organization's network may use both public and private IPv4 addresses. This affects how you will subnet your network.

The figure shows a typical enterprise network:

- **Intranet -** This is the internal part of a company's network, accessible only within the organization. Devices in the intranet use private IPv4 addresses.
- **DMZ -** This is part of the company's network containing resources available to the internet such as a web server. Devices in the DMZ use public IPv4 addresses.

The diagram is a network topology showing a router in the center with three connections; one to the company Intranet, one to a DMZ, and one to the Internet. On the left is the Intranet with devices using private IPv4 addresses. At the top, is the DMZ with two servers using public IPv4 addresses. The router is labeled router to the Internet and has a connection to the Internet cloud.

Public and Private IPv4 Address Space

InternetRouter to the InternetPrivate IPv4 Addresses**Intranet**Public IPv4 Addresses**DMZ**

Both the intranet and the DMZ have their own subnetting requirements and challenges.

The intranet uses private IPv4 addressing space. This allows an organization to use any of the private IPv4 network addresses including the 10.0.0.0/8 prefix with 24 host bits and over 16 million hosts. Using a network address with 24 host bits makes subnetting easier and more flexible. This includes subnetting on an octet boundary using a /16 or /24.

For example, the private IPv4 network address 10.0.0.0/8 can be subnetted using a /16 mask. As shown in the table, this results in 256 subnets, with 65,534 hosts per subnet. If an organization has a need for fewer than 200 subnets, allowing for some growth, this gives each subnet more than enough host addresses.

Subnetting Network 10.0.0.0/8 using a /16

Subnet Address (256 Possible Subnets)Host Range (65,534 possible hosts per subnet)Broadcast10.0.0.0/1610.0.0.1 - 10.0.255.25410.0.255.25510.1.0.0/1610.1.0.1 - 10.1.255.25410.1.255.25510.2.0.0/1610.2.0.1 - 10.2.255.25410.2.255.25510.3.0.0/1610.3.0.1 - 10.3.255.25410.3.255.25510.4.0.0/1610.4.0.1 - 10.4.255.25410.4.255.25510.5.0.0/1610.5.0.1 - 10.5.255.25410.5.255.25510.6.0.0/1610.6.0.1 - 10.6.255.25410.6.255.25510.7.0.0/1610.7.0.1 - 10.7.255.25410.7.255.255.....10.255.0.0/1610.255.0.1 - 10.255.255.25410.255.255.255

Subnet Address (256 Possible Subnets)	**Host Range (65,534 possible hosts per subnet)**	**Broadcast**
10.0.0.0**/16**	**10.0**.0.1 - **10.0**.255.254	**10.0**.255.255
10.1.0.0**/16**	**10.1**.0.1 - **10.1**.255.254	**10.1**.255.255
10.2.0.0/16	**10.2**.0.1 - **10.2**.255.254	**10.2**.255.255
10.3.0.0**/16**	**10.3**.0.1 - **10.3**.255.254	**10.3**.255.255
10.4.0.0**/16**	**10.4**.0.1 - **10.4**.255.254	**10.4**.255.255
10.5.0.0**/16**	**10.5**.0.1 - **10.5**.255.254	**10.5**.255.255

Subnet Address (256 Possible Subnets)Host Range (65,534 possible hosts per subnet)Broadcast10.0.0.0/1610.0.0.1 - 10.0.255.25410.0.255.25510.1.0.0/1610.1.0.1 - 10.1.255.25410.1.255.25510.2.0.0/1610.2.0.1 - 10.2.255.25410.2.255.25510.3.0.0/1610.3.0.1 - 10.3.255.25410.3.255.25510.4.0.0/1610.4.0.1 - 10.4.255.25410.4.255.25510.5.0.0/1610.5.0.1 - 10.5.255.25410.5.255.25510.6.0.0/1610.6.0.1 - 10.6.255.25410.6.255.25510.7.0.0/1610.7.0.1 - 10.7.255.25410.7.255.255.....10.255.0.0/1610.255.0.1 - 10.255.255.25410.255.255.255

Subnet Address (256 Possible Subnets)	**Host Range (65,534 possible hosts per subnet)**	**Broadcast**
10.6.0.0**/16**	**10.6**.0.1 - **10.6.**255.254	**10.6**.255.255
10.7.0.0**/16**	**10.7**.0.1 - **10.7**.255.254	**10.7**.255.255
...	...	...
10.255.0.0**/16**	**10.255**.0.1 - **10.255**.255.254	**10.255**.255.255

Another option using the 10.0.0.0/8 private IPv4 network address is to subnet using a /24 mask. As shown in the table, this results in 65,536 subnets, with 254 hosts per subnet. If an organization needs more than 256 subnets, then using a /24 can be used with 254 hosts per subnet.

Subnetting Network 10.0.0.0/8 using a /24

Subnet Address (65,536 Possible Subnets)Host Range(254 possible hosts per subnet)Broadcast10.0.0.0/2410.0.0.1 - 10.0.0.25410.0.0.25510.0.1.0/2410.0.1.1 - 10.0.1.25410.0.1.25510.0.2.0/2410.0.2.1 - 10.0.2.25410.0.2.255.....10.0.255.0/2410.0.255.1 - 10.0.255.25410.0.255.25510.1.0.0/2410.1.0.1 - 10.1.0.25410.1.0.25510.1.1.0/2410.1.1.1 - 10.1.1.25410.1.1.25510.1.2.0/2410.1.2.1 - 10.1.2.25410.1.2.255.....10.100.0.0/2410.100.0.1 - 10.100.0.25410.100.0.255.....10.255.255.0/2410.255.255.1 - 10.2255.255.25410.255.255.255

Subnet Address (65,536 Possible Subnets)	**Host Range (254 possible hosts per subnet)**	**Broadcast**
10.0.0.0**/24**	**10.0.0**.1 - **10.0.0**.254	**10.0.0**.255
10.0.1.0**/24**	**10.0.1**.1 - **10.0.1**.254	**10.0.1**.255
10.0.2.0**/24**	**10.0.2**.1 - **10.0.2**.254	**10.0.2**.255
...	...	...
10.0.255.0**/24**	**10.0.255**.1 - **10.0.255**.254	**10.0.255**.255
10.1.0.0**/24**	**10.1.0**.1 - **10.1.0**.254	**10.1.0**.255
10.1.1.0**/24**	**10.1.1**.1 - **10.1.1**.254	**10.1.1**.255
10.1.2.0**/24**	**10.1.2**.1 - **10.1.2**.254	**10.1.2**.255
...	...	...
10.100.0.0**/24**	**10.100.0**.1 - **10.100.0**.254	**10.100.0**.255
...	...	...
10.255.255.0**/24**	**10.255.255**.1 - **10.2255.255**.254	**10.255.255**.255

The 10.0.0.0/8 can also be subnetted using any other number of prefix lengths, such as /12, /18, /20, etc. This would give the network administrator a wide variety of options. Using a 10.0.0.0/8 private IPv4 network address makes subnet planning and implementation easy.

What about the DMZ?

Because these devices need to be publicly accessible from the internet, the devices in the DMZ require public IPv4 addresses. The depletion of public IPv4 address space became an issue beginning in the mid-1990s. Since 2011, IANA and four out of five RIRs have run out of IPv4 address space. Although organizations are making the transition to IPv6, the remaining IPv4 address space remains severely limited. This means an organization must maximize its own limited number of public IPv4 addresses. This requires the network administrator to subnet their public address space into subnets with different subnet masks, in order to minimize the number of unused host addresses per subnet. This is known as Variable Subnet Length Masking (VLSM).

11.7.2

Minimize Unused Host IPv4 Addresses and Maximize Subnets

To minimize the number of unused host IPv4 addresses and maximize the number of available subnets, there are two considerations when planning subnets: the number of host addresses required for each network and the number of individual subnets needed.

The table displays the specifics for subnetting a /24 network. Notice how there is an inverse relationship between the number of subnets and the number of hosts. The more bits that are borrowed to create subnets, the fewer host bits remain available. If more host addresses are needed, more host bits are required, resulting in fewer subnets.

The number of host addresses required in the largest subnet will determine how many bits must be left in the host portion. Recall that two of the addresses cannot be used, so the usable number of addresses can be calculated as 2n-2.

Subnetting a /24 Network

Prefix LengthSubnet MaskSubnet Mask in Binary (n = network, h = host)# of subnets# of hosts per subnet/25255.255.255.128nnnnnnnn.nnnnnnnn.nnnnnnnn.nhhhhhhh 11111111.11111111.11111111.100000002126/26255.255.255.192nnnnnnnn.nnnnnnnn.nnnnnnnn.nnhhhhhh 11111111.11111111.11111111.11000000462/27255.255.255.224nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnhhhhh 11111111.11111111.11111111.11100000830/28255.255.255.240nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnhhhh 11111111.11111111.11111111.111100001614/29255.255.255.248nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnhhh 11111111.11111111.11111111.11111000326/30255.255.255.252nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnnhh 11111111.11111111.11111111.11111100642

Prefix Length	Subnet Mask	Subnet Mask in Binary (n = network, h = host)	# of subnets	# of hosts per subnet
/25	255.255.255.128	nnnnnnnn.nnnnnnnn.nnnnnnnn.**n**hhhhhhh 11111111.11111111.11111111.**1**0000000	**2**	126
/26	255.255.255.192	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nn**hhhhhh 11111111.11111111.11111111.**11**000000	**4**	62
/27	255.255.255.224	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnn**hhhhh 11111111.11111111.11111111.**111**00000	**8**	30
/28	255.255.255.240	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnnn**hhhh 11111111.11111111.11111111.**1111**0000	**16**	14
/29	255.255.255.248	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnnnn**hhh 11111111.11111111.11111111.**11111**000	**32**	6
/30	255.255.255.252	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnnnnn**hh 11111111.11111111.11111111.**111111**00	**64**	2

Network administrators must devise the network addressing scheme to accommodate the maximum number of hosts for each network and the number of subnets. The addressing scheme should allow for growth in both the number of host addresses per subnet and the total number of subnets.

11.7.3

Example: Efficient IPv4 Subnetting

In this example, corporate headquarters has been allocated a public network address of 172.16.0.0/22 (10 host bits) by its ISP. As shown in the figure, this will provide 1,022 host addresses.

Note: 172.16.0.0/22 is part of the IPv4 private address space. We are using this address instead of an actual public IPv4 address.

The graphic shows the number of hosts provided when using a 172.16.0.0/22 network. The network portion of the address in binary is: 10101100.00010100.000000. The host portion in binary is: 00.00000000. The host portion consists of 10 host bits therefore 2 to the power of 10 - 2 = 1,022 hosts.

Network Address

172.16.0.0/2210101100.00010000.00000000. 00000000

10 host bits
2^{10} - 2 = 1,022 hostsNetwork portionHost portion

The corporate headquarters has a DMZ and four branch offices, each needing its own public IPv4 address space. Corporate headquarters needs to make best use of its limited IPv4 address space.

The topology shown in the figure consists of five sites; a corporate office and four branch sites. Each site requires internet connectivity and therefore, five internet connections. This means that the organization requires 10 subnets from the company's 172.16.0.0/22 public address. The largest subnet requires 40 addresses.

The diagram is a corporate network topology with five sites. In the middle is the ISP cloud. Connected to the cloud are five sites, each shown with a router, several servers, and the public IPv4 addressing requirements. The sites are: Corporate headquarters with 40 addresses; Branch 1 with 25 addresses; Branch 2 with 30 addresses; Branch 3 with 10 addresses; and Branch 4 with 15 addresses.

Corporate Topology with Five Sites

Corporate Headquarters40 public IPv4 addressesBranch 125 public IPv4 addresses30 public IPv4 addresses10 public IPv4 addresses15 public IPv4 addressesBranch 2ISPPublic IPv4 addressesBranch 3Branch 4

The 172.16.0.0/22 network address has 10 host bits, as shown in the figure. Because the largest subnet requires 40 hosts, a minimum of 6 host bits are needed to provide addressing for 40 hosts. This is determined by using this formula: 2^6 - 2 = 62 hosts.

The diagram shows the subnet scheme for the given address 172.16.0.0/22 with 4 bits borrowed from the host portion to create subnets. All four octets are shown in binary followed by the dotted decimal format for the given network address and for several subnets created. The given network address in binary is 10101100.00010000.000000 (network portion highlighted in gray) 00.00000000 (host portion highlighted in purple) = 172.16.0.0/22. For the subnets listed below, the first 22 bits are highlighted in gray (network portion), the next 4 bits are shaded in blue, and the last 6 bits are the remaining host portion shaded in purple. Subnet 0 is 10101100.00010000.00000000.00000000 = 172.16.0.0/26. Subnet 1 is 10101100.00010000.0000000.01000000 = 172.16.0.64/26. Subnet 2 is 10101100.00010000.00000000.10000000 = 172.16.0.128/26. Subnet 3 is 10101100.00010000.00000000.11000000 = 172.16.0.192/26. Subnet 4 is 10101100.00010000.00000001.00000000 = 172.16.1.0/26. Subnet 5 is 10101100.00010000.00000001.01000000 = 172.16.1.64/26. Subnet 6 is 10101100.00010000.00000001.10000000 = 172.16.1.128/26. Subnet 7 - 13 are not shown. Subnet 14 is 10101100.00010000.00000011.10000000 = 172.16.3.128/26. Subnet 15 is 10101100.00010000.00000011.11000000 = 172.16.3.192/26.

Subnet Scheme

10101100.00010000.00000014172.16.0.0/2211.11000000172.16.3.192/2611.10000000172.16.3.128/2610101100.00010000.00000010101100.00010000.00000000.0000000015501.01172.16.1.64/2600000010101100.00010000.000000601.10000000172.16.1.128/2610101100.00010000.0000004000000172.16.1.0/2601.0010101100.00010000.00000300.11172.16.0.192/2600000010101100.00010000.000000200.10172.16.0.128/2600000010101100.00010000.000000100.01172.16.0.64/2600000010101100.00010000.000000000.00000000172.16.0.0/2610101100.00010000.000000

Network portionHost portionDotted DecimalNets 7 - 13 not shown4-bits borrowed from host portion to create subnets

Using the formula for determining subnets results in 16 subnets: 2^4 = 16. Because the example internetwork requires 10 subnets, this will meet the requirement and allow for some additional growth.

Therefore, the first 4 host bits can be used to allocate subnets. This means two bits from the 3rd octet and two bits from the 4th octet will be borrowed. When 4 bits are borrowed from the 172.16.0.0/22 network, the new prefix length is /26 with a subnet mask of 255.255.255.192.

As shown in this figure, the subnets can be assigned to each location and router-to-ISP connections.

The diagram shows the subnet assignments for a corporate topology with five sites connected to an ISP cloud. Each site shows a router connected to the ISP, several servers, the public IPv4 addressing requirements, and the assigned subnet address. Each router-to-ISP connection has also been assigned a subnet address. The Corporate headquarters connection is assigned subnet 172.16.0.0/26 and the site with 40 addresses is assigned subnet 172.16.0.64/26. The Branch 1 connection is assigned subnet 172.16.0.128/26 and the site with 25 addresses is assigned 172.16.0.192/26. The Branch 2 connection is assigned subnet 172.16.1.0/26 and the site with 30 addresses is assigned subnet 172.16.1.64/26. The Branch 3 connection is assigned subnet 172.16.1.128/26 and the site with 10 addresses is assigned subnet 172.16.1.192/26. The Branch 4 connection is assigned subnet 172.16.2.0/26 and the site with 15 addresses is assigned subnet 172.16.2.64/26.

Subnet Assignments to each Site and ISP

172.16.2.64/26172.16.1.192/26172.16.1.64/26172.16.0.0/26172.16.1.0/26172.16.1.128/26172.16.2.0/26172.16.0.128/26172.16.0.192/26172.16.0.64/26

Corporate Headquarters40 public IPv4 addressesBranch 125 public IPv4 addresses30 public IPv4 addresses10 public IPv4 addresses15 public IPv4 addressesBranch 2ISPPublic IPv4 addressesBranch 3Branch 4

11.7.4

Video - VLSM Basics

As mentioned in the previous topic, public and private addresses affect the way you would subnet your network. There are also other issues that affect subnetting schemes. A standard /16 subnetting scheme creates subnets that each have the same number of hosts. Not every subnet you create will need this many hosts, leaving many IPv4 addresses unused. Perhaps you will need one subnet that contains many more hosts. This is why the variable-length subnet mask (VLSM) was developed.

Click Play to view a demonstration of basic VLSM techniques.

Play Video

11.8.2

Video - VLSM Example

Click Play to view a demonstration of VLSM subnetting.

Play Video

11.8.3

IPv4 Address Conservation

Because of the depletion of public IPv4 address space, making the most out of the available host addresses is a primary concern when subnetting IPv4 networks.

Note: The larger IPv6 address allows for much easier address planning and allocation than IPv4 allows. Conserving IPv6 addresses is not an issue. This is one of the driving forces for transitioning to IPv6.

Using traditional subnetting, the same number of addresses is allocated for each subnet. If all the subnets have the same requirements for the number of hosts, or if conserving IPv4 address space is not an issue, these fixed-size address blocks would be efficient. Typically, with public IPv4 addresses, that is not the case.

For example, the topology shown in the figure requires seven subnets, one for each of the four LANs, and one for each of the three connections between the routers.

The diagram shows a network topology consisting of seven subnets. There are four routers, each with an attached LAN and host addressing requirements, and three router-to-router connections requiring 2 hosts each.

The R1 router LAN is Building A with 25 hosts; the R2 router LAN is Building B with 20 hosts; the R3 router LAN is Building C with 15 hosts; and the R4 router LAN is Building D with 28 hosts.

R1R2R3R4

Building ABuilding BBuilding CBuilding D25 hosts20 hosts15 hosts28 hosts2 hosts2 hosts2 hosts

Using traditional subnetting with the given address of 192.168.20.0/24, three bits can be borrowed from the host portion in the last octet to meet the subnet requirement of seven subnets. As shown in the figure, borrowing 3 bits creates 8 subnets and leaves 5 host bits with 30 usable hosts per subnet. This scheme creates the needed subnets and meets the host requirement of the largest LAN.

The diagram shows the basic subnet scheme for a given address of 192.168.20.0/24 with three bits borrowed for subnetting. Having 3 bits for subnetting results in 2 to the power of 3 = 8 subnets. Having 5 bits for hosts results in 2 to the power of 5 - 2 = 30 host IP addresses per subnet. All four octets are shown in binary followed by the dotted decimal format for the given address and for all the subnets created. The given network address in binary is 11000000.10101000.00010100 (network portion highlighted in gray) .00000000 (host portion highlighted in purple) = 192.168.20.0/24. For the subnets listed below, the first 24 bits are highlighted in gray (network portion), the next three bits are highlighted in blue (subnet portion), and the last five bits are the remaining host bits highlighted in purple. Subnet 0 is 11000000.10101000.00010100.00000000 = 192.168.20.0/27. Subnet 1 is 11000000.10101000.00010100.00100000 = 192.168.20.32/27. Subnet 2 is 11000000.10101000.00010100.01000000 = 192.168.20.64/27. Subnet 3 is 11000000.10101000.00010100.01100000 = 192.168.20.96/27. Subnets 0 - 3 are assigned to building LANs A, B, C, and D. Subnet 4 is 11000000.10101000.00010100.10000000 = 192.168.20.128/27. Subnet 5 is 11000000.10101000.00010100.10100000 = 192.168.20.160/27. Subnet 6 is 11000000.10101000.00010100.11000000 = 192.168.20.192/27. Subnets 4, 5, and 6 are assigned to the site to site WANs. Subnet 7 is 11000000.10101000.00010100.11100000 = 192.168.20.224/27. Subnet 7 is unused/available.

Basic Subnet Scheme

11000000.10101000.00010100.00000000192.168.20.0/246.1100000011000000.10101000.00010100192.168.20.192/275.1010000011000000.10101000.00010100192.168.20.160/274.1000000011000000.10101000.00010100192.168.20.128/273.0110000011000000.10101000.00010100192.168.20.96/272.0100000011000000.10101000.00010100192.168.20.64/271.0010000011000000.10101000.00010100192.168.20.32/270.0000000011000000.10101000.00010100192.168.20.0/277.1110000011000000.10101000.00010100192.168.20.224/27

Building LANs A, B, C, and DSite to Site WANsUnused / AvailableNetwork PortionHost Portion Subnet portion

2^3 = 8 subnetsHost portion

2^5 – 2 = 30 host IP addresses per subnet

These seven subnets could be assigned to the LAN and WAN networks, as shown in the figure.

The diagram shows the subnet assignments for a network topology consisting of seven subnets. There are four routers, each with an attached LAN and host addressing requirements, and three router-to-router connections requiring 2 hosts each. The R1 router Building A LAN has 25 hosts and is assigned subnet 192.168.20.0/27. The R2 router Building B LAN has 20 hosts and is assigned the subnet 192.168.20.32/27. The R3 router Building C LAN has 15 hosts and is assigned subnet 192.168.20.64/27. The R4 router Building D LAN has 28 hosts and is assigned subnet 192.168.20.96/27. The R1 to R2 connection is assigned subnet 192.168.20.128/27. The R2 to R3 connection is assigned subnet 192.168.20.160/27. The R3 to R4 connection is assigned subnet 192.168.20.192/27.

R1R2R3R4192.168.20.128/27192.168.20.160/27192.168.20.192/27

Building A
192.168.20.0/27Building B
192.168.20.32/27Building C
192.168.20.64/27Building D
192.168.20.96/2725 hosts20 hosts15 hosts28 hosts2 hosts2 hosts2 hosts

Although this traditional subnetting meets the needs of the largest LAN and divides the address space into an adequate number of subnets, it results in significant waste of unused addresses.

For example, only two addresses are needed in each subnet for the three WAN links. Because each subnet has 30 usable addresses, there are 28 unused addresses in each of these subnets. As shown in the figure, this results in 84 unused addresses (28x3).

The graphic shows the unused addresses of four WAN subnets using a /27 subnet mask. All four octets are shown in binary followed by the dotted decimal format for the subnet. The first 24 bits are highlighted in gray (network portion), the next three bits are highlighted in blue, and the last five bits are the remaining host bits highlighted in purple. Subnet 4 is 11000000.10101000.00010100.10000000 = 192.168.20.128/27. Subnet 5 is 11000000.10101000.00010100.10100000 = 192.168.20.160/27. Subnet 6 is 11000000.10101000.00010100.11000000 = 192.168.20.192/27. Having 5 bits for hosts results in 2 to the power of 5 - 2 = 30 host IP addresses per subnet. 30 - 2 = 28; each WAN subnet wastes 28 addresses. 28 x 3 = 84; 84 addresses are unused.

Unused Addresses on WAN Subnets

5.1010000011000000.10101000.00010100192.168.20.160/276.1100000011000000.10101000.00010100192.168.20.192/2711000000.10101000.00010100.100000004192.168.20.128/27

Network PortionHost Portion Host portion

2^5 – 2 = 30 host IP addresses per subnet

30 – 2 = 28

Each WAN subnet wastes 28 addresses

28 x 3 = 84

84 addresses are unused Dotted Decimal

Further, this limits future growth by reducing the total number of subnets available. This inefficient use of addresses is characteristic of traditional subnetting. Applying a traditional subnetting scheme to this scenario is not very efficient and is wasteful.

The variable-length subnet mask (VLSM) was developed to avoid wasting addresses by enabling us to subnet a subnet.

11.8.4

VLSM

In all of the previous subnetting examples, the same subnet mask was applied for all the subnets. This means that each subnet has the same number of available host addresses. As illustrated in the left side of the figure, traditional subnetting creates subnets of equal size. Each subnet in a traditional scheme uses the same subnet mask. As shown in the right side of the figure, VLSM allows a network space to be divided into unequal parts. With VLSM, the subnet mask will vary depending on how many bits have been borrowed for a particular subnet, thus the “variable” part of the VLSM.

The graphic shows two pie charts that compare traditional subnetting to VLSM. On the right is a pie chart titled: traditional subnetting creates equal sized subnets. The pie is divided into 8 equal sized slices, each with 30 hosts. On the left is a pie chart titled: subnets of varying sizes. This pie has 7 slices identical to the first pie. The 8th slice has been further divided into 8 smaller slices. Text pointing to 8th slice reads: one subnet was further divided using a /30 subnet mask to create 8 smaller subnets of 2 hosts each.

One subnet was further divided using a /30 subnet mask to create 8 smaller subnets of 2 hosts each. 30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 HostsTraditional Subnetting Creates Equal Sized SubnetsSubnets of Varying Sizes

VLSM is just subnetting a subnet. The same topology used previously is shown in the figure. Again, we will use the 192.168.20.0/24 network and subnet it for seven subnets, one for each of the four LANs, and one for each of the three connections between the routers.

The diagram shows a network topology consisting of seven subnets. There are four routers, each with an attached LAN and host addressing requirements, and three router-to-router connections requiring 2 hosts each. The R1 router LAN is Building A with 25 hosts; the R2 router LAN is Building B with 20 hosts; the R3 router LAN is Building C with 15 hosts; and the R4 router LAN is Building D with 28 hosts.

R1R2R3R4

Building ABuilding BBuilding CBuilding D25 hosts20 hosts15 hosts28 hosts2 hosts2 hosts2 hosts

The figure shows how network 192.168.20.0/24 subnetted into eight equal-sized subnets with 30 usable host addresses per subnet. Four subnets are used for the LANs and three subnets could be used for the connections between the routers.

The diagram shows the basic subnet scheme for a given address of 192.168.20.0/24 with three bits borrowed for subnetting. All four octets are shown in binary followed by the dotted decimal format for the given address and for all the subnets created. The given network address in binary is 11000000.10101000.00010100 (network portion highlighted in gray) .00000000 (host portion highlighted in purple) = 192.168.20.0/24. For the subnets listed below, the first 24 bits are highlighted in gray (network portion), the next three bits are highlighted in blue (subnet portion), and the last five bits are the remaining host bits highlighted in purple. Subnet 0 is 11000000.10101000.00010100.00000000 = 192.168.20.0/27. Subnet 1 is 11000000.10101000.00010100.00100000 = 192.168.20.32/27. Subnet 2 is 11000000.10101000.00010100.01000000 = 192.168.20.64/27. Subnet 3 is 11000000.10101000.00010100.01100000 = 192.168.20.96/27. Subnets 0 - 3 are assigned to building LANs A, B, C, and D. Subnet 4 is 11000000.10101000.00010100.10000000 = 192.168.20.128/27. Subnet 5 is 11000000.10101000.00010100.10100000 = 192.168.20.160/27. Subnet 6 is 11000000.10101000.00010100.11000000 = 192.168.20.192/27. Subnets 4, 5, and 6 are unused/available. Subnet 7 is 11000000.10101000.00010100.11100000 = 192.168.20.224/27. Subnet 7 will be subnetted further.

Basic Subnetting Scheme

11000000.10101000.00010100 .00000000 192.168.20.0/24 0 11000000.10101000.00010100 .000 00000 192.168.20.0/27 1 11000000.10101000.00010100 .001 00000 192.168.20.32/27 2 11000000.10101000.00010100 .010 00000 192.168.20.64/27 3 11000000.10101000.00010100 .011 00000 192.168.20.96/27 4 11000000.10101000.00010100 .100 00000 192.168.20.128/27 5 11000000.10101000.00010100 .101 00000 192.168.20.160/27 6 11000000.10101000.00010100 .110 00000 192.168.20.192/27 7 11000000.10101000.00010100 .111 00000 192.168.20.224/27

Network portionHost portionDotted DecimalLAN's A, B, C, DUnused / Available Subnet 7 will be subnetted further.

However, the connections between the routers require only two host addresses per subnet (one host address for each router interface). Currently all subnets have 30 usable host addresses per subnet. To avoid wasting 28 addresses per subnet, VLSM can be used to create smaller subnets for the inter-router connections.

To create smaller subnets for the inter-router links, one of the subnets will be divided. In this example, the last subnet, 192.168.20.224/27, will be further subnetted. The figure shows the last subnet has been subnetted further by using the subnet mask 255.255.255.252 or /30.

The diagram show the VLSM subnetting scheme when the subnet 192.168.20.224/27 is further subnetted by borrowing 3 more bits. For the original subnet, the first 24 bits represent the network portion and are 11000000.10101000.00010100. The next three bits represent the subnet portion and are 111. The last five bits represent the host portion and are 00000. The address in dotted decimal is 192.168.20.224/27. Borrowing 3 additional bits, subnetting a subnet, results in dividing the original subnet into 8 smaller subnets. For the smaller subnets, the first 24 bits are the network portion, the next six bits are the subnet portion, and the last two bits are the remaining host portion. Subnet 7:0 is 11000000.10101000.00010100.11100000 = 192.168.20.224/30. Subnet 7:1 is 11000000.10101000.00010100.11100100 = 192.168.20.228/30. Subnet 7:2 is 11000000.10101000.00010100.11101000 = 192.168.20.232/30. Subnet 7:3 is 11000000.10101000.00010100.11101100 = 192.168.20.236/30. Subnet 7:4 is 11000000.10101000.00010100.11110000 = 192.168.20.240/30. Subnet 7:5 is 11000000.10101000.00010100.11110100 = 192.168.20.244/30. Subnet 7:6 is 11000000.10101000.00010100.11111000 = 192.168.20.248/30. Subnet 7:7 is 11000000.10101000.00010100.11111100 = 192.168.20.252/30. Subnets 7:0, 7:1, and 7:2 are assigned to the WANs and the remaining subnets are unused/available.

VLSM Subnetting Scheme

7 11000000.10101000.00010100 .111 00000 192.168.20.224/277:0 11000000.10101000.00010100 .111000 00 192.168.20.224/30 7:1 11000000.10101000.00010100 .111001 00 192.168.20.228/30 7:2 11000000.10101000.00010100 .111010 00 192.168.20.232/30 7:3 11000000.10101000.00010100 .111011 00 192.168.20.236/30 7:4 11000000.10101000.00010100 .111100 00 192.168.20.240/30 7:5 11000000.10101000.00010100 .111101 00 192.168.20.244/30 7:6 11000000.10101000.00010100 .111110 00 192.168.20.248/30 7:7 11000000.10101000.00010100 .111111 00 192.168.20.252/30

Network portionHost portionDotted DecimalWANsUnused / AvailableSubnetting a subnet3 more bits borrowed from subnet ?

Why /30? Recall that when the number of needed host addresses is known, the formula 2n-2 (where n equals the number of host bits remaining) can be used. To provide two usable addresses, two host bits must be left in the host portion.

Because there are five host bits in the subnetted 192.168.20.224/27 address space, three more bits can be borrowed, leaving two bits in the host portion. The calculations at this point are exactly the same as those used for traditional subnetting. The bits are borrowed, and the subnet ranges are determined. The figure shows how the four /27 subnets have been assigned to the LANs and three of the /30 subnets have been assigned to the inter-router links.

The diagram shows the VLSM subnet assignments for a network topology consisting of four LANs and three WANs. There are four routers, each with an attached LAN and host addressing requirements, and three router-to-router connections requiring 2 hosts each. The R1 router Building A LAN has 25 hosts and is assigned subnet 192.168.20.0/27. The R2 router Building B LAN has 20 hosts and is assigned the subnet 192.168.20.32/27. The R3 router Building C LAN has 15 hosts and is assigned subnet 192.168.20.64/27. The R4 router Building D LAN has 28 hosts and is assigned subnet 192.168.20.96/27. The R1 to R2 connection is assigned subnet 192.168.20.224/30. The R2 to R3 connection is assigned subnet 192.168.20.228/30. The R3 to R4 connection is assigned subnet 192.168.20.232/30.

192.168.20.0/27192.168.20.32/27192.168.20.64/27192.168.20.96/27192.168.20.224/30192.168.20.228/30192.168.20.232/30R1R2R3R4

Building ABuilding BBuilding CBuilding D25 hosts20 hosts15 hosts28 hosts2 hosts2 hosts2 hosts

This VLSM subnetting scheme reduces the number of addresses per subnet to a size appropriate for the networks that require fewer subnets. Subnetting subnet 7 for inter-router links, allows subnets 4, 5, and 6 to be available for future networks, as well as five additional subnets available for inter-router connections.

Note: When using VLSM, always begin by satisfying the host requirements of the largest subnet. Continue subnetting until the host requirements of the smallest subnet are satisfied.

11.8.5

VLSM Topology Address Assignment

Using the VLSM subnets, the LAN and inter-router networks can be addressed without unnecessary waste.

The figure shows the network address assignments and the IPv4 addresses assigned to each router interface.

The diagram shows the VLSM subnet assignments and interface IP addressing for a network topology consisting of four LANs and three WANs. There are four routers, each with an attached LAN and host addressing requirements, and three router-to-router connections requiring 2 hosts each. The R1 router Building A LAN is connected to the G0/0/0 interface of R1 at 192.168.20.1/27, has 25 hosts, and is assigned subnet 192.168.20.0/27. The R2 router Building B LAN is connected to the G0/0/0 interface of R2 at 192.168.20.33/27, has 20 hosts, and is assigned the subnet 192.168.20.32/27. The R3 router Building C LAN is connected to the G0/0/0 interface of R3 at 192.168.20.65/27, has 15 hosts, and is assigned subnet 192.168.20.64/27. The R4 router Building D LAN is connected to the G0/0/0 interface of R4 at 192.168.20.97/27, has 28 hosts, and is assigned subnet 192.168.20.96/27. The R1 to R2 connection, assigned subnet 192.168.20.224/30, connects the G0/0/1 interface of R1 with address .225 to the G0/0/1 interface of R2 with address .226. The R2 to R3 connection, assigned subnet 192.168.20.228/30, connects the G0/1/0 interface of R2 with address .229 to the G0/0/1 interface of R3 with address .230. The R3 to R4 connection, assigned subnet 192.168.20.232/30, connects the G0/1/0 interface of R2 with address .233 to the G0/0/1 interface of R4 with address .234.

G0/0/1G0/0/1G0/1/0G0/0/1G0/0/1G0/1/0.225.234.233.230
.229.226G0/0/0G0/0/0G0/0/0G0/0/0192.168.20.224/30192.168.20.228/30192.168.20.232/30R1-R2R2-R3R3-R4192.168.20.1/27192.168.20.33/27192.168.20.65/27192.168.20.97/27192.168.20.0/27192.168.20.32/27192.168.20.64/27192.168.20.96/27R1R2R3R4

Building ABuilding BBuilding CBuilding D25 hosts20 hosts15 hosts28 hosts2 hosts2 hosts2 hosts

Using a common addressing scheme, the first host IPv4 address for each subnet is assigned to the LAN interface of the router. Hosts on each subnet will have a host IPv4 address from the range of host addresses for that subnet and an appropriate mask. Hosts will use the address of the attached router LAN interface as the default gateway address.

The table shows the network addresses and range of host addresses for each network. The default gateway address is displayed for the four LANs.

Network AddressRange of Host AddressesDefault Gateway AddressBuilding A192.168.20.0/27192.168.20.1/27 to 192.168.20.30/27192.168.20.1/27Building B192.168.20.32/27192.168.20.33/27 to 192.168.20.62/27192.168.20.33/27Building C192.168.20.64/27192.168.20.65/27 to 192.168.20.94/27192.168.20.65/27Building D192.168.20.96/27192.168.20.97/27 to 192.168.20.126/27192.168.20.97/27R1-R2192.168.20.224/30192.168.20.225/30 to 192.168.20.226/30R2-R3192.168.20.228/30192.168.20.229/30 to 192.168.20.230/30R3-R4192.168.20.232/30192.168.20.233/30 to 192.168.20.234/30

	Network Address	Range of Host Addresses	Default Gateway Address
Building A	192.168.20.0/27	192.168.20.1/27 to 192.168.20.30/27	192.168.20.1/27
Building B	192.168.20.32/27	192.168.20.33/27 to 192.168.20.62/27	192.168.20.33/27
Building C	192.168.20.64/27	192.168.20.65/27 to 192.168.20.94/27	192.168.20.65/27
Building D	192.168.20.96/27	192.168.20.97/27 to 192.168.20.126/27	192.168.20.97/27
R1-R2	192.168.20.224/30	192.168.20.225/30 to 192.168.20.226/30	
R2-R3	192.168.20.228/30	192.168.20.229/30 to 192.168.20.230/30	
R3-R4	192.168.20.232/30	192.168.20.233/30 to 192.168.20.234/30	

Video - VLSM Basics

As mentioned in the previous topic, public and private addresses affect the way you would subnet your network. There are also other issues that affect subnetting schemes. A standard /16 subnetting scheme creates subnets that each have the same number of hosts. Not every subnet you create will need this many hosts, leaving many IPv4 addresses unused. Perhaps you will need one subnet that contains many more hosts. This is why the variable-length subnet mask (VLSM) was developed.

Click Play to view a demonstration of basic VLSM techniques.

Play Video

11.8.2

Video - VLSM Example

Click Play to view a demonstration of VLSM subnetting.

Play Video

11.8.3

IPv4 Address Conservation

Because of the depletion of public IPv4 address space, making the most out of the available host addresses is a primary concern when subnetting IPv4 networks.

Note: The larger IPv6 address allows for much easier address planning and allocation than IPv4 allows. Conserving IPv6 addresses is not an issue. This is one of the driving forces for transitioning to IPv6.

Using traditional subnetting, the same number of addresses is allocated for each subnet. If all the subnets have the same requirements for the number of hosts, or if conserving IPv4 address space is not an issue, these fixed-size address blocks would be efficient. Typically, with public IPv4 addresses, that is not the case.

For example, the topology shown in the figure requires seven subnets, one for each of the four LANs, and one for each of the three connections between the routers.

The diagram shows a network topology consisting of seven subnets. There are four routers, each with an attached LAN and host addressing requirements, and three router-to-router connections requiring 2 hosts each. The R1 router LAN is Building A with 25 hosts; the R2 router LAN is Building B with 20 hosts; the R3 router LAN is Building C with 15 hosts; and the R4 router LAN is Building D with 28 hosts.

R1R2R3R4

Building ABuilding BBuilding CBuilding D25 hosts20 hosts15 hosts28 hosts2 hosts2 hosts2 hosts

Using traditional subnetting with the given address of 192.168.20.0/24, three bits can be borrowed from the host portion in the last octet to meet the subnet requirement of seven subnets. As shown in the figure, borrowing 3 bits creates 8 subnets and leaves 5 host bits with 30 usable hosts per subnet. This scheme creates the needed subnets and meets the host requirement of the largest LAN.

The diagram shows the basic subnet scheme for a given address of 192.168.20.0/24 with three bits borrowed for subnetting. Having 3 bits for subnetting results in 2 to the power of 3 = 8 subnets. Having 5 bits for hosts results in 2 to the power of 5 - 2 = 30 host IP addresses per subnet. All four octets are shown in binary followed by the dotted decimal format for the given address and for all the subnets created. The given network address in binary is 11000000.10101000.00010100 (network portion highlighted in gray) .00000000 (host portion highlighted in purple) = 192.168.20.0/24. For the subnets listed below, the first 24 bits are highlighted in gray (network portion), the next three bits are highlighted in blue (subnet portion), and the last five bits are the remaining host bits highlighted in purple. Subnet 0 is 11000000.10101000.00010100.00000000 = 192.168.20.0/27. Subnet 1 is 11000000.10101000.00010100.00100000 = 192.168.20.32/27. Subnet 2 is 11000000.10101000.00010100.01000000 = 192.168.20.64/27. Subnet 3 is 11000000.10101000.00010100.01100000 = 192.168.20.96/27. Subnets 0 - 3 are assigned to building LANs A, B, C, and D. Subnet 4 is 11000000.10101000.00010100.10000000 = 192.168.20.128/27. Subnet 5 is 11000000.10101000.00010100.10100000 = 192.168.20.160/27. Subnet 6 is 11000000.10101000.00010100.11000000 = 192.168.20.192/27. Subnets 4, 5, and 6 are assigned to the site to site WANs. Subnet 7 is 11000000.10101000.00010100.11100000 = 192.168.20.224/27. Subnet 7 is unused/available.

Basic Subnet Scheme

11000000.10101000.00010100.00000000192.168.20.0/246.1100000011000000.10101000.00010100192.168.20.192/275.1010000011000000.10101000.00010100192.168.20.160/274.1000000011000000.10101000.00010100192.168.20.128/273.0110000011000000.10101000.00010100192.168.20.96/272.0100000011000000.10101000.00010100192.168.20.64/271.0010000011000000.10101000.00010100192.168.20.32/270.0000000011000000.10101000.00010100192.168.20.0/277.1110000011000000.10101000.00010100192.168.20.224/27

Building LANs A, B, C, and DSite to Site WANsUnused / AvailableNetwork PortionHost Portion Subnet portion

2^3 = 8 subnetsHost portion

2^5 – 2 = 30 host IP addresses per subnet

These seven subnets could be assigned to the LAN and WAN networks, as shown in the figure.

The diagram shows the subnet assignments for a network topology consisting of seven subnets. There are four routers, each with an attached LAN and host addressing requirements, and three router-to-router connections requiring 2 hosts each. The R1 router Building A LAN has 25 hosts and is assigned subnet 192.168.20.0/27. The R2 router Building B LAN has 20 hosts and is assigned the subnet 192.168.20.32/27. The R3 router Building C LAN has 15 hosts and is assigned subnet 192.168.20.64/27. The R4 router Building D LAN has 28 hosts and is assigned subnet 192.168.20.96/27. The R1 to R2 connection is assigned subnet 192.168.20.128/27. The R2 to R3 connection is assigned subnet 192.168.20.160/27. The R3 to R4 connection is assigned subnet 192.168.20.192/27.

R1R2R3R4192.168.20.128/27192.168.20.160/27192.168.20.192/27

Building A
192.168.20.0/27Building B
192.168.20.32/27Building C
192.168.20.64/27Building D
192.168.20.96/2725 hosts20 hosts15 hosts28 hosts2 hosts2 hosts2 hosts

Although this traditional subnetting meets the needs of the largest LAN and divides the address space into an adequate number of subnets, it results in significant waste of unused addresses.

For example, only two addresses are needed in each subnet for the three WAN links. Because each subnet has 30 usable addresses, there are 28 unused addresses in each of these subnets. As shown in the figure, this results in 84 unused addresses (28x3).

The graphic shows the unused addresses of four WAN subnets using a /27 subnet mask. All four octets are shown in binary followed by the dotted decimal format for the subnet. The first 24 bits are highlighted in gray (network portion), the next three bits are highlighted in blue, and the last five bits are the remaining host bits highlighted in purple. Subnet 4 is 11000000.10101000.00010100.10000000 = 192.168.20.128/27. Subnet 5 is 11000000.10101000.00010100.10100000 = 192.168.20.160/27. Subnet 6 is 11000000.10101000.00010100.11000000 = 192.168.20.192/27. Having 5 bits for hosts results in 2 to the power of 5 - 2 = 30 host IP addresses per subnet. 30 - 2 = 28; each WAN subnet wastes 28 addresses. 28 x 3 = 84; 84 addresses are unused.

Unused Addresses on WAN Subnets

5.1010000011000000.10101000.00010100192.168.20.160/276.1100000011000000.10101000.00010100192.168.20.192/2711000000.10101000.00010100.100000004192.168.20.128/27

Network PortionHost Portion Host portion
2^5 – 2 = 30 host IP addresses per subnet

30 – 2 = 28
Each WAN subnet wastes 28 addresses

28 x 3 = 84
84 addresses are unused Dotted Decimal

Further, this limits future growth by reducing the total number of subnets available. This inefficient use of addresses is characteristic of traditional subnetting. Applying a traditional subnetting scheme to this scenario is not very efficient and is wasteful.

The variable-length subnet mask (VLSM) was developed to avoid wasting addresses by enabling us to subnet a subnet.

11.8.4

VLSM

In all of the previous subnetting examples, the same subnet mask was applied for all the subnets. This means that each subnet has the same number of available host addresses. As illustrated in the left side of the figure, traditional subnetting creates subnets of equal size. Each subnet in a traditional scheme uses the same subnet mask. As shown in the right side of the figure, VLSM allows a network space to be divided into unequal parts. With VLSM, the subnet mask will vary depending on how many bits have been borrowed for a particular subnet, thus the “variable” part of the VLSM.

The graphic shows two pie charts that compare traditional subnetting to VLSM. On the right is a pie chart titled: traditional subnetting creates equal sized subnets. The pie is divided into 8 equal sized slices, each with 30 hosts. On the left is a pie chart titled: subnets of varying sizes. This pie has 7 slices identical to the first pie. The 8th slice has been further divided into 8 smaller slices. Text pointing to 8th slice reads: one subnet was further divided using a /30 subnet mask to create 8 smaller subnets of 2 hosts each.

One subnet was further divided using a /30 subnet mask to create 8 smaller subnets of 2 hosts each. 30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 Hosts30 HostsTraditional Subnetting Creates Equal Sized SubnetsSubnets of Varying Sizes

VLSM is just subnetting a subnet. The same topology used previously is shown in the figure. Again, we will use the 192.168.20.0/24 network and subnet it for seven subnets, one for each of the four LANs, and one for each of the three connections between the routers.

The diagram shows a network topology consisting of seven subnets. There are four routers, each with an attached LAN and host addressing requirements, and three router-to-router connections requiring 2 hosts each. The R1 router LAN is Building A with 25 hosts; the R2 router LAN is Building B with 20 hosts; the R3 router LAN is Building C with 15 hosts; and the R4 router LAN is Building D with 28 hosts.

R1R2R3R4

Building ABuilding BBuilding CBuilding D25 hosts20 hosts15 hosts28 hosts2 hosts2 hosts2 hosts

The figure shows how network 192.168.20.0/24 subnetted into eight equal-sized subnets with 30 usable host addresses per subnet. Four subnets are used for the LANs and three subnets could be used for the connections between the routers.

The diagram shows the basic subnet scheme for a given address of 192.168.20.0/24 with three bits borrowed for subnetting. All four octets are shown in binary followed by the dotted decimal format for the given address and for all the subnets created. The given network address in binary is 11000000.10101000.00010100 (network portion highlighted in gray) .00000000 (host portion highlighted in purple) = 192.168.20.0/24. For the subnets listed below, the first 24 bits are highlighted in gray (network portion), the next three bits are highlighted in blue (subnet portion), and the last five bits are the remaining host bits highlighted in purple. Subnet 0 is 11000000.10101000.00010100.00000000 = 192.168.20.0/27. Subnet 1 is 11000000.10101000.00010100.00100000 = 192.168.20.32/27. Subnet 2 is 11000000.10101000.00010100.01000000 = 192.168.20.64/27. Subnet 3 is 11000000.10101000.00010100.01100000 = 192.168.20.96/27. Subnets 0 - 3 are assigned to building LANs A, B, C, and D. Subnet 4 is 11000000.10101000.00010100.10000000 = 192.168.20.128/27. Subnet 5 is 11000000.10101000.00010100.10100000 = 192.168.20.160/27. Subnet 6 is 11000000.10101000.00010100.11000000 = 192.168.20.192/27. Subnets 4, 5, and 6 are unused/available. Subnet 7 is 11000000.10101000.00010100.11100000 = 192.168.20.224/27. Subnet 7 will be subnetted further.

Basic Subnetting Scheme

11000000.10101000.00010100 .00000000 192.168.20.0/24 0 11000000.10101000.00010100 .000 00000 192.168.20.0/27 1 11000000.10101000.00010100 .001 00000 192.168.20.32/27 2 11000000.10101000.00010100 .010 00000 192.168.20.64/27 3 11000000.10101000.00010100 .011 00000 192.168.20.96/27 4 11000000.10101000.00010100 .100 00000 192.168.20.128/27 5 11000000.10101000.00010100 .101 00000 192.168.20.160/27 6 11000000.10101000.00010100 .110 00000 192.168.20.192/27 7 11000000.10101000.00010100 .111 00000 192.168.20.224/27

Network portionHost portionDotted DecimalLAN's
A, B, C, DUnused /
Available Subnet 7 will be subnetted further.

However, the connections between the routers require only two host addresses per subnet (one host address for each router interface). Currently all subnets have 30 usable host addresses per subnet. To avoid wasting 28 addresses per subnet, VLSM can be used to create smaller subnets for the inter-router connections.

To create smaller subnets for the inter-router links, one of the subnets will be divided. In this example, the last subnet, 192.168.20.224/27, will be further subnetted. The figure shows the last subnet has been subnetted further by using the subnet mask 255.255.255.252 or /30.

The diagram show the VLSM subnetting scheme when the subnet 192.168.20.224/27 is further subnetted by borrowing 3 more bits. For the original subnet, the first 24 bits represent the network portion and are 11000000.10101000.00010100. The next three bits represent the subnet portion and are 111. The last five bits represent the host portion and are 00000. The address in dotted decimal is 192.168.20.224/27. Borrowing 3 additional bits, subnetting a subnet, results in dividing the original subnet into 8 smaller subnets. For the smaller subnets, the first 24 bits are the network portion, the next six bits are the subnet portion, and the last two bits are the remaining host portion. Subnet 7:0 is 11000000.10101000.00010100.11100000 = 192.168.20.224/30. Subnet 7:1 is 11000000.10101000.00010100.11100100 = 192.168.20.228/30. Subnet 7:2 is 11000000.10101000.00010100.11101000 = 192.168.20.232/30. Subnet 7:3 is 11000000.10101000.00010100.11101100 = 192.168.20.236/30. Subnet 7:4 is 11000000.10101000.00010100.11110000 = 192.168.20.240/30. Subnet 7:5 is 11000000.10101000.00010100.11110100 = 192.168.20.244/30. Subnet 7:6 is 11000000.10101000.00010100.11111000 = 192.168.20.248/30. Subnet 7:7 is

11000000.10101000.00010100.11111100 = 192.168.20.252/30. Subnets 7:0, 7:1, and 7:2 are assigned to the WANs and the remaining subnets are unused/available.

VLSM Subnetting Scheme

7 11000000.10101000.00010100 .111 00000 192.168.20.224/277:0 11000000.10101000.00010100 .111000 00 192.168.20.224/30 7:1 11000000.10101000.00010100 .111001 00 192.168.20.228/30 7:2 11000000.10101000.00010100 .111010 00 192.168.20.232/30 7:3 11000000.10101000.00010100 .111011 00 192.168.20.236/30 7:4 11000000.10101000.00010100 .111100 00 192.168.20.240/30 7:5 11000000.10101000.00010100 .111101 00 192.168.20.244/30 7:6 11000000.10101000.00010100 .111110 00 192.168.20.248/30 7:7 11000000.10101000.00010100 .111111 00 192.168.20.252/30

Network portionHost portionDotted DecimalWANsUnused / AvailableSubnetting a subnet3 more bits borrowed from subnet ?

Why /30? Recall that when the number of needed host addresses is known, the formula 2n-2 (where n equals the number of host bits remaining) can be used. To provide two usable addresses, two host bits must be left in the host portion.

Because there are five host bits in the subnetted 192.168.20.224/27 address space, three more bits can be borrowed, leaving two bits in the host portion. The calculations at this point are exactly the same as those used for traditional subnetting. The bits are borrowed, and the subnet ranges are determined. The figure shows how the four /27 subnets have been assigned to the LANs and three of the /30 subnets have been assigned to the inter-router links.

The diagram shows the VLSM subnet assignments for a network topology consisting of four LANs and three WANs. There are four routers, each with an attached LAN and host addressing requirements, and three router-to-router connections requiring 2 hosts each. The R1 router Building A LAN has 25 hosts and is assigned subnet 192.168.20.0/27. The R2 router Building B LAN has 20 hosts and is assigned the subnet 192.168.20.32/27. The R3 router Building C LAN has 15 hosts and is assigned subnet 192.168.20.64/27. The R4 router Building D LAN has 28 hosts and is assigned subnet 192.168.20.96/27. The R1 to R2 connection is assigned subnet 192.168.20.224/30. The R2 to R3 connection is assigned subnet 192.168.20.228/30. The R3 to R4 connection is assigned subnet 192.168.20.232/30.

192.168.20.0/27192.168.20.32/27192.168.20.64/27192.168.20.96/27192.168.20.224/30192.168.20.228/30192.168.20.232/30R1R2R3R4

Building ABuilding BBuilding CBuilding D25 hosts20 hosts15 hosts28 hosts2 hosts2 hosts2 hosts

This VLSM subnetting scheme reduces the number of addresses per subnet to a size appropriate for the networks that require fewer subnets. Subnetting subnet 7 for inter-router links, allows subnets 4, 5, and 6 to be available for future networks, as well as five additional subnets available for inter-router connections.

Note: When using VLSM, always begin by satisfying the host requirements of the largest subnet. Continue subnetting until the host requirements of the smallest subnet are satisfied.

11.8.5

VLSM Topology Address Assignment

Using the VLSM subnets, the LAN and inter-router networks can be addressed without unnecessary waste.

The figure shows the network address assignments and the IPv4 addresses assigned to each router interface.

The diagram shows the VLSM subnet assignments and interface IP addressing for a network topology consisting of four LANs and three WANs. There are four routers, each with an attached LAN and host addressing requirements, and three router-to-router connections requiring 2 hosts each. The R1 router Building A LAN is connected to the G0/0/0 interface of R1 at 192.168.20.1/27, has 25 hosts, and is assigned subnet 192.168.20.0/27. The R2 router Building B LAN is connected to the G0/0/0 interface of R2 at 192.168.20.33/27, has 20 hosts, and is assigned the subnet 192.168.20.32/27. The R3 router Building C LAN is connected to the G0/0/0 interface of R3 at 192.168.20.65/27, has 15 hosts, and is assigned subnet 192.168.20.64/27. The R4 router Building D LAN is connected to the G0/0/0 interface of R4 at 192.168.20.97/27, has 28 hosts, and is assigned subnet 192.168.20.96/27. The R1 to R2 connection, assigned subnet 192.168.20.224/30, connects the G0/0/1 interface of R1 with address .225 to the G0/0/1 interface of R2 with address .226. The R2 to R3 connection, assigned subnet 192.168.20.228/30, connects the G0/1/0 interface of R2 with address .229 to the G0/0/1 interface of R3 with address .230. The R3 to R4 connection, assigned

subnet 192.168.20.232/30, connects the G0/1/0 interface of R2 with address .233 to the G0/0/1 interface of R4 with address .234.

G0/0/1G0/0/1G0/1/0G0/0/1G0/0/1G0/1/0.225.234.233.230
.229.226G0/0/0G0/0/0G0/0/0G0/0/0192.168.20.224/30192.168.20.228/30192.168.20.232/30R1-R2R2-R3R3-R4192.168.20.1/27192.168.20.33/27192.168.20.65/27192.168.20.97/27192.168.20.0/27192.168.20.32/27192.168.20.64/27192.168.20.96/27R1R2R3R4

Building ABuilding BBuilding CBuilding D25 hosts20 hosts15 hosts28 hosts2 hosts2 hosts2 hosts

Using a common addressing scheme, the first host IPv4 address for each subnet is assigned to the LAN interface of the router. Hosts on each subnet will have a host IPv4 address from the range of host addresses for that subnet and an appropriate mask. Hosts will use the address of the attached router LAN interface as the default gateway address.

The table shows the network addresses and range of host addresses for each network. The default gateway address is displayed for the four LANs.

Network AddressRange of Host AddressesDefault Gateway AddressBuilding A192.168.20.0/27192.168.20.1/27 to 192.168.20.30/27192.168.20.1/27Building B192.168.20.32/27192.168.20.33/27 to 192.168.20.62/27192.168.20.33/27Building C192.168.20.64/27192.168.20.65/27 to 192.168.20.94/27192.168.20.65/27Building D192.168.20.96/27192.168.20.97/27 to 192.168.20.126/27192.168.20.97/27R1-R2192.168.20.224/30192.168.20.225/30 to 192.168.20.226/30R2-R3192.168.20.228/30192.168.20.229/30 to 192.168.20.230/30R3-R4192.168.20.232/30192.168.20.233/30 to 192.168.20.234/30

	Network Address	Range of Host Addresses	Default Gateway Address
Building A	192.168.20.0/27	192.168.20.1/27 to 192.168.20.30/27	192.168.20.1/27
Building B	192.168.20.32/27	192.168.20.33/27 to 192.168.20.62/27	192.168.20.33/27
Building C	192.168.20.64/27	192.168.20.65/27 to 192.168.20.94/27	192.168.20.65/27
Building D	192.168.20.96/27	192.168.20.97/27 to 192.168.20.126/27	192.168.20.97/27
R1-R2	192.168.20.224/30	192.168.20.225/30 to 192.168.20.226/30	
R2-R3	192.168.20.228/30	192.168.20.229/30 to 192.168.20.230/30	
R3-R4	192.168.20.232/30	192.168.20.233/30 to 192.168.20.234/30	

12.1.1

Need for IPv6

You already know that IPv4 is running out of addresses. That is why you need to learn about IPv6.

IPv6 is designed to be the successor to IPv4. IPv6 has a larger 128-bit address space, providing 340 undecillion (i.e., 340 followed by 36 zeroes) possible addresses. However, IPv6 is more than just larger addresses.

When the IETF began its development of a successor to IPv4, it used this opportunity to fix the limitations of IPv4 and include enhancements. One example is Internet Control Message Protocol version 6 (ICMPv6), which includes address resolution and address autoconfiguration not found in ICMP for IPv4 (ICMPv4).

The depletion of IPv4 address space has been the motivating factor for moving to IPv6. As Africa, Asia and other areas of the world become more connected to the internet, there are not enough IPv4 addresses to accommodate this growth. As shown in the figure, all five RIRs have run out of IPv4 addresses.

The graphic shows a global map of the five regional internet registries and there IPv4 exhaustion dates. ARINs IPv4 exhaustion date is July 2015, RIPE NCCs exhaustion data is September 2012, APNICs exhaustion date is June 2014, LACNICs exhaustion date is April 2011, and AfriNICs projected exhaustion date is 2020.

RIR IPv4 Exhaustion Dates

IPv4 exhaustion date

July 2015IPv4 exhaustion date

September 2012IPv4 exhaustion date

June 2014IPv4 exhaustion date

2020IPv4 exhaustion date

April 2011

IPv4 has a theoretical maximum of 4.3 billion addresses. Private addresses in combination with Network Address Translation (NAT) have been instrumental in slowing the depletion of IPv4 address space. However, NAT is problematic for many applications, creates latency, and has limitations that severely impede peer-to-peer communications.

With the ever-increasing number of mobile devices, mobile providers have been leading the way with the transition to IPv6. The top two mobile providers in the United States report that over 90% of their traffic is over IPv6.

Most top ISPs and content providers such as YouTube, Facebook, and NetFlix, have also made the transition. Many companies like Microsoft, Facebook, and LinkedIn are transitioning to IPv6-only internally. In 2018, broadband ISP Comcast reported a deployment of over 65% and British Sky Broadcasting over 86%.

Internet of Things

The internet of today is significantly different than the internet of past decades. The internet of today is more than email, web pages, and file transfers between computers. The evolving internet is becoming an Internet of Things (IoT). No longer will the only devices accessing the internet be computers, tablets, and smartphones. The sensor-equipped, internet-ready devices of tomorrow will include everything from automobiles and biomedical devices, to household appliances and natural ecosystems.

With an increasing internet population, a limited IPv4 address space, issues with NAT and the IoT, the time has come to begin the transition to IPv6.

12.1.2

IPv4 and IPv6 Coexistence

There is no specific date to move to IPv6. Both IPv4 and IPv6 will coexist in the near future and the transition will take several years. The IETF has created various protocols and tools to help network administrators migrate their networks to IPv6. The migration techniques can be divided into three categories:

Click each button for more information.
Dual Stack

Tunneling

Translation

Dual stack allows IPv4 and IPv6 to coexist on the same network segment. Dual stack devices run both IPv4 and IPv6 protocol stacks simultaneously. Known as native IPv6, this means the customer network has an IPv6 connection to their ISP and is able to access content found on the internet over IPv6.

Physical topology showing three dual stack PCs and a dual stack router

Dual Stack
IPv4 and IPv6Dual Stack
IPv4 and IPv6Dual Stack
IPv4 and IPv6Dual Stack
IPv4 and IPv6

Note: Tunneling and translation are for transitioning to native IPv6 and should only be used where needed. The goal should be native IPv6 communications from source to destination.

12.2.1

IPv6 Addressing Formats

The first step to learning about IPv6 in networks is to understand the way an IPv6 address is written and formatted. IPv6 addresses are much larger than IPv4 addresses, which is why we are unlikely to run out of them.

IPv6 addresses are 128 bits in length and written as a string of hexadecimal values. Every four bits is represented by a single hexadecimal digit; for a total of 32 hexadecimal values, as shown in the figure. IPv6 addresses are not case-sensitive and can be written in either lowercase or uppercase.

16-bit Segments or Hextets

X:X:X:X:X:X:X:X0000000000000000000000000000000ffffffffffffffffffffffffffffffff:::::::1111111111111111000000000000000000

4 hexadecimal digits = 16 binary digitstotototototototototototo

Preferred Format

The previous figure also shows that the preferred format for writing an IPv6 address is x:x:x:x:x:x:x:x, with each “x” consisting of four hexadecimal values. The term octet refers to the eight bits of an IPv4 address. In IPv6, a hextet is the unofficial term used to refer to a segment of 16 bits, or four hexadecimal values. Each “x” is a single hextet which is 16 bits or four hexadecimal digits.

Preferred format means that you write IPv6 address using all 32 hexadecimal digits. It does not necessarily mean that it is the ideal method for representing the IPv6 address. In this module, you will see two rules that help to reduce the number of digits needed to represent an IPv6 address.

These are examples of IPv6 addresses in the preferred format.

```
2001 : 0db8 : 0000 : 1111 : 0000 : 0000 : 0000: 0200
2001 : 0db8 : 0000 : 00a3 : abcd : 0000 : 0000: 1234
2001 : 0db8 : 000a : 0001 : c012 : 9aff : fe9a: 19ac
2001 : 0db8 : aaaa : 0001 : 0000 : 0000 : 0000: 0000
fe80 : 0000 : 0000 : 0000 : 0123 : 4567 : 89ab: cdef
fe80 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000: 0001
fe80 : 0000 : 0000 : 0000 : c012 : 9aff : fe9a: 19ac
fe80 : 0000 : 0000 : 0000 : 0123 : 4567 : 89ab: cdef
0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000: 0001
0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000: 0000
```

12.2.2

Rule 1 – Omit Leading Zeros

The first rule to help reduce the notation of IPv6 addresses is to omit any leading 0s (zeros) in any hextet. Here are four examples of ways to omit leading zeros:

- 01ab can be represented as 1ab
- 09f0 can be represented as 9f0
- 0a00 can be represented as a00
- 00ab can be represented as ab

This rule only applies to leading 0s, NOT to trailing 0s, otherwise the address would be ambiguous. For example, the hextet “abc” could be either “0abc” or “abc0”, but these do not represent the same value.

Omitting Leading 0s

TypeFormatPreferred2001 : 0db8 : 0000 : 1111 : 0000 : 0000 : 0000 : 0200No leading 0s2001 : db8 : 0 : 1111 : 0 : 0 : 0 : 200Prefe
0db8 : 0000 : 00a3 : ab00 : 0ab0 : 00ab : 1234No leading 0s2001 : db8 : 0 : a3 : ab00 : ab0 : ab : 1234Preferred2001 : 0db8 : 000
c012 : 90ff : fe90 : 0001No leading 0s2001 : db8 : a : 1 : c012 : 90ff : fe90 : 1Preferred2001 : 0db8 : aaaa : 0001 : 0000 : 0000 :
0000No leading 0s2001 : db8 : aaaa : 1 : 0 : 0 : 0 : 0Preferredfe80 : 0000 : 0000 : 0000 : 0123 : 4567 : 89ab : cdefNo leading 0sfe8
: 123 : 4567 : 89ab : cdefPreferredfe80 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0001No leading 0sfe80 : 0 : 0 : 0 : 0 : 0 :
1Preferred0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0001No leading 0s 0 : 0 : 0 : 0 : 0 : 0 : 0 : 1Preferred0000 : 0000 : 000
0000 : 0000 : 0000 : 0000No leading 0s 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0

Type	Format
Preferred	2001 : **0**db8 : **000**0 : 1111 : **000**0 : **000**0 : **000**0 : **0**200
No leading 0s	2001 : db8 : 0 : 1111 : 0 : 0 : 0 : 200
Preferred	2001 : **0**db8 : **000**0 : **00**a3 : ab00 : **0**ab0 : **00**ab : 1234
No leading 0s	2001 : db8 : 0 : a3 : ab00 : ab0 : ab : 1234
Preferred	2001 : **0**db8 : **000**a : **000**1 : c012 : 90ff : fe90 : **000**1
No leading 0s	2001 : db8 : a : 1 : c012 : 90ff : fe90 : 1
Preferred	2001 : **0**db8 : aaaa : **000**1 : **000**0 : **000**0 : **000**0 : **000**0
No leading 0s	2001 : db8 : aaaa : 1 : 0 : 0 : 0 : 0
Preferred	fe80 : **000**0 : **000**0 : **000**0 : **0**123 : 4567 : 89ab : cdef
No leading 0s	fe80 : 0 : 0 : 0 : 123 : 4567 : 89ab : cdef
Preferred	fe80 : **000**0 : **000**0 : **000**0 : **000**0 : **000**0 : **000**0 : **000**1
No leading 0s	fe80 : 0 : 0 : 0 : 0 : 0 : 0 : 1
Preferred	**000**0 : **000**0 : **000**0 : **000**0 : **000**0 : **000**0 : **000**0 : **000**1
No leading 0s	0 : 0 : 0 : 0 : 0 : 0 : 0 : 1
Preferred	**000**0 : **000**0 : **000**0 : **000**0 : **000**0 : **000**0 : **000**0 : **000**0
No leading 0s	0 : 0 : 0 : 0 : 0 : 0 : 0 : 0

12.2.3

Rule 2- Double Colon

The second rule to help reduce the notation of IPv6 addresses is that a double colon (::) can replace any single, contiguous string of one or more 16-bit hextets consisting of all zeros. For example, 2001:db8:cafe:1:0:0:0:1 (leading 0s omitted) could be represented as 2001:db8:cafe:1::1. The double colon (::) is used in place of the three all-0 hextets (0:0:0).

The double colon (::) can only be used once within an address, otherwise there would be more than one possible resulting address. When used with the omitting leading 0s technique, the notation of IPv6 address can often be greatly reduced. This is commonly known as the compressed format.

Here is an example of the incorrect use of the double colon: 2001:db8::abcd::1234.

The double colon is used twice in the example above. Here are the possible expansions of this incorrect compressed format address:

- 2001:db8::abcd:0000:0000:1234

- 2001:db8::abcd:0000:0000:0000:1234
- 2001:db8:0000:abcd::1234
- 2001:db8:0000:0000:abcd::1234

If an address has more than one contiguous string of all-0 hextets, best practice is to use the double colon (::) on the longest string. If the strings are equal, the first string should use the double colon (::).

Omitting Leading 0s and All 0 Segments

TypeFormatPreferred2001 : 0db8 : 0000 : 1111 : 0000 : 0000 : 0000 : 0200Compressed/spaces2001 : db8 : 0 : 1111 : : 200Compressed2001:db8:0:1111::200Preferred2001 : 0db8 : 0000 : 0000 : ab00 : 0000 : 0000 : 0000Compressed/spaces2001 : db8 : 0 : 0 : ab00 :: Compressed2001:db8:0:0:ab00::Preferred2001 : 0db8 : aaaa : 0001 : 0000 : 0000 : 0000 : 0000Compressed/spaces2001 : db8 : aaaa : 1 ::Compressed2001:db8:aaaa:1::Preferredfe80 : 0000 : 0000 : 0000 : 0123 : 4567 : 89ab : cdefCompressed/spacesfe80 : : 123 : 4567 : 89ab : cdefCompressedfe80::123:4567:89ab:cdefPreferredfe80 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0001Compressed/spaces fe80 : 1Compressedfe80::0Preferred0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0001Compressed/spaces:: 1Compressed::1Preferred0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000Compressed/spaces::Compressed::

Type	Format
Preferred	2001 : 0db8 : 0000 1111 : 0000 : 000 0000 : 0200
Compressed/spaces	2001 : db8 : 1111 : : 200
Compressed	2001:db8:0:1111::
Preferred	2001 : 0db8 : 000 0000 : ab00 : 000 0000 : 0000
Compressed/spaces	2001 : db8 : 0 : ab00 ::
Compressed	2001:db8:0:0:ab00
Preferred	2001 : 0db8 : aaa 0001 : 0000 : 000 0000 : 0000
Compressed/spaces	2001 : db8 : aaa 1 ::
Compressed	2001:db8:aaaa:1::
Preferred	fe80 : 0000 : 000 0000 : 0123 : 456 89ab : cdef
Compressed/spaces	fe80 : : 123 : 4567 : 8 cdef
Compressed	fe80::123:4567:89
Preferred	fe80 : 0000 : 000 0000 : 0000 : 000 0000 : 0001
Compressed/spaces	fe80 : : 1
Compressed	fe80::1
Preferred	0000 : 0000 : 000 0000 : 0000 : 000 0000 : 0001
Compressed/spaces	:: 1
Compressed	::1

TypeFormatPreferred2001 : 0db8 : 0000 : 1111 : 0000 : 0000 : 0000 : 0200Compressed/spaces2001 : db8 : 0 : 1111 : : 200Compressed2001:db8:0:1111::200Preferred2001 : 0db8 : 0000 : 0000 : ab00 : 0000 : 0000 : 0000Compressed/spaces2001 : db8 : 0 : 0 : ab00 :: Compressed2001:db8:0:0:ab00::Preferred2001 : 0db8 : aaaa : 0001 : 0000 : 0000 : 0000 : 0000Compressed/spaces2001 : db8 : aaaa : 1 : : 1 ::Compressed2001:db8:aaaa:1::Preferredfe80 : 0000 : 0000 : 0000 : 0123 : 4567 : 89ab : cdefCompressed/spacesfe80 : : 123 : 4567 : 89ab : cdefCompressedfe80::123:4567:89ab:cdefPreferredfe80 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0001Compressed/spaces fe80 : : 1Compressedfe80::0Preferred0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0001Compressed/spaces:: 1Compressed::1Preferred 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000Compressed/spaces::Compressed::

Type	Format
Preferred	0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000
Compressed/spaces	: :
Compressed	: :

12.3.1

Unicast, Multicast, Anycast

As with IPv4, there are different types of IPv6 addresses. In fact, there are three broad categories of IPv6 addresses:

- **Unicast** - An IPv6 unicast address uniquely identifies an interface on an IPv6-enabled device.
- **Multicast** - An IPv6 multicast address is used to send a single IPv6 packet to multiple destinations.
- **Anycast** - An IPv6 anycast address is any IPv6 unicast address that can be assigned to multiple devices. A packet sent to an anycast address is routed to the nearest device having that address. Anycast addresses are beyond the scope of this course.

Unlike IPv4, IPv6 does not have a broadcast address. However, there is an IPv6 all-nodes multicast address that essentially gives the same result.

12.3.2

IPv6 Prefix Length

The prefix, or network portion, of an IPv4 address can be identified by a dotted-decimal subnet mask or prefix length (slash notation). For example, an IPv4 address of 192.168.1.10 with dotted-decimal subnet mask 255.255.255.0 is equivalent to 192.168.1.10/24.

In IPv6 it is only called the prefix length. IPv6 does not use the dotted-decimal subnet mask notation. Like IPv4, the prefix length is represented in slash notation and is used to indicate the network portion of an IPv6 address.

The prefix length can range from 0 to 128. The recommended IPv6 prefix length for LANs and most other types of networks is /64, as shown in the figure.

The graphic shows an IPv6 address divided into a 64-bit prefix and a 64-bit interface ID. The 64-bit prefix is 2001:0db8:000a:0000. The 64-bit interface ID is 0000:0000:0000:0000.

IPv6 Prefix Length

64 bits64 bitsExample: 2001:db8:a::/64

The prefix or network portion of the address is 64 bits in length, leaving another 64 bits for the interface ID (host portion) of the address.

It is strongly recommended to use a 64-bit Interface ID for most networks. This is because stateless address autoconfiguration (SLAAC) uses 64 bits for the Interface ID. It also makes subnetting easier to create and manage.

12.3.3

Types of IPv6 Unicast Addresses

An IPv6 unicast address uniquely identifies an interface on an IPv6-enabled device. A packet sent to a unicast address is received by the interface which is assigned that address. Similar to IPv4, a source IPv6 address must be a unicast address. The destination IPv6 address can be either a unicast or a multicast address. The figure shows the different types of IPv6 unicast addresses.

The graphic shows a chart of six types of IPv6 unicast addresses. From top to bottom, the types of Ipv6 addresses in the chart are: Global Unicast, Link-local, Loopback ::1/128, Unspecified ::/128, Unique local fc00::/7 - fdff::/7, and Embedded IPv4.

IPv6 Unicast Addresses

::1/128::fc00::/7 - fdff::/7

Unlike IPv4 devices that have only a single address, IPv6 addresses typically have two unicast addresses:

- **Global Unicast Address (GUA)** - This is similar to a public IPv4 address. These are globally unique, internet-routable addresses. GUAs can be configured statically or assigned dynamically.
- **Link-local Address (LLA)** - This is required for every IPv6-enabled device. LLAs are used to communicate with other devices on the same local link. With IPv6, the term link refers to a subnet. LLAs are confined to a single link. Their uniqueness must only be confirmed on that link because they are not routable beyond the link. In other words, routers will not forward packets with a link-local source or destination address.

12.3.4

A Note About the Unique Local Address

Unique local addresses (range fc00::/7 to fdff::/7) are not yet commonly implemented. Therefore, this module only covers GUA and LLA configuration. However, unique local addresses may eventually be used to address devices that should not be accessible from the outside, such as internal servers and printers.

The IPv6 unique local addresses have some similarity to RFC 1918 private addresses for IPv4, but there are significant differences:

- Unique local addresses are used for local addressing within a site or between a limited number of sites.
- Unique local addresses can be used for devices that will never need to access another network.
- Unique local addresses are not globally routed or translated to a global IPv6 address.

Note: Many sites also use the private nature of RFC 1918 addresses to attempt to secure or hide their network from potential security risks. However, this was never the intended use of these technologies, and the IETF has always recommended that sites take the proper security precautions on their internet-facing router.

12.3.5

IPv6 GUA

IPv6 global unicast addresses (GUAs) are globally unique and routable on the IPv6 internet. These addresses are equivalent to public IPv4 addresses. The Internet Committee for Assigned Names and Numbers (ICANN), the operator for IANA, allocates IPv6 address blocks to the five RIRs. Currently, only GUAs with the first three bits of 001 or 2000::/3 are being assigned, as shown in the figure.

The figure shows the range of values for the first hextet where the first hexadecimal digit for currently available GUAs begins with a 2 or a 3. This is only 1/8th of the total available IPv6 address space, excluding only a very small portion for other types of unicast and multicast addresses.

Note: The 2001:db8::/32 address has been reserved for documentation purposes, including use in examples.

The graphic shows the three parts of a GUA: First is the global routing prefix, then the Subnet ID, then finally the Interface ID. The first three bits of the Global routing prefix are 001. The Range of the first hextext is shown to be from 0010 0000 0000 0000 (2000) to 0011 1111 1111 1111 (3fff)

0010 0000 0000 0000 (2000) 0011 1111 1111 1111 (3fff)001

Range of first hextet:to

The next figure shows the structure and range of a GUA.

The graphic shows the three parts of a GUA: First is the global routing prefix which is 48 bits in length, then the Subnet ID which is 16 bits in length, then finally the Interface ID which is 64 bits in length. Text under the graphic states A /48 routing prefix + 16 bit Subnet ID = /64 prefix.

IPv6 Address with a /48 Global Routing Prefix and /64 Prefix

48 bits16 bits64 bitsA /48 routing prefix + 16 bit Subnet ID = /64 prefix.64 bits

A GUA has three parts:

- Global Routing Prefix
- Subnet ID
- Interface ID

12.3.6

IPv6 GUA Structure

Global Routing Prefix

The global routing prefix is the prefix, or network, portion of the address that is assigned by the provider, such as an ISP, to a customer or site. For example, it is common for ISPs to assign a /48 global routing prefix to its customers. The global routing prefix will usually vary depending on the policies of the ISP.

The previous figure shows a GUA using a /48 global routing prefix. /48 prefixes are a common global routing prefix that is assigned and will be used in most of the examples throughout this course.

For example, the IPv6 address 2001:db8:acad::/48 has a global routing prefix that indicates that the first 48 bits (3 hextets) (2001:db8:acad) is how the ISP knows of this prefix (network). The double colon (::) following the /48 prefix length means the rest of the address contains all 0s. The size of the global routing prefix determines the size of the subnet ID.

Subnet ID

The Subnet ID field is the area between the Global Routing Prefix and the Interface ID. Unlike IPv4 where you must borrow bits from the host portion to create subnets, IPv6 was designed with subnetting in mind. The Subnet ID is used by an organization to identify subnets within its site. The larger the subnet ID, the more subnets available.

Note: Many organizations are receiving a /32 global routing prefix. Using the recommended /64 prefix in order to create a 64-bit Interface ID, leaves a 32 bit Subnet ID. This means an organization with a /32 global routing prefix and a 32-bit Subnet ID will have 4.3 billion subnets, each with 18 quintillion devices per subnet. That is as many subnets as there are public IPv4 addresses!

The IPv6 address in the previous figure has a /48 Global Routing Prefix, which is common among many enterprise networks. This makes it especially easy to examine the different parts of the address. Using a typical /64 prefix length, the first four hextets are for the network portion of the address, with the fourth hextet indicating the Subnet ID. The remaining four hextets are for the Interface ID.

Interface ID

The IPv6 interface ID is equivalent to the host portion of an IPv4 address. The term Interface ID is used because a single host may have multiple interfaces, each having one or more IPv6 addresses. The figure shows an example of the structure of an IPv6 GUA. It is strongly recommended that in most cases /64 subnets should be used, which creates a 64-bit interface ID. A 64-bit interface ID allows for 18 quintillion devices or hosts per subnet.

A /64 subnet or prefix (Global Routing Prefix + Subnet ID) leaves 64 bits for the interface ID. This is recommended to allow SLAAC-enabled devices to create their own 64-bit interface ID. It also makes developing an IPv6 addressing plan simple and effective.

Note: Unlike IPv4, in IPv6, the all-0s and all-1s host addresses can be assigned to a device. The all-1s address can be used because broadcast addresses are not used within IPv6. The all-0s address can also be used, but is reserved as a Subnet-Router anycast address, and should be assigned only to routers.

12.3.7

IPv6 LLA

An IPv6 link-local address (LLA) enables a device to communicate with other IPv6-enabled devices on the same link and only on that link (subnet). Packets with a source or destination LLA cannot be routed beyond the link from which the packet originated.

The GUA is not a requirement. However, every IPv6-enabled network interface must have an LLA.

If an LLA is not configured manually on an interface, the device will automatically create its own without communicating with a DHCP server. IPv6-enabled hosts create an IPv6 LLA even if the device has not been assigned a global unicast IPv6 address. This allows IPv6-enabled devices to communicate with other IPv6-enabled devices on the same subnet. This includes communication with the default gateway (router).

IPv6 LLAs are in the fe80::/10 range. The /10 indicates that the first 10 bits are 1111 1110 10xx xxxx. The first hextet has a range of 1111 1110 10**00 0000** (fe80) to 1111 1110 10**11 1111** (febf).

The figure shows an example of communication using IPv6 LLAs. The PC is able to communicate directly with the printer using the LLAs.

Physical topology showing two PCs, a server, a printer, a switch, and a router. It depicts that link-local communications are not routed outside the network.

IPv6 Link-Local Communications

fe80::bbbb/64fe80::cccc/64fe80::dddd/64fe80::1/64fe80::aaaafe80::ddddfe80::aaaa/64

IPv6 Packet

The next figure shows some of the uses for IPv6 LLAs.

The graphic shows two routers connect by a link with their LLA addresses. The number 1 with a bidirectional arrow pointing to each router is over the link along with the text Routing Protocol Messages. A PC is connected to the router on the left with a number 2 an arrow pointing from the PC to the router. Text under the graphic reads 1. Routers use the LLA of neighbor routers to send routing updates. 2. Hosts use the LLA of a local router as the default-gateway.

12

Routing Protocol
MessagesLLA
AddressLLA
AddressLLA
Address

1. Routers use the LLA of neighbor routers to send routing updates.
2. Hosts use the LLA of a local router as the default-gateway.

Note: Typically, it is the LLA of the router, and not the GUA, that is used as the default gateway for other devices on the link.

There are two ways that a device can obtain an LLA:

- **Statically** - This means the device has been manually configured.
- **Dynamically** - This means the device creates its own interface ID by using randomly generated values or using the Extended Unique Identifier (EUI) method, which uses the client MAC address along with additional bits.

12.4.1

Static GUA Configuration on a Router

As you learned in the previous topic, IPv6 GUAs are the same as public IPv4 addresses. They are globally unique and routable on the IPv6 internet. An IPv6 LLA lets two IPv6-enabled devices communicate with each other on the same link (subnet). It is easy to statically configure IPv6 GUAs and LLAs on routers to help you create an IPv6 network. This topic teaches you how to do just that!

Most IPv6 configuration and verification commands in the Cisco IOS are similar to their IPv4 counterparts. In many cases, the only difference is the use of **ipv6** in place of **ip** within the commands.

For example, the Cisco IOS command to configure an IPv4 address on an interface is **ip address** *ip-address subnet-mask.* In contrast, the command to configure an IPv6 GUA on an interface is **ipv6 address** *ipv6-address/prefix-length.*

Notice that there is no space between *ipv6-address* and *prefix-length.*

The example configuration uses the topology shown in the figure and these IPv6 subnets:

- 2001:db8:acad:1::/64
- 2001:db8:acad:2::/64
- 2001:db8:acad:3::/64

The graphic shows two PCs, PC1 and PC2. PC1 is connected to a switch and has the IPv6 address 2001:db8:acad:1::10/64. PC2 is connected to a switch and has the IPv6 address 2001:db8:acad:2::10/64. The two switches are connected to a router, R1. PC1 is connected through the switch to R1s G0/0/0 interface which has IPv6 address 2001:db8:acad:1::1/64. PC2 is connected through the switch to R1s G0/0/1 interface which has IPv6 address 2001:db8:acad:2::1/64. R1 connects to the cloud through its S0/1/0 interface which has IPv6 address 2001:db8:acad:3::1/64.

Example Topology

R1::10::10S0/1/0::1PC1PC22001:db8.acad:2::/642001:db8.acad:1::/642001:db8.acad:3::/64G0/0/0::1G0/0/1::1R1

The example shows the commands required to configure the IPv6 GUA on GigabitEthernet 0/0/0, GigabitEthernet 0/0/1, and the Serial 0/1/0 interface of R1.

IPv6 GUA Configuration on Router R1

```
R1(config)# interface gigabitethernet 0/0/0
```

```
R1(config-if)# ipv6 address 2001:db8:acad:1::1/64
```

```
R1(config-if)# no shutdown
```

```
R1(config-if)# exit
```

```
R1(config)# interface gigabitethernet 0/0/1
```

```
R1(config-if)# ipv6 address 2001:db8:acad:2::1/64
```

```
R1(config-if)# no shutdown
```

```
R1(config-if)# exit
```

```
R1(config)# interface serial 0/1/0
```

```
R1(config-if)# ipv6 address 2001:db8:acad:3::1/64
```

```
R1(config-if)# no shutdown
```

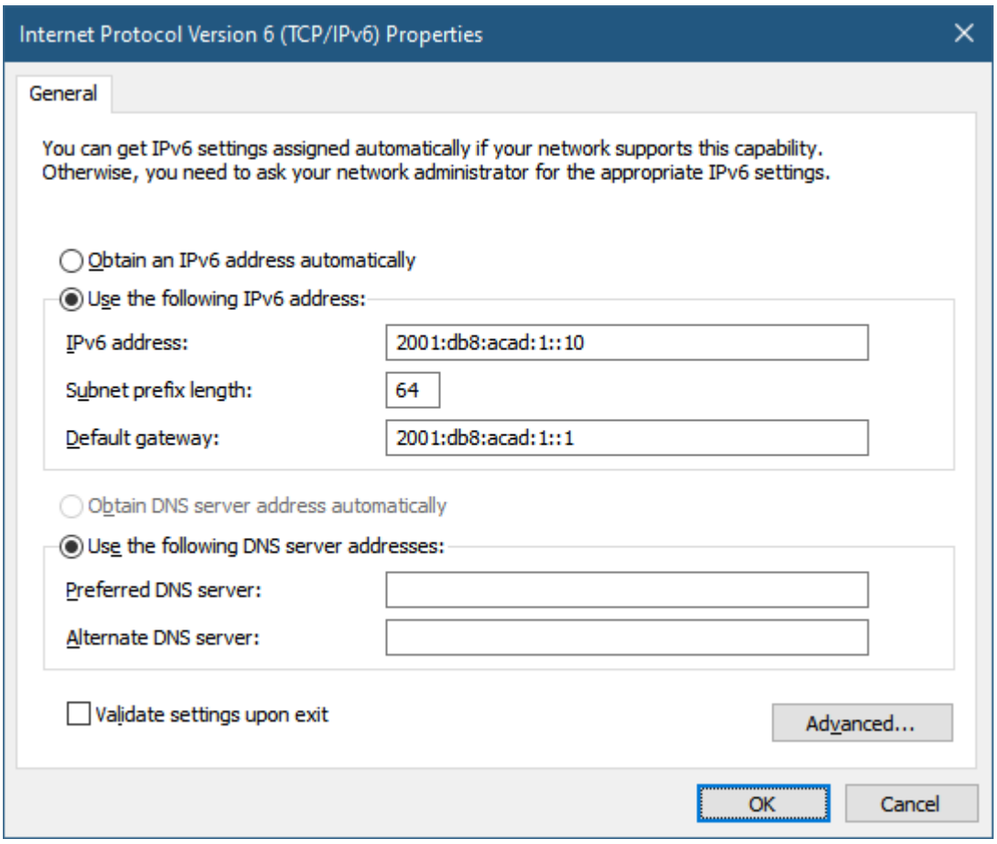
12.4.2

Static GUA Configuration on a Windows Host

Manually configuring the IPv6 address on a host is similar to configuring an IPv4 address.

As shown in the figure, the default gateway address configured for PC1 is 2001:db8:acad:1::1. This is the GUA of the R1 GigabitEthernet interface on the same network. Alternatively, the default gateway address can be configured to match the LLA of the GigabitEthernet interface. Using the LLA of the router as the default gateway address is considered best practice. Either configuration will work.

The graphic shows the Windows Internet Protocol Version 6 (TCP/IPv6) Prosperities window. The Use the following IPv6 address button is selected. The IPv6 address is 2001:db8:acad:1::1. The Subnet prefix length is 64. The Default gateway is 2001:db8:acad:1::1. The Use the following DNS server address button is selected.



Just as with IPv4, configuring static addresses on clients does not scale to larger environments. For this reason, most network administrators in an IPv6 network will enable dynamic assignment of IPv6 addresses.

There are two ways in which a device can obtain an IPv6 GUA automatically:

- Stateless Address Autoconfiguration (SLAAC)
- Stateful DHCPv6

SLAAC and DHCPv6 are covered in the next topic.

Note: When DHCPv6 or SLAAC is used, the LLA of the router will automatically be specified as the default gateway address.

12.4.3

Static Configuration of a Link-Local Unicast Address

Configuring the LLA manually lets you create an address that is recognizable and easier to remember. Typically, it is only necessary to create recognizable LLAs on routers. This is beneficial because router LLAs are used as default gateway addresses and in routing advertisement messages.

LLAs can be configured manually using the **ipv6 address** *ipv6-link-local-address* **link-local** command. When an address begins with this hextet within the range of fe80 to febf, the **link-local** parameter must follow the address.

The figure shows an example topology with LLAs on each interface.

The graphic shows two PCs, PC1 and PC2. PC1 is connected to a switch and has the IPv6 address 2001:db8:acad:1::10/64. PC2 is connected to a switch and has the IPv6 address 2001:db8:acad:2::10/64. The two switches are connected to a router, R1. PC1 is connected through the switch to R1s G0/0/0 interface which

has IPv6 address 2001:db8:acad:1::1/64 and the LLA address of fe80::1:1. PC2 is connected through the switch to R1s G0/0/1 interface which has IPv6 address 2001:db8:acad:2::1/64 and the LLA address of fe80::2:1. R1 connects to the cloud through its S0/1/0 interface which has IPv6 address 2001:db8:acad:3::1/64 and the LLA address of fe80::3:1.

Example Topology with LLAs

R1::10::10S0/1/0::1fe80::3:1PC12001:db8.acad:2::/642001:db8.acad:1::/642001:db8.acad:3::/64G0/0/0::1fe80::1:1G0/0/1::1fe80::2:1PC2R1

The example shows the configuration of an LLA on router R1.

```
R1(config)# interface gigabitethernet 0/0/0
```

```
R1(config-if)# ipv6 address fe80::1:1 link-local
```

```
R1(config-if)# exit
```

```
R1(config)# interface gigabitethernet 0/0/1
```

```
R1(config-if)# ipv6 address fe80::2:1 link-local
```

```
R1(config-if)# exit
```

```
R1(config)# interface serial 0/1/0
```

```
R1(config-if)# ipv6 address fe80::3:1 link-local
```

```
R1(config-if)# exit
```

Statically configured LLAs are used to make them more easily recognizable as belonging to router R1. In this example, all the interfaces of router R1 have been configured with an LLA that begins with **fe80::*n*:1**.

Note: The exact same LLA could be configured on each link as long as it is unique on that link. This is because LLAs only have to be unique on that link. However, common practice is to create a different LLA on each interface of the router to make it easy to identify the router and the specific interface.

12.4.4

Syntax Checker - GUA and LLA Static Configuration

Assign IPv6 GUAs and LLAs to the specified interfaces on router R1.

The graphic shows two PCs, PC1 and PC2. PC1 is connected to a switch and has the IPv6 address 2001:db8:acad:1::10/64. PC2 is connected to a switch and has the IPv6 address 2001:db8:acad:2::10/64. The two switches are connected to a router, R1. PC1 is connected through the switch to R1s G0/0/0 interface which has IPv6 address 2001:db8:acad:1::1/64 and the LLA address of fe80::1:1. PC2 is connected through the switch to R1s G0/0/1 interface which has IPv6 address 2001:db8:acad:2::1/64 and the LLA address of fe80::2:1. R1 connects to the cloud through its S0/1/0 interface which has IPv6 adress 2001:db8:acad:3::1/64 and the LLA address of fe80::3:1.

R1::10::10S0/1/0::1fe80::3:1PC12001:db8.acad:2::/642001:db8.acad:1::/642001:db8.acad:3::/64G0/0/0::1fe80::1:1G0/0/1::1fe80::2:1PC2R1

Configure and activate IPv6 on the GigabitEthernet 0/0/0 interface with the following addresses:

- Use **g0/0/0** as the interface name
- LLA - fe80::1:1
- GUA - 2001:db8:acad:1::1/64

- Activate the interface
- Exit interface configuration mode

```
R1(config)#
```

12.5.1

RS and RA Messages

If you do not want to statically configure IPv6 GUAs, no need to worry. Most devices obtain their IPv6 GUAs dynamically. This topic explains how this process works using Router Advertisement (RA) and Router Solicitation (RS) messages. This topic gets rather technical, but when you understand the difference between the three methods that a router advertisement can use, as well as how the EUI-64 process for creating an interface ID differs from a randomly generated process, you will have made a huge leap in your IPv6 expertise!

For the GUA, a device obtains the address dynamically through Internet Control Message Protocol version 6 (ICMPv6) messages. IPv6 routers periodically send out ICMPv6 RA messages, every 200 seconds, to all IPv6-enabled devices on the network. An RA message will also be sent in response to a host sending an ICMPv6 RS message, which is a request for an RA message. Both messages are shown in the figure.

The graphic shows LAN with a host sending an RS message towards are router and the router sending an RA message in return towards the PC. Also on the LAN is a DHCPv6 Server. Text under the gaphic reads 1. RS messages are sent to all IPv6 rotues by hosts requesting addressing information. 2. RA messages are sent to all IPv6 nodes. If Method 1 (SLAAC only) is used, the RA includes the prefix, refix-lenght and default-gateway information.

ICMPv6 RS and RA Messages

12

RA MessagesRS Messages**DHCPv6 Server**

1. RS messages are sent to all IPv6 routers by hosts requesting addressing information.
2. RA messages are sent to all IPv6 nodes. If Method 1 (SLAAC only) is used, the RA includes network prefix, prefix-length, and default-gateway information.

RA messages are on IPv6 router Ethernet interfaces. The router must be enabled for IPv6 routing, which is not enabled by default. To enable a router as an IPv6 router, the **ipv6 unicast-routing** global configuration command must be used.

The ICMPv6 RA message is a suggestion to a device on how to obtain an IPv6 GUA. The ultimate decision is up to the device operating system. The ICMPv6 RA message includes the following:

- **Network prefix and prefix length** - This tells the device which network it belongs to.
- **Default gateway address** - This is an IPv6 LLA, the source IPv6 address of the RA message.
- **DNS addresses and domain name** - These are the addresses of DNS servers and a domain name.

There are three methods for RA messages:

- **Method 1: SLAAC** - “I have everything you need including the prefix, prefix length, and default gateway address.”
- **Method 2: SLAAC with a stateless DHCPv6 server** - “Here is my information but you need to get other information such as DNS addresses from a stateless DHCPv6 server.”
- **Method 3: Stateful DHCPv6 (no SLAAC)** - “I can give you your default gateway address. You need to ask a stateful DHCPv6 server for all your other information.”

12.5.2

Method 1: SLAAC

SLAAC is a method that allows a device to create its own GUA without the services of DHCPv6. Using SLAAC, devices rely on the ICMPv6 RA messages of the local router to obtain the necessary information.

By default, the RA message suggests that the receiving device use the information in the RA message to create its own IPv6 GUA and all other necessary information. The services of a DHCPv6 server are not required.

SLAAC is stateless, which means there is no central server (for example, a stateful DHCPv6 server) allocating GUAs and keeping a list of devices and their addresses. With SLAAC, the client device uses the information in the RA message to create its own GUA. As shown in the figure, the two parts of the address are created as follows:

- **Prefix** - This is advertised in the RA message.
- **Interface ID** - This uses the EUI-64 process or by generating a random 64-bit number, depending on the device operating system.

The graphic shows LAN with a router sending an ICMPv6 Router Advertisement message towards a PC. The PC has the IPv6 address of 2001:db8:acad:1:fc99:47ff:fe75:cee0/64. The is a graphic indicating that the network prefix recieved in the RA message is 2001:db8:acad:1: and the Interface ID which was created by the client device EUI-64 or random 64-bit number is fc99:47ff:fe75:cee0. Text under the graphic reads 1. The router sends an RA message with the prefix for the local link. 2. The PC uses SLAAC to obtain a prefix from the RA message and creates its own Interface ID.

/642001:db8:acad:1::/642001:db8:acad:1:fc99:47ff:fe75:cee0/642001:db8:acad:1:fc99:47ff:fe75:cee012

ICMPv6 Router
AdvertisementFrom RA MessageCreated by client device
EUI-64 or random 64-bit number

1. The router sends an RA message with the prefix for the local link.
2. The PC uses SLAAC to obtain a prefix from the RA message and creates its own Interface ID.

12.5.3

Method 2: SLAAC and Stateless DHCPv6

A router interface can be configured to send a router advertisement using SLAAC and stateless DHCPv6.

As shown in the figure, with this method, the RA message suggests devices use the following:

- SLAAC to create its own IPv6 GUA
- The router LLA, which is the RA source IPv6 address, as the default gateway address
- A stateless DHCPv6 server to obtain other information such as a DNS server address and a domain name

Note: A stateless DHCPv6 server distributes DNS server addresses and domain names. It does not allocate GUAs.

The graphic shows LAN with a PC that is sending an RS Messages (labled #1) to the router. The router is sending in return an RA message (labled #2). The PC is also sending a DHCPv6 Solicit message (labled #3) to a stateless DHCPv6 server. Text under the graphic reads 1. The PC sends an RS to all IPv6 routers, I need addressing information. 2. The router sends an RA message to all IPv6 nodes with Method 2 (SLAAC and DHCPv6) specified. Here is your prefix prefix-length, and default gateway information. but you will need to get DNS information form a DHCPv6 server. 3. The PC sends a DHCPv6 solicit message to all DHCPv6 servers. I used SLAAC to create my IPv6 address and get my default gateway address but I need other information from a stateless DHCPv6 server.

123

RA MessageRS Message**Stateless DHCPv6 Server**DHCPv6 Solicit

1. The PC sends an RS to all IPv6 routers, “I need addressing information.”
2. The router sends an RA message to all IPv6 nodes with Method 2 (SLAAC and DHCPv6) specified. “Here is your prefix, prefix-length, and default gateway information. But you will need to get DNS information from a DHCPv6 server.”
3. The PC sends a DHCPv6 Solicit message to all DHCPv6 servers. "I used SLAAC to create my IPv6 address and get my default gateway address, but I need other information from a stateless DHCPv6 server."

12.5.4

Method 3: Stateful DHCPv6

A router interface can be configured to send an RA using stateful DHCPv6 only.

Stateful DHCPv6 is similar to DHCP for IPv4. A device can automatically receive its addressing information including a GUA, prefix length, and the addresses of DNS servers from a stateful DHCPv6 server.

As shown in the figure, with this method, the RA message suggests devices use the following:

- The router LLA, which is the RA source IPv6 address, for the default gateway address.
- A stateful DHCPv6 server to obtain a GUA, DNS server address, domain name and other necessary information.

The graphic shows LAN with a PC that is sending (labled #1) to a router. The router is sending a message (labled #2) to the PC. The PC is also sending a message (labled #3) to s server. Text under the graphic reads 1. The PC sends an RS to all IPv6 routers, I need addressing information. 2. The router sends an RA message to all IPv6 nodes with Method 3 (statefull DHCPv6) specified I am your default gateway, but you need to ask a statefull DHCPv6 server for your IPv6 addressign information. 3. The PC sends a DHCPv6 solicit message to all DHCPv6 servers, I received my default gateway address from the RA message, but I need an IPv6 address and all other addressing information from a stateful DHCPv6 server.

123

RA MessageRS Message**Stateful DHCPv6 Server**DHCPv6 Solicit

1. The PC sends an RS to all IPv6 routers, “I need addressing information.”
2. The router sends an RA message to all IPv6 nodes with Method 3 (Stateful DHCPv6) specified, “I am your default gateway, but you need to ask a stateful DHCPv6 server for your IPv6 address and other addressing information."
3. The PC sends a DHCPv6 Solicit message to all DHCPv6 servers, " I received my default gateway address from the RA message, but I need an IPv6 address and all other addressing information from a stateful DHCPv6 server."

A stateful DHCPv6 server allocates and maintains a list of which device receives which IPv6 address. DHCP for IPv4 is stateful.

Note: The default gateway address can only be obtained dynamically from the RA message. The stateless or stateful DHCPv6 server does not provide the default gateway address.

12.5.5

EUI-64 Process vs. Randomly Generated

When the RA message is either SLAAC or SLAAC with stateless DHCPv6, the client must generate its own interface ID. The client knows the prefix portion of the address from the RA message, but must create its own interface ID. The interface ID can be created using the EUI-64 process or a randomly generated 64-bit number, as shown in the figure.

The graphic shows a router sending an ICMPv6 Router Advertisement message (labled #1) to a PC. The PC is shown in a step labled #2 creating its /64 prefix from the RA messageand creating its interface ID using EUI-64 or random 64-bit number. Text under the graphic reads 1. The router sends and RA message. 2. The PC uses the prefix in the RA message and uses either EUI-64 or a random 64-bit number to generate an interface ID

Dynamically Creating an Interface ID

/6412

ICMPv6 Router AdvertisementFrom RA MessageCreated by client device EUI-64 or random 64-bit number

1. The router sends an RA message.
2. The PC uses the prefix in the RA message and uses either EUI-64 or a random 64-bit number to generate an interface ID.

12.5.6

EUI-64 Process

IEEE defined the Extended Unique Identifier (EUI) or modified EUI-64 process. This process uses the 48-bit Ethernet MAC address of a client, and inserts another 16 bits in the middle of the 48-bit MAC address to create a 64-bit interface ID.

Ethernet MAC addresses are usually represented in hexadecimal and are made up of two parts:

- **Organizationally Unique Identifier (OUI)** - The OUI is a 24-bit (6 hexadecimal digits) vendor code assigned by IEEE.
- **Device Identifier** - The device identifier is a unique 24-bit (6 hexadecimal digits) value within a common OUI.

An EUI-64 Interface ID is represented in binary and is made up of three parts:

- 24-bit OUI from the client MAC address, but the 7th bit (the Universally/Locally (U/L) bit) is reversed. This means that if the 7th bit is a 0, it becomes a 1, and vice versa.
- The inserted 16-bit value fffe (in hexadecimal).
- 24-bit Device Identifier from the client MAC address.

The EUI-64 process is illustrated in the figure, using the R1 GigabitEthernet MAC address of fc99:4775:cee0.

The graphic shows the steps in the EUI-64 process. At the top of the graphic is the MAC address fc:99:47:75:ce:e0. In step 1: Split the MAC address, The first 24-bits of the OUI fc:99:47 in binary is 1111 1100 1001 1001 0100 0111. The final 24-bits of the device identifiere 75:ce:e0 in binary is 1111 0101 1100 1110 1110 0000. Instep 2: Insert ffee the binary representation becomes 1111 1100 1001 1001 0100 0111 1111 1111 1111 1110 1111 0101 1100 1110 1110 0000. In step three Flip the u/l bit. the 7th bit from the left is changed from a 0 to a 1. The address in binary is now 1111 1110 1001 1001 0100 0111 1111 1111 1111 1110 1111 0101 1100 1110 1110 0000. The modified EUI-64 Interface ID in Hexadecimal notation is now: fe:99:47:ff:fe:75:ce:e0. Test under the graphic reads: Step 1: Divide the MAC address between the OUI and device identifier. Step 2: Insert the hexadecimal value fffe, which in binary is: 1111 1111 1111 1110. Step 3. Convert the first 2 hexadecimalvalues of the OUI to binary and flip the U/L bit (bit 7). In this example, the 0 in bit 7 is change to at 1. The result is an EUI-64 generated interface ID of fe99:47ff:fe75:cee0. Note: The use of the U/L bit, and the reasons for reversign its value, are discussed in RFC 5342.

fc994775cee0fe994775cee0fffe

OUI
24 bitsDevice Identifier
24 bitsEUI-64 ProcessBinaryBinaryBinaryBinaryStep 1: Split the MAC addressStep 2: Insert fffeStep 3: Flip the u/l bitModified EUI-64 Interface ID in Hexadecimal Notation 0
1

Step 1: Divide the MAC address between the OUI and device identifier.

Step 2: Insert the hexadecimal value fffe, which in binary is: 1111 1111 1111 1110.

Step 3: Convert the first 2 hexadecimal values of the OUI to binary and flip the U/L bit (bit 7). In this example, the 0 in bit 7 is changed to a 1.

The result is an EUI-64 generated interface ID of fe99:47ff:fe75:cee0.

Note: The use of the U/L bit, and the reasons for reversing its value, are discussed in RFC 5342.

The example output for the **ipconfig** command shows the IPv6 GUA being dynamically created using SLAAC and the EUI-64 process. An easy way to identify that an address was probably created using EUI-64 is the **fffe** located in the middle of the interface ID.

The advantage of EUI-64 is that the Ethernet MAC address can be used to determine the interface ID. It also allows network administrators to easily track an IPv6 address to an end-device using the unique MAC address. However, this has caused privacy concerns among many users who worried that their packets could be traced to the actual physical computer. Due to these concerns, a randomly generated interface ID may be used instead.

EUI-64 Generated Interface ID

```
C:\> ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :

   IPv6 Address. . . . . . . . . . . : 2001:db8:acad:1:fc99:47ff:fe75:cee0

   Link-local IPv6 Address . . . . . : fe80::fc99:47ff:fe75:cee0

   Default Gateway . . . . . . . . . : fe80::1

C:\>
```

12.5.7

Randomly Generated Interface IDs

Depending upon the operating system, a device may use a randomly generated interface ID instead of using the MAC address and the EUI-64 process. Beginning with Windows Vista, Windows uses a randomly generated interface ID instead of one created with EUI-64. Windows XP and previous Windows operating systems used EUI-64.

After the interface ID is established, either through the EUI-64 process or through random generation, it can be combined with an IPv6 prefix in the RA message to create a GUA, as shown in the figure.

Random 64-Bit Generated Interface ID

```
C:\> ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :

   IPv6 Address. . . . . . . . . . . : 2001:db8:acad:1:50a5:8a35:a5bb:66e1

   Link-local IPv6 Address . . . . . : fe80::50a5:8a35:a5bb:66e1

   Default Gateway . . . . . . . . . : fe80::1

C:\>
```

Note: To ensure the uniqueness of any IPv6 unicast address, the client may use a process known as Duplicate Address Detection (DAD). This is similar to an ARP request for its own address. If there is no reply, then the address is unique.

12.6.1

Dynamic LLAs

All IPv6 devices must have an IPv6 LLA. Like IPv6 GUAs, you can also create LLAs dynamically. Regardless of how you create your LLAs (and your GUAs), it is important that you verify all IPv6 address configuration. This topic explains dynamically generated LLAs and IPv6 configuration verification.

The figure shows the LLA is dynamically created using the fe80::/10 prefix and the interface ID using the EUI-64 process, or a randomly generated 64-bit number.

The graphic shows the Prefix of an LLA in binary: 1111 1110 10, and in hexadecimal: fe80::/10.

fe80::/101111 1110 10/64

EUI-64 Process64 bitsRandomly Generated Number

12.6.2

Dynamic LLAs on Windows

Operating systems, such as Windows, will typically use the same method for both a SLAAC-created GUA and a dynamically assigned LLA. See the highlighted areas in the following examples that were shown previously.

EUI-64 Generated Interface ID

```
C:\> ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . :

IPv6 Address. . . . . . . . . . . : 2001:db8:acad:1:fc99:47ff:fe75:cee0

Link-local IPv6 Address . . . . . : fe80::fc99:47ff:fe75:cee0

Default Gateway . . . . . . . . . : fe80::1

C:\>
```

Random 64-Bit Generated Interface ID

```
C:\> ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :

   IPv6 Address. . . . . . . . . . . : 2001:db8:acad:1:50a5:8a35:a5bb:66e1

   Link-local IPv6 Address . . . . . : fe80::50a5:8a35:a5bb:66e1

   Default Gateway . . . . . . . . . : fe80::1

C:\>
```

12.6.3

Dynamic LLAs on Cisco Routers

Cisco routers automatically create an IPv6 LLA whenever a GUA is assigned to the interface. By default, Cisco IOS routers use EUI-64 to generate the interface ID for all LLAs on IPv6 interfaces. For serial interfaces, the router will use the MAC address of an Ethernet interface. Recall that an LLA must be unique only on that link or network. However, a drawback to using the dynamically assigned LLA is its long interface ID, which makes it challenging to identify and remember assigned addresses. The example displays the MAC address on the GigabitEthernet 0/0/0 interface of router R1. This address is used to dynamically create the LLA on the same interface, and also for the Serial 0/1/0 interface.

To make it easier to recognize and remember these addresses on routers, it is common to statically configure IPv6 LLAs on routers.

IPv6 LLA Using EUI-64 on Router R1

```
R1# show interface gigabitEthernet 0/0/0

GigabitEthernet0/0/0 is up, line protocol is up

  Hardware is ISR4221-2x1GE, address is 7079.b392.3640 (bia 7079.b392.3640)

(Output omitted)

R1# show ipv6 interface brief

GigabitEthernet0/0/0    [up/up]

    FE80::7279:B3FF:FE92:3640

    2001:DB8:ACAD:1::1

GigabitEthernet0/0/1    [up/up]

    FE80::7279:B3FF:FE92:3641

    2001:DB8:ACAD:2::1

Serial0/1/0             [up/up]

    FE80::7279:B3FF:FE92:3640

    2001:DB8:ACAD:3::1

Serial0/1/1             [down/down]

    unassigned

R1#
```

12.6.4

Verify IPv6 Address Configuration

The figure shows the example topology.

The graphic shows two PCs, PC1 and PC2. PC1 is connected to a switch and has the IPv6 address 2001:db8:acad:1::10/64. PC2 is connected to a switch and has the IPv6 address 2001:db8:acad:2::10/64. The two switches are connected to a router, R1. PC1 is connected through the switch to R1s G0/0/0 interface which has IPv6 address 2001:db8:acad:1::1/64 and the LLA address of fe80::1:1. PC2 is connected through the switch to R1s G0/0/1 interface which has IPv6 address 2001:db8:acad:2::1/64 and the LLA address of fe80::2:1. R1 connects to the cloud through its S0/1/0 interface which has IPv6 adress 2001:db8:acad:3::1/64 and the LLA address of fe80::3:1.

Click each button for the output and a description of the command.

show ipv6 interface brief

show ipv6 route

ping

The **show ipv6 interface brief** command displays the IPv6 address of the Ethernet interfaces. EUI-64 uses this MAC address to generate the interface ID for the LLA. Additionally, the **show ipv6 interface brief** command displays abbreviated output for each of the interfaces. The [up/up] output on the same line as the interface indicates the Layer 1/Layer 2 interface state. This is the same as the Status and Protocol columns in the equivalent IPv4 command.

Notice that each interface has two IPv6 addresses. The second address for each interface is the GUA that was configured. The first address, the one that begins with fe80, is the link-local unicast address for the interface. Recall that the LLA is automatically added to the interface when a GUA is assigned.

Also, notice that the R1 Serial 0/1/0 LLA is the same as its GigabitEthernet 0/0/0 interface. Serial interfaces do not have Ethernet MAC addresses, so Cisco IOS uses the MAC address of the first available Ethernet interface. This is possible because link-local interfaces only have to be unique on that link.

The show ipv6 interface brief Command on R1

```
R1# show ipv6 interface brief

GigabitEthernet0/0/0    [up/up]

    FE80::7279:b3ff:fe92:3640

    2001:DB8:ACAD:1::1

GigabitEthernet0/0/1    [up/up]

    FE80::8320:e4ff:fe79:2830

    2001:DB8:ACAD:2::1

Serial0/1/0             [up/up]

    FE80::7279:b3ff:fe92:3640

    2001:DB8:ACAD:3::1

Serial0/1/1             [down/down]

    unassigned

R1#
```

12.6.5

Syntax Checker - Verify IPv6 Address Configuration

Use **show** commands to verify IPv6 address configuration on router R1 interfaces.

The graphic shows two PCs, PC1 and PC2. PC1 is connected to a switch and has the IPv6 address 2001:db8:acad:1::10/64. PC2 is connected to a switch and has the IPv6 address 2001:db8:acad:2::10/64. The two switches are connected to a router, R1. PC1 is connected through the switch to R1s G0/0/0 interface which has IPv6 address 2001:db8:acad:1::1/64 and the LLA address of fe80::1:1. PC2 is connected through the switch to R1s G0/0/1 interface which has IPv6 address 2001:db8:acad:2::1/64 and the LLA address of fe80::2:1. R1 connects to the cloud through its S0/1/0 interface which has IPv6 adress 2001:db8:acad:3::1/64 and the LLA address of fe80::3:1.

R1::10::10S0/1/0::1fe80::3:1PC12001.db8.acad:2::/642001.db8.acad:1::/642001.db8.acad:3::/64G0/0/0::1fe80::1:1G0/0/1::1fe80::2:1PC2R1

Enter the show command that will display a brief summary of the IPv6 interface status.

```
R1#
```

12.7.1

Assigned IPv6 Multicast Addresses

Earlier in this module, you learned that there are three broad categories of IPv6 addresses: unicast, anycast, and multicast. This topic goes into more detail about multicast addresses.

IPv6 multicast addresses are similar to IPv4 multicast addresses. Recall that a multicast address is used to send a single packet to one or more destinations (multicast group). IPv6 multicast addresses have the prefix ff00::/8.

Note: Multicast addresses can only be destination addresses and not source addresses.

There are two types of IPv6 multicast addresses:

- Well-known multicast addresses
- Solicited node multicast addresses

12.7.2

Well-Known IPv6 Multicast Addresses

Well-known IPv6 multicast addresses are assigned. Assigned multicast addresses are reserved multicast addresses for predefined groups of devices. An assigned multicast address is a single address used to reach a group of devices running a common protocol or service. Assigned multicast addresses are used in context with specific protocols such as DHCPv6.

These are two common IPv6 assigned multicast groups:

- **ff02::1 All-nodes multicast group** - This is a multicast group that all IPv6-enabled devices join. A packet sent to this group is received and processed by all IPv6 interfaces on the link or network. This has the same effect as a broadcast address in IPv4. The figure shows an example of communication using the all-nodes multicast address. An IPv6 router sends ICMPv6 RA messages to the all-node multicast group.
- **ff02::2 All-routers multicast group** - This is a multicast group that all IPv6 routers join. A router becomes a member of this group when it is enabled as an IPv6 router with the **ipv6 unicast-routing** global configuration command. A packet sent to this group is received and processed by all IPv6 routers on the link or network.

The graphic shows two PCs wtih IPv6 addresses of 2001:db8:acad:1::10/64 and 2001:db8:acad:1::20/64, a server with IPv6 address 2001:db8:acad:1::8/64, and a printer with IPv6 address 2001:db8:acad:1::9/64 connected to a switch which is connected to a router. Above the graphic is indicated the source IPv6 address of fe80::1 and the destination IPv6 address of ff02::1. Text under the graphic reads IPv6-enabled devices send ICMPv6 RS messages to the all-routers multicast address. The RS message requests an RA message form the IPv6 router assist the device in its address configuation. The IPv6 router responds with an RA message, as shown.

IPv6 All-Nodes Multicast: RA Message

2001:db8:acad:1::20/642001:db8:acad:1::8/642001:db8:acad:1::9/642001:db8:acad:1::10/64ff02::1fe80::1

IPv6-enabled devices send ICMPv6 RS messages to the all-routers multicast address. The RS message requests an RA message from the IPv6 router to assist the device in its address configuration. The IPv6 router responds with an RA message, as shown.

12.7.3

Solicited-Node IPv6 Multicast Addresses

A solicited-node multicast address is similar to the all-nodes multicast address. The advantage of a solicited-node multicast address is that it is mapped to a special Ethernet multicast address. This allows the Ethernet NIC to filter the frame by examining the destination MAC address without sending it to the IPv6 process to see if the device is the intended target of the IPv6 packet.

The graphic shows three PCs recieving a message from a router. Each PC has the following informational text: My Ethernet NIC determined this multicast is not for me. Above the graphic is indicated that the Destination MAC address is a multicast and the Destination IPv6 address is a Solicited-Node multicast.

IPv6Ethernet My Ethernet NIC determined this multicast is not for me.My Ethernet NIC determined this multicast is not for me.My Ethernet NIC determined this multicast is for me!

12.7.4

12.8.1

Subnet Using the Subnet ID

The introduction to this module mentioned subnetting an IPv6 network. It also said that you might discover that it is a bit easier than subnetting an IPv4 network. You are about to find out!

Recall that with IPv4, we must borrow bits from the host portion to create subnets. This is because subnetting was an afterthought with IPv4. However, IPv6 was designed with subnetting in mind. A separate subnet ID field in the IPv6 GUA is used to create subnets. As shown in the figure, the subnet ID field is the area between the Global Routing Prefix and the interface ID.

The graphic shows the parts of a GUA. First is the 48 bit Global Routing Prefix followed by the 16 bit Subnet ID, then finally the 64 bit Interface ID. Text under the graphic reads A /48 routing prefix + 16 bit Subnet ID = /64 prefix.

GUA with a 16-bit Subnet ID

48 bits16 bits64 bits A /48 routing prefix + 16 bit Subnet ID = /64 prefix

The benefit of a 128-bit address is that it can support more than enough subnets and hosts per subnet, for each network. Address conservation is not an issue. For example, if the global routing prefix is a /48, and using a typical 64 bits for the interface ID, this will create a 16-bit subnet ID:

- **16-bit subnet ID** - Creates up to 65,536 subnets.
- **64-bit interface ID** - Supports up to 18 quintillion host IPv6 addresses per subnet (i.e., 18,000,000,000,000,000,000).

Note: Subnetting into the 64-bit interface ID (or host portion) is also possible but it is rarely required.

IPv6 subnetting is also easier to implement than IPv4, because there is no conversion to binary required. To determine the next available subnet, just count up in hexadecimal.

12.8.2

IPv6 Subnetting Example

For example, assume an organization has been assigned the 2001:db8:acad::/48 global routing prefix with a 16 bit subnet ID. This would allow the organization to create 65,536 /64 subnets, as shown in the figure. Notice how the global routing prefix is the same for all subnets. Only the subnet ID hextet is incremented in hexadecimal for each subnet.

The graphic shows the IPv6 address prefix 2001:db8:acad::/48 subnetted into /64 subnets. A note reads Increment subnet ID to create 65,536 subnets. The subnets are: 2001:db8:acad:0000::/64, 2001:db8:acad:0001::/64, 2001:db8:acad:0002::/64, 2001:db8:acad:0003::/64, 2001:db8:acad:0004::/64, 2001:db8:acad:0005::/64, 2001:db8:acad:0006::/64, 2001:db8:acad:0007::/64, 2001:db8:acad:0008::/64, 2001:db8:acad:0009::/64, 2001:db8:acad:000a::/64, 2001:db8:acad:000b::/64, 2001:db8:acad:000c::/64. Subnets 13-65,534 not shown, 2001:db8:acad:ffff::/64.

Subnetting using a 16-bit Subnet ID

2001:db8:acad:0000::/642001:db8:acad:0001::/642001:db8:acad:0002::/642001:db8:acad:0003::/642001:db8:acad:0004::/642001:db8:acad:0005::/642001:db8:acad:0006::/642001:db8:acad:0007::/642001:db8:acad:0008::/642001:

db8:acad:0009::/642001:db8:acad:000a::/642001:db8:acad:000b::/642001:db8:acad:000c::/642001:db8:acad:ffff::/64

Increment subnet ID to create 65,536 subnets Subnets 13 – 65,534 not shown

12.8.3

IPv6 Subnet Allocation

With over 65,536 subnets to choose from, the task of the network administrator becomes one of designing a logical scheme to address the network.

As shown in the figure, the example topology requires five subnets, one for each LAN as well as for the serial link between R1 and R2. Unlike the example for IPv4, with IPv6 the serial link subnet will have the same prefix length as the LANs. Although this may seem to “waste” addresses, address conservation is not a concern when using IPv6.

The graphic shows four PCs, PC1, PC2, PC3, and PC4, each with the interface ID of ::10. Each PC is connected to a switch. PC1 is in network 2001:db8:acad:1::/64 and connects through a switch to the G0/0/0 interface, with interface ID ::1, of router 1. PC2 is in network 2001:db8:acad:2::/64 and connects through a switch to the G0/0/1 interface, with interface ID ::1, of router 1. PC3 is in network 2001:db8:acad:4::/64 and connects through a switch to the G0/0/0 interface, with interface ID ::1, of router 2. PC4 is in network 2001:db8:acad:5::/64 and connects through a switch to the G0/0/1 interface, with interface ID ::1 of router 2. Router 1 and 2 are connected over their S0/1/0 interfaces with R1 having an interface ID of ::1 and R2 having an interface ID of ::2 in the 2001:db8:acad:3::/64 network.

Example Topology

2001:db8:acad:1::/642001:db8:acad:2::/642001:db8:acad:4::/642001:db8:acad:5::/642001:db8:acad:3::/64G0/0/1::1::10::10::10::10S0/1/0::1S0/1/0::2G0/0/0::1G0/0/0::1G0/0/1::1PC1PC2PC3PC4

As shown in the next figure, the five IPv6 subnets were allocated, with the subnet ID field 0001 through 0005 used for this example. Each /64 subnet will provide more addresses than will ever be needed.

The graphic shows subnets from the address block: 2001:db8:acad::/48. The subnets are: 2001:db8:acad:0000::/64, 2001:db8:acad:0001::/64, 2001:db8:acad:0002::/64, 2001:db8:acad:0003::/64, 2001:db8:acad:0004::/64, 2001:db8:acad:0005::/64, 2001:db8:acad:0006::/64, 2001:db8:acad:0007::/64, 2001:db8:acad:0008::/64, 2001:db8:acad:ffff::/64. A note reads 5 subnets allocated from 65,536 available and indicates the five subnets allocated are: 2001:db8:acad:0001::/64, 2001:db8:acad:0002::/64, 2001:db8:acad:0003::/64, 2001:db8:acad:0004::/64, 2001:db8:acad:0005::/64.

...2001:db8:acad:0000::/642001:db8:acad:0001::/642001:db8:acad:0002::/642001:db8:acad:0003::/642001:db8:acad:0004::/642001:db8:acad:0005::/642001:db8:acad:0006::/642001:db8:acad:0007::/642001:db8:acad:0008::/642001:db8:acad:ffff::/64

5 subnets allocated from 65,536 available subnetsAddress Block: 2001:0db8:acad::/48

12.8.4

Router Configured with IPv6 Subnets

Similar to configuring IPv4, the example shows that each of the router interfaces has been configured to be on a different IPv6 subnet.

IPv6 Address Configuration on Router R1

```
R1(config)# interface gigabitethernet 0/0/0
```

```
R1(config-if)# ipv6 address 2001:db8:acad:1::1/64
```

```
R1(config-if)# no shutdown
```

```
R1(config-if)# exit
```

```
R1(config)# interface gigabitethernet 0/0/1
```

```
R1(config-if)# ipv6 address 2001:db8:acad:2::1/64
```

```
R1(config-if)# no shutdown
```

```
R1(config-if)# exit
```

```
R1(config)# interface serial 0/1/0
```

```
R1(config-if)# ipv6 address 2001:db8:acad:3::1/64
```

```
R1(config-if)# no shutdown
```

13.1.1

ICMPv4 and ICMPv6 Messages

In this topic, you will learn about the different types of Internet Control Message Protocols (ICMPs), and the tools that are used to send them.

Although IP is only a best-effort protocol, the TCP/IP suite does provide for error messages and informational messages when communicating with another IP device. These messages are sent using the services of ICMP. The purpose of these messages is to provide feedback about issues related to the processing of IP packets under certain conditions, not to make IP reliable. ICMP messages are not required and are often not allowed within a network for security reasons.

ICMP is available for both IPv4 and IPv6. ICMPv4 is the messaging protocol for IPv4. ICMPv6 provides these same services for IPv6 but includes additional functionality. In this course, the term ICMP will be used when referring to both ICMPv4 and ICMPv6.

The types of ICMP messages, and the reasons why they are sent, are extensive. The ICMP messages common to both ICMPv4 and ICMPv6 and discussed in this module include:

- Host reachability
- Destination or Service Unreachable
- Time exceeded

13.1.2

Host Reachability

An ICMP Echo Message can be used to test the reachability of a host on an IP network. The local host sends an ICMP Echo Request to a host. If the host is available, the destination host responds with an Echo Reply. In the figure, click the Play button to see an animation of the ICMP Echo Request/Echo Reply. This use of the ICMP Echo messages is the basis of the **ping** utility.

animation of host 1 sending a ping ICMP echo request to host 2 and the ICMP echo reply from host 2 back to host 1

Yes, I am here.

ping 192.168.30.1

13.1.3

Destination or Service Unreachable

When a host or gateway receives a packet that it cannot deliver, it can use an ICMP Destination Unreachable message to notify the source that the destination or service is unreachable. The message will include a code that indicates why the packet could not be delivered.

Some of the Destination Unreachable codes for ICMPv4 are as follows:

- 0 - Net unreachable
- 1 - Host unreachable
- 2 - Protocol unreachable
- 3 - Port unreachable

Some of the Destination Unreachable codes for ICMPv6 are as follows:

- 0 - No route to destination
- 1 - Communication with the destination is administratively prohibited (e.g., firewall)
- 2 – Beyond scope of the source address
- 3 - Address unreachable
- 4 - Port unreachable

Note: ICMPv6 has similar but slightly different codes for Destination Unreachable messages.

13.1.4

Time Exceeded

An ICMPv4 Time Exceeded message is used by a router to indicate that a packet cannot be forwarded because the Time to Live (TTL) field of the packet was decremented to 0. If a router receives a packet and decrements the TTL field in the IPv4 packet to zero, it discards the packet and sends a Time Exceeded message to the source host.

ICMPv6 also sends a Time Exceeded message if the router cannot forward an IPv6 packet because the packet has expired. Instead of the IPv4 TTL field, ICMPv6 uses the IPv6 Hop Limit field to determine if the packet has expired.

Note: Time Exceeded messages are used by the **traceroute** tool.

13.1.5

ICMPv6 Messages

The informational and error messages found in ICMPv6 are very similar to the control and error messages implemented by ICMPv4. However, ICMPv6 has new features and improved functionality not found in ICMPv4. ICMPv6 messages are encapsulated in IPv6.

ICMPv6 includes four new protocols as part of the Neighbor Discovery Protocol (ND or NDP).

Messaging between an IPv6 router and an IPv6 device, including dynamic address allocation are as follows:

- Router Solicitation (RS) message
- Router Advertisement (RA) message

Messaging between IPv6 devices, including duplicate address detection and address resolution are as follows:

- Neighbor Solicitation (NS) message
- Neighbor Advertisement (NA) message

Note: ICMPv6 ND also includes the redirect message, which has a similar function to the redirect message used in ICMPv4.

Click each for an illustration and explanation of ICMPv6 messages.

RA Message

RS Message

NS Message

NA Message

RA messages are sent by IPv6-enabled routers every 200 seconds to provide addressing information to IPv6-enabled hosts. The RA message can include addressing information for the host such as the prefix, prefix length, DNS address, and domain name. A host using Stateless Address Autoconfiguration (SLAAC) will set its default gateway to the link-local address of the router that sent the RA.

R1 sends an RA router advertisement message to FF02::1 the all-nodes multicast address which will reach PC1.

2001:db8:acad:1::1/64PC12001:db8:acad:1::1/64fe80::1R1

RA Message

R1 sends an RA message, “Hi all IPv6-enabled devices. I’m R1 and you can use SLAAC to create an IPv6 global unicast address. The prefix is 2001:db8:acad:1::/64. By the way, use my link-local address fe80::1 as your default gateway."

13.2.1

Ping - Test Connectivity

In the previous topic, you were introduced to the **ping** and traceroute (**tracert**) tools. In this topic, you will learn about the situations in which each tool is used, and how to use them. Ping is an IPv4 and IPv6 testing utility that uses ICMP echo request and echo reply messages to test connectivity between hosts.

To test connectivity to another host on a network, an echo request is sent to the host address using the **ping** command. If the host at the specified address receives the echo request, it responds with an echo reply. As each echo reply is received, **ping** provides feedback on the time between when the request was sent and when the reply was received. This can be a measure of network performance.

Ping has a timeout value for the reply. If a reply is not received within the timeout, ping provides a message indicating that a response was not received. This may indicate that there is a problem, but could also indicate that security features blocking ping messages have been enabled on the network. It is common for the first ping to timeout if address resolution (ARP or ND) needs to be performed before sending the ICMP Echo Request.

After all the requests are sent, the **ping** utility provides a summary that includes the success rate and average round-trip time to the destination.

Type of connectivity tests performed with **ping** include the following:

- Pinging the local loopback
- Pinging the default gateway
- Pinging the remote host

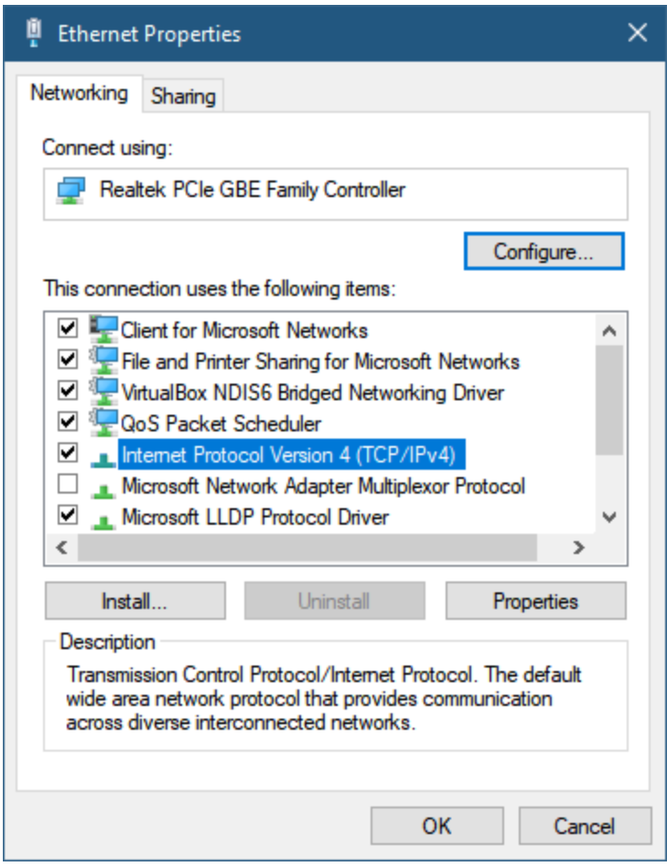
13.2.2

Ping the Loopback

Ping can be used to test the internal configuration of IPv4 or IPv6 on the local host. To perform this test, **ping** the local loopback address of 127.0.0.1 for IPv4 (::1 for IPv6).

A response from 127.0.0.1 for IPv4, or ::1 for IPv6, indicates that IP is properly installed on the host. This response comes from the network layer. This response is not, however, an indication that the addresses, masks, or gateways are properly configured. Nor does it indicate anything about the status of the lower layer of the network stack. This simply tests IP down through the network layer of IP. An error message indicates that TCP/IP is not operational on the host.

shows the Ethernet properties dialogue box shows that Internet Protocol Version 4 (TCP/IPv4) is installed and active which is proved with a ping to 127.0.0.1




```
C:\>ping 127.0.0.1
```

- Pinging the local host confirms that TCP/IP is installed and working on the local host.
- Pinging 127.0.0.1 causes a device to ping itself.

13.2.3

Ping the Default Gateway

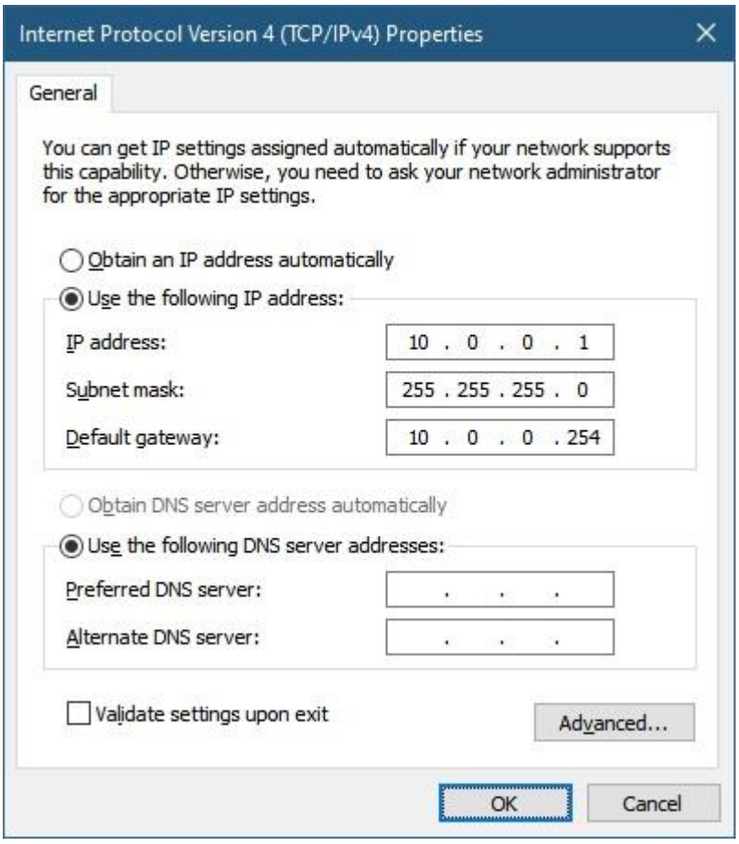
You can also use **ping** to test the ability of a host to communicate on the local network. This is generally done by pinging the IP address of the default gateway of the host. A successful **ping** to the default gateway indicates that the host and the router interface serving as the default gateway are both operational on the local network.

For this test, the default gateway address is most often used because the router is normally always operational. If the default gateway address does not respond, a **ping** can be sent to the IP address of another host on the local network that is known to be operational.

If either the default gateway or another host responds, then the local host can successfully communicate over the local network. If the default gateway does not respond but another host does, this could indicate a problem with the router interface serving as the default gateway.

One possibility is that the wrong default gateway address has been configured on the host. Another possibility is that the router interface may be fully operational but have security applied to it that prevents it from processing or responding to ping requests.

The graphic shows the Ethernet properties dialogue box configured with a static IP address, subnet mask, and default gateway. The topology shows the PC sending an echo request to the router default gateway and the routers echo response reply.



ECHO REQUESTECHO REPLY10.0.0.1
255.255.255.010.0.0.254
255.255.255.0G0/0/0

The host pings its default gateway, sending an ICMP echo request. The default gateway sends an echo reply confirming connectivity.

13.2.4

Ping a Remote Host

Ping can also be used to test the ability of a local host to communicate across an internetwork. The local host can ping an operational IPv4 host of a remote network, as shown in the figure. The router uses its IP routing table to forward the packets.

If this ping is successful, the operation of a large piece of the internetwork can be verified. A successful **ping** across the internetwork confirms communication on the local network, the operation of the router serving as the default gateway, and the operation of all other routers that might be in the path between the local network and the network of the remote host.

Additionally, the functionality of the remote host can be verified. If the remote host could not communicate outside of its local network, it would not have responded.

Note: Many network administrators limit or prohibit the entry of ICMP messages into the corporate network; therefore, the lack of a **ping** response could be due to security restrictions.

animation shows a ping echo request to a remote network that is routed through a router and the echo reply that is routed back from the remote network

Echo request

Echo reply

IP Routing Table

13.2.5

Traceroute - Test the Path

Ping is used to test connectivity between two hosts but does not provide information about the details of devices between the hosts. Traceroute (**tracert**) is a utility that generates a list of hops that were successfully reached along the path. This list can provide important verification and troubleshooting information. If the data reaches the destination, then the trace lists the interface of every router in the path between the hosts. If the data fails at some

hop along the way, the address of the last router that responded to the trace can provide an indication of where the problem or security restrictions are found.

Round Trip Time (RTT)

Using traceroute provides round-trip time for each hop along the path and indicates if a hop fails to respond. The round-trip time is the time a packet takes to reach the remote host and for the response from the host to return. An asterisk (*) is used to indicate a lost or unreplied packet.

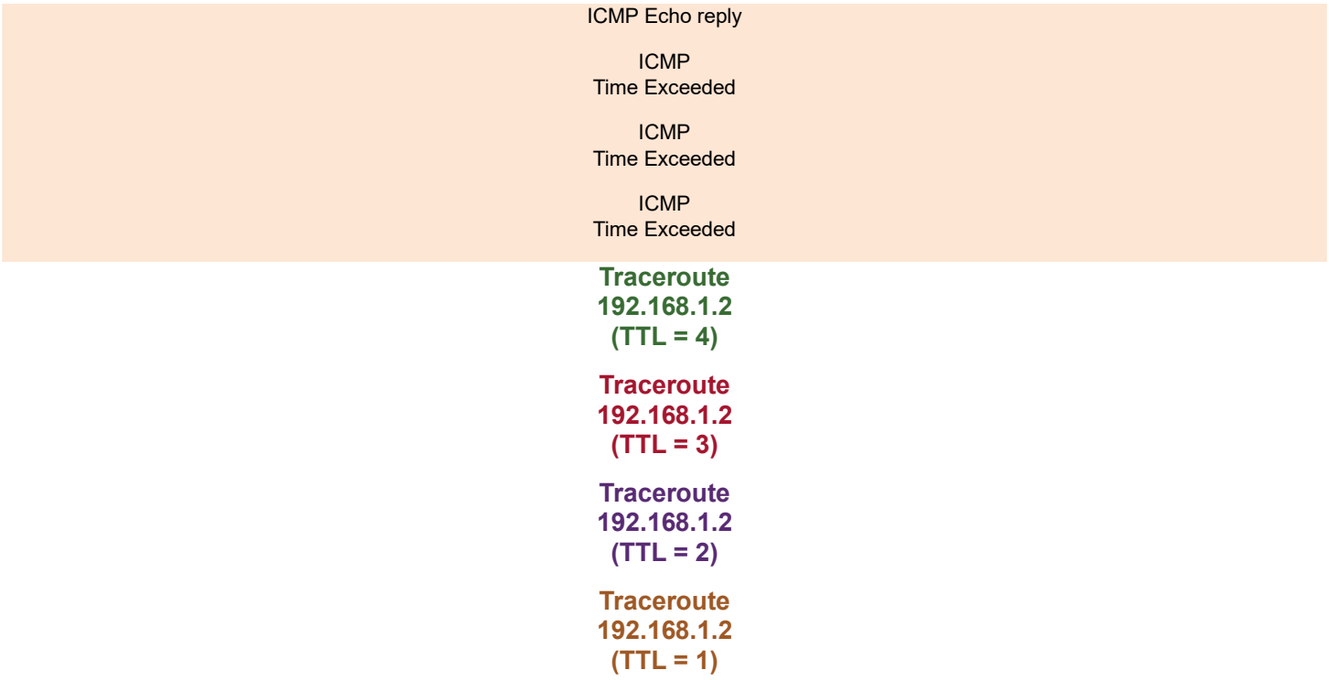
This information can be used to locate a problematic router in the path or may indicate that the router is configured not to reply. If the display shows high response times or data losses from a particular hop, this is an indication that the resources of the router or its connections may be stressed.

IPv4 TTL and IPv6 Hop Limit

Traceroute makes use of a function of the TTL field in IPv4 and the Hop Limit field in IPv6 in the Layer 3 headers, along with the ICMP Time Exceeded message.

Play the animation in the figure to see how traceroute takes advantage of TTL.

animation shows a traceroute to a remote network that crosses three routers. The traceroute will take 4 echo requests to reach its destination



The first sequence of messages sent from traceroute will have a TTL field value of 1. This causes the TTL to time out the IPv4 packet at the first router. This router then responds with an ICMPv4 Time Exceeded message. Traceroute now has the address of the first hop.

Traceroute then progressively increments the TTL field (2, 3, 4...) for each sequence of messages. This provides the trace with the address of each hop as the packets time out further down the path. The TTL field continues to be increased until the destination is reached, or it is incremented to a predefined maximum.

After the final destination is reached, the host responds with either an ICMP Port Unreachable message or an ICMP Echo Reply message instead of the ICMP Time Exceeded message.

14.1.1

Role of the Transport Layer

Application layer programs generate data that must be exchanged between source and destination hosts. The transport layer is responsible for logical communications between applications running on different hosts. This may include services such as establishing a temporary session between two hosts and the reliable transmission of information for an application.

As shown in the figure, the transport layer is the link between the application layer and the lower layers that are responsible for network transmission.

shows a diagram of how devices use the transport layer to move data between applications in the TCP/IP model

The transport layer moves data between applications on devices in the network. **TCP/IP Model** **TCP/IP Model**

The transport layer has no knowledge of the destination host type, the type of media over which the data must travel, the path taken by the data, the congestion on a link, or the size of the network.

The transport layer includes two protocols:

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

14.1.2

Transport Layer Responsibilities

The transport layer has many responsibilities.

Select each tab for more information.
Tracking Individual Conversations

Segmenting Data and Reassembling Segments

Add Header Information

Identifying the Applications

Conversation Multiplexing

Tracking Individual Conversations

At the transport layer, each set of data flowing between a source application and a destination application is known as a conversation and is tracked separately. It is the responsibility of the transport layer to maintain and track these multiple conversations.

As illustrated in the figure, a host may have multiple applications that are communicating across the network simultaneously.

Most networks have a limitation on the amount of data that can be included in a single packet. Therefore, data must be divided into manageable pieces.

The PC simultaneously runs multiple network applications including an email client, instant messaging client, web browser web pages, streaming video, and a video conference client.

EmailOnline Video ChattingStreaming VideoMultiple Web PagesInstant MessagingNetwork

14.1.3

Transport Layer Protocols

IP is concerned only with the structure, addressing, and routing of packets. IP does not specify how the delivery or transportation of the packets takes place.

Transport layer protocols specify how to transfer messages between hosts, and are responsible for managing reliability requirements of a conversation. The transport layer includes the TCP and UDP protocols.

Different applications have different transport reliability requirements. Therefore, TCP/IP provides two transport layer protocols, as shown in the figure.

shows how application layer protocols like FTP, HTTP, SMTP use TCP at the transport layer and DNS and TFTP use UDP. How they all use IP at the internet layer regardless of whether they connect to a LAN or a WAN at the network access layer

ApplicationTransportInternetNetwork AccessFTPHTTP
(www)SMTP
(email)DNSTFTPTCPUDPIPLAN
connectionsWAN
connections

14.1.4

Transmission Control Protocol (TCP)

IP is concerned only with the structure, addressing, and routing of packets, from original sender to final destination. IP is not responsible for guaranteeing delivery or determining whether a connection between the sender and receiver needs to be established.

TCP is considered a reliable, full-featured transport layer protocol, which ensures that all of the data arrives at the destination. TCP includes fields which ensure the delivery of the application data. These fields require additional processing by the sending and receiving hosts.

Note: TCP divides data into segments.

TCP transport is analogous to sending packages that are tracked from source to destination. If a shipping order is broken up into several packages, a customer can check online to see the order of the delivery.

TCP provides reliability and flow control using these basic operations:

- Number and track data segments transmitted to a specific host from a specific application
- Acknowledge received data
- Retransmit any unacknowledged data after a certain amount of time
- Sequence data that might arrive in wrong order
- Send data at an efficient rate that is acceptable by the receiver

In order to maintain the state of a conversation and track the information, TCP must first establish a connection between the sender and the receiver. This is why TCP is known as a connection-oriented protocol.

Click Play in the figure to see how TCP segments and acknowledgments are transmitted between sender and receiver.

The animation shows a connection to an FTP server initiated with a TCP 3-way handshake and the data segments being accounted for by using sequence numbers and acknowledgements

A file is sent to a server using the File Transfer Protocol (FTP) application. TCP tracks the conversation and divides the data to be sent into 6 segments.

The first 3 out of 6 segments are forwarded to the server.

The file server acknowledges the first 3 segments received.

The client forwards the next 3 segments.

No segments are received, no acknowledgement is sent.

The client resends the final 3 segments.

The final 3 segments are received and acknowledged.

ISP 1

FTP

Server Farm

Internet

ISP 2

14.1.5

User Datagram Protocol (UDP)

UDP is a simpler transport layer protocol than TCP. It does not provide reliability and flow control, which means it requires fewer header fields. Because the sender and the receiver UDP processes do not have to manage reliability and flow control, this means UDP datagrams can be processed faster than TCP segments. UDP provides the basic functions for delivering datagrams between the appropriate applications, with very little overhead and data checking.

Note: UDP divides data into datagrams that are also referred to as segments.

UDP is a connectionless protocol. Because UDP does not provide reliability or flow control, it does not require an established connection. Because UDP does not track information sent or received between the client and server, UDP is also known as a stateless protocol.

UDP is also known as a best-effort delivery protocol because there is no acknowledgment that the data is received at the destination. With UDP, there are no transport layer processes that inform the sender of a successful delivery.

UDP is like placing a regular, nonregistered, letter in the mail. The sender of the letter is not aware of the availability of the receiver to receive the letter. Nor is the post office responsible for tracking the letter or informing the sender if the letter does not arrive at the final destination.

Click Play in the figure to see an animation of UDP datagrams being transmitted from sender to receiver.

animation shows a connection to a TFTP server using UDP datagrams which are sent without sequence numbers or acknowledgments

A file is sent to a server using the Trivial File Transfer Protocol (TFTP) application. UDP divides the data into datagrams and sends them using best-effort delivery.

The file server receives all 6 segments, no acknowledgment is sent.

ISP 1

TFTP

Server Farm

Internet

ISP 2

14.1.6

The Right Transport Layer Protocol for the Right Application

Some applications can tolerate some data loss during transmission over the network, but delays in transmission are unacceptable. For these applications, UDP is the better choice because it requires less network overhead. UDP is preferable for applications such as Voice over IP (VoIP). Acknowledgments and retransmission would slow down delivery and make the voice conversation unacceptable.

UDP is also used by request-and-reply applications where the data is minimal, and retransmission can be done quickly. For example, Domain Name System (DNS) uses UDP for this type of transaction. The client requests IPv4 and IPv6 addresses for a known domain name from a DNS server. If the client does not receive a response in a predetermined amount of time, it simply sends the request again.

For example, if one or two segments of a live video stream fail to arrive, it creates a momentary disruption in the stream. This may appear as distortion in the image or sound, but may not be noticeable to the user. If the destination device had to account for lost data, the stream could be delayed while waiting for retransmissions, therefore causing the image or sound to be greatly degraded. In this case, it is better to render the best media possible with the segments received, and forego reliability.

For other applications it is important that all the data arrives and that it can be processed in its proper sequence. For these types of applications, TCP is used as the transport protocol. For example, applications such as databases, web browsers, and email clients, require that all data that is sent arrives at the destination in its original condition. Any missing data could corrupt a communication, making it either incomplete or unreadable. For example, it is important when accessing banking information over the web to make sure all the information is sent and received correctly.

Application developers must choose which transport protocol type is appropriate based on the requirements of the applications. Video may be sent over TCP or UDP. Applications that stream stored audio and video typically use TCP. The application uses TCP to perform buffering, bandwidth probing, and congestion control, in order to better control the user experience.

Real-time video and voice usually use UDP, but may also use TCP, or both UDP and TCP. A video conferencing application may use UDP by default, but because many firewalls block UDP, the application can also be sent over TCP.

Applications that stream stored audio and video use TCP. For example, if your network suddenly cannot support the bandwidth needed to watch an on-demand movie, the application pauses the playback. During the pause, you might see a “buffering...” message while TCP works to re-establish the stream. When all the segments are in order and a minimum level of bandwidth is restored, your TCP session resumes, and the movie resumes playing.

The figure summarizes differences between UDP and TCP.

lists the differences between UDP: fast, low overhead, no acknowledgements, no resending and TCP: reliable, acknowledges data, resends lost data, and delivers data with sequence numbers

TCPUDPVoIP
(IP telephony)DNS
(Domain Name Resolution)SMTP/IMAP
(Email)HTTP/HTTPS
(World Wide WebRequired protocol properties:

- Fast
- Low overhead
- Does not require acknowledgements
- Does not resend lost data
- Delivers data as it arrives

Required protocol properties:

- Reliable
- Acknowledges data
- Resends lost data
- Delivers data in sequenced order

14.1.7

14.2.1

TCP Features

In the previous topic, you learned that TCP and UDP are the two transport layer protocols. This topic gives more details about what TCP does and when it is a good idea to use it instead of UDP.

To understand the differences between TCP and UDP, it is important to understand how each protocol implements specific reliability features and how each protocol tracks conversations.

In addition to supporting the basic functions of data segmentation and reassembly, TCP also provides the following services:

- **Establishes a Session -** TCP is a connection-oriented protocol that negotiates and establishes a permanent connection (or session) between source and destination devices prior to forwarding any traffic. Through session establishment, the devices negotiate the amount of traffic that can be forwarded at a given time, and the communication data between the two can be closely managed.
- **Ensures Reliable Delivery -** For many reasons, it is possible for a segment to become corrupted or lost completely, as it is transmitted over the network. TCP ensures that each segment that is sent by the source arrives at the destination.
- **Provides Same-Order Delivery -** Because networks may provide multiple routes that can have different transmission rates, data can arrive in the wrong order. By numbering and sequencing the segments, TCP ensures segments are reassembled into the proper order.
- **Supports Flow Control -** Network hosts have limited resources (i.e., memory and processing power). When TCP is aware that these resources are overtaxed, it can request that the sending application reduce the rate of data flow. This is done by TCP regulating the amount of data the source transmits. Flow control can prevent the need for retransmission of the data when the resources of the receiving host are overwhelmed.

For more information on TCP, search the internet for the RFC 793.

14.2.2

TCP Header

TCP is a stateful protocol which means it keeps track of the state of the communication session. To track the state of a session, TCP records which information it has sent and which information has been acknowledged. The stateful session begins with the session establishment and ends with the session termination.

A TCP segment adds 20 bytes (i.e., 160 bits) of overhead when encapsulating the application layer data. The figure shows the fields in a TCP header.

shows the fields in the TCP header

20 Bytes

14.2.3

TCP Header Fields

The table identifies and describes the ten fields in a TCP header.

TCP Header Field DescriptionSource Port A 16-bit field used to identify the source application by port number.Destination PortA 16-bit field used to identify the destination application by port number.Sequence Number A 32-bit field used for data reassembly purposes.Acknowledgment Number A 32-bit field used to indicate that data has been received and the next byte expected from the source.Header Length A 4-bit field known as ”data offset” that indicates the length of the TCP segment header.Reserved A 6-bit field that is reserved for future use.Control bits A 6-bit field used that includes bit codes, or flags, which indicate the purpose and function of the TCP segment.Window size A 16-bit field used to indicate the number of bytes that can be accepted at one time.Checksum A 16-bit field used for error checking of the segment header and data.Urgent A 16-bit field used to indicate if the contained data is urgent.	
TCP Header Field	**Description**
Source Port	A 16-bit field used to identify the source application by port number.
Destination Port	A 16-bit field used to identify the destination application by port number.
Sequence Number	A 32-bit field used for data reassembly purposes.
Acknowledgment Number	A 32-bit field used to indicate that data has been received and the next byte expected from the source.
Header Length	A 4-bit field known as ”data offset” that indicates the length of the TCP segment header.
Reserved	A 6-bit field that is reserved for future use.
Control bits	A 6-bit field that includes bit codes, or flags, which indicate the purpose and function of the TCP segment.
Window size	A 16-bit field used to indicate the number of bytes that can be accepted at one time.
Checksum	A 16-bit field used for error checking of the segment header and data.
Urgent	A 16-bit field used to indicate if the contained data is urgent.

14.2.4

Applications that use TCP

TCP is a good example of how the different layers of the TCP/IP protocol suite have specific roles. TCP handles all tasks associated with dividing the data stream into segments, providing reliability, controlling data flow, and reordering segments. TCP frees the application from having to manage any of these tasks. Applications, like those shown in the figure, can simply send the data stream to the transport layer and use the services of TCP.

shows arrows pointing both directions from HTTP, FTP, SMTP, and SSH to TCP and then from TCP to IP

14.2.5

14.3.1

UDP Features

This topic will cover UDP, what it does, and when it is a good idea to use it instead of TCP. UDP is a best-effort transport protocol. UDP is a lightweight transport protocol that offers the same data segmentation and reassembly as TCP, but without TCP reliability and flow control.

UDP is such a simple protocol that it is usually described in terms of what it does not do compared to TCP.

UDP features include the following:

- Data is reconstructed in the order that it is received.
- Any segments that are lost are not resent.
- There is no session establishment.
- The sending is not informed about resource availability.

For more information on UDP, search the internet for the RFC.

14.3.2

UDP Header

UDP is a stateless protocol, meaning neither the client, nor the server, tracks the state of the communication session. If reliability is required when using UDP as the transport protocol, it must be handled by the application.

One of the most important requirements for delivering live video and voice over the network is that the data continues to flow quickly. Live video and voice applications can tolerate some data loss with minimal or no noticeable effect, and are perfectly suited to UDP.

The blocks of communication in UDP are called datagrams, or segments. These datagrams are sent as best effort by the transport layer protocol.

The UDP header is far simpler than the TCP header because it only has four fields and requires 8 bytes (i.e., 64 bits). The figure shows the fields in a UDP header.

UDP datagram diagram shows 4 header fields: source port, destination port, length, and checksum as well as the non header application layer data

8 Bytes

14.3.3

UDP Header Fields

The table identifies and describes the four fields in a UDP header.

UDP Header Field DescriptionSource Port A 16-bit field used to identify the source application by port number.Destination PortA 16-bit field used to identify the destination application by port number.Length A 16-bit field that indicates the length of the UDP datagram header.Checksum A 16-bit field used for error checking of the datagram header and data.

UDP Header Field	Description
Source Port	A 16-bit field used to identify the source application by port number.

UDP Header Field DescriptionSource Port A 16-bit field used to identify the source application by port number.Destination PortA 16-bit field used to identify the destination application by port number.Length A 16-bit field that indicates the length of the UDP datagram header.Checksum A 16-bit field used for error checking of the datagram header and data.

UDP Header Field	Description
Destination Port	A 16-bit field used to identify the destination application by port number.
Length	A 16-bit field that indicates the length of the UDP datagram header.
Checksum	A 16-bit field used for error checking of the datagram header and data.

14.3.4

Applications that use UDP

There are three types of applications that are best suited for UDP:

- **Live video and multimedia applications** - These applications can tolerate some data loss, but require little or no delay. Examples include VoIP and live streaming video.
- **Simple request and reply applications** - Applications with simple transactions where a host sends a request and may or may not receive a reply. Examples include DNS and DHCP.
- **Applications that handle reliability themselves** - Unidirectional communications where flow control, error detection, acknowledgments, and error recovery is not required, or can be handled by the application. Examples include SNMP and TFTP.

The figure identifies applications that require UDP.

shows arrows pointing both directions from DHCP, DNS, SNMP, TFTP, VoIP, and IPTV to UDP and then from UDP to IP

Although DNS and SNMP use UDP by default, both can also use TCP. DNS will use TCP if the DNS request or DNS response is more than 512 bytes, such as when a DNS response includes many name resolutions. Similarly, under some situations the network administrator may want to configure SNMP to use TCP.

14.4.1

Multiple Separate Communications

As you have learned, there are some situations in which TCP is the right protocol for the job, and other situations in which UDP should be used. No matter what type of data is being transported, both TCP and UDP use port numbers.

The TCP and UDP transport layer protocols use port numbers to manage multiple, simultaneous conversations. As shown in the figure, the TCP and UDP header fields identify a source and destination application port number.

shows the source port and destination port header fields which are 2 bytes each

The source port number is associated with the originating application on the local host whereas the destination port number is associated with the destination application on the remote host.

For instance, assume a host is initiating a web page request from a web server. When the host initiates the web page request, the source port number is dynamically generated by the host to uniquely identify the conversation. Each request generated by a host will use a different dynamically created source port number. This process allows multiple conversations to occur simultaneously.

In the request, the destination port number is what identifies the type of service being requested of the destination web server.. For example, when a client specifies port 80 in the destination port, the server that receives the message knows that web services are being requested.

A server can offer more than one service simultaneously such as web services on port 80 while it offers File Transfer Protocol (FTP) connection establishment on port 21.

14.4.2

Socket Pairs

The source and destination ports are placed within the segment. The segments are then encapsulated within an IP packet. The IP packet contains the IP address of the source and destination. The combination of the source IP address and source port number, or the destination IP address and destination port number is known as a socket.

In the example in the figure, the PC is simultaneously requesting FTP and web services from the destination server.

The figure depicts a PC making both an FTP connection and a web connection to a server. The requests have source and destination port numbers which identify the host PC and the requested application service respectively.

In the example, the FTP request generated by the PC includes the Layer 2 MAC addresses and the Layer 3 IP addresses. The request also identifies the source port number 1305 (i.e., dynamically generated by the host) and destination port, identifying the FTP services on port 21. The host also has requested a web page from the server using the same Layer 2 and Layer 3 addresses. However, it is using the source port number 1099 (i.e., dynamically generated by the host) and destination port identifying the web service on port 80.

The socket is used to identify the server and service being requested by the client. A client socket might look like this, with 1099 representing the source port number: 192.168.1.5:1099

The socket on a web server might be 192.168.1.7:80

Together, these two sockets combine to form a *socket pair*: 192.168.1.5:1099, 192.168.1.7:80

Sockets enable multiple processes, running on a client, to distinguish themselves from each other, and multiple connections to a server process to be distinguished from each other.

The source port number acts as a return address for the requesting application. The transport layer keeps track of this port and the application that initiated the request so that when a response is returned, it can be forwarded to the correct application.

14.4.3

Port Number Groups

The Internet Assigned Numbers Authority (IANA) is the standards organization responsible for assigning various addressing standards, including the 16-bit port numbers. The 16 bits used to identify the source and destination port numbers provides a range of ports from 0 through 65535.

The IANA has divided the range of numbers into the following three port groups.

Port GroupNumber RangeDescriptionWell-known Ports0 to 1,023These port numbers are reserved for common or popular servi and applications such as web browsers, email clients, and remote access clients. Defined well-known ports for common serve applications enables clients to easily identify the associated service required.Registered Ports1,024 to 49,151These port numbe are assigned by IANA to a requesting entity to use with specific processes or applications. These processes are primarily individ applications that a user has chosen to install, rather than common applications that would receive a well-known port number. F example, Cisco has registered port 1812 for its RADIUS server authentication process.Private and/or Dynamic Ports49,152 to 65,535These ports are also known as ephemeral ports.The client's OS usually assign port numbers dynamically when a connectio a service is initiated. The dynamic port is then used to identify the client application during communication.

Port Group	Number Range	Description
Well-known Ports	**0 to 1,023**	• These port numbers are reserved for common or popular services and applications such as web browsers, email clients, and remote access clients. • Defined well-known ports for common server applications enables clients to easily identify the associated service required.
Registered Ports	**1,024 to 49,151**	• These port numbers are assigned by IANA to a requesting entity to use with specific processes or applications. • These processes are primarily individual applications that a user has chosen install, rather than common applications that would receive a well-known po number. • For example, Cisco has registered port 1812 for its RADIUS server authentica process.
Private and/or **Dynamic Ports**	**49,152 to 65,535**	• These ports are also known as *ephemeral ports*. • The client's OS usually assign port numbers dynamically when a connection t service is initiated. • The dynamic port is then used to identify the client application during communication.

Note: Some client operating systems may use registered port numbers instead of dynamic port numbers for assigning source ports.

The table displays some common well-known port numbers and their associated applications.

Well-Known Port Numbers

Port NumberProtocolApplication20TCPFile Transfer Protocol (FTP) - Data 21TCPFile Transfer Protocol (FTP) - Control 22TCPSecure Shell (SSH)23TCPTelnet25TCPSimple Mail Transfer Protocol (SMTP)53UDP, TCPDomain Name Service (DNS) 67UDPDynamic Host Configuration Protocol (DHCP) - Server68UDPDynamic Host Configuration Protocol - Client 69UDPTrivial File Transfer Protocol (TFTP)80TCPHypertext Transfer Protocol (HTTP)110TCPPost Office Protocol version 3 (POP3)143TCPInternet Message Access Protocol (IMAP)161UDPSimple Network Management Protocol (SNMP) 443TCPHypertext Transfer Protocol Secure (HTTPS)

Port Number	Protocol	Application
20	TCP	File Transfer Protocol (FTP) - Data
21	TCP	File Transfer Protocol (FTP) - Control
22	TCP	Secure Shell (SSH)
23	TCP	Telnet
25	TCP	Simple Mail Transfer Protocol (SMTP)
53	UDP, TCP	Domain Name System (DNS)

Port NumberProtocolApplication20TCPFile Transfer Protocol (FTP) - Data 21TCPFile Transfer Protocol (FTP) - Control 22TCPSecure Shell (SSH)23TCPTelnet25TCPSimple Mail Transfer Protocol (SMTP)53UDP, TCPDomain Name Service (DNS) 67UDPDynamic Host Configuration Protocol (DHCP) - Server68UDPDynamic Host Configuration Protocol - Client 69UDPTrivial File Transfer Protocol (TFTP)80TCPHypertext Transfer Protocol (HTTP)110TCPPost Office Protocol version 3 (POP3)143TCPInternet Message Access Protocol (IMAP)161UDPSimple Network Management Protocol (SNMP) 443TCPHypertext Transfer Protocol Secure (HTTPS)

Port Number	Protocol	Application
67	UDP	Dynamic Host Configuration Protocol (DHCP) - Server
68	UDP	Dynamic Host Configuration Protocol - Client
69	UDP	Trivial File Transfer Protocol (TFTP)
80	TCP	Hypertext Transfer Protocol (HTTP)
110	TCP	Post Office Protocol version 3 (POP3)
143	TCP	Internet Message Access Protocol (IMAP)
161	UDP	Simple Network Management Protocol (SNMP)
443	TCP	Hypertext Transfer Protocol Secure (HTTPS)

Some applications may use both TCP and UDP. For example, DNS uses UDP when clients send requests to a DNS server. However, communication between two DNS servers always uses TCP.

Search the IANA website for port registry to view the full list of port numbers and associated applications.

14.4.4

The netstat Command

Unexplained TCP connections can pose a major security threat. They can indicate that something or someone is connected to the local host. Sometimes it is necessary to know which active TCP connections are open and running on a networked host. Netstat is an important network utility that can be used to verify those connections. As shown below, enter the command **netstat** to list the protocols in use, the local address and port numbers, the foreign address and port numbers, and the connection state.

```
C:\> netstat
```

```
Active Connections
```

```
  Proto  Local Address          Foreign Address        State
```

```
  TCP    192.168.1.124:3126     192.168.0.2:netbios-ssn    ESTABLISHED
```

```
  TCP    192.168.1.124:3158     207.138.126.152:http       ESTABLISHED
```

```
  TCP    192.168.1.124:3159     207.138.126.169:http       ESTABLISHED

  TCP    192.168.1.124:3160     207.138.126.169:http       ESTABLISHED

  TCP    192.168.1.124:3161     sc.msn.com:http            ESTABLISHED

  TCP    192.168.1.124:3166     www.cisco.com:http         ESTABLISHED

(output omitted)

C:\>
```

By default, the **netstat** command will attempt to resolve IP addresses to domain names and port numbers to well-known applications. The **-n** option can be used to display IP addresses and port numbers in their numerical form.

14.5.1

TCP Server Processes

You already know the fundamentals of TCP. Understanding the role of port numbers will help you to grasp the details of the TCP communication process. In this topic, you will also learn about the TCP three-way handshake and session termination processes.

Each application process running on a server is configured to use a port number. The port number is either automatically assigned or configured manually by a system administrator.

An individual server cannot have two services assigned to the same port number within the same transport layer services. For example, a host running a web server application and a file transfer application cannot have both configured to use the same port, such as TCP port 80.

An active server application assigned to a specific port is considered open, which means that the transport layer accepts, and processes segments addressed to that port. Any incoming client request addressed to the correct socket is accepted, and the data is passed to the server application. There can be many ports open simultaneously on a server, one for each active server application.

Select each tab for more information about TCP server processes.
Clients Sending TCP Requests

Request Destination Ports

Request Source Ports

Response Destination Ports

Response Source Ports

Clients Sending TCP Requests

Client 1 is requesting web services and Client 2 is requesting email service of the same sever.

HTTP Request:
Source Port: 49152
Destination Port: 80SMTP Request:
Source Port: 51152
Destination Port: 25 ServerClient 1Client 2 HTTP: Port 80
SMTP: Port 25 Client requests to TCP server

14.5.2

TCP Connection Establishment

In some cultures, when two persons meet, they often greet each other by shaking hands. Both parties understand the act of shaking hands as a signal for a friendly greeting. Connections on the network are similar. In TCP connections, the host client establishes the connection with the server using the three-way handshake process.

Select each tab for more information about each TCP connection establishment step.
Step 1. SYN

Step 2. ACK and SYN

Step 3. ACK

Step 1. SYN

The initiating client requests a client-to-server communication session with the server.

PCA initiates a three way handshake by sending a syn segment to PCB.

1AB

Send SYN(SEQ=100 CTL=SYN)SYN receivedA sends SYN request to B

The three-way handshake validates that the destination host is available to communicate. In this example, host A has validated that host B is available.

14.5.3

Session Termination

To close a connection, the Finish (FIN) control flag must be set in the segment header. To end each one-way TCP session, a two-way handshake, consisting of a FIN segment and an Acknowledgment (ACK) segment, is used. Therefore, to terminate a single conversation supported by TCP, four exchanges are needed to end both sessions. Either the client or the server can initiate the termination.

In the example, the terms client and server are used as a reference for simplicity, but any two hosts that have an open session can initiate the termination process.

Select each tab for more information about the session termination steps.
Step 1. FIN

Step 2. ACK

Step 3. FIN

Step 4. ACK

Step 1. FIN

When the client has no more data to send in the stream, it sends a segment with the FIN flag set.

PCA sends a fin segment to PCB to end the session when there is no more data to send

1AB

Send FINFIN receivedA sends FIN request to B.

When all segments have been acknowledged, the session is closed.

14.5.4

TCP Three-way Handshake Analysis

Hosts maintain state, track each data segment within a session, and exchange information about what data is received using the information in the TCP header. TCP is a full-duplex protocol, where each connection represents two one-way communication sessions. To establish the connection, the hosts perform a three-way handshake. As shown in the figure, control bits in the TCP header indicate the progress and status of the connection.

These are the functions of the three-way handshake:

- It establishes that the destination device is present on the network.
- It verifies that the destination device has an active service and is accepting requests on the destination port number that the initiating client intends to use.

- It informs the destination device that the source client intends to establish a communication session on that port number.

After the communication is completed the sessions are closed, and the connection is terminated. The connection and session mechanisms enable TCP reliability function.

shows the tcp segment header fields with the control bits field of 6 bits highlighted

Control Bits Field

20 Bytes

The six bits in the Control Bits field of the TCP segment header are also known as flags. A flag is a bit that is set to either on or off.

The six control bits flags are as follows:

- **URG** - Urgent pointer field significant
- **ACK** - Acknowledgment flag used in connection establishment and session termination
- **PSH** - Push function
- **RST** - Reset the connection when an error or timeout occurs
- **SYN** - Synchronize sequence numbers used in connection establishment
- **FIN** - No more data from sender and used in session termination

Search the internet to learn more about the PSH and URG flags.

14.6.1

TCP Reliability - Guaranteed and Ordered Delivery

The reason that TCP is the better protocol for some applications is because, unlike UDP, it resends dropped packets and numbers packets to indicate their proper order before delivery. TCP can also help maintain the flow of packets so that devices do not become overloaded. This topic covers these features of TCP in detail.

There may be times when TCP segments do not arrive at their destination. Other times, the TCP segments might arrive out of order. For the original message to be understood by the recipient, all the data must be received and the data in these segments must be reassembled into the original order. Sequence numbers are assigned in the header of each packet to achieve this goal. The sequence number represents the first data byte of the TCP segment.

During session setup, an initial sequence number (ISN) is set. This ISN represents the starting value of the bytes that are transmitted to the receiving application. As data is transmitted during the session, the sequence number is incremented by the number of bytes that have been transmitted. This data byte tracking enables each segment to be uniquely identified and acknowledged. Missing segments can then be identified.

The ISN does not begin at one but is effectively a random number. This is to prevent certain types of malicious attacks. For simplicity, we will use an ISN of 1 for the examples in this chapter.

Segment sequence numbers indicate how to reassemble and reorder received segments, as shown in the figure.

shows that even though segments may take different routes and arrive out of order at the destination, TCP has the ability to reorder the segments

TCP Segments Are Reordered at the Destination

Having taken different routes to the destination, segments arrive out of order.TCP reorders the segments to the original order. Different segments may take different routes.

The receiving TCP process places the data from a segment into a receiving buffer. Segments are then placed in the proper sequence order and passed to the application layer when reassembled. Any segments that arrive with sequence numbers that are out of order are held for later processing. Then, when the segments with the missing bytes arrive, these segments are processed in order.

14.6.2

Video - TCP Reliability - Sequence Numbers and Acknowledgments

One of the functions of TCP is to ensure that each segment reaches its destination. The TCP services on the destination host acknowledge the data that have been received by the source application.

Click Play in the figure to view a lesson on TCP sequence numbers and acknowledgments.

Play Video

14.6.3

TCP Reliability - Data Loss and Retransmission

No matter how well designed a network is, data loss occasionally occurs. TCP provides methods of managing these segment losses. Among these is a mechanism to retransmit segments for unacknowledged data.

The sequence (SEQ) number and acknowledgement (ACK) number are used together to confirm receipt of the bytes of data contained in the transmitted segments. The SEQ number identifies the first byte of data in the segment being transmitted. TCP uses the ACK number sent back to the source to indicate the next byte that the receiver expects to receive. This is called expectational acknowledgement.

Prior to later enhancements, TCP could only acknowledge the next byte expected. For example, in the figure, using segment numbers for simplicity, host A sends segments 1 through 10 to host B. If all the segments arrive except for segments 3 and 4, host B would reply with acknowledgment specifying that the next segment expected is segment 3. Host A has no idea if any other segments arrived or not. Host A would, therefore, resend segments 3 through 10. If all the resent segments arrived successfully, segments 5 through 10 would be duplicates. This can lead to delays, congestion, and inefficiencies.

shows PCA sending 10 segments to PCB, but segments 3 and 4 fail to arrive. So starting with segment 3, PCA resends segments 3 through 10, even though PCB only needed segments 3 and 4

ABAB

Segment 1Segment 2Segment 3Segment 5Segment 6Segment 7Segment 8Segment 9Segment 10Segment 4Segment 3Segment 4Segment 5Segment 6Segment 7Segment 8Segment 9Segment 10ACK 3ACK 11Duplicate segments

Host operating systems today typically employ an optional TCP feature called selective acknowledgment (SACK), negotiated during the three-way handshake. If both hosts support SACK, the receiver can explicitly acknowledge which segments (bytes) were received including any discontinuous segments. The sending host would therefore only need to retransmit the missing data. For example, in the next figure, again using segment numbers for simplicity, host A sends segments 1 through 10 to host B. If all the segments arrive except for segments 3 and 4, host B can acknowledge that it has received segments 1 and 2 (ACK 3), and selectively acknowledge segments 5 through 10 (SACK 5-10). Host A would only need to resend segments 3 and 4.

show PCA sending 10 segments to PCB, but segments 3 and 4 fail to arrive. This time PCB sends an ack 3 and a sack 5-10 letting PCA know to resend missing segments 3 and 4 and the continue with segment 11

ABAB

Segment 1Segment 2Segment 3Segment 5Segment 6Segment 7Segment 8Segment 9Segment 10Segment 4Segment 3Segment 4Segment 11Segment 12ACK 3,SACK 5-10ACK 13

Note: TCP typically sends ACKs for every other packet, but other factors beyond the scope of this topic may alter this behavior.

TCP uses timers to know how long to wait before resending a segment. In the figure, play the video and click the link to download the PDF file. The video and PDF file examine TCP data loss and retransmission.

14.6.4

Video - TCP Reliability - Data Loss and Retransmission

Click Play in the figure to view a lesson on TCP retransmission.

Play Video

14.6.5

TCP Flow Control - Window Size and Acknowledgments

TCP also provides mechanisms for flow control. Flow control is the amount of data that the destination can receive and process reliably. Flow control helps maintain the reliability of TCP transmission by adjusting the rate of data flow between source and destination for a given session. To accomplish this, the TCP header includes a 16-bit field called the window size.

The figure shows an example of window size and acknowledgments.

shows PCB sending PCA a negotiated window size of 10,000 bytes and a maximum segment size of 1,460 bytes. PCA starts sending segments starting with sequence number 1. An acknowledgement from PCB can be sent without waiting until the window size is reached and the window size can be adjusted by PCA creating a sliding window

TCP Window Size Example

AB

MSS = Maximum Segment Size During three-way handshake Window size 10,000, MSS 1,460Sequence number 1,4611,460 bytesACK 2,921 Window size 10,0001,460 bytesACK 4,381
Window size 10,000Send window 10,000Sequence number 11,460 bytesReceive acknowledgement
Send window 12,920Sequence number 2,921Receive acknowledgement
Send window 14,380Receive 1 – 1,460Receive 1,461 – 2,920Receive 2,921 – 4,380

The window size determines the number of bytes that can be sent before expecting an acknowledgment. The acknowledgment number is the number of the next expected byte.

The window size is the number of bytes that the destination device of a TCP session can accept and process at one time. In this example, the PC B initial window size for the TCP session is 10,000 bytes. Starting with the first byte, byte number 1, the last byte PC A can send without receiving an acknowledgment is byte 10,000. This is known as the send window of PC A. The window size is included in every TCP segment so the destination can modify the window size at any time depending on buffer availability.

The initial window size is agreed upon when the TCP session is established during the three-way handshake. The source device must limit the number of bytes sent to the destination device based on the window size of the destination. Only after the source device receives an acknowledgment that the bytes have been received, can it continue sending more data for the session. Typically, the destination will not wait for all the bytes for its window size to be received before replying with an acknowledgment. As the bytes are received and processed, the destination will send acknowledgments to inform the source that it can continue to send additional bytes.

For example, it is typical that PC B would not wait until all 10,000 bytes have been received before sending an acknowledgment. This means PC A can adjust its send window as it receives acknowledgments from PC B. As shown in the figure, when PC A receives an acknowledgment with the acknowledgment number 2,921, which is the next expected byte. The PC A send window will increment 2,920 bytes. This changes the send window from 10,000 bytes to 12,920. PC A can now continue to send up to another 10,000 bytes to PC B as long as it does not send more than its new send window at 12,920.

A destination sending acknowledgments as it processes bytes received, and the continual adjustment of the source send window, is known as sliding windows. In the previous example, the send window of PC A increments or slides over another 2,921 bytes from 10,000 to 12,920.

If the availability of the destination's buffer space decreases, it may reduce its window size to inform the source to reduce the number of bytes it should send without receiving an acknowledgment.

Note: Devices today use the sliding windows protocol. The receiver typically sends an acknowledgment after every two segments it receives. The number of segments received before being acknowledged may vary. The advantage of sliding windows is that it allows the sender to continuously transmit segments, as long as the receiver is acknowledging previous segments. The details of sliding windows are beyond the scope of this course.

14.6.6

TCP Flow Control - Maximum Segment Size (MSS)

In the figure, the source is transmitting 1,460 bytes of data within each TCP segment. This is typically the Maximum Segment Size (MSS) that the destination device can receive. The MSS is part of the options field in the TCP header that specifies the largest amount of data, in bytes, that a device can receive in a single TCP segment. The MSS size does not include the TCP header. The MSS is typically included during the three-way handshake.

shows the same diagram as before but the emphasis is on the MSS of maximum segment size of 1460

AB

MSS = Maximum Segment Size During three-way handshake Window size 10,000, MSS 1,460Sequence number 1,4611,460 bytesACK 2,921 Window size 10,0001,460 bytesACK 4,381
Window size 10,000Send window 10,000Sequence number 11,460 bytesReceive acknowledgement
Send window 12,920Sequence number 2,921Receive acknowledgement
Send window 14,380Receive 1 – 1,460Receive 1,461 – 2,920Receive 2,921 – 4,380

A common MSS is 1,460 bytes when using IPv4. A host determines the value of its MSS field by subtracting the IP and TCP headers from the Ethernet maximum transmission unit (MTU). On an Ethernet interface, the default MTU is 1500 bytes. Subtracting the IPv4 header of 20 bytes and the TCP header of 20 bytes, the default MSS size will be 1460 bytes, as shown in the figure.

shows a diagram of an entire Ethernet frame of which the MTU is 1500 bytes, with 20 bytes being the IP header, and 20 bytes being the TCP header, this leaves 1460 bytes which is the TCP maximum segment size MSS

Ethernet MTUIP MTU20 bytes20 bytes1460 bytes1500 bytesTCP MSS

14.6.7

TCP Flow Control - Congestion Avoidance

When congestion occurs on a network, it results in packets being discarded by the overloaded router. When packets containing TCP segments do not reach their destination, they are left unacknowledged. By determining the rate at which TCP segments are sent but not acknowledged, the source can assume a certain level of network congestion.

Whenever there is congestion, retransmission of lost TCP segments from the source will occur. If the retransmission is not properly controlled, the additional retransmission of the TCP segments can make the congestion even worse. Not only are new packets with TCP segments introduced into the network, but the feedback effect of the retransmitted TCP segments that were lost will also add to the congestion. To avoid and control congestion, TCP employs several congestion handling mechanisms, timers, and algorithms.

If the source determines that the TCP segments are either not being acknowledged or not acknowledged in a timely manner, then it can reduce the number of bytes it sends before receiving an acknowledgment. As illustrated in the figure, PC A senses there is congestion and therefore, reduces the number of bytes it sends before receiving an acknowledgment from PC B.

shows PCA sending segments to PCB where lost segments and retransmission can cause congestion

TCP Congestion Control

AB

TCP segment 1TCP segment 2TCP segment 3TCP segment 4Acknowledgement segment 2TCP segment 2TCP segment 3 I'm not getting the acknowledgments I expect from PC B so I will reduce the number of bytes I send before getting an acknowledgement.

Acknowledgment numbers are for the next expected byte and not for a segment. The segment numbers used are simplified for illustration purposes.

Notice that it is the source that is reducing the number of unacknowledged bytes it sends and not the window size determined by the destination.

Note: Explanations of actual congestion handling mechanisms, timers, and algorithms are beyond the scope of this course.

14.7.1

UDP Low Overhead versus Reliability

As explained before, UDP is perfect for communications that need to be fast, like VoIP. This topic explains in detail why UDP is perfect for some types of transmissions. As shown in the figure, UDP does not establish a connection. UDP provides low overhead data transport because it has a small datagram header and no network management traffic.

shows a host sender needing to send voice and video data which is sent with UDP which requires no prior negotiated connection

NetworkSenderReceiverData UDP does not establish a connection before sending data.

14.7.2

UDP Datagram Reassembly

Like segments with TCP, when UDP datagrams are sent to a destination, they often take different paths and arrive in the wrong order. UDP does not track sequence numbers the way TCP does. UDP has no way to reorder the datagrams into their transmission order, as shown in the figure.

Therefore, UDP simply reassembles the data in the order that it was received and forwards it to the application. If the data sequence is important to the application, the application must identify the proper sequence and determine how the data should be processed.

shows UDP datagrams being sent in order but arriving out of order due to the possibility of different routes to reach the destination

UDP: Connectionless and Unreliable

Having taken different routes to the destination, datagrams arrive out of order.Out of order datagrams are not re-ordered. Different datagrams may take different routes. Lost datagrams are not re-sent.

14.7.3

UDP Server Processes and Requests

Like TCP-based applications, UDP-based server applications are assigned well-known or registered port numbers, as shown in the figure. When these applications or processes are running on a server, they accept the data matched with the assigned port number. When UDP receives a datagram destined for one of these ports, it forwards the application data to the appropriate application based on its port number.

shows that a RADIUS server application uses UDP to listen for requests on port 53

UDP Server Listening for Requests

ServerClient 1Client 2**Server Applications**
Client DNS requests will be received on Port 53.

Client RADIUS requests will be received on Port 1812.DNS requestRADIUS request

Note: The Remote Authentication Dial-in User Service (RADIUS) server shown in the figure provides authentication, authorization, and accounting services to manage user access. The operation of RADIUS is beyond the scope for this course.

14.7.4

UDP Client Processes

As with TCP, client-server communication is initiated by a client application that requests data from a server process. The UDP client process dynamically selects a port number from the range of port numbers and uses this as the source port for the conversation. The destination port is usually the well-known or registered port number assigned to the server process.

After a client has selected the source and destination ports, the same pair of ports are used in the header of all datagrams in the transaction. For the data returning to the client from the server, the source and destination port numbers in the datagram header are reversed.

Select each tab for an illustration of two hosts requesting services from the DNS and RADIUS authentication server.

Clients Sending UDP Requests

UDP Request Destination Ports

UDP Request Source Ports

UDP Response Destination

UDP Response Source Ports

Clients Sending UDP Requests

Client 1 is sending a DNS request while Client 2 is requesting RADIUS authentication services of the same server.

Two different PC clients need to make a request to a DNS server

Client 1 DNS Request:
Source Port 49152
Destination Port 53Client 2 RADIUS User Authentication Request:
Source Port 51152
Destination Port 1812 Client 1ServerClient 2 DNS: Port 53
RADIUS: Port 1812

15.1.1

Application Layer

In the OSI and the TCP/IP models, the application layer is the closest layer to the end user. As shown in the figure, it is the layer that provides the interface between the applications used to communicate, and the underlying network over which messages are transmitted. Application layer protocols are used to exchange data between programs running on the source and destination hosts.

The figure is a comparision of the OSI and TCP/IP model layers. The OSI model is shown on the left. From top to bottom are the following layer numbers and names: 7) Application, 6) Presentation, 5) Session, 4) Transport, 3) Network, 2) Data Link, and 1) Physical. The TCP/IP Model is shown on the right. From top to bottom the layer names and the associated OSI model layer numbers are: Application (OSI Layers 7, 6, and 5), Transport (OSI Layer 4), Internet (OSI Layer 3), and Network Access (OSI Layers 2 and 1). Text at the bottom reads: The key similarities are in the transport and network layers; however, the two models differ in how they relate to the layers above and below each layer:OSI Layer 3, the network layer, maps directly to the TCP/IP internet layer. This layer is used to describe protocols that address and route messages through an internetwork. OSI Layer 4, the transport layer, maps directly to the TCP/IP transport layer. This layer describes general services and functions that provide ordered and reliable delivery of data between source and destination hosts. The TCP/IP application layer includes several protocols that provide specific functionality to a variety of end user applications. The OSI model Layers 5, 6, and 7 are used as references for application software developers and vendors to produce applications that operate on networks. Both the TCP/IP and OSI models are commonly used when referring to protocols at various layers. Because the OSI model separates the data link layer from the physical layer, it is commonly used when referring to these lower layers. The Application layer ofthe OSI model specifically identifies Domain Name System, Hypertest Transfer Protocol, Simple Mail Tansfer Protocol, Post Office Protocol, Dynamic Host Configuration Protocol, FIle transfer Protocol, and Internet Message Access Protocol.

OSI Model **TCP/IP**
Model
Application
LayersData Flow
Layers Domain Name System

Hypertext Transfer Protocol

Simple Mail Transfer Protocol

Post Office Protocol

Dynamic Host Configuration Protocol

File Transfer Protocol

Internet Message Access Protocol

Based on the TCP/IP model, the upper three layers of the OSI model (application, presentation, and session) define functions of the TCP/IP application layer.

There are many application layer protocols, and new protocols are always being developed. Some of the most widely known application layer protocols include Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP), Internet Message Access Protocol (IMAP), and Domain Name System (DNS) protocol.

15.1.2

Presentation and Session Layer

Presentation Layer

The presentation layer has three primary functions:

- Formatting, or presenting, data at the source device into a compatible format for receipt by the destination device.
- Compressing data in a way that can be decompressed by the destination device.
- Encrypting data for transmission and decrypting data upon receipt.

As shown in the figure, the presentation layer formats data for the application layer, and it sets standards for file formats. Some well-known standards for video include Matroska Video (MKV), Motion Picture Experts Group (MPG), and QuickTime Video (MOV). Some well-known graphic image formats are Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPG), and Portable Network Graphics (PNG) format.

The figure is a comparision of the OSI and TCP/IP model layers. The OSI model is shown on the left. From top to bottom are the following layer numbers and names: 7) Application, 6) Presentation, 5) Session, 4) Transport, 3) Network, 2) Data Link, and 1) Physical. The TCP/IP Model is shown on the right. From top to bottom the layer names and the associated OSI model layer numbers are: Application (OSI Layers 7, 6, and 5), Transport (OSI Layer 4), Internet (OSI Layer 3), and Network Access (OSI Layers 2 and 1). Text at the bottom reads: The key similarities are in the transport and network layers; however, the two models differ in how they relate to the layers above and below each layer:OSI Layer 3, the network layer, maps directly to the TCP/IP internet layer. This layer is used to describe protocols that address and route messages through an internetwork. OSI Layer 4, the transport layer, maps directly to the TCP/IP transport layer. This layer describes general services and functions that provide ordered and reliable delivery of data between source and destination hosts. The TCP/IP application layer includes several protocols that provide specific functionality to a variety of end user applications. The OSI model Layers 5, 6, and 7 are used as references for application software developers and vendors to produce applications that operate on networks. Both the TCP/IP and OSI models are commonly used when referring to protocols at various layers. Because the OSI model separates the data link layer from the physical layer, it is commonly used when referring to these lower layers. The Presentation layer specifically identifies Matroska Video(MKV), Motion Pictures Expert Group(MPG), QuickTime(MOV), Graphics Interchange Format(GIF), Joint Photographic Experts Group(JPG), and Portable Network Graphics(PNG).

OSI Model **TCP/IP**
Model Application
Application
Layers Data Flow
Layers Matroska video (MKV)

Motion Pictures Expert Group (MPG)

Quick Time (MOV)

Graphics Interchange Format (GIF)

Joint Photographic Experts Group (JPG)

Portable Nerwork Graphics (PNG)

Session Layer

As the name implies, functions at the session layer create and maintain dialogs between source and destination applications. The session layer handles the exchange of information to initiate dialogs, keep them active, and to restart sessions that are disrupted or idle for a long period of time.

15.1.3

TCP/IP Application Layer Protocols

The TCP/IP application protocols specify the format and control information necessary for many common internet communication functions. Application layer protocols are used by both the source and destination devices during a communication session. For the communications to be successful, the application layer protocols that are implemented on the source and destination host must be compatible.

Click each application protocol type to learn more about each protocol.

Name System

Host Config

Email

File Transfer

Web

Name System

DNS - Domain Name System (or Service)

- TCP, UDP 53
- Translates domain names, such as cisco.com, into IP addresses.

15.2.1

Client-Server Model

In the previous topic, you learned that TCP/IP application layer protocols implemented on both the source and destination host must be compatible. In this topic you will learn about the client/server model and the processes used, which are in the application layer. The same is true for a peer-to-peer network. In the client/server model, the device requesting the information is called a client and the device responding to the request is called a server. The client is a hardware/software combination that people use to directly access the resources that are stored on the server.

Client and server processes are considered to be in the application layer. The client begins the exchange by requesting data from the server, which responds by sending one or more streams of data to the client. Application layer protocols describe the format of the requests and responses between clients and servers. In addition to the actual data transfer, this exchange may also require user authentication and the identification of a data file to be transferred.

One example of a client/server network is using the email service of an ISP to send, receive, and store email. The email client on a home computer issues a request to the email server of the ISP for any unread mail. The server responds by sending the requested email to the client. Data transfer from a client to a server is referred to as an upload and data from a server to a client as a download.

As shown in the figure, files are downloaded from the server to the client.

The figure depicts the client server model. A client which is depicted as a cell phone, computer, or VoIP phone is connected to a server and downloading files from the server.

DownloadNetworkClientServer

15.2.2

Peer-to-Peer Networks

In the peer-to-peer (P2P) networking model, the data is accessed from a peer device without the use of a dedicated server.

The P2P network model involves two parts: P2P networks and P2P applications. Both parts have similar features, but in practice work quite differently.

In a P2P network, two or more computers are connected via a network and can share resources (such as printers and files) without having a dedicated server. Every connected end device (known as a peer) can function as both a server and a client. One computer might assume the role of server for one transaction while simultaneously serving as a client for another. The roles of client and server are set on a per request basis.

In addition to sharing files, a network such as this one would allow users to enable networked games or share an internet connection.

In a peer-to-peer exchange, both devices are considered equal in the communication process. Peer 1 has files that are shared with Peer 2 and can access the shared printer that is directly connected to Peer 2 to print files. Peer 2 is sharing the directly connected printer with Peer 1 while accessing the shared files on Peer 1, as shown in the figure.

The figure depicts the peer to peer model. Two computers are connected by a switch and are communicating directly with each other. In addition a printer is shared by one of the computers and can be accessed by either computer in the figure.

Peer 1Print Client
File ServerPeer 2Directly connected printerPrinterPrint Server
File Client

15.2.3

Peer-to-Peer Applications

A P2P application allows a device to act as both a client and a server within the same communication, as shown in the figure. In this model, every client is a server and every server is a client. P2P applications require that each end device provide a user interface and run a background service.

Some P2P applications use a hybrid system where resource sharing is decentralized, but the indexes that point to resource locations are stored in a centralized directory. In a hybrid system, each peer accesses an index server to get the location of a resource stored on another peer.

The figure depicts a Instant Message conversation with two machines communicating with each other through the network. Each machine is acting as both a client and server. The caption states Both clients can simultaneously send and receive messages.

Good.I'll be there.Meeting tonight.I'll be there.Good.Meeting tonight.**Client and Server**Network**Client and Server**

Both clients can simultaneously send and receive messages.

15.2.4

Common P2P Applications

With P2P applications, each computer in the network that is running the application can act as a client or a server for the other computers in the network that are also running the application. Common P2P networks include the following:

- BitTorrent
- Direct Connect
- eDonkey
- Freenet

Some P2P applications are based on the Gnutella protocol, where each user shares whole files with other users. As shown in the figure, Gnutella-compatible client software allows users to connect to Gnutella services over the internet, and to locate and access resources shared by other Gnutella peers. Many Gnutella client applications are available, including µTorrent, BitComet, DC++, Deluge, and emule.

The figure shows a P2P application searching for shared resources. The P2papplication is asking its pers if the have the resource in this case mysong.mp3.

Where is: mysong.mp3I've got it.I've got it. Network I've got it.

Gnutella P2P applications search for shared resources on multiple peers.

Many P2P applications allow users to share pieces of many files with each other at the same time. Clients use a torrent file to locate other users who have pieces that they need so that they can then connect directly to them. This file also contains information about tracker computers that keep track of which users have specific pieces of certain files. Clients ask for pieces from multiple users at the same time. This is known as a swarm and the technology is called BitTorrent. BitTorrent has its own client. But there are many other BitTorrent clients including uTorrent, Deluge, and qBittorrent.

Note: Any type of file can be shared between users. Many of these files are copyrighted, meaning that only the creator has the right to use and distribute them. It is against the law to download or distribute copyrighted files without permission from the copyright holder. Copyright violation can result in criminal charges and civil lawsuits.

15.3.1

Hypertext Transfer Protocol and Hypertext Markup Language

There are application layer-specific protocols that are designed for common uses such as web browsing and email. The first topic gave you an overview of these protocols. This topic goes into more detail.

When a web address or Uniform Resource Locator (URL) is typed into a web browser, the web browser establishes a connection to the web service. The web service is running on the server that is using the HTTP protocol. URLs and Uniform Resource Identifiers (URIs) are the names most people associate with web addresses.

To better understand how the web browser and web server interact, examine how a web page is opened in a browser. For this example, use the http://www.cisco.com/index.html URL.

Click each button for more information.

Step 1

Step 2

Step 3

Step 4

Step 1

The browser interprets the three parts of the URL:

- http (the protocol or scheme)
- www.cisco.com (the server name)
- index.html (the specific filename requested)

shows a http server connected thru the internet to a client computer. The client computer is requesting the url www.cisco.com which is the HTTP server.

HTTP ServerNetworkClienthttp://www.cisco.com/index.html

15.3.2

HTTP and HTTPS

HTTP is a request/response protocol. When a client, typically a web browser, sends a request to a web server, HTTP specifies the message types used for that communication. The three common message types are GET (see figure), POST, and PUT:

- **GET** - This is a client request for data. A client (web browser) sends the GET message to the web server to request HTML pages.
- **POST** - This uploads data files to the web server, such as form data.
- **PUT** - This uploads resources or content to the web server, such as an image.

The figure depicts a client performing a HTTP request to a HTTP server. THe file requested is a Fully Qualified Domain Name. The request uses a Get to retrieve the web page. The URL field is shown on the client computer as a http://www.cisco.com request.

HTTP ServerClient Uniform Resource Locator (URL)HTTP RequestFile Requested Host : www.cisco.com

GET /index.html HTTP/1.1 Fully qualified domain name

Although HTTP is remarkably flexible, it is not a secure protocol. The request messages send information to the server in plaintext that can be intercepted and read. The server responses, typically HTML pages, are also unencrypted.

For secure communication across the internet, the HTTP Secure (HTTPS) protocol is used. HTTPS uses authentication and encryption to secure data as it travels between the client and server. HTTPS uses the same client request-server response process as HTTP, but the data stream is encrypted with Transport Layer Security (TLS) or its predecessor Secure Socket Layer (SSL) before being transported across the network.

15.3.3

Email Protocols

One of the primary services offered by an ISP is email hosting. To run on a computer or other end device, email requires several applications and services, as shown in the figure. Email is a store-and-forward method of sending, storing, and retrieving electronic messages across a network. Email messages are stored in databases on mail servers.

The figure depicts an email transaction from a sender using the SMTP protocol sending an email to recipient@cisco.com through an I S P mail server A arriving at the recipients I S P mail server B and the recipient reading the email using either an I MAP or POP protocol.

InternetIMAP or POP3SMTPISP AISP BSenderRecipientMail ServerMail ServerSMTP Send to recipient@cisco.com

Email clients communicate with mail servers to send and receive email. Mail servers communicate with other mail servers to transport messages from one domain to another. An email client does not communicate directly with another email client when sending email. Instead, both clients rely on the mail server to transport messages.

Email supports three separate protocols for operation: Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), and IMAP. The application layer process that sends mail uses SMTP. A client retrieves email using one of the two application layer protocols: POP or IMAP.

15.3.4

SMTP, POP, and IMAP

Click each button for more information.

SMTP

POP

IMAP

SMTP

SMTP message formats require a message header and a message body. Although the message body can contain any amount of text, the message header must have a properly formatted recipient email address and a sender address.

When a client sends email, the client SMTP process connects with a server SMTP process on well-known port 25. After the connection is made, the client attempts to send the email to the server across the connection. When the server receives the message, it either places the message in a local account, if the recipient is local, or forwards the message to another mail server for delivery.

The destination email server may not be online, or may be busy, when email messages are sent. Therefore, SMTP spools messages to be sent at a later time. Periodically, the server checks the queue for messages and attempts to send them again. If the message is still not delivered after a predetermined expiration time, it is returned to the sender as undeliverable.

This is a figure with a sender computer and recipient computer. Two SMTP/POP3 servers are connected between the two. A mail message is sent from the sender computer labeled recipient@domain.com using smtp protocol. The first SMTP/POP3 receives the message from the sender and asks Is the recipient in my list of recipients? No. forward email to another server. The second SMTP/POP3 server receives the message via the SMTP protocol and forwards the message to the recipient.

Recipientrecipient@domain.comSend EmailSenderSMTP/POP3
ServerSMTP/POP3
Server Is the recipient in my list of recipients ?

No.

Forward email to another server. SMTPSMTP

15.4.1

Domain Name System

There are other application layer-specific protocols that were designed to make it easier to obtain addresses for network devices. These services are essential because it would be very time consuming to remember IP addresses instead of URLs or manually configure all of the devices in a medium to large network. The first topic in this module gave you an overview of these protocols. This topic goes into more detail about the IP addressing services, DNS and DHCP.

In data networks, devices are labeled with numeric IP addresses to send and receive data over networks. Domain names were created to convert the numeric address into a simple, recognizable name.

On the internet, fully-qualified domain names (FQDNs), such as http://www.cisco.com, are much easier for people to remember than 198.133.219.25, which is the actual numeric address for this server. If Cisco decides to change the numeric address of www.cisco.com, it is transparent to the user because the domain name remains the same. The new address is simply linked to the existing domain name and connectivity is maintained.

The DNS protocol defines an automated service that matches resource names with the required numeric network address. It includes the format for queries, responses, and data. The DNS protocol communications use a single format called a message. This message format is used for all types of client queries and server responses, error messages, and the transfer of resource record information between servers.

Select each tab for more information.
Step 1

Step 2

Step 3

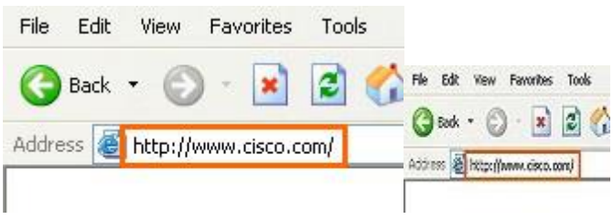
Step 4

Step 5

Step 1

The user types an FQDN into a browser application Address field.

this is a figure with a client contacting a DNS sever thru the network with a FQDN typed in a browser URL field because the name of a website is easier for people to used



The name is easy for people to use.DNS ServerNetworkClient

15.4.2

DNS Message Format

The DNS server stores different types of resource records that are used to resolve names. These records contain the name, address, and type of record. Some of these record types are as follows:

- **A** - An end device IPv4 address
- **NS** - An authoritative name server
- **AAAA** - An end device IPv6 address (pronounced quad-A)
- **MX** - A mail exchange record

When a client makes a query, the server DNS process first looks at its own records to resolve the name. If it is unable to resolve the name by using its stored records, it contacts other servers to resolve the name. After a match is found and returned to the original requesting server, the server temporarily stores the numbered address in the event that the same name is requested again.

The DNS cient service on Windows PCs also stores previously resolved names in memory. The **ipconfig /displaydns** command displays all of the cached DNS entries.

As shown in the table, DNS uses the same message format between servers, consisting of a question, answer, authority, and additional information for all types of client queries and server responses, error messages, and transfer of resource record information.

DNS message sectionDescriptionQuestionThe question for the name serverAnswerResource Records answering the questionAuthorityResource Records pointing toward an authorityAdditionalResource Records holding additional information	
DNS message section	**Description**
Question	The question for the name server
Answer	Resource Records answering the question
Authority	Resource Records pointing toward an authority
Additional	Resource Records holding additional information

15.4.3

DNS Hierarchy

The DNS protocol uses a hierarchical system to create a database to provide name resolution, as shown in the figure. DNS uses domain names to form the hierarchy.

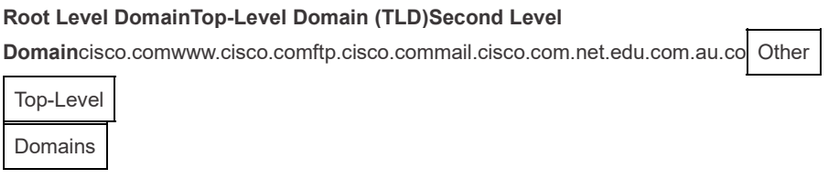
The naming structure is broken down into small, manageable zones. Each DNS server maintains a specific database file and is only responsible for managing name-to-IP mappings for that small portion of the entire DNS structure. When a DNS server receives a request for a name translation that is not within its DNS zone, the DNS

server forwards the request to another DNS server within the proper zone for translation. DNS is scalable because hostname resolution is spread across multiple servers.

The different top-level domains represent either the type of organization or the country of origin. Examples of top-level domains are the following:

- **.com** - a business or industry
- **.org** - a non-profit organization
- **.au** - Australia
- **.co** - Colombia

The figure shows the DNS Hierachy tree. At the top is the Root Level Domain with the TOp-Level Domains(TLD) connected undererneath the Root Level Domainmain. THe TLDs are .net, .edu, .com,.au, .co, and other top-level doamins. Under the .com TLD is the Second Level domain www.cisco.com and under cisco.com are www.cisco.com , ftp.cisco.com, and mail.cisco.com.



15.4.4

The nslookup Command

When configuring a network device, one or more DNS Server addresses are provided that the DNS client can use for name resolution. Usually the ISP provides the addresses to use for the DNS servers. When a user application requests to connect to a remote device by name, the requesting DNS client queries the name server to resolve the name to a numeric address.

Computer operating systems also have a utility called Nslookup that allows the user to manually query the name servers to resolve a given host name. This utility can also be used to troubleshoot name resolution issues and to verify the current status of the name servers.

In this figure, when the **nslookup** command is issued, the default DNS server configured for your host is displayed. The name of a host or domain can be entered at the **nslookup** prompt. The Nslookup utility has many options available for extensive testing and verification of the DNS process.

```
C:\Users> nslookup
Default Server:  dns-sj.cisco.com
Address:  171.70.168.183
> www.cisco.com
Server:  dns-sj.cisco.com
Address:  171.70.168.183
Name:    origin-www.cisco.com
Addresses:  2001:420:1101:1::a
          173.37.145.84
Aliases:  www.cisco.com
> cisco.netacad.net
Server:  dns-sj.cisco.com
Address:  171.70.168.183
Name:    cisco.netacad.net
Address:  72.163.6.223
>
```

15.4.5

Syntax Checker - The nslookup Command

Practice entering the nslookup command in both Windows and Linux

From the Windows command prompt, enter the **nslookup** command to begin a manual query of the name servers.

```
C:\>
```

ResetShow MeShow All

15.4.6

Dynamic Host Configuration Protocol

The Dynamic Host Configuration Protocol (DHCP) for IPv4 service automates the assignment of IPv4 addresses, subnet masks, gateways, and other IPv4 networking parameters. This is referred to as dynamic addressing. The alternative to dynamic addressing is static addressing. When using static addressing, the network administrator manually enters IP address information on hosts.

When a host connects to the network, the DHCP server is contacted, and an address is requested. The DHCP server chooses an address from a configured range of addresses called a pool and assigns (leases) it to the host.

On larger networks, or where the user population changes frequently, DHCP is preferred for address assignment. New users may arrive and need connections; others may have new computers that must be connected. Rather than use static addressing for each connection, it is more efficient to have IPv4 addresses assigned automatically using DHCP.

DHCP can allocate IP addresses for a configurable period of time, called a lease period. The lease period is an important DHCP setting, When the lease period expires or the DHCP server gets a DHCPRELEASE message the address is returned to the DHCP pool for reuse. Users can freely move from location to location and easily re-establish network connections through DHCP.

As the figure shows, various types of devices can be DHCP servers. The DHCP server in most medium-to-large networks is usually a local, dedicated PC-based server. With home networks, the DHCP server is usually located on the local router that connects the home network to the ISP.

The figure depicts a ISP DHCP server connected to the Internet with three ISP routers labelled ISP1, ISP2, ISP#. Each ISP reouter is connected to a different network. ISP1 connects to a wireless antenna to a mobile worker who is the DCHP client. ISP2 is connected to a coporate network router which connects to a coporate LAN with its own local DHCP server connected to a swith connected to six DHCP clients. ISP3 is connected to a wireless DHCP server for a Home and Small Business network the three DHCP clients connected.

InternetISP 2ISP 1ISP 3**ISP DHCP ServerRouter DHCP ServerMobile Worker**Home and Small Business NetworkDHCP Client**Corporate Network**DHCP ClientsDHCP Clients**Local DHCP ServerDHCP Server**

Many networks use both DHCP and static addressing. DHCP is used for general purpose hosts, such as end user devices. Static addressing is used for network devices, such as gateway routers, switches, servers, and printers.

DHCP for IPv6 (DHCPv6) provides similar services for IPv6 clients. One important difference is that DHCPv6 does not provide a default gateway address. This can only be obtained dynamically from the Router Advertisement message of the router.

15.4.7

DHCP Operation

As shown in the figure, when an IPv4, DHCP-configured device boots up or connects to the network, the client broadcasts a DHCP discover (DHCPDISCOVER) message to identify any available DHCP servers on the network. A DHCP server replies with a DHCP offer (DHCPOFFER) message, which offers a lease to the client. The offer message contains the IPv4 address and subnet mask to be assigned, the IPv4 address of the DNS server, and the IPv4 address of the default gateway. The lease offer also includes the duration of the lease.

The figure shows a protocol ladder with a DHCP client on one side and a DHCP client on the other. The DHCP client sends a DHCPDISCOVER message to the DHCP Server. The DHCP server sends a DHCPOFFER message to the DHCP client. The DHCP client sends a DHCPREQUEST message in repoonse to the DHCPOFFER from the DHCP server. THE DHCP server sends a DHCPACK message back to the DHCP client. The process is called DORA.

The client may receive multiple DHCPOFFER messages if there is more than one DHCP server on the local network. Therefore, it must choose between them, and sends a DHCP request (DHCPREQUEST) message that identifies the explicit server and lease offer that the client is accepting. A client may also choose to request an address that it had previously been allocated by the server.

Assuming that the IPv4 address requested by the client, or offered by the server, is still available, the server returns a DHCP acknowledgment (DHCPACK) message that acknowledges to the client that the lease has been finalized. If the offer is no longer valid, then the selected server responds with a DHCP negative acknowledgment (DHCPNAK) message. If a DHCPNAK message is returned, then the selection process must begin again with a new DHCPDISCOVER message being transmitted. After the client has the lease, it must be renewed prior to the lease expiration through another DHCPREQUEST message.

The DHCP server ensures that all IP addresses are unique (the same IP address cannot be assigned to two different network devices simultaneously). Most ISPs use DHCP to allocate addresses to their customers.

DHCPv6 has a set of messages that is similar to those for DHCPv4. The DHCPv6 messages are SOLICIT, ADVERTISE, INFORMATION REQUEST, and REPLY.

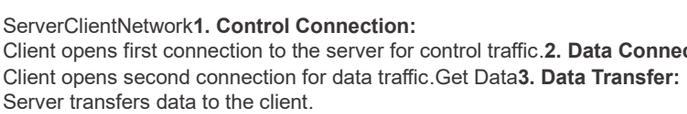
15.5.1

File Transfer Protocol

As you learned in previous topics, in the client/server model, the client can upload data to a server, and download data from a server, if both devices are using a file transfer protocol (FTP). Like HTTP, email, and addressing protocols, FTP is commonly used application layer protocol. This topic discusses FTP in more detail.

FTP was developed to allow for data transfers between a client and a server. An FTP client is an application which runs on a computer that is being used to push and pull data from an FTP server.

The figure depicts an FTP transaction between a client and a server. A client is contacting a server thru a network. The first message from the client is a control connection: the client opens first connection to the server for control traffic. The second message from the client is a data connection: the client opens a second connection for data traffic. Data can then be downloaded from the server of uploaded from the client.



Based on commands sent across the control connection, data can be downloaded from the server or uploaded from the client.

The client establishes the first connection to the server for control traffic using TCP port 21. The traffic consists of client commands and server replies.

The client establishes the second connection to the server for the actual data transfer using TCP port 20. This connection is created every time there is data to be transferred.

The data transfer can happen in either direction. The client can download (pull) data from the server, or the client can upload (push) data to the server.

15.5.2

Server Message Block

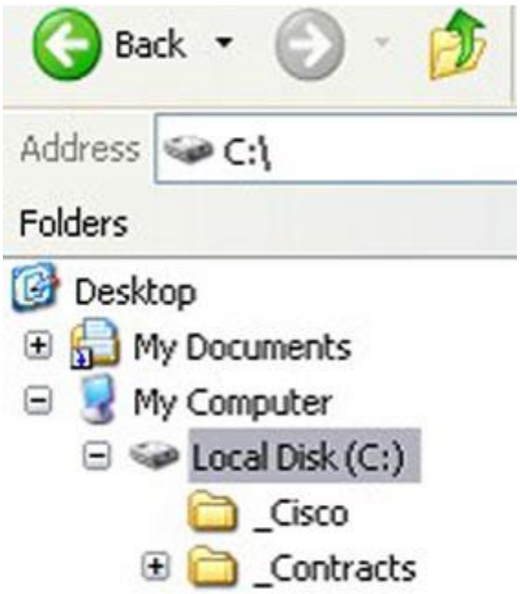
The Server Message Block (SMB) is a client/server file sharing protocol that describes the structure of shared network resources, such as directories, files, printers, and serial ports. It is a request-response protocol. All SMB messages share a common format. This format uses a fixed-sized header, followed by a variable-sized parameter and data component.

Here are three functions of SMB messages:

- Start, authenticate, and terminate sessions.
- Control file and printer access.
- Allow an application to send or receive messages to or from another device.

SMB file-sharing and print services have become the mainstay of Microsoft networking. With the introduction of the Windows 2000 software series, Microsoft changed the underlying structure for using SMB. In previous versions of Microsoft products, the SMB services used a non-TCP/IP protocol to implement name resolution. Beginning with Windows 2000, all subsequent Microsoft products use DNS naming, which allows TCP/IP protocols to directly support SMB resource sharing, as shown in the figure.

The first figure shows a Microsoft Windows shared resource of My Documents with a client requests from a Server My Documents. The client sends a SMB request and receives an SMB response of the shared resource My Documents. Shared Resources include File systems, Pronters shown as a icon, Mail slots, and APIs.



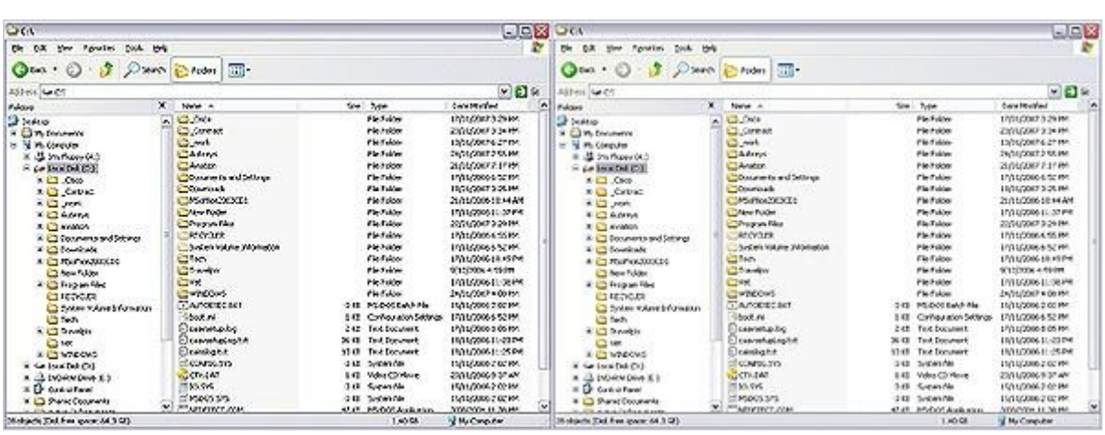
ClientSMB RequestsSMB ResponsesServerPrinterShared Resources

- File systems
- Printers
- Mail slots
- APIs

SMB is a client/server, request-response protocol. Servers can make their own resources available to clients on the network.

The SMB file exchange process between Windows PCs is shown in the next figure.

The second figure depicts a copy file taking place between two Windows computers form one filesystem to the other thru the network.



NetworkCopy File

A file may be copied from PC to PC with Windows Explorer using the SMB protocol.

Unlike the file sharing supported by FTP, clients establish a long-term connection to servers. After the connection is established, the user of the client can access the resources on the server as though the resource is local to the client host.

The LINUX and UNIX operating systems also provide a method of sharing resources with Microsoft networks by using a version of SMB called SAMBA. The Apple Macintosh operating systems also support resource sharing by using the SMB protocol.

16.1.1

Types of Threats

Wired and wireless computer networks are essential to everyday activities. Individuals and organizations depend on their computers and networks. Intrusion by an unauthorized person can result in costly network outages and loss of work. Attacks on a network can be devastating and can result in a loss of time and money due to damage, or theft of important information or assets.

Intruders can gain access to a network through software vulnerabilities, hardware attacks, or through guessing someone's username and password. Intruders who gain access by modifying software or exploiting software vulnerabilities are called threat actors.

After the threat actor gains access to the network, four types of threats may arise.

Click each button for information about each threat.
Information Theft

Data Loss and Manipulation

Identity Theft

Disruption of Service

Information theft is breaking into a computer to obtain confidential information. Information can be used or sold for various purposes such as when someone is stealing proprietary information of an organization, like research and development data.

The figure shows an open folder

16.1.2

Types of Vulnerabilities

Vulnerability is the degree of weakness in a network or a device. Some degree of vulnerability is inherent in routers, switches, desktops, servers, and even security devices. Typically, the network devices under attack are the endpoints, such as servers and desktop computers.

There are three primary vulnerabilities or weaknesses: technological, configuration, and security policy. All three of these sources of vulnerabilities can leave a network or device open to various attacks, including malicious code attacks and network attacks.

Click each button for a table with examples and a description of each type of vulnerability.
Technological Vulnerabilities

Configuration Vulnerabilities

Policy Vulnerabilities

Technological Vulnerabilities

Table caption

Vulnerability	Description
TCP/IP Protocol Weakness	• Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), and Internet Control Mess Protocol (ICMP) are inherently insecure. • Simple Network Management Protocol (SNMP) and Simple Mail Transfer Protocol (SMTP) a to the inherently insecure structure upon which TCP was designed.
Operating System Weakness	• Each operating system has security problems what must be addressed. • UNIX, Linux, Mac OS, Mac OS X, Windows Server 2012, Windows 7, Windows 8

Table caption	
Vulnerability	**Description**
	• They are documented in the Computer Emergency Response Team (CERT) archives at http://www.cert.org
Network Equipment Weakness	Various types of network equipment, such as routers, firewalls, and switches have security weakne must be recognized and protected against. Their weaknesses include password protection, lack of authentication, routing protocols, and firewall holes.

16.1.3

Physical Security

An equally important vulnerable area of the network to consider is the physical security of devices. If network resources can be physically compromised, a threat actor can deny the use of network resources.

The four classes of physical threats are as follows:

- **Hardware threats -** This includes physical damage to servers, routers, switches, cabling plant, and workstations.
- **Environmental threats -** This includes temperature extremes (too hot or too cold) or humidity extremes (too wet or too dry).
- **Electrical threats -** This includes voltage spikes, insufficient supply voltage (brownouts), unconditioned power (noise), and total power loss.
- **Maintenance threats -** This includes poor handling of key electrical components (electrostatic discharge), lack of critical spare parts, poor cabling, and poor labeling.

A good plan for physical security must be created and implemented to address these issues. The figure shows an example of physical security plan.

The figure is a square that depicts a computer room. Inside the computer room on the top left corner, is a small rectangle labeled, AC. On the top right corner, four squares are connected and labeled, UPS BAY. In the center of the computer room there are three rows of squares, labeled servers, WAN, and LAN. The lower part of the computer room is divided to create a separate room. There is a dotted section of the divider labeled, locked door. Inside the separate room is a Help desk, a card reader as well as another door on the exterior.

Plan Physical Security to Limit Damage to Equipment

Locked DoorDoorCard
ReaderHelp DeskACWANSVRSUPS BAYLAN

- Secure computer room.
- Implement physical security to limit damage to the equipment.

Step 1. Lock up equipment and prevent unauthorized access from the doors, ceiling, raised floor, windows, ducts, and vents.

Step 2. Monitor and control closet entry with electronic logs.

Step 3. Use security cameras.

16.2.1

Types of Malware

The previous topic explained the types of network threats and the vulnerabilities that make threats possible. This topic goes into more detail about how threat actors gain access to network or restrict authorized users from having access.

Malware is short for malicious software. It is code or software specifically designed to damage, disrupt, steal, or inflict “bad” or illegitimate action on data, hosts, or networks. Viruses, worms, and Trojan horses are types of malware.

Viruses

A computer virus is a type of malware that propagates by inserting a copy of itself into, and becoming part of, another program. It spreads from one computer to another, leaving infections as it travels. Viruses can range in severity from causing mildly annoying effects, to damaging data or software and causing denial of service (DoS) conditions. Almost all viruses are attached to an executable file, which means the virus may exist on a system but will not be active or able to spread until a user runs or opens the malicious host file or program. When the host code is executed, the viral code is executed as well. Normally, the host program keeps functioning after the virus infects it. However, some viruses overwrite other programs with copies of themselves, which destroys the host program altogether. Viruses spread when the software or document they are attached to is transferred from one computer to another using the network, a disk, file sharing, or infected email attachments.

Worms

Computer worms are similar to viruses in that they replicate functional copies of themselves and can cause the same type of damage. In contrast to viruses, which require the spreading of an infected host file, worms are standalone software and do not require a host program or human help to propagate. A worm does not need to attach to a program to infect a host and enter a computer through a vulnerability in the system. Worms take advantage of system features to travel through the network unaided.

Trojan Horses

A Trojan horse is another type of malware named after the wooden horse the Greeks used to infiltrate Troy. It is a harmful piece of software that looks legitimate. Users are typically tricked into loading and executing it on their systems. After it is activated, it can achieve any number of attacks on the host, from irritating the user (with excessive pop-up windows or changing the desktop) to damaging the host (deleting files, stealing data, or activating and spreading other malware, such as viruses). Trojan horses are also known to create back doors to give malicious users access to the system.

Unlike viruses and worms, Trojan horses do not reproduce by infecting other files. Trojan horses must spread through user interaction such as opening an email attachment or downloading and running a file from the internet.

Click Play in the figure to view an animated explanation of the three types of malware.

The animation shows a network with two PCs and two routers with the routers are connected to each other sit between the two PCs with each PC connected to one of the routers. The PC on the left has an attacker. As the animation plays a text box opens that reads “The primary vulnerabilities for end-user workstations are virus, worm, and Trojan Horse attacks. As the animation continues to play the attacker at the PC on the left sends a virus attack on the network that travels over the network routers to the PC on the right. A text box opens that reads “A virus is malicious software which executes a specific unwanted, and often harmful, function on a computer”. As the animation continues to play the attacker at the PC on the left sends a worm attack on the network that travels over the network routers to the PC on the right. A text box opens that reads “A worm executes arbitrary code and installs copies of itself in the memory of the infected computer. The main purpose of a worm is to automatically replicate itself and spread across the network from system to system”. As the animation continues to play the attacker at the PC on the left sends a Trojan Horse attack on the network that travels over the network routers to the PC on the right. A text box opens that reads “A Trojan horse is a non-self-replicating type of malware. It often contains malicious code that is designed to look like something else, such as a legitimate application or file. When an infected application or file is downloaded and opened, the Trojan horse can attack the end device from within”.

The primary vulnerabilities for end-user workstations are virus, worm, and Trojan Horse attacks.

A virus is malicious software which executes a specific unwanted, and often harmful, function on a computer.

A worm executes arbitrary code and installs copies of itself in the memory of the infected computer. The main purpose of a worm is to automatically replicate itself and spread across the network from system to system.

A Trojan horse is a non-self-replicating type of malware. It often contains malicious code that is designed to look like something else, such as a legitimate application or file. When an infected application or file is downloaded and opened, the Trojan horse can attack the end device from within.

16.2.2

Reconnaissance Attacks

In addition to malicious code attacks, it is also possible for networks to fall prey to various network attacks. Network attacks can be classified into three major categories:

- **Reconnaissance attacks** - The discovery and mapping of systems, services, or vulnerabilities.
- **Access attacks** - The unauthorized manipulation of data, system access, or user privileges.
- **Denial of service** - The disabling or corruption of networks, systems, or services.

For reconnaissance attacks, external threat actors can use internet tools, such as the **nslookup** and **whois** utilities, to easily determine the IP address space assigned to a given corporation or entity. After the IP address space is determined, a threat actor can then ping the publicly available IP addresses to identify the addresses that are active. To help automate this step, a threat actor may use a ping sweep tool, such as **fping** or **gping**. This systematically pings all network addresses in a given range or subnet. This is similar to going through a section of a telephone book and calling each number to see who answers.

Click each type of reconnaissance attack tool to see an animation of the attack.
Internet Queries

Ping Sweeps

Port Scans

Internet Queries

Click Play in the figure to view an animation. The threat actor is looking for initial information about a target. Various tools can be used, including Google search, the websites of organizations, whois, and more.

Threat Actor

16.2.3

Access Attacks

Access attacks exploit known vulnerabilities in authentication services, FTP services, and web services to gain entry to web accounts, confidential databases, and other sensitive information. An access attack allows individuals to gain unauthorized access to information that they have no right to view. Access attacks can be classified into four types: password attacks, trust exploitation, port redirection, and man-in-the middle.

Click each button for an explanation of each type of attack.
Password Attacks

Trust Exploitation

Port Redirection

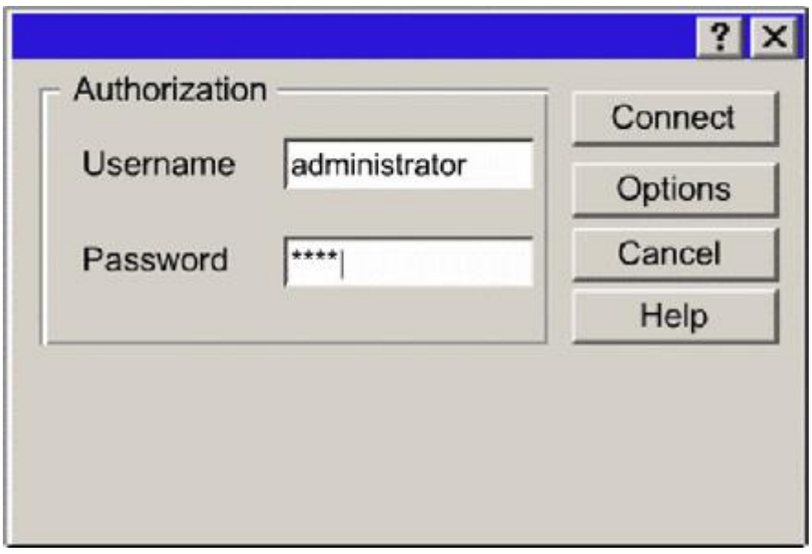
Man-in-the-Middle

Password Attacks

Threat actors can implement password attacks using several different methods:

- Brute-force attacks
- Trojan horse attacks
- Packet sniffers

The first figure shows a login prompt box with the username, administrator and the password, ****.



16.2.4

Denial of Service Attacks

Denial of service (DoS) attacks are the most publicized form of attack and among the most difficult to eliminate. However, because of their ease of implementation and potentially significant damage, DoS attacks deserve special attention from security administrators.

DoS attacks take many forms. Ultimately, they prevent authorized people from using a service by consuming system resources. To help prevent DoS attacks it is important to stay up to date with the latest security updates for operating systems and applications.

Click each button for an example of DoS and distributed DoS (DDoS) attacks.

DoS Attack

DDoS Attack

DoS Attack

DoS attacks are a major risk because they interrupt communication and cause significant loss of time and money. These attacks are relatively simple to conduct, even by an unskilled threat actor.

Click Play in the figure to view the animation of a DoS attack.

This animation shows a Web server, www.XYZcorp.com, an internet user, two PCs, and a threat actor all connected to an internet cloud. As the animation plays a text box appears above the threat actor stating “I’ll send so many pings that the server can’t respond to anyone else. A series of pings leaves the treat actor’s PC toward the Internet. From the internet the series of pings are sent to the www.XYZcorp.com web server. A text box appearing above the web server states “Help, I can’t accomplish any work!”. As the pings continue from the threat actor to the web server a text box appears above the internet user stating “This website is very slow today!”.

I'll send so many pings that the server can't respond to anyone else.

Help, I can't accomplish any work!

Internet

Web Server www.XYZcorp.com

Threat Actor

This website is very slow today!

16.3.1

The Defense-in-Depth Approach

Now that you know more about how threat actors can break into networks, you need to understand what to do to prevent this unauthorized access. This topic details several actions you can take to make your network more secure.

To mitigate network attacks, you must first secure devices including routers, switches, servers, and hosts. Most organizations employ a defense-in-depth approach (also known as a layered approach) to security. This requires a combination of networking devices and services working in tandem.

Consider the network in the figure. There are several security devices and services that have been implemented to protect its users and assets against TCP/IP threats.

All network devices including the router and switches are also hardened as indicated by the combination locks on their respective icons. This indicates that they have been secured to prevent threat actors from gaining access and tampering with the devices.

The figure depicts a campus area network. A cloud representing the Internet is connected to a router, labeled VPN. The VPN router is connected to an ASA firewall. The firewall has two additional connections; one to an IPS and another to a switch. The switch is connected to a DHCP server, email server, web server, and ESA/WSA. The IPS is connected to a multilayer switch. The multilayer switch has a connection to an AAA server as well as to two layer 2 switches and a to another multilayer switch. The second multilayer switch also has connections to the same layer 2 switches, creating redundancy. Below the layer 2 switches are three laptops and three pcs which are labeled as hosts.

Campus Area Network InternetAAA ServerASA
FirewallLayer 3 SwitchesVPNIPSESA/WSALayer 2 SwitchesDHCP ServerEmail ServerWeb ServerHosts

Several security devices and services are implemented to protect an organization’s users and assets against TCP/IP threats.

- **VPN** - A router is used to provide secure VPN services with corporate sites and remote access support for remote users using secure encrypted tunnels.
- **ASA Firewall** - This dedicated device provides stateful firewall services. It ensures that internal traffic can go out and come back, but external traffic cannot initiate connections to inside hosts.
- **IPS** - An intrusion prevention system (IPS) monitors incoming and outgoing traffic looking for malware, network attack signatures, and more. If it recognizes a threat, it can immediately stop it.
- **ESA/WSA** - The email security appliance (ESA) filters spam and suspicious emails. The web security appliance (WSA) filters known and suspicious internet malware sites.
- **AAA Server** - This server contains a secure database of who is authorized to access and manage network devices. Network devices authenticate administrative users using this database.

16.3.2

Keep Backups

Backing up device configurations and data is one of the most effective ways of protecting against data loss. A data backup stores a copy of the information on a computer to removable backup media that can be kept in a safe place. Infrastructure devices should have backups of configuration files and IOS images on an FTP or similar file server. If the computer or a router hardware fails, the data or configuration can be restored using the backup copy.

Backups should be performed on a regular basis as identified in the security policy. Data backups are usually stored offsite to protect the backup media if anything happens to the main facility. Windows hosts have a backup and restore utility. It is important for users to back up their data to another drive, or to a cloud-based storage provider.

The table shows backup considerations and their descriptions.

Table caption

Consideration	Description
Frequency	• Perform backups on a regular basis as identified in the security policy. • Full backups can be time-consuming, therefore perform monthly or weekly backups with frequent partial bac changed files.
Validation	• Always validate backups to ensure the integrity of the data and validate the file restoration procedures.
Storage	• Backups should be transported to an approved offsite storage location on a daily, weekly, or monthly rotation required by the security policy.

Table caption	
Consideration	**Description**
Security	• Backups should be protected using strong passwords. The password is required to restore the data.

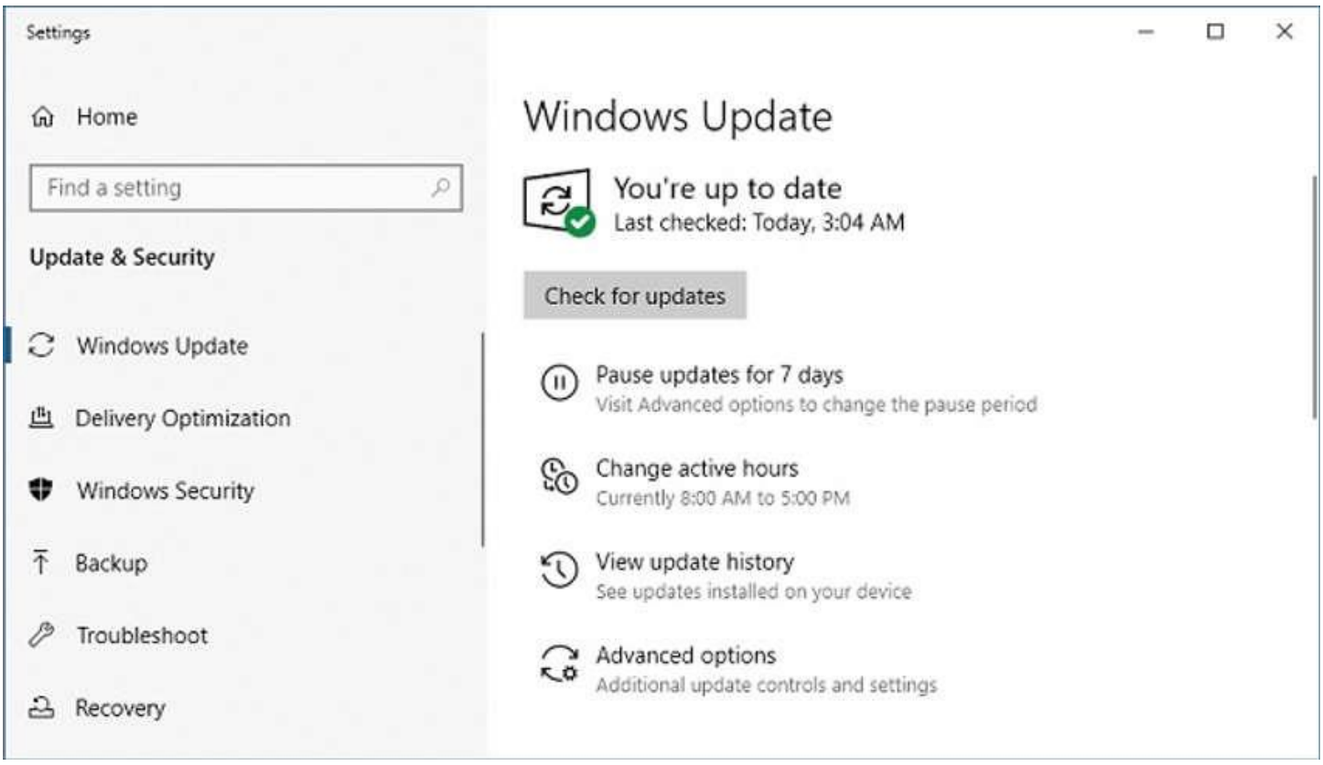
16.3.3

Upgrade, Update, and Patch

Keeping up to date with the latest developments can lead to a more effective defense against network attacks. As new malware is released, enterprises need to keep current with the latest versions of antivirus software.

The most effective way to mitigate a worm attack is to download security updates from the operating system vendor and patch all vulnerable systems. Administering numerous systems involves the creation of a standard software image (operating system and accredited applications that are authorized for use on client systems) that is deployed on new or upgraded systems. However, security requirements change, and already deployed systems may need to have updated security patches installed.

One solution to the management of critical security patches is to make sure all end systems automatically download updates, as shown for Windows 10 in the figure. Security patches are automatically downloaded and installed without user intervention.



16.3.4

Authentication, Authorization, and Accounting

All network devices should be securely configured to provide only authorized individuals with access. Authentication, authorization, and accounting (AAA, or “triple A”) network security services provide the primary framework to set up access control on network devices.

AAA is a way to control who is permitted to access a network (authenticate), what actions they perform while accessing the network (authorize), and making a record of what was done while they are there (accounting).

The concept of AAA is similar to the use of a credit card. The credit card identifies who can use it, how much that user can spend, and keeps account of what items the user spent money on, as shown in the figure.

The figure shows a credit card next to a credit card statement. There is a rectangle around the numbers on the credit card with the text, Authentication Who are you? A second rectangle is around the credit limit on the credit card statement with text, Authorization How much can you spend? A third rectangle is around the transaction portion of the credit card summary with the text Accounting What did you spend on it?

Account Number	Statement Closing Date	Current Amount Due
1234-567-890	01-31-01	$278.50

JOE EMPLOYEE
456 SKYVIEW DRIVE
HOMETOWN, USA 99900-1234

MAIL PAYMENT TO :
THE BANK
132 VINE STREET
ANYTOWN, USA 67500-0010

872919345 00178255000000003

Detach here and return upper portion with check or money order. Do not staple or fold.

Statement of Personal Credit Card Account

Retain this portion for your files.

THE BANK

Cardmember Name	Account Number	Statement Closing Date
JOE EMPLOYEE	1234-456-890	01-31-01

Statement Date:	02-01-01	Payment Due Date:	03-01-01
Closing Date:	01-31-01		
Credit Limit	$1,500.00	Credit Available:	$1221.50
New Balance:	$278.50	Minimum Payment Due:	$20.00

Account Summary

Previous Balance:	+74.24	Transaction Fees:	+3.00
Purchases:	+250.50	Annual Fees:	+25.00
Cash Advances:	+0	Current Amount Due:	+250.50
Payments:	-74.25	Amount Past Due:	+0
Finance Charge:	+0	Amount Over Credit Line:	+0
Late Charge:	+0	NEW BALANCE:	$278.50

Reference Number	Sold	Posted	Activity Since Last Statement		Amount
43210987	01-03	01-13	Payment, Thank You		-$74.25
01234567	01-12	01-13	Wings 'N' Things	Anytown, USA	$25.25
78901234	01-14	01-17	Record Release	Anytown, USA	$40.00
45678901	01-14	01-17	Sports Stadium	Anytown, USA	$75.25
3210987	01-22	01-23	Tie Tack	Anytown, USA	$20.75
76543210	01-29	01-30	Electronic World	Anytown, USA	$89.25
23455678		01-30	Transaction Fees		$3.00
34567890		01-01	Annual Fee		$25.00

PAGE 1 OF 1

Authentication

Who are you?**Authorization**

How much can you spend?**Accounting**

What did you spend it on?

16.3.5

Firewalls

A firewall is one of the most effective security tools available for protecting users from external threats. A firewall protects computers and networks by preventing undesirable traffic from entering internal networks.

Network firewalls reside between two or more networks, control the traffic between them, and help prevent unauthorized access. For example, the top topology in the figure illustrates how the firewall enables traffic from an internal network host to exit the network and return to the inside network. The bottom topology illustrates how traffic initiated by the outside network (i.e., the internet) is denied access to the internal network.

The figure shows a rectangle, labeled Inside. Inside the rectangle there is a pc. Outside of and to the right of the rectangle, there is a firewall. To the right of the firewall, there is a cloud labeled, Internet. There are two arrows, one signifying traffic leaving the pc going through the firewall and out to the Internet. The second arrow signifies the firewall permitting traffic from the Internet to the pc. The figure shows another rectangle, labeled Inside. Inside the rectangle there is a pc. Outside of and to the right of the rectangle, there is a firewall. To the right of the firewall, there is a cloud labeled, Internet. There is an arrow pointing from the Internet to the firewall with an X signifying that traffic is being denied from the Internet to the internal network.

Firewall Operation

Firewall permits traffic from users in the inside network to exit and return.**InsideInternetFirewall**Firewall denies outside traffic access to the inside network.**InsideInternetFirewall**

A firewall could allow outside users controlled access to specific services. For example, servers accessible to outside users are usually located on a special network referred to as the demilitarized zone (DMZ), as shown in the figure. The DMZ enables a network administrator to apply specific policies for hosts connected to that network.

The figure shows a rectangle, labeled Inside. Inside the rectangle there is a pc. Outside of and to the right of the rectangle, there is a firewall. To the right of the firewall, there is a cloud labeled Internet. Above the firewall, theres a DMZ server inside of a rectangle. There are twos arrows, one going from the pc through the firewall to the DMZ server and another going from the Internet through the firewall to the DMZ sever.

Firewall Topology with DMZ

DMZServerInsideInternet

16.3.6

Types of Firewalls

Firewall products come packaged in various forms. These products use different techniques for determining what will be permitted or denied access to a network. They include the following:

- **Packet filtering** - Prevents or allows access based on IP or MAC addresses
- **Application filtering** - Prevents or allows access by specific application types based on port numbers
- **URL filtering** - Prevents or allows access to websites based on specific URLs or keywords
- **Stateful packet inspection (SPI)** - Incoming packets must be legitimate responses to requests from internal hosts. Unsolicited packets are blocked unless permitted specifically. SPI can also include the capability to recognize and filter out specific types of attacks, such as denial of service (DoS)

16.3.7

Endpoint Security

An endpoint, or host, is an individual computer system or device that acts as a network client. Common endpoints are laptops, desktops, servers, smartphones, and tablets. Securing endpoint devices is one of the most challenging jobs of a network administrator because it involves human nature. A company must have well-documented policies in place and employees must be aware of these rules. Employees need to be trained on proper use of the network. Policies often include the use of antivirus software and host intrusion prevention. More comprehensive endpoint security solutions rely on network access control.

16.4.1

Cisco AutoSecure

One area of networks that requires special attention to maintain security is the devices. You probably already have a password for your computer, smart phone, or tablet. Is it as strong as it could be? Are you using other tools to enhance the security of your devices? This topic tells you how.

The security settings are set to the default values when a new operating system is installed on a device. In most cases, this level of security is inadequate. For Cisco routers, the Cisco AutoSecure feature can be used to assist securing the system, as shown in the example.

```
Router# auto secure

                --- AutoSecure Configuration ---

*** AutoSecure configuration enhances the security of

the router but it will not make router absolutely secure

from all security attacks ***
```

In addition, there are some simple steps that should be taken that apply to most operating systems:

- Default usernames and passwords should be changed immediately.
- Access to system resources should be restricted to only the individuals that are authorized to use those resources.
- Any unnecessary services and applications should be turned off and uninstalled when possible.

Often, devices shipped from the manufacturer have been sitting in a warehouse for a period of time and do not have the most up-to-date patches installed. It is important to update any software and install any security patches prior to implementation.

16.4.2

Passwords

To protect network devices, it is important to use strong passwords. Here are standard guidelines to follow:

- Use a password length of at least eight characters, preferably 10 or more characters. A longer password is a more secure password.
- Make passwords complex. Include a mix of uppercase and lowercase letters, numbers, symbols, and spaces, if allowed.
- Avoid passwords based on repetition, common dictionary words, letter or number sequences, usernames, relative or pet names, biographical information, such as birthdates, ID numbers, ancestor names, or other easily identifiable pieces of information.
- Deliberately misspell a password. For example, Smith = Smyth = 5mYth or Security = 5ecur1ty.
- Change passwords often. If a password is unknowingly compromised, the window of opportunity for the threat actor to use the password is limited.
- Do not write passwords down and leave them in obvious places such as on the desk or monitor.

The tables show examples of strong and weak passwords.

Weak Passwords

Table caption

Weak Password	Why it is Weak
secret	Simple dictionary password
smith	Maiden name of mother
toyota	Make of a car
bob1967	Name and birthday of the user
Blueleaf23	Simple words and numbers

Strong Passwords

Table caption	
Strong Password	**Why it is Strong**
b67n42d39c	Combines alphanumeric characters
12^h u4@1p7	Combines alphanumeric characters, symbols, and includes a space

On Cisco routers, leading spaces are ignored for passwords, but spaces after the first character are not. Therefore, one method to create a strong password is to use the space bar and create a phrase made of many words. This is called a passphrase. A passphrase is often easier to remember than a simple password. It is also longer and harder to guess.

16.4.3

Additional Password Security

Strong passwords are only useful if they are secret. There are several steps that can be taken to help ensure that passwords remain secret on a Cisco router and switch including these:

- Encrypting all plaintext passwords
- Setting a minimum acceptable password length
- Deterring brute-force password guessing attacks
- Disabling an inactive privileged EXEC mode access after a specified amount of time.

As shown in the sample configuration in the figure, the **service password-encryption** global configuration command prevents unauthorized individuals from viewing plaintext passwords in the configuration file. This command encrypts all plaintext passwords. Notice in the example, that the password “cisco” has been encrypted as “03095A0F034F”.

To ensure that all configured passwords are a minimum of a specified length, use the **security passwords min-length** *length* command in global configuration mode. In the figure, any new password configured would have to have a minimum length of eight characters.

Threat actors may use password cracking software to conduct a brute-force attack on a network device. This attack continuously attempts to guess the valid passwords until one works. Use the **login block-for # attempts # within #** global configuration command to deter this type of attack. In the figure for example, the **login block-for 120 attempts 3 within 60** command will block vty login attempts for 120 seconds if there are three failed login attempts within 60 seconds.

Network administrators can become distracted and accidently leave a privileged EXEC mode session open on a terminal. This could enable an internal threat actor access to change or erase the device configuration.

By default, Cisco routers will logout an EXEC session after 10 minutes of inactivity. However, you can reduce this setting using the **exec-timeout** *minutes seconds* line configuration command. This command can be applied online console, auxiliary, and vty lines. In the figure, we are telling the Cisco device to automatically disconnect an inactive user on a vty line after the user has been idle for 5 minutes and 30 seconds.

```
R1(config)# service password-encryption
R1(config)# security passwords min-length 8
R1(config)# login block-for 120 attempts 3 within 60
R1(config)# line vty 0 4
R1(config-line)# password cisco123
R1(config-line)# exec-timeout 5 30
R1(config-line)# transport input ssh
R1(config-line)# end
R1#
```

```
R1# show running-config | section line vty
line vty 0 4
 password 7 094F471A1A0A
 exec-timeout 5 30
 login
 transport input ssh
R1#
```

16.4.4

Enable SSH

Telnet simplifies remote device access, but it is not secure. Data contained within a Telnet packet is transmitted unencrypted. For this reason, it is highly recommended to enable Secure Shell (SSH) on devices for secure remote access.

It is possible to configure a Cisco device to support SSH using the following six steps:

Step 1. Configure a unique device hostname. A device must have a unique hostname other than the default.

Step 2. Configure the IP domain name. Configure the IP domain name of the network by using the global configuration mode command **ip domain name** *name*.

Step 3. Generate a key to encrypt SSH traffic. SSH encrypts traffic between source and destination. However, to do so, a unique authentication key must be generated by using the global configuration command **crypto key generate rsa general-keys modulus** *bits*. The modulus *bits* determines the size of the key and can be configured from 360 bits to 2048 bits. The larger the bit value, the more secure the key. However, larger bit values also take longer to encrypt and decrypt information. The minimum recommended modulus length is 1024 bits.

Step 4. Verify or create a local database entry. Create a local database username entry using the **username** global configuration command. In the example, the parameter **secret** is used so that the password will be encrypted using MD5.

Step 5. Authenticate against the local database. Use the **login local** line configuration command to authenticate the vty line against the local database.

Step 6. Enable vty inbound SSH sessions. By default, no input session is allowed on vty lines. You can specify multiple input protocols including Telnet and SSH using the **transport input {ssh | telnet}** command.

As shown in the example, router R1 is configured in the span.com domain. This information is used along with the bit value specified in the **crypto key generate rsa general-keys modulus** command to create an encryption key.

Next, a local database entry for a user named Bob is created. Finally, the vty lines are configured to authenticate against the local database and to only accept incoming SSH sessions.

```
Router# configure terminal
Router(config)# hostname R1
R1(config)# ip domain name span.com
R1(config)# crypto key generate rsa general-keys modulus 1024
The name for the keys will be: R1.span.com % The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
Dec 13 16:19:12.079: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#
R1(config)# username Bob secret cisco
R1(config)# line vty 0 4
```

```
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# exit
R1(config)#
```

16.4.5

Disable Unused Services

Cisco routers and switches start with a list of active services that may or may not be required in your network. Disable any unused services to preserve system resources, such as CPU cycles and RAM, and prevent threat actors from exploiting these services. The type of services that are on by default will vary depending on the IOS version. For example, IOS-XE typically will have only HTTPS and DHCP ports open. You can verify this with the **show ip ports all** command, as shown in the example.

```
Router# show ip ports all

Proto Local Address               Foreign Address             State       PID/Program
Name

TCB       Local Address               Foreign Address             (state)

tcp   :::443                       :::*                        LISTEN      309/[IOS]HTTP
CORE

tcp   *:443                        *:*                         LISTEN      309/[IOS]HTTP
CORE

udp   *:67                         0.0.0.0:0
387/[IOS]DHCPD Receive

Router#
```

IOS versions prior to IOS-XE use the **show control-plane host open-ports** command. We mention this command because you may see it on older devices. The output is similar. However, notice that this older router has an insecure HTTP server and Telnet running. Both of these services should be disabled. As shown in the example, disable HTTP with the **no ip http server** global configuration command. Disable Telnet by specifying only SSH in the line configuration command, **transport input ssh**.

```
Router# show control-plane host open-ports

Active internet connections (servers and established)

Prot        Local Address      Foreign Address                   Service    State

tcp                *:23                *:0                     Telnet   LISTEN

tcp                *:80                *:0                  HTTP CORE   LISTEN

udp                *:67                *:0             DHCPD Receive   LISTEN

Router# configure terminal
```

```
Router(config)# no ip http server
```

```
Router(config)# line vty 0 15
```

```
Router(config-line)# transport input ssh
```

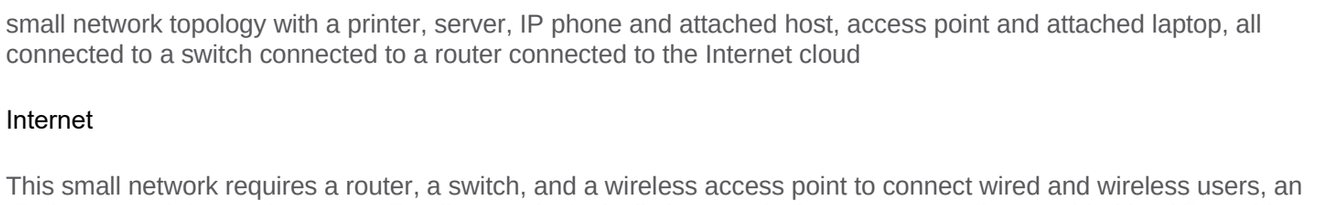
17.1.1

Small Network Topologies

The majority of businesses are small; therefore, it is not surprising that the majority of business networks are also small.

A small network design is usually simple. The number and type of devices included are significantly reduced compared to that of a larger network.

For instance, refer to the sample small-business network shown in the figure.



This small network requires a router, a switch, and a wireless access point to connect wired and wireless users, an IP phone, a printer, and a server. Small networks typically have a single WAN connection provided by DSL, cable, or an Ethernet connection.

Large networks require an IT department to maintain, secure, and troubleshoot network devices and to protect organizational data. Managing a small network requires many of the same skills as those required for managing a larger one. Small networks are managed by a local IT technician or by a contracted professional.

17.1.2

Device Selection for a Small Network

Like large networks, small networks require planning and design to meet user requirements. Planning ensures that all requirements, cost factors, and deployment options are given due consideration.

One of the first design considerations is the type of intermediary devices to use to support the network.

Click each button for more information about the factors that must be considered when selecting network devices.
Cost

Speed and Types of Ports/Interfaces

Expandability

Operating System Features and Services

Cost

The cost of a switch or router is determined by its capacity and features. This includes the number and types of ports available and the backplane speed. Other factors that influence the cost are network management capabilities, embedded security technologies, and optional advanced switching technologies. The expense of cable runs required to connect every device on the network must also be considered. Another key element affecting cost considerations is the amount of redundancy to incorporate into the network.

17.1.3

IP Addressing for a Small Network

When implementing a network, create an IP addressing scheme and use it. All hosts and devices within an internetwork must have a unique address.

Devices that will factor into the IP addressing scheme include the following:

- End user devices - The number and type of connection (i.e., wired, wireless, remote access)
- Servers and peripherals devices (e.g., printers and security cameras)
- Intermediary devices including switches and access points

It is recommended that you plan, document, and maintain an IP addressing scheme based on device type. The use of a planned IP addressing scheme makes it easier to identify a type of device and to troubleshoot problems, as for instance, when troubleshooting network traffic issues with a protocol analyzer.

For example, refer to the topology of a small to medium sized organization in the figure.

network topology consisting of three LANs - 192.168.1.0/24, 192.168.2.0/24, and 192.168.3.0/24 - with various end devices, connected to a router connected to the Internet cloud

192.168.1.0/24192.168.2.0/24192.168.3.0/24

Internet

The organization requires three user LANs (i.e., 192.168.1.0/24, 192.168.2.0/24, and 192.168.3.0/24). The organization has decided to implement a consistent IP addressing scheme for each 192.168.x.0/24 LAN using the following plan:

Device TypeAssignable IP Address Range Summarized as …Default gateway (Router)192.168.x.1 - 192.168.x.2192.168.x.0/30Switches (max 2)192.168.x.5 - 192.168.x.6192.168.x.4/30Access points (max 6)192.168.x.9 - 192.168.x.14192.168.x.8/29Servers (max 6)192.168.x.17 - 192.168.x.22192.168.x.16/29Printers (max 6)192.168.x.25 - 192.168.x.30192.168.x.24/29IP Phones (max 6)192.168.x.33 - 192.168.x.38192.168.x.32/29Wired devices (max 62)192.168.x.65 - 192.168.x.126192.168.x.64/26Wireless devices (max 62)192.168.x.193 - 192.168.x.254192.168.x.192/26

Device Type	**Assignable IP Address Range**	**Summarized as …**
Default gateway (Router)	192.168.x.**1** - 192.168.x.**2**	192.168.x.**0/30**
Switches (max 2)	192.168.x.**5** - 192.168.x.**6**	192.168.x.**4/30**
Access points (max 6)	192.168.x.**9** - 192.168.x.**14**	192.168.x.**8/29**
Servers (max 6)	192.168.x.**17** - 192.168.x.**22**	192.168.x.**16/29**
Printers (max 6)	192.168.x.**25** - 192.168.x.**30**	192.168.x.**24/29**
IP Phones (max 6)	192.168.x.**33** - 192.168.x.**38**	192.168.x.**32/29**
Wired devices (max 62)	192.168.x.**65** - 192.168.x.**126**	192.168.x.**64/26**
Wireless devices (max 62)	192.168.x.**193** - 192.168.x.**254**	192.168.x.**192/26**

The figure displays an example of the 192.168.2.0/24 network devices with assigned IP addresses using the predefined IP addressing scheme.

The diagram is a small LAN topology with a network address of 192.168.2.0/24. It shows various end devices all connected to a switch, with address .5, connected to a router, at address .1, connected to the Internet cloud. All devices have been assigned an IP address. A printer has an address of .25; server has an address of .17; a PC has an address of .65 connected to an IP phone with an address of .33; and a laptop has an address of .193 connected to an access point with an address of .9.

.193.9.5.33.65.25.17.1192.168.2.0/24

Internet

For instance, the default gateway IP address is 192.168.2.1/24, the switch is 192.168.2.5/24, the server is 192.168.2.17/24, etc..

Notice that the assignable IP address ranges were deliberately allocated on subnetnetwork boundaries to simplify summarizing the group type. For instance, assume another switch with IP address 192.168.2.6 is added to the

network. To identify all switches in a network policy, the administrator could specify the summarized network address 192.168.x.4/30.

17.1.4

Redundancy in a Small Network

Another important part of network design is reliability. Even small businesses often rely heavily on their network for business operation. A failure of the network can be very costly.

In order to maintain a high degree of reliability, *redundancy* is required in the network design. Redundancy helps to eliminate single points of failure.

There are many ways to accomplish redundancy in a network. Redundancy can be accomplished by installing duplicate equipment, but it can also be accomplished by supplying duplicate network links for critical areas, as shown in the figure.

The diagram illustrates the use of redundant servers, links, switches, and routers in a network. Shown are four layers with an explanation of the redundancy achieved at each. The top layer has three servers and text reads: Redundant servers are available in case of server failure. The next layer shows that each server has two connections leading to two switches and text reads: Redundant links are present to provide alternate paths in case of a link failure. The next layer shows two switches connected to each other with each connected to all three servers above and text reads: Redundant switches are present in case of switch failure. The bottom layer shows two routers connected to each other with each connected to one of the switches and text reads: Redundant routers are available in case of router or route failure.

Redundant servers are available in case of server failure.Redundant links are present to provide alternate paths in case of a link failure.Redundant switches are present in case of switch failure.Redundant routers are available in case of router or route failure.

Small networks typically provide a single exit point toward the internet via one or more default gateways. If the router fails, the entire network loses connectivity to the internet. For this reason, it may be advisable for a small business to pay for a second service provider as backup.

17.1.5

Traffic Management

The goal for a good network design, even for a small network, is to enhance the productivity of the employees and minimize network downtime. The network administrator should consider the various types of traffic and their treatment in the network design.

The routers and switches in a small network should be configured to support real-time traffic, such as voice and video, in an appropriate manner relative to other data traffic. In fact, a good network design will implement quality of service (QoS) to classify traffic carefully according to priority during times of congestion, as shown in the figure.

The diagram shows how a router prioritizes network traffic. An arrow shows that traffic is sent to a router without any priority. The router then sends traffic to the backbone in order of priority. The router has layers of different traffic with varying levels of priority, the highest priority at the top. Voice is listed first with high priority, then SMTP with medium priority, then instant messaging with normal priority, and lastly FTP with low priority.

Traffic sent to router without any priorityBackbone NetworkTraffic sent to backbone in order of priority

Priority queuing has four queues. The high-priority queue is always emptied first.

17.2.1

Common Applications

The previous topic discussed the components of a small network, as well as some of the design considerations. These considerations are necessary when you are just setting up a network. After you have set it up, your network still needs certain types of applications and protocols in order to work.

The network is only as useful as the applications that are on it. There are two forms of software programs or processes that provide access to the network: network applications and application layer services.

Network Applications

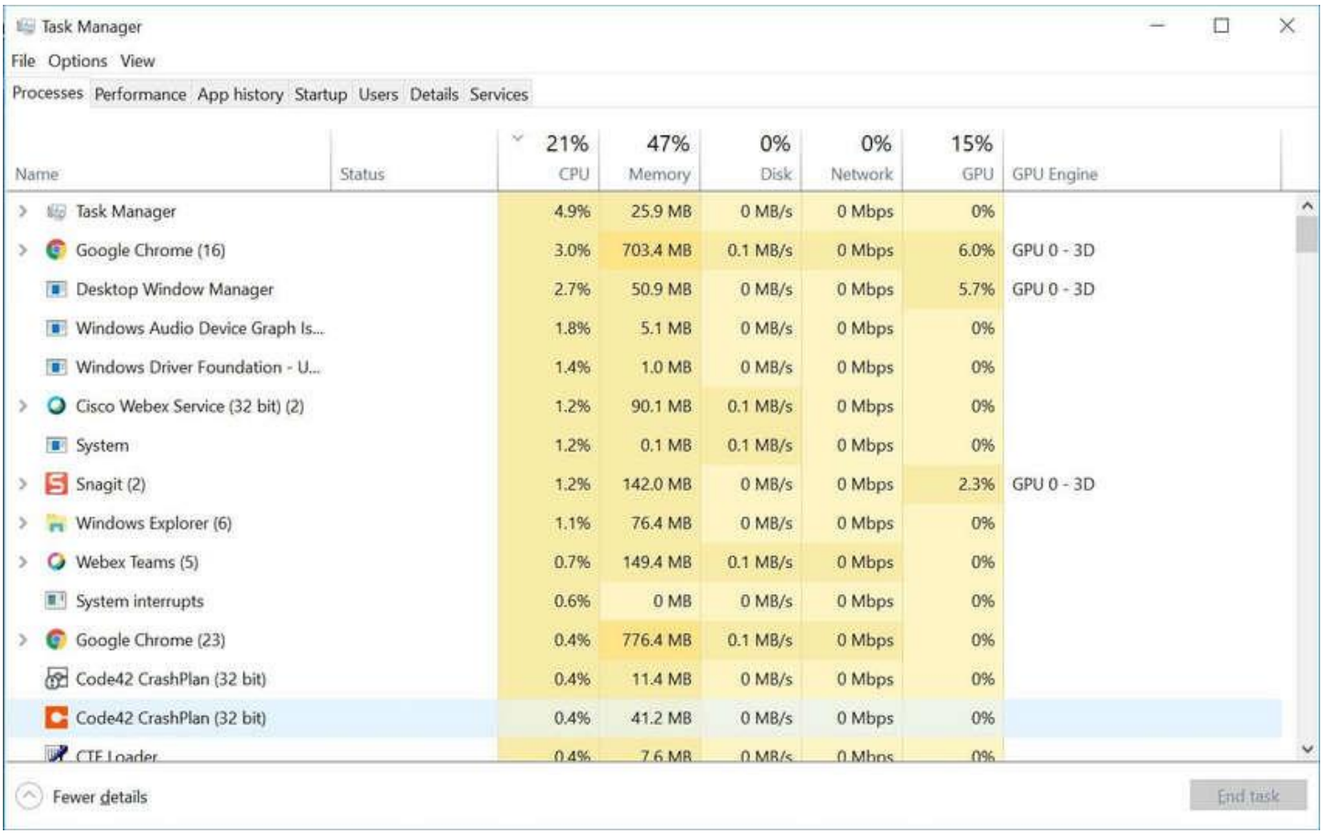
Applications are the software programs used to communicate over the network. Some end-user applications are network-aware, meaning that they implement application layer protocols and are able to communicate directly with the lower layers of the protocol stack. Email clients and web browsers are examples of this type of application.

Application Layer Services

Other programs may need the assistance of application layer services to use network resources like file transfer or network print spooling. Though transparent to an employee, these services are the programs that interface with the network and prepare the data for transfer. Different types of data, whether text, graphics or video, require different network services to ensure that they are properly prepared for processing by the functions occurring at the lower layers of the OSI model.

Each application or network service uses protocols, which define the standards and data formats to be used. Without protocols, the data network would not have a common way to format and direct data. In order to understand the function of various network services, it is necessary to become familiar with the underlying protocols that govern their operation.

Use the Task Manager to view the current applications, processes, and services running on a Windows PC, as shown in the figure.



17.2.2

Common Protocols

Most of a technician’s work, in either a small or a large network, will in some way be involved with network protocols. Network protocols support the applications and services used by employees in a small network.

Network administrators commonly require access to network devices and servers. The two most common remote access solutions are Telnet and Secure Shell (SSH). SSH service is a secure alternative to Telnet. When connected, administrators can access the SSH server device as though they were logged in locally.

SSH is used to establish a secure remote access connection between an SSH client and other SSH-enabled devices:

- **Network device** - The network device (e.g., router, switch, access point, etc.) must support SSH to provide remote access SSH server services to clients.
- **Server** - The server (e.g., web server, email server, etc.) must support remote access SSH server services to clients.

Network administrators must also support common network servers and their required related network protocols, as shown in the figure.

Web ServerEmail ServerFTP ServerDHCP ServerDNS Server

Click each button for more information about common network servers and their required related network protocols.

Web Server

Email Server

FTP Server

DHCP Server

DNS Server

Web Server

- Web clients and web servers exchange web traffic using the Hypertext Transfer Protocol (HTTP).
- Hypertext Transfer Protocol Secure (HTTPS) is used for secure web communication.

Note: A server could provide multiple network services. For instance, a server could be an email, FTP, and SSH server.

These network protocols comprise the fundamental toolset of a network professional. Each of these network protocols define:

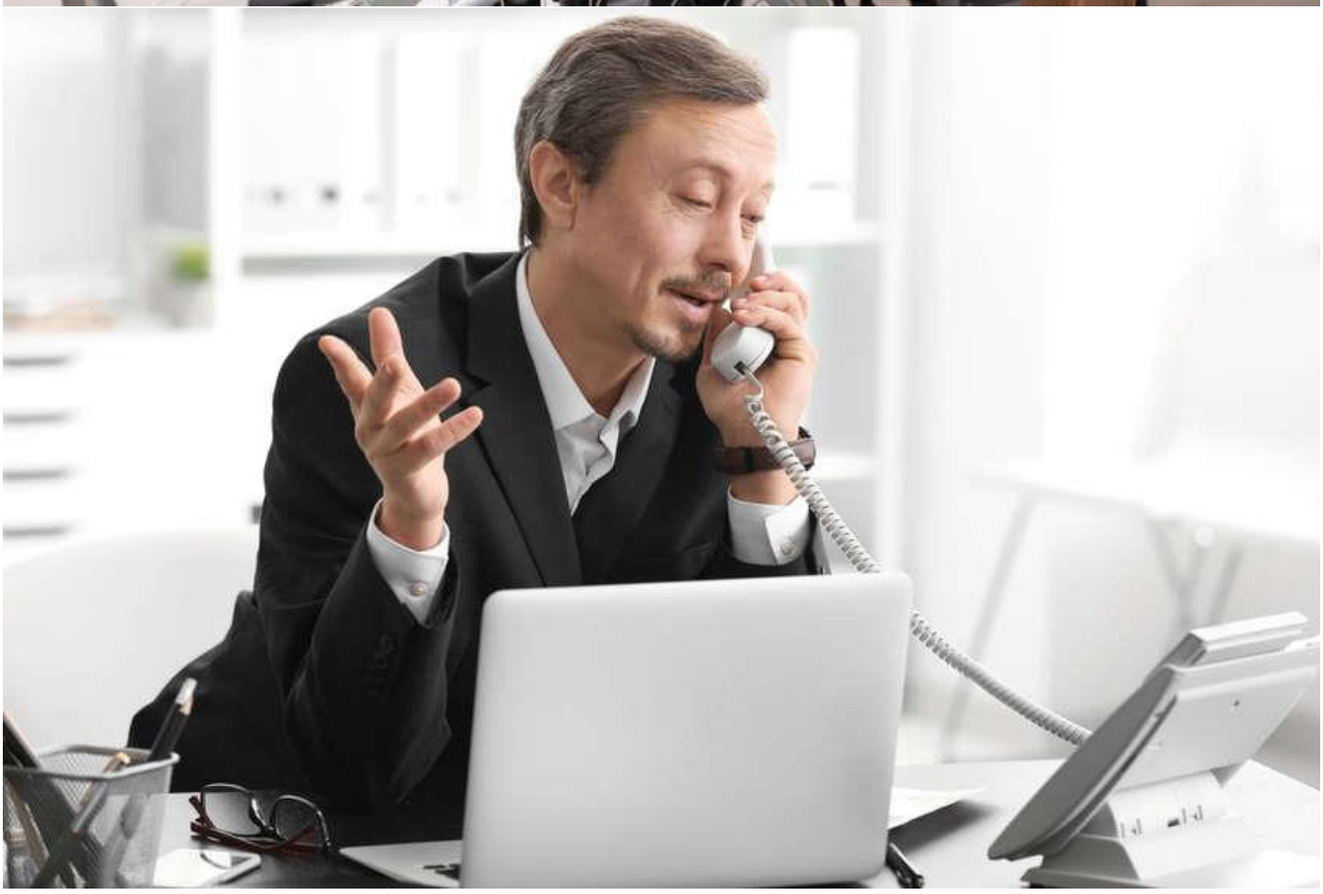
- Processes on either end of a communication session
- Types of messages
- Syntax of the messages
- Meaning of informational fields
- How messages are sent and the expected response
- Interaction with the next lower layer

Many companies have established a policy of using secure versions (e.g., SSH, SFTP, and HTTPS) of these protocols whenever possible.

17.2.3

Voice and Video Applications

Businesses today are increasingly using IP telephony and streaming media to communicate with customers and business partners. Many organizations are enabling their employees to work remotely. As the figure shows, many of their users still require access to corporate software and files, as well as support for voice and video applications.



The network administrator must ensure the proper equipment is installed in the network and that the network devices are configured to ensure priority delivery.

Click each button for more information about the factors that a small network administrator must consider when supporting real-time applications.

Infrastructure

VoIP

IP Telephony

Real-Time Applications

Infrastructure

- The network infrastructure must support the real-time applications.
- Existing devices and cabling must be tested and validated.
- Newer networking products may be required.

17.3.1

Small Network Growth

If your network is for a small business, presumably, you want that business to grow, and your network to grow along with it. This is called scaling a network, and there are some best practices for doing this.

Growth is a natural process for many small businesses, and their networks must grow accordingly. Ideally, the network administrator has enough lead-time to make intelligent decisions about growing the network in alignment with the growth of the company.

To scale a network, several elements are required:

- **Network documentation** - Physical and logical topology
- **Device inventory** - List of devices that use or comprise the network
- **Budget** - Itemized IT budget, including fiscal year equipment purchasing budget
- **Traffic analysis** - Protocols, applications, and services and their respective traffic requirements should be documented

These elements are used to inform the decision-making that accompanies the scaling of a small network.

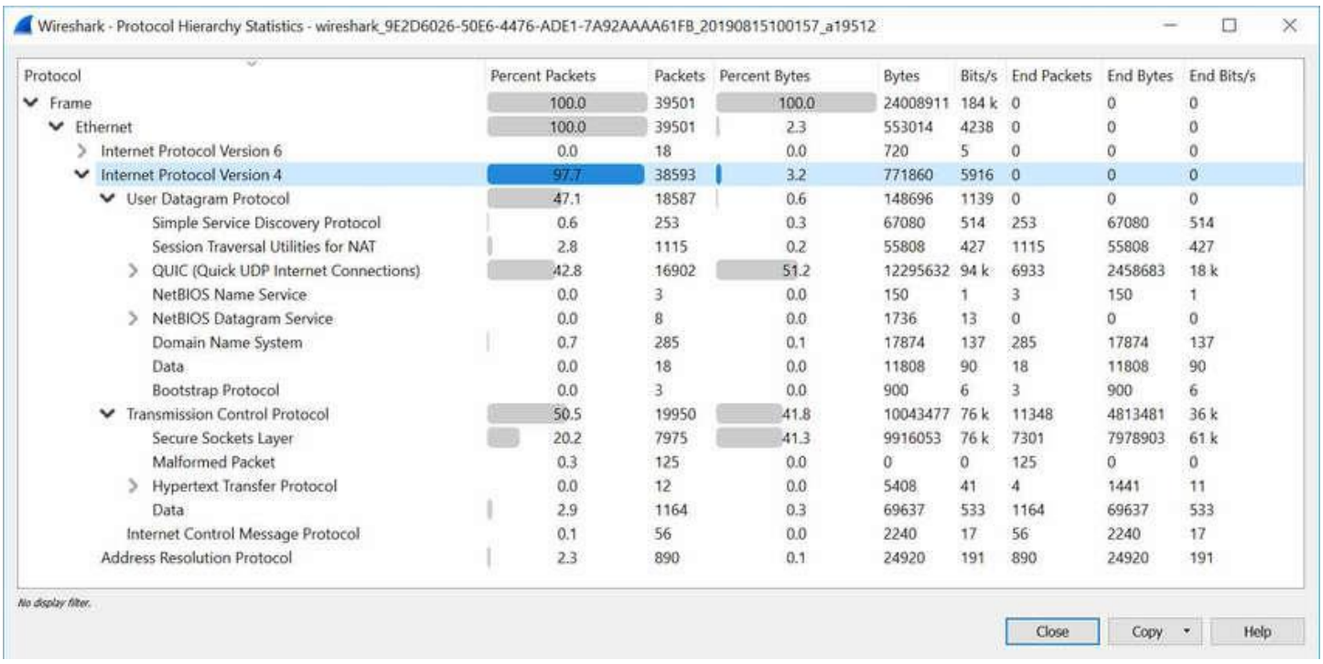
17.3.2

Protocol Analysis

As the network grows, it becomes important to determine how to manage network traffic. It is important to understand the type of traffic that is crossing the network as well as the current traffic flow. There are several network management tools that can be used for this purpose. However, a simple protocol analyzer such as Wireshark can also be used.

For instance, running Wireshark on several key hosts can reveal the types of network traffic flowing through the network. The following figure displays Wireshark protocol hierarchy statistics for a Windows host on a small network.

screen capture of Wireshark protocol hierarchy statistics for traffic captured by a host



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	39501	100.0	24008911	184 k	0	0	0
Ethernet	100.0	39501	2.3	553014	4238	0	0	0
Internet Protocol Version 6	0.0	18	0.0	720	5	0	0	0
Internet Protocol Version 4	97.7	38593	3.2	771860	5916	0	0	0
User Datagram Protocol	47.1	18587	0.6	148696	1139	0	0	0
Simple Service Discovery Protocol	0.6	253	0.3	67080	514	253	67080	514
Session Traversal Utilities for NAT	2.8	1115	0.2	55808	427	1115	55808	427
QUIC (Quick UDP Internet Connections)	42.8	16902	51.2	12295632	94 k	6933	2458683	18 k
NetBIOS Name Service	0.0	3	0.0	150	1	3	150	1
NetBIOS Datagram Service	0.0	8	0.0	1736	13	0	0	0
Domain Name System	0.7	285	0.1	17874	137	285	17874	137
Data	0.0	18	0.0	11808	90	18	11808	90
Bootstrap Protocol	0.0	3	0.0	900	6	3	900	6
Transmission Control Protocol	50.5	19950	41.8	10043477	76 k	11348	4813481	36 k
Secure Sockets Layer	20.2	7975	41.3	9916053	76 k	7301	7978903	61 k
Malformed Packet	0.3	125	0.0	0	0	125	0	0
Hypertext Transfer Protocol	0.0	12	0.0	5408	41	4	1441	11
Data	2.9	1164	0.3	69637	533	1164	69637	533
Internet Control Message Protocol	0.1	56	0.0	2240	17	56	2240	17
Address Resolution Protocol	2.3	890	0.1	24920	191	890	24920	191

The screen capture reveals the host is using IPv6 and IPv4 protocols. The IPv4 specific output also reveals that the host has used DNS, SSL, HTTP, ICMP, and other protocols.

To determine traffic flow patterns, it is important to do the following:

- Capture traffic during peak utilization times to get a good representation of the different traffic types.
- Perform the capture on different network segments and devices as some traffic will be local to a particular segment.

Information gathered by the protocol analyzer is evaluated based on the source and destination of the traffic, as well as the type of traffic being sent. This analysis can be used to make decisions on how to manage the traffic more efficiently. This can be done by reducing unnecessary traffic flows or changing flow patterns altogether by moving a server, for example.

Sometimes, simply relocating a server or service to another network segment improves network performance and accommodates the growing traffic needs. At other times, optimizing the network performance requires major network redesign and intervention.

17.3.3

Employee Network Utilization

In addition to understanding changing traffic trends, a network administrator must be aware of how network use is changing. Many operating systems provide built-in tools to display such information. For example, a Windows host provides tools such as the Task Manager, Event Viewer, and Data Usage tools.

These tools can be used to capture a “snapshot” of information such as the following:

- OS and OS Version
- CPU utilization
- RAM utilization
- Drive utilization
- Non-Network applications
- Network applications

Documenting snapshots for employees in a small network over a period of time is very useful to identify evolving protocol requirements and associated traffic flows. A shift in resource utilization may require the network administrator to adjust network resource allocations accordingly.

The Windows 10 Data Usage tool is especially useful to determine which applications are using network services on a host. The Data Usage tool is accessed using **Settings > Network & Internet > Data usage > network interface** (from the last 30 days).

The example in the figure is displaying the applications running on a remote user Windows 10 host using the local Wi-Fi network connection.

screen capture of the Windows 10 Data Usage Tool showing usage from a local Wi-Fi connection

17.4.1

Verify Connectivity with Ping

Whether your network is small and new, or you are scaling an existing network, you will always want to be able to verify that your components are properly connected to each other and to the internet. This topic discusses some utilities that you can use to ensure that your network is connected.

The **ping** command is the most effective way to quickly test Layer 3 connectivity between a source and destination IP address. The command also displays various round-trip time statistics.

Specifically, the **ping** command uses the Internet Control Message Protocol (ICMP) echo request (ICMP Type 8) and echo reply (ICMP Type 0) messages. The **ping** command is available in most operating systems including Windows, Linux, macOS, and Cisco IOS.

On a Windows 10 host, the **ping** command sends four consecutive ICMP echo request messages and expects four consecutive ICMP echo replies from the destination.

For example, assume PC A pings PC B. As shown in the figure, the PC A Windows host sends four consecutive ICMP echo request messages. sometimes referred to as an ICMP echo, to PC B (i.e., 203.0.113.8).

The diagram shows host PC A, at address 192.168.10.10, using the ping ping 203.0.113.8 command from the command prompt to send four ICMP echo messages with a source IP of 198.168.10.10 (should read 192.168.10.10) and a destination IP of ping 203.0.113.8, which is host PC B on another network.

192.168.10.10PC APC BC:\>ping 203.0.113.8203.0.113.8

Source IP	Destination IP	ICMP
192.168.10.10	203.0.113.8	**Echo**

Internet

The destination host receives and processes the ICMP echos. As shown in the figure, PC B responds by sending four ICMP echo reply messages to PC A.

The diagram shows host PC B, at address 203.0.113.8, sending four ICMP echo replies with source IP 203.0.113.8 and destination IP 198.168.10.10 (should read 192.168.10.10) in response to a ping from host PC A at address 192.168.10.10.

PC APC BC:\>ping 203.0.113.8203.0.113.8192.168.10.10

SourceIP	Destination IP	ICMP
203.0.113.8	192.168.10.10	**Echo Replies**

Internet

As shown in the command output, PC A has received echo replies from PC-B verifying the Layer 3 network connection.

```
C:\Users\PC-A> ping 10.1.1.10

Pinging 10.1.1.10 with 32 bytes of data:

Reply from 10.1.1.10: bytes=32 time=47ms TTL=51

Reply from 10.1.1.10: bytes=32 time=60ms TTL=51

Reply from 10.1.1.10: bytes=32 time=53ms TTL=51

Reply from 10.1.1.10: bytes=32 time=50ms TTL=51

Ping statistics for 10.1.1.10:

    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

    Minimum = 47ms, Maximum = 60ms, Average = 52ms

C:\Users\PC-A>
```

The output validates Layer 3 connectivity between PC A and PC B.

A Cisco IOS **ping** command output varies from a Windows host. For instance, the IOS ping sends five ICMP echo messages, as shown in the output.

```
R1# ping 10.1.1.10

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.10, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms

R1#
```

Notice the **!!!!!** output characters. The IOS **ping** command displays an indicator for each ICMP echo reply received. The table lists the most common output characters from the **ping** command.

IOS Ping Indicators

Table caption

Element	Description
!	• Exclamation mark indicates successful receipt of an echo reply message.

Table caption	
Element	**Description**
	• It validates a Layer 3 connection between source and destination.
.	• A period means that time expired waiting for an echo reply message. • This indicates a connectivity problem occurred somewhere along the path.
U	• Uppercase **U** indicates a router along the path responded with an ICMP Type 3 “destination unreachable” error mes • Possible reasons include the router does not know the direction to the destination network or it could not find the the destination network.

Note: Other possible ping replies include Q, M, ?, or &. However, the meaning of these are out of scope for this module.

17.4.2

Extended Ping

A standard **ping** uses the IP address of the interface closest to the destination network as the source of the **ping**. The source IP address of the **ping 10.1.1.10** command on R1 would be that of the G0/0/0 interface (i.e., 209.165.200.225), as illustrated in the example.

The diagram shows how a router uses a standard ping to ping a host by sending four consecutive ICMP echo messages sourced from the interface closest to the destination. Router R1 is connected to two networks: on the left is 192.168.10.0/24 on interface G0/0/1 with address .1 and on the right is network 209.165.200.224/30 on interface G0/0/0 with address .225. The latter network is connected to R2 which is connected to network 10.1.1.0/24 on which host PC B is attached with address .10. R1 is sending PC B four ICMP echo messages with a source IP of 209.165.200.225 and a destination IP of 10.1.1.10.

R2.10.1G0/0/0.10209.165.200.224 /30192.168.10.0 /2410.1.1.0/24.225G0/0/1PC APC BR1

Source IP	Destination IP	ICMP
209.165.200.225	10.1.1.10	**Echo**

The Cisco IOS offers an "extended" mode of the **ping** command. This mode enables the user to create special type of pings by adjusting parameters related to the command operation.

Extended ping is entered in privileged EXEC mode by typing **ping** without a destination IP address. You will then be given several prompts to customize the extended **ping**.

Note: Pressing **Enter** accepts the indicated default values.

For example, assume you wanted to test connectivity from the R1 LAN (i.e., 192.168.10.0/24) to the 10.1.1.0 LAN. This could be verified from the PC A. However, an extended **ping** could be configured on R1 to specify a different source address.

As illustrated in the example, the source IP address of the extended **ping** command on R1 could be configured to use the G0/0/1 interface IP address (i.e., 192.168.10.1).

The diagram shows how a router uses an extended ping command to ping a host by sending four consecutive ICMP echo messages with a specified source IP address. Router R1 is connected to two networks: on the left is 192.168.10.0/24 on interface G0/0/1 with address .1 and on the right is network 209.165.200.224/30 on interface G0/0/0 with address .225. The latter network is connected to R2 which is connected to network 10.1.1.0/24 on which host PC B is attached with address .10. R1 is sending PC B four ICMP echo messages with a source IP of 192.168.10.1 and a destination IP of 10.1.1.10.

R2.10.1G0/0/0.10209.165.200.224 /30192.168.10.0 /2410.1.1.0/24.225G0/0/1PC APC BR1

Source IP	Destination IP	ICMP
192.168.10.1	10.1.1.10	**Echo**

The following command output configures an extended **ping** on R1 and specifies the source IP address to be that of the G0/0/1 interface (i.e., 192.168.10.1).

```
R1# ping
```

```
Protocol [ip]:
```

```
Target IP address: 10.1.1.10
```

```
Repeat count [5]:
```

```
Datagram size [100]:
```

```
Timeout in seconds [2]:
```

```
Extended commands [n]: y
```

```
Ingress ping [n]:
```

```
Source address or interface: 192.168.10.1
```

```
DSCP Value [0]:
```

```
Type of service [0]:
```

```
Set DF bit in IP header? [no]:
```

```
Validate reply data? [no]:
```

```
Data pattern [0x0000ABCD]:
```

```
Loose, Strict, Record, Timestamp, Verbose[none]:
```

```
Sweep range of sizes [n]:
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
```

```
Packet sent with a source address of 192.168.10.1
```

```
!!!!!
```

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

```
R1#
```

Note: The **ping ipv6** command is used for IPv6 extended pings.

17.4.3

Verify Connectivity with Traceroute

The **ping** command is useful to quickly determine if there is a Layer 3 connectivity problem. However, it does not identify where the problem is located along the path.

Traceroute can help locate Layer 3 problem areas in a network. A trace returns a list of hops as a packet is routed through a network. It could be used to identify the point along the path where the problem can be found.

The syntax of the trace command varies between operating systems, as illustrated in the figure.

The diagram shows the difference between the trace command as issued from a Windows host versus a Cisco IOS router. The network topology consists of a host PC A connected to a switch connected to router R1 connected to router R2 connected to router R3 connected to a switch connected to host PC B. PC A, at IP address 192.168.10.10, is issuing the following command from a Windows command prompt: C:>:tracert 10.1.1.10. R1 is issuing the following command from the Cisco IOS CLI: R#traceroute 10.1.1.10.

Windows and Cisco IOS Trace Commands

PC APC B10.1.1.10192.168.10.10.1R3R2R1

Trace from Windows host

```
c:\>:tracert 10.1.1.10
```

Trace from a Cisco IOS router

```
R# traceroute 10.1.1.10
```

The following is a sample output of **tracert** command on a Windows 10 host.

```
C:\Users\PC-A> tracert 10.1.1.10
```

```
Tracing route to 10.1.10 over a maximum of 30 hops:
```

```
  1     2 ms     2 ms     2 ms  192.168.10.1
```

```
  2     *        *        *     Request timed out.
```

```
  3     *        *        *     Request timed out.
```

```
  4     *        *        *     Request timed out.
```

```
^C
```

```
C:\Users\PC-A>
```

Note: Use **Ctrl-C** to interrupt a **tracert** in Windows.

The only successful response was from the gateway on R1. Trace requests to the next hop timed out as indicated by the asterisk (*), meaning that the next hop router did not respond. The timed out requests indicate that there is a failure in the internetwork beyond the LAN, or that these routers have been configured to not respond to echo requests used in the trace. In this example there appears to be a problem between R1 and R2.

A Cisco IOS **traceroute** command output varies from the Windows **tracert** command. For instance, refer to the following topology.

The diagram shows a network topology with the IP addressing of router interfaces and a traceroute command issued from a Cisco IOS router. The topology consists of the following devices and networks, from left to right. A

switch on network 192.168.10.0/24 is connected to router R1 at an interface with an address of .1. R1 is connected to router R2 by network 209.165.200.224/30. The interface on R1 has an address of .225 and the interface on R2 has an address of .226. R2 is connected to router R3 by network 209.165.200.228/30. The interface on R2 has an address of .229 and the interface on R3 has an address of .230. R3 is connected to a switch which is connected to host PC B with address 10.1.1.10. R1 is issuing the following trace command from the CLI: R1# traceroute 10.1.1.10.

PC B10.1.1.10192.168.10.0 /24.1R3R1.225.226.229R2.230209.165.200.224 /30209.165.200.228 /30R1

Trace from a Cisco IOS router

```
R1# traceroute 10.1.1.10
```

The following is a sample output of traceroute command from R1.

```
R1# traceroute 10.1.1.10

Type escape sequence to abort.

Tracing the route to 10.1.1.10

VRF info: (vrf in name/id, vrf out name/id)

  1 209.165.200.226 1 msec 0 msec 1 msec

  2 209.165.200.230 1 msec 0 msec 1 msec

  3 10.1.1.10 1 msec 0 msec

R1#
```

In this example, the trace validated that it could successfully reach PC B.

Timeouts indicate a potential problem. For instance, if the 10.1.1.10 host was not available, the **traceroute** command would display the following output.

```
R1# traceroute 10.1.1.10

Type escape sequence to abort.

Tracing the route to 10.1.1.10

VRF info: (vrf in name/id, vrf out name/id)

  1 209.165.200.226 1 msec 0 msec 1 msec

  2 209.165.200.230 1 msec 0 msec 1 msec

  3  *  *  *

  4  *  *  *

  5  *
```

Use **Ctrl-Shift-6** to interrupt a **traceroute** in Cisco IOS.

Note: Windows implementation of traceroute (tracert) sends ICMP Echo Requests. Cisco IOS and Linux use UDP with an invalid port number. The final destination will return an ICMP port unreachable message.

17.4.4

Extended Traceroute

Like the extended **ping** command, there is also an extended **traceroute** command. It allows the administrator to adjust parameters related to the command operation. This is helpful in locating the problem when troubleshooting routing loops, determining the exact next-hop router, or determining where packets are getting dropped or denied by a router or firewall.

The Windows **tracert** command allows the input of several parameters through options in the command line. However, it is not guided like the extended traceroute IOS command. The following output displays the available options for the Windows **tracert** command.

```
C:\Users\PC-A> tracert /?

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
               [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -d                  Do not resolve addresses to hostnames.
    -h maximum_hops     Maximum number of hops to search for target.
    -j host-list        Loose source route along host-list (IPv4-only).
    -w timeout          Wait timeout milliseconds for each reply.
    -R                  Trace round-trip path (IPv6-only).
    -S srcaddr          Source address to use (IPv6-only).
    -4                  Force using IPv4.
    -6                  Force using IPv6.

C:\Users\PC-A>
```

The Cisco IOS extended **traceroute** option enables the user to create a special type of trace by adjusting parameters related to the command operation. Extended traceroute is entered in privileged EXEC mode by typing **traceroute** without a destination IP address. IOS will guide you through the command options by presenting a number of prompts related to the setting of all the different parameters.

Note: Pressing **Enter** accepts the indicated default values.

For example, assume you want to test connectivity to PC B from the R1 LAN. Although this could be verified from PC A, an extended **traceroute** could be configured on R1 to specify a different source address.

The diagram shows a network topology with the IP addressing of router interfaces and an extended traceroute command issued from a Cisco IOS router. The topology consists of the following devices and networks, from left to right. A switch on network 192.168.10.0/24 is connected to router R1 at an interface with an address of .1. R1 is connected to router R2 by network 209.165.200.224/30. The interface on R1 has an address of .225 and the interface on R2 has an address of .226. R2 is connected to router R3 by network 209.165.200.228/30. The interface on R2 has an address of .229 and the interface on R3 has an address of .230. R3 is connected to a switch which is connected to host PC B with address 10.1.1.10. R1 is issuing the following trace command from the CLI: R1# traceroute.

PC B192.168.10.0/24209.165.200.224/30209.165.200.228/3010.1.1.10R1R3R2.1.225.226.229.230

Extended trace from a Cisco IOS router

```
R1# traceroute
```

As illustrated in the example, the source IP address of the extended **traceroute** command on R1 could be configured to use the R1 LAN interface IP address (i.e., 192.168.10.1).

```
R1# traceroute

Protocol [ip]:

Target IP address: 10.1.1.10

Ingress traceroute [n]:

Source address: 192.168.10.1

DSCP Value [0]:

Numeric display [n]:

Timeout in seconds [3]:

Probe count [3]:

Minimum Time to Live [1]:

Maximum Time to Live [30]:

Port Number [33434]:

Loose, Strict, Record, Timestamp, Verbose[none]:

Type escape sequence to abort.

Tracing the route to 192.168.10.10

VRF info: (vrf in name/id, vrf out name/id)

  1 209.165.200.226 1 msec 1 msec 1 msec

  2 209.165.200.230 0 msec 1 msec 0 msec

  3  *

    10.1.1.10 2 msec 2 msec

R1#
```

17.4.5

Network Baseline

One of the most effective tools for monitoring and troubleshooting network performance is to establish a network baseline. Creating an effective network performance baseline is accomplished over a period of time. Measuring performance at varying times and loads will assist in creating a better picture of overall network performance.

The output derived from network commands contributes data to the network baseline. One method for starting a baseline is to copy and paste the results from an executed **ping**, **trace**, or other relevant commands into a text file. These text files can be time stamped with the date and saved into an archive for later retrieval and comparison.

Among items to consider are error messages and the response times from host to host. If there is a considerable increase in response times, there may be a latency issue to address.

For example, the following **ping** output was captured and pasted into a text file.

August 19, 2019 at 08:14:43

```
C:\Users\PC-A> ping 10.1.1.10
```

```
Pinging 10.1.1.10 with 32 bytes of data:
```

```
Reply from 10.1.1.10: bytes=32 time<1ms TTL=64
```

```
Reply from 10.1.1.10: bytes=32 time<1ms TTL=64
```

```
Reply from 10.1.1.10: bytes=32 time<1ms TTL=64
```

```
Reply from 10.1.1.10: bytes=32 time<1ms TTL=64
```

```
Ping statistics for 10.1.1.10:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\Users\PC-A>
```

Notice the **ping** round-trip times are less than 1 ms.

A month later, the ping is repeated and captured.

September 19, 2019 at 10:18:21

```
C:\Users\PC-A> ping 10.1.1.10
```

```
Pinging 10.1.1.10 with 32 bytes of data:
```

```
Reply from 10.1.1.10: bytes=32 time=50ms TTL=64
```

```
Reply from 10.1.1.10: bytes=32 time=49ms TTL=64
```

```
Reply from 10.1.1.10: bytes=32 time=46ms TTL=64
```

```
Reply from 10.1.1.10: bytes=32 time=47ms TTL=64

Ping statistics for 10.1.1.10:

    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

    Minimum = 46ms, Maximum = 50ms, Average = 48ms

C:\Users\PC-A>
```

Notice this time that the **ping** round-trip times are much longer indicating a potential problem.

Corporate networks should have extensive baselines; more extensive than we can describe in this course. Professional-grade software tools are available for storing and maintaining baseline information. In this course, we cover a few basic techniques and discuss the purpose of baselines.

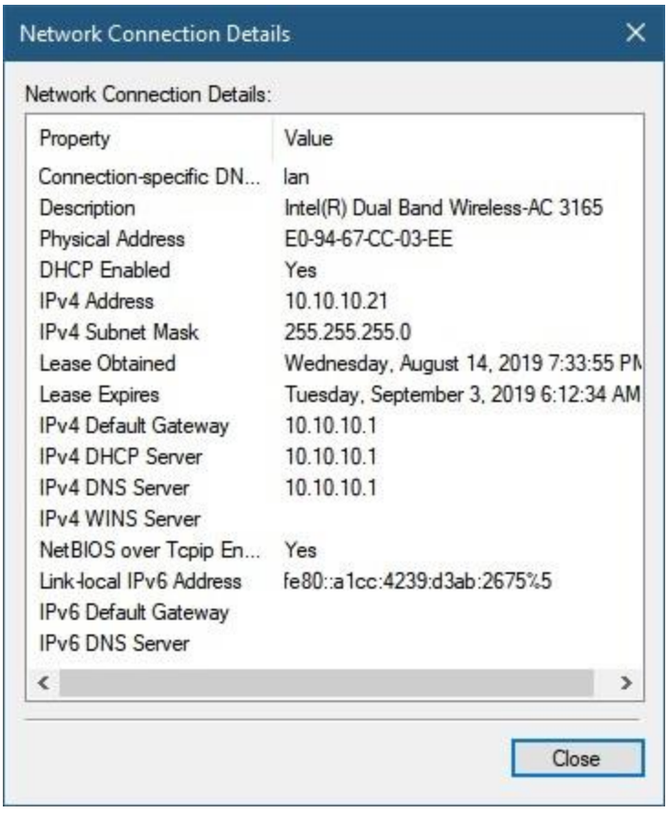
Cisco's best practices for baseline processes can be found by searching the internet for "Baseline Process Best Practices".

17.5.1

IP Configuration on a Windows Host

If you have used any of the tools in the previous topic to verify connectivity and found that some part of your network is not working as it should, now is the time to use some commands to troubleshoot your devices. Host and IOS commands can help you determine if the problem is with the IP addressing of your devices, which is a common network problem.

Checking the IP addressing on host devices is a common practice in networking for verifying and troubleshooting end-to-end connectivity. In Windows 10, you can access the IP address details from the **Network and Sharing Center**, as shown in the figure, to quickly view the four important settings: address, mask, router, and DNS.



However, network administrators typically view the IP addressing information on a Windows host by issuing the **ipconfig** command at the command line of a Windows computer, as shown in the sample output.

```
C:\Users\PC-A> ipconfig

Windows IP Configuration

(Output omitted)

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :

   Link-local IPv6 Address . . . . . : fe80::a4aa:2dd1:ae2d:a75e%16

   IPv4 Address. . . . . . . . . . . : 192.168.10.10

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : 192.168.10.1

(Output omitted)
```

Use the **ipconfig /all** command to view the MAC address, as well as a number of details regarding the Layer 3 addressing of the device, as shown in the example output.

```
C:\Users\PC-A> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : PC-A-00H20

   Primary Dns Suffix  . . . . . . . : cisco.com

   Node Type . . . . . . . . . . . . : Hybrid

   IP Routing Enabled. . . . . . . . : No

   WINS Proxy Enabled. . . . . . . . : No

   DNS Suffix Search List. . . . . . : cisco.com

(Output omitted)

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :

   Description . . . . . . . . . . . : Intel(R) Dual Band Wireless-AC 8265

   Physical Address. . . . . . . . . : F8-94-C2-E4-C5-0A

   DHCP Enabled. . . . . . . . . . . : Yes
```

```
   Autoconfiguration Enabled . . . . : Yes

   Link-local IPv6 Address . . . . . : fe80::a4aa:2dd1:ae2d:a75e%16(Preferred)

   IPv4 Address. . . . . . . . . . . : 192.168.10.10(Preferred)

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Lease Obtained. . . . . . . . . . : August 17, 2019 1:20:17 PM

   Lease Expires . . . . . . . . . . : August 18, 2019 1:20:18 PM

   Default Gateway . . . . . . . . . : 192.168.10.1

   DHCP Server . . . . . . . . . . . : 192.168.10.1

   DHCPv6 IAID . . . . . . . . . . . : 100177090

   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-21-F3-76-75-54-E1-AD-DE-DA-9A

   DNS Servers . . . . . . . . . . . : 192.168.10.1

   NetBIOS over Tcpip. . . . . . . . : Enabled
```

If a host is configured as a DHCP client, the IP address configuration can be renewed using the **ipconfig /release** and **ipconfig /renew** commands, as shown in the sample output.

```
C:\Users\PC-A> ipconfig /release

(Output omitted)

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :

   Link-local IPv6 Address . . . . . : fe80::a4aa:2dd1:ae2d:a75e%16

   Default Gateway . . . . . . . . . :

(Output omitted)

C:\Users\PC-A> ipconfig /renew

(Output omitted)

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :

   Link-local IPv6 Address . . . . . : fe80::a4aa:2dd1:ae2d:a75e%16
```

```
   IPv4 Address. . . . . . . . . . . : 192.168.1.124
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1
(Output omitted)
C:\Users\PC-A>
```

The DNS Client service on Windows PCs also optimizes the performance of DNS name resolution by storing previously resolved names in memory. The **ipconfig /displaydns** command displays all of the cached DNS entries on a Windows computer system, as shown in the example output.

```
C:\Users\PC-A> ipconfig /displaydns
Windows IP Configuration
(Output omitted)
    netacad.com
    ----------------------------------------
    Record Name . . . . . : netacad.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 602
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 54.165.95.219
(Output omitted)
```

17.5.2

IP Configuration on a Linux Host

Verifying IP settings using the GUI on a Linux machine will differ depending on the Linux distribution (distro) and desktop interface. The figure shows the **Connection Information** dialog box on the Ubuntu distro running the Gnome desktop.

On the command line, network administrators use the **ifconfig** command to display the status of the currently active interfaces and their IP configuration, as shown in the output.

```
[analyst@secOps ~]$ ifconfig
```

```
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:b5:d6:cb
```

```
          inet addr: 10.0.2.15  Bcast:10.0.2.255  Mask: 255.255.255.0
```

```
          inet6 addr: fe80::57c6:ed95:b3c9:2951/64 Scope:Link
```

```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
          RX packets:1332239 errors:0 dropped:0 overruns:0 frame:0
```

```
          TX packets:105910 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:1000
```

```
          RX bytes:1855455014 (1.8 GB)  TX bytes:13140139 (13.1 MB)
```

```
lo: flags=73  mtu 65536
```

```
        inet 127.0.0.1  netmask 255.0.0.0
```

```
        inet6 ::1  prefixlen 128  scopeid 0x10
```

```
        loop  txqueuelen 1000  (Local Loopback)
```

```
        RX packets 0  bytes 0 (0.0 B)
```

```
        RX errors 0  dropped 0  overruns 0  frame 0
```

```
        TX packets 0  bytes 0 (0.0 B)
```

```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

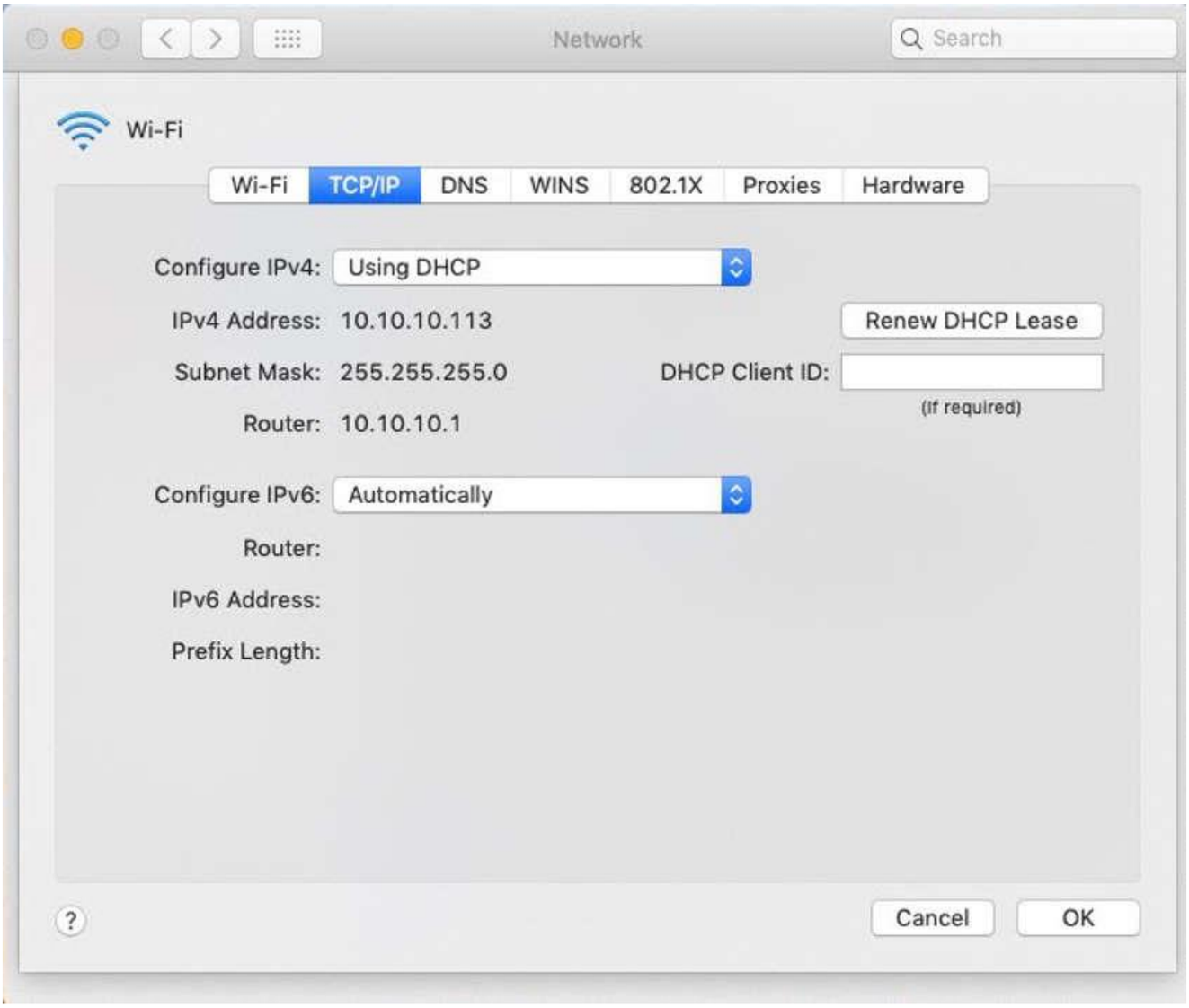
The Linux **ip address** command is used to display addresses and their properties. It can also be used to add or delete IP addresses.

Note: The output displayed may vary depending on the Linux distribution.

17.5.3

IP Configuration on a macOS Host

In the GUI of a Mac host, open **Network Preferences > Advanced** to get the IP addressing information, as shown in the figure.



However, the **ifconfig** command can also be used to verify the interface IP configuration a shown in the output.

```
MacBook-Air:~ Admin$ ifconfig en0
```

```
en0: flags=8863 mtu 1500
```

```
        ether c4:b3:01:a0:64:98
```

```
        inet6 fe80::c0f:1bf4:60b1:3adb%en0 prefixlen 64 secured scopeid 0x5
        inet 10.10.10.113 netmask 0xffffff00 broadcast 10.10.10.255
        nd6 options=201
        media: autoselect
        status: active
MacBook-Air:~ Admin$
```

Other useful macOS commands to verify the host IP settings include **networksetup -listallnetworkservices** and the **networksetup -getinfo <***network service***>,** as shown in the following output.

```
MacBook-Air:~ Admin$ networksetup -listallnetworkservices
An asterisk (*) denotes that a network service is disabled.
iPhone USB
Wi-Fi
Bluetooth PAN
Thunderbolt Bridge
MacBook-Air:~ Admin$
MacBook-Air:~ Admin$ networksetup -getinfo Wi-Fi
DHCP Configuration
IP address: 10.10.10.113
Subnet mask: 255.255.255.0
Router: 10.10.10.1
Client ID:
IPv6: Automatic
IPv6 IP address: none
IPv6 Router: none
Wi-Fi ID: c4:b3:01:a0:64:98
MacBook-Air:~ Admin$
```

The arp Command

The **arp** command is executed from the Windows, Linux, or Mac command prompt. The command lists all devices currently in the ARP cache of the host, which includes the IPv4 address, physical address, and the type of addressing (static/dynamic), for each device.

For instance, refer to the topology in the figure.

five hosts with IP addresses 10.0.0.1/24, 10.0.0.2/24, 10.0.0.3/24, 10.0.0.4/24, and 10.0.0.5/24 are connected to a switch connected to a router with an IP address of 10.0.0.254/24

10.0.0.254/2410.0.0.1/2410.0.0.2/2410.0.0.3/2410.0.0.4/2410.0.0.5/24PC-A

The output of the **arp -a** command on the Windows PC-A host is displayed.

```
C:\Users\PC-A> arp -a
```

```
Interface: 192.168.93.175 --- 0xc
```

Internet Address	Physical Address	Type
10.0.0.2	d0-67-e5-b6-56-4b	dynamic
10.0.0.3	78-48-59-e3-b4-01	dynamic
10.0.0.4	00-21-b6-00-16-97	dynamic
10.0.0.254	00-15-99-cd-38-d9	dynamic

The **arp -a** command displays the known IP address and MAC address binding. Notice how IP address 10.0.0.5 is not included in the list. This is because the ARP cache only displays information from devices that have been recently accessed.

To ensure that the ARP cache is populated, **ping** a device so that it will have an entry in the ARP table. For instance, if PC-A pinged 10.0.0.5, then the ARP cache would contain an entry for that IP address.

The cache can be cleared by using the **netsh interface ip delete arpcache** command in the event the network administrator wants to repopulate the cache with updated information.

Note: You may need administrator access on the host to be able to use the **netsh interface ip delete arpcache** command.

17.5.5

Common show Commands Revisited

In the same way that commands and utilities are used to verify a host configuration, commands can be used to verify the interfaces of intermediary devices. The Cisco IOS provides commands to verify the operation of router and switch interfaces.

The Cisco IOS CLI **show** commands display relevant information about the configuration and operation of the device. Network technicians use **show** commands extensively for viewing configuration files, checking the status of device interfaces and processes, and verifying the device operational status. The status of nearly every process or function of the router can be displayed using a **show** command.

Commonly used **show** commands and when to use them are listed in the table.

Command	Useful for …
show running-config	To verify the current configuration and settings
show interfaces	To verify the interface status and see if there are any error messages
show ip interface	To verify the Layer 3 information of an interface
show arp	To verify the list of known hosts on the local Ethernet LANs
show ip route	To verify the Layer 3 routing information
show protocols	To verify which protocols are operational
show version	To verify the memory, interfaces, and licences of the device

Click the buttons to see example output from each of these show commands.Note: The output of some commands has been edited to focus on pertinent settings and reduce content.
show running-config

show interfaces

show ip interface

show arp

show ip route

show protocols

show version

show running-config

Verifies the current configuration and settings

```
R1# show running-config
```

```
(Output omitted)
```

```
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname R1
!
interface GigabitEthernet0/0/0
 description Link to R2
 ip address 209.165.200.225 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 description Link to LAN
 ip address 192.168.10.1 255.255.255.0
 negotiation auto
```

```
!
router ospf 10
 network 192.168.10.0 0.0.0.255 area 0
 network 209.165.200.224 0.0.0.3 area 0
!
banner motd ^C Authorized access only! ^C
!
line con 0
 password 7 14141B180F0B
 login
line vty 0 4
 password 7 00071A150754
 login
 transport input telnet ssh
!
end
R1#
```

17.5.6

The show cdp neighbors Command

There are several other IOS commands that are useful. The Cisco Discovery Protocol (CDP) is a Cisco proprietary protocol that runs at the data link layer. Because CDP operates at the data link layer, two or more Cisco network devices, such as routers that support different network layer protocols, can learn about each other even if Layer 3 connectivity has not been established.

When a Cisco device boots, CDP starts by default. CDP automatically discovers neighboring Cisco devices running CDP, regardless of which Layer 3 protocol or suites are running. CDP exchanges hardware and software device information with its directly connected CDP neighbors.

CDP provides the following information about each CDP neighbor device:

- **Device identifiers** - The configured host name of a switch, router, or other device
- **Address list** - Up to one network layer address for each protocol supported
- **Port identifier** - The name of the local and remote port in the form of an ASCII character string, such as FastEthernet 0/0
- **Capabilities list** - For example, whether a specific device is a Layer 2 switch or a Layer 3 switch
- **Platform** - The hardware platform of the device--for example, a Cisco 1841 series router.

Refer to the topology and the **show cdp neighbor** command output.

router R3 is connected via interface G0/0/1 to switch S3 at port F0/5 which is connected to switch S4

G0/0/1F0/5R3S3S4

```
R3# show cdp neighbors

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

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

```
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID

S3               Gig 0/0/1         122              S I   WS-C2960+ Fas 0/5

Total cdp entries displayed : 1

R3#
```

The output displays that the R3 GigabitEthernet 0/0/1 interface is connected to the FastEthernet 0/5 interface of S3, which is a Cisco Catalyst 2960+ switch. Notice that R3 has not gathered information about S4. This is because CDP can only discover directly connected Cisco devices. S4 is not directly connected to R3 and therefore is not listed in the output.

The **show cdp neighbors detail** command reveals the IP address of a neighboring device, as shown in the output. CDP will reveal the IP address of the neighbor regardless of whether or not you can ping that neighbor. This command is very helpful when two Cisco routers cannot route across their shared data link. The **show cdp neighbors detail** command will help determine if one of the CDP neighbors has an IP configuration error.

As helpful as CDP is, it can also be a security risk because it can provide useful network infrastructure information to threat actors. For example, by default many IOS versions send CDP advertisements out all enabled ports. However, best practices suggest that CDP should be enabled only on interfaces that are connecting to other infrastructure Cisco devices. CDP advertisements should be disabled on user-facing ports.

Because some IOS versions send out CDP advertisements by default, it is important to know how to disable CDP. To disable CDP globally, use the global configuration command **no cdp run**. To disable CDP on an interface, use the interface command **no cdp enable**.

17.5.7

The show ip interface brief Command

One of the most frequently used commands is the **show ip interface brief** command. This command provides a more abbreviated output than the **show ip interface** command. It provides a summary of the key information for all the network interfaces on a router.

For example, the **show ip interface brief** output displays all interfaces on the router, the IP address assigned to each interface, if any, and the operational status of the interface.

```
R1# show ip interface brief

Interface              IP-Address      OK? Method Status                Protocol

GigabitEthernet0/0/0   209.165.200.225 YES manual up                    up

GigabitEthernet0/0/1   192.168.10.1    YES manual up                    up

Serial0/1/0            unassigned      NO  unset  down                  down

Serial0/1/1            unassigned      NO  unset  down                  down

GigabitEthernet0       unassigned      YES unset  administratively down down

R1#
```

Verify Switch Interfaces

The **show ip interface brief** command can also be used to verify the status of the switch interfaces, as shown in the output.

```
S1# show ip interface brief

Interface              IP-Address      OK? Method Status                Protocol

Vlan1                  192.168.254.250 YES manual up                    up

FastEthernet0/1        unassigned      YES unset  down                  down

FastEthernet0/2        unassigned      YES unset  up                    up

FastEthernet0/3        unassigned      YES unset  up                    up
```

The VLAN1 interface is assigned an IPv4 address of 192.168.254.250, has been enabled, and is operational.

The output also shows that the FastEthernet0/1 interface is down. This indicates that either no device is connected to the interface or the device that is connected has a network interface that is not operational.

In contrast, the output shows that the FastEthernet0/2 and FastEthernet0/3 interfaces are operational. This is indicated by both the status and protocol being shown as up.

17.6.1

Basic Troubleshooting Approaches

In the previous two topics, you learned about some utilities and commands that you can use to help identify problem areas in your network. This is an important part of troubleshooting. There are many ways to troubleshoot a network problem. This topic details a structured troubleshooting process that can help you to become a better network administrator. It also provides a few more commands to help you resolve problems. Network problems can be simple or complex, and can result from a combination of hardware, software, and connectivity issues. Technicians must be able to analyze the problem and determine the cause of the error before they can resolve the network issue. This process is called troubleshooting.

A common and efficient troubleshooting methodology is based on the scientific method.

The table shows the six main steps in the troubleshooting process.

Table caption

Step	Description
Step 1. Identify the Problem	• This is the first step in the troubleshooting process. • Although tools can be used in this step, a conversation with the user is of helpful.
Step 2. Establish a Theory of Probable Causes	• After the problem is identified, try to establish a theory of probable caus • This step often yields more than a few probable causes to the problem.
Step 3. Test the Theory to Determine Cause	• Based on the probable causes, test your theories to determine which one cause of the problem. • A technician will often apply a quick procedure to test and see if it solves problem. • If a quick procedure does not correct the problem, you might need to res problem further to establish the exact cause.

Table caption	
Step	**Description**
Step 4. Establish a Plan of Action and Implement the Solution	After you have determined the exact cause of the problem, establish a plan of act resolve the problem and implement the solution.
Step 5. Verify Solution and Implement Preventive Measures	• After you have corrected the problem, verify full functionality. • If applicable, implement preventive measures.
Step 6. Document Findings, Actions, and Outcomes	• In the final step of the troubleshooting process, document your findings, and outcomes. • This is very important for future reference.

To assess the problem, determine how many devices on the network are experiencing the problem. If there is a problem with one device on the network, start the troubleshooting process at that device. If there is a problem with all devices on the network, start the troubleshooting process at the device where all other devices are connected. You should develop a logical and consistent method for diagnosing network problems by eliminating one problem at a time.

17.6.2

Resolve or Escalate?

In some situations, it may not be possible to resolve the problem immediately. A problem should be escalated when it requires a manager decision, some specific expertise, or network access level unavailable to the troubleshooting technician.

For example, after troubleshooting, the technician concludes a router module should be replaced. This problem should be escalated for manager approval. The manager may have to escalate the problem again as it may require the approval of the financial department before a new module can be purchased.

A company policy should clearly state when and how a technician should escalate a problem.

17.6.3

The debug Command

OS processes, protocols, mechanisms and events generate messages to communicate their status. These messages can provide valuable information when troubleshooting or verifying system operations. The IOS **debug** command allows the administrator to display these messages in real-time for analysis. It is a very important tool for monitoring events on a Cisco IOS device.

All **debug** commands are entered in privileged EXEC mode. The Cisco IOS allows for narrowing the output of **debug** to include only the relevant feature or subfeature. This is important because debugging output is assigned high priority in the CPU process and it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems.

For example, to monitor the status of ICMP messages in a Cisco router, use **debug ip icmp**, as shown in the example.

```
R1# debug ip icmp
```

```
ICMP packet debugging is on
```

```
R1#
```

```
R1# ping 10.1.1.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms

R1#

*Aug 20 14:18:59.605: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology
BASE, dscp 0 topoid 0

*Aug 20 14:18:59.606: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology
BASE, dscp 0 topoid 0

*Aug 20 14:18:59.608: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology
BASE, dscp 0 topoid 0

*Aug 20 14:18:59.609: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology
BASE, dscp 0 topoid 0

*Aug 20 14:18:59.611: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology
BASE, dscp 0 topoid 0

R1#
```

To list a brief description of all the debugging command options, use the **debug ?** command in privileged EXEC mode at the command line.

To turn off a specific debugging feature, add the **no** keyword in front of the **debug** command:

```
Router# no debug ip icmp
```

Alternatively, you can enter the **undebug** form of the command in privileged EXEC mode:

```
Router# undebug ip icmp
```

To turn off all active debug commands at once, use the **undebug all** command:

```
Router# undebug all
```

Be cautious using some **debug** command. Commands such as **debug all** and **debug ip packet** generate a substantial amount of output and can use a large portion of system resources. The router could get so busy displaying **debug** messages that it would not have enough processing power to perform its network functions, or even listen to commands to turn off debugging. For this reason, using these command options is not recommended and should be avoided.

17.6.4

The terminal monitor Command

Connections to grant access to the IOS command line interface can be established in the following two ways:

- **Locally** - Local connections (i.e., console connection) require physical access to the router or switch console port using a rollover cable.

- **Remotely** - Remote connections require the use of Telnet or SSH to establish a connection to an IP configured device.

Certain IOS messages are automatically displayed on a console connection but not on a remote connection. For instance, **debug** output is displayed by default on console connections. However, **debug** output is not automatically displayed on remote connections. This is because **debug** messages are log messages which are prevented from being displayed on vty lines.

In the following output for instance, the user established a remote connection using Telnet from R2 to R1. The user then issued the **debug ip icmp** command. However, the command failed to display **debug** output.

```
R2# telnet 209.165.200.225
```

```
Trying 209.165.200.225 ... Open
```

```
 Authorized access only!
```

```
User Access Verification
```

```
Password:
```

```
R1> enable
```

```
Password:
```

```
R1# debug ip icmp
```

```
ICMP packet debugging is on
```

```
R1# ping 10.1.1.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

```
R1#
```

```
! No debug output displayed>
```

To display log messages on a terminal (virtual console), use the **terminal monitor** privileged EXEC command. To stop logging messages on a terminal, use the **terminal no monitor** privileged EXEC command.

For instance, notice how the **terminal monitor** command has now been entered and the **ping** command displays the **debug** output.

```
R1#  terminal monitor
```

```
R1# ping 10.1.1.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

```
R1#
```

```
*Aug 20 16:03:49.735: ICMP: echo reply rcvd, src 10.1.1.1, dst 209.165.200.225,topology
BASE, dscp 0 topoid 0
```

```
**Aug 20 16:03:49.737: ICMP: echo reply rcvd, src 10.1.1.1, dst
209.165.200.225,topology BASE, dscp 0 topoid 0
```

```
**Aug 20 16:03:49.738: ICMP: echo reply rcvd, src 10.1.1.1, dst
209.165.200.225,topology BASE, dscp 0 topoid 0
```

```
**Aug 20 16:03:49.740: ICMP: echo reply rcvd, src 10.1.1.1, dst
209.165.200.225,topology BASE, dscp 0 topoid 0
```

```
**Aug 20 16:03:49.741: ICMP: echo reply rcvd, src 10.1.1.1, dst
209.165.200.225,topology BASE, dscp 0 topoid 0
```

```
R1# no debug ip icmp
```

```
ICMP packet debugging is off
```

```
R1#
```

Note: The intent of the **debug** command is to capture live output for a short period of time (i.e., a few seconds to a minute or so). Always disable **debug** when not required.

17.7.1

Duplex Operation and Mismatch Issues

Many common network problems can be identified and resolved with little effort. Now that you have the tools and the process for troubleshooting a network, this topic reviews some common networking issues that you are likely to find as a network administrator.

In data communications, *duplex* refers to the direction of data transmission between two devices.

There are two duplex communication modes:

- **Half-duplex** - Communication is restricted to the exchange of data in one direction at a time.
- **Full-duplex** - Communications is permitted to be sent and received simultaneously.

The figure illustrates how each duplex method operates.

The figure is a comparison of half-duplex versus full-duplex communications. The top diagram shows half-duplex communication. Switch S1 is connected to switch S2 with an arrow flowing from S1 to S2 indicating a device can

send or receive. The bottom diagram shows-full duplex communication. Switch S1 is connected to switch S2 with arrows pointing in both directions indicating a device can send AND receive simultaneously.

S2S1S2S1

Full-Duplex CommunicationHalf-Duplex Communication Send **OR** receiveSend **AND** receive simultaneously

Interconnecting Ethernet interfaces must operate in the same duplex mode for best communication performance and to avoid inefficiency and latency on the link.

The Ethernet autonegotiation feature facilitates configuration, minimizes problems and maximizes link performance between two interconnecting Ethernet links. The connected devices first announce their supported capabilities and then choose the highest performance mode supported by both ends. For example, the switch and router in the figure have successfully autonegotiated full-duplex mode.

S1R1

F0/5G0/0/1

If one of the two connected devices is operating in full-duplex and the other is operating in half-duplex, a duplex mismatch occurs. While data communication will occur through a link with a duplex mismatch, link performance will be very poor.

Duplex mismatches are typically caused by a misconfigured interface or in rare instances by a failed autonegotiation. Duplex mismatches may be difficult to troubleshoot as the communication between devices still occurs.

17.7.2

IP Addressing Issues on IOS Devices

IP address-related problems will likely keep remote network devices from communicating. Because IP addresses are hierarchical, any IP address assigned to a network device must conform to that range of addresses in that network. Wrongly assigned IP addresses create a variety of issues, including IP address conflicts and routing problems.

Two common causes of incorrect IPv4 assignment are manual assignment mistakes or DHCP-related issues.

Network administrators often have to manually assign IP addresses to devices such as servers and routers. If a mistake is made during the assignment, then communications issues with the device are very likely to occur.

On an IOS device, use the **show ip interface** or **show ip interface brief** commands to verify what IPv4 addresses are assigned to the network interfaces. For example, issuing the **show ip interface brief** command as shown would validate the interface status on R1.

```
R1# show ip interface brief

Interface              IP-Address      OK? Method Status                Protocol

GigabitEthernet0/0/0   209.165.200.225 YES manual up                    up

GigabitEthernet0/0/1   192.168.10.1    YES manual up                    up

Serial0/1/0            unassigned      NO  unset  down                  down

Serial0/1/1            unassigned      NO  unset  down                  down

GigabitEthernet0       unassigned      YES unset  administratively down down

R1#
```

17.7.3

IP Addressing Issues on End Devices

In Windows-based machines, when the device cannot contact a DHCP server, Windows will automatically assign an address belonging to the 169.254.0.0/16 range. This feature is called Automatic Private IP Addressing (APIPA) and is designed to facilitate communication within the local network. Think of it as Windows saying, “I will use this address from the 169.254.0.0/16 range because I could not get any other address”.

Often, a computer with an APIPA address will not be able to communicate with other devices in the network because those devices will most likely not belong to the 169.254.0.0/16 network. This situation indicates an automatic IPv4 address assignment problem that should be fixed.

Note: Other operating systems, such Linux and OS X, will not assign an IPv4 address to the network interface if communication with a DHCP server fails.

Most end devices are configured to rely on a DHCP server for automatic IPv4 address assignment. If the device is unable to communicate with the DHCP server, then the server cannot assign an IPv4 address for the specific network and the device will not be able to communicate.

To verify the IP addresses assigned to a Windows-based computer, use the **ipconfig** command, as shown in the output.

```
C:\Users\PC-A> ipconfig

Windows IP Configuration

(Output omitted)

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :

   Link-local IPv6 Address . . . . . : fe80::a4aa:2dd1:ae2d:a75e%16

   IPv4 Address. . . . . . . . . . . : 192.168.10.10

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : 192.168.10.1

(Output omitted)
```

17.7.4

Default Gateway Issues

The default gateway for an end device is the closest networking device that can forward traffic to other networks. If a device has an incorrect or nonexistent default gateway address, it will not be able to communicate with devices in remote networks. Because the default gateway is the path to remote networks, its address must belong to the same network as the end device.

The address of the default gateway can be manually set or obtained from a DHCP server. Similar to IPv4 addressing issues, default gateway problems can be related to misconfiguration (in the case of manual assignment) or DHCP problems (if automatic assignment is in use).

To solve misconfigured default gateway issues, ensure that the device has the correct default gateway configured. If the default address was manually set but is incorrect, simply replace it with the proper address. If the default gateway address was automatically set, ensure the device can communicate with the DHCP server. It is also important to verify that the proper IPv4 address and subnet mask were configured on the interface of the router and that the interface is active.

To verify the default gateway on Windows-based computers, use the **ipconfig** command as shown.

```
C:\Users\PC-A> ipconfig
```

```
Windows IP Configuration

(Output omitted)

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :

   Link-local IPv6 Address . . . . . : fe80::a4aa:2dd1:ae2d:a75e%16

   IPv4 Address. . . . . . . . . . . : 192.168.10.10

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : 192.168.10.1

(Output omitted)
```

On a router, use the **show ip route** command to list the routing table and verify that the default gateway, known as a default route, has been set. This route is used when the destination address of the packet does not match any other routes in its routing table.

For example, the output verifies that R1 has a default gateway (i.e., Gateway of last resort) configured pointing to IP address 209.168.200.226.

```
R1# show ip route | begin Gateway

Gateway of last resort is 209.165.200.226 to network 0.0.0.0

O*E2  0.0.0.0/0 [110/1] via 209.165.200.226, 02:19:50, GigabitEthernet0/0/0

      10.0.0.0/24 is subnetted, 1 subnets

O        10.1.1.0 [110/3] via 209.165.200.226, 02:05:42, GigabitEthernet0/0/0

      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks

C        192.168.10.0/24 is directly connected, GigabitEthernet0/0/1

L        192.168.10.1/32 is directly connected, GigabitEthernet0/0/1

      209.165.200.0/24 is variably subnetted, 3 subnets, 2 masks

C        209.165.200.224/30 is directly connected, GigabitEthernet0/0/0

L        209.165.200.225/32 is directly connected, GigabitEthernet0/0/0

O        209.165.200.228/30

           [110/2] via 209.165.200.226, 02:07:19, GigabitEthernet0/0/0

R1#
```

The first highlighted line basically states that the gateway to any (i.e., 0.0.0.0) should be sent to IP address 209.165.200.226. The second highlighted displays how R1 learned about the default gateway. In this case, R1 received the information from another OSPF-enabled router.

17.7.5

Troubleshooting DNS Issues

Domain Name System (DNS) defines an automated service that matches names, such as www.cisco.com, with the IP address. Although DNS resolution is not crucial to device communication, it is very important to the end user.

It is common for users to mistakenly relate the operation of an internet link to the availability of the DNS. User complaints such as “the network is down” or “the internet is down” are often caused by an unreachable DNS server. While packet routing and all other network services are still operational, DNS failures often lead the user to the wrong conclusion. If a user types in a domain name such as www.cisco.com in a web browser and the DNS server is unreachable, the name will not be translated into an IP address and the website will not display.

DNS server addresses can be manually or automatically assigned. Network administrators are often responsible for manually assigning DNS server addresses on servers and other devices, while DHCP is used to automatically assign DNS server addresses to clients.

Although it is common for companies and organizations to manage their own DNS servers, any reachable DNS server can be used to resolve names. Small office and home office (SOHO) users often rely on the DNS server maintained by their ISP for name resolution. ISP-maintained DNS servers are assigned to SOHO customers via DHCP. Additionally, Google maintains a public DNS server that can be used by anyone and it is very useful for testing. The IPv4 address of Google’s public DNS server is 8.8.8.8 and 2001:4860:4860::8888 for its IPv6 DNS address.

Cisco offers OpenDNS which provides secure DNS service by filtering phishing and some malware sites. You can change your DNS address to 208.67.222.222 and 208.67.220.220 in the Preferred DNS server and Alternate DNS server fields. Advanced features such as web content filtering and security are available to families and businesses.

Use the **ipconfig /all** as shown to verify which DNS server is in use by the Windows computer.

```
C:\Users\PC-A> ipconfig /all
```

```
(Output omitted)
```

```
Wireless LAN adapter Wi-Fi:
```

```
   Connection-specific DNS Suffix  . :
```

```
   Description . . . . . . . . . . . : Intel(R) Dual Band Wireless-AC 8265
```

```
   Physical Address. . . . . . . . . : F8-94-C2-E4-C5-0A
```

```
   DHCP Enabled. . . . . . . . . . . : Yes
```

```
   Autoconfiguration Enabled . . . . : Yes
```

```
   Link-local IPv6 Address . . . . . : fe80::a4aa:2dd1:ae2d:a75e%16(Preferred)
```

```
   IPv4 Address. . . . . . . . . . . : 192.168.10.10(Preferred)
```

```
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
```

```
   Lease Obtained. . . . . . . . . . : August 17, 2019 1:20:17 PM
```

```
   Lease Expires . . . . . . . . . . : August 18, 2019 1:20:18 PM
```

```
   Default Gateway . . . . . . . . . : 192.168.10.1

   DHCP Server . . . . . . . . . . . : 192.168.10.1

   DHCPv6 IAID . . . . . . . . . . . : 100177090

   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-21-F3-76-75-54-E1-AD-DE-DA-9A

   DNS Servers . . . . . . . . . . . : 208.67.222.222

   NetBIOS over Tcpip. . . . . . . . : Enabled

(Output omitted)
```

The **nslookup** command is another useful DNS troubleshooting tool for PCs. With **nslookup** a user can manually place DNS queries and analyze the DNS response. The **nslookup** command shows the output for a query for www.cisco.com. Notice you can also simply enter an IP address and **nslookup** will resolve the name.

Note: It is not always possible to type an IP address in **nslookup** and receive the domain name. One of the most common reasons for this is that most websites run on servers that support multiple sites.

```
C:\Users\PC-A> nslookup

Default Server:  Home-Net

Address:  192.168.1.1

> cisco.com

Server:  Home-Net

Address:  192.168.1.1

Non-authoritative answer:

Name:    cisco.com

Addresses:  2001:420:1101:1::185

          72.163.4.185

> 8.8.8.8

Server:  Home-Net

Address:  192.168.1.1

Name:    dns.google

Address:  8.8.8.8

>
```

```
> 208.67.222.222

Server:  Home-Net

Address:  192.168.1.1

Name:    resolver1.opendns.com

Address:  208.67.222.222

>
```