LearnSmart

Exam Manual

CISCO (640-802) CCNA

Cisco Certified Network Associate

Smarter Training

LearnSmart's **CCNA Exam Manual** equips readers with all the knowledge and skill sets required to become Cisco certified. This guide breaks down concepts with which candidates must be familiar in order to successfully complete the 640-802 exam and become Cisco certified. Topics covered in this guide include:

- Network Foundations
- IOS Foundations
- Cisco Switches
- And more!

Sharpen your competitive edge today by purchasing this exam manual and moving one step closer towards earning your CCNA!

CCNA (640-802)
LearnSmart Exam Manual

Copyright © 2011 by PrepLogic, LLC.
Product ID: 011088
Production Date: June 28 , 2011

Warning and Disclaimer

Every effort has been made to make this document as complete and as accurate as possible, but no warranty or fitness is implied. The publisher and authors assume no responsibility for errors or omissions. The information provided is on an "as is" basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this document.

Volume, Corporate, and Educational Sales

Favorable discounts are offered on all products when ordered in quantity. For more information, please contact us directly:

1-800-418-6789
solutions@learnsmartsystems.com

International Contact Information

International: +1 (813) 769-0920

United Kingdom: (0) 20 8816 8036

Table of Contents

Abstract

The Cisco Certified Network Associate is the most well-recognized and respected Networking certification in the world. By attaining it, students and candidates signify themselves as extremely accomplished and capable Network Administrators. The exam, created by Cisco Systems, is extremely difficult and not to be taken lightly. It covers a myriad of topics, from the basics of the OSI model, all the way to the most detailed analysis of routing packets across multiple subnetted networks. It is multiple choice, simulative, and incorporates test strategies such as "drag and drop" and "hot area" questions to verify a candidate's knowledge.

Before taking this exam, you should be very familiar with both Cisco technology and networking. Most candidates for this exam have already passed other certifications, such as the CompTIA A+ or Network+ exams. If it is your first certification, we recommend you study for the exam very thoroughly. It will not be easy.

Your Product

This CCNA Exam Manual has been designed from the ground up with you, the student, in mind. It is lean, strong, and specifically targeted toward the candidate. Unlike many other CCNA products, the LearnSmart CCNA Exam Manual does not waste time with excessive explanations. Instead, it is packed full of valuable techniques, priceless information, and brief, but precisely worded, explanations. While we do not recommend using only this product to pass the exam, but rather a combination of LearnSmart Audio Training, Practice Exams, and Video Training, we have designed the product so that it and it alone can be used to pass the exam.

About the Author

Jeremy Cioara is an accomplished Cisco Certified Internetwork Expert (CCIE), MCSE, CNE, and full-time trainer. In addition, he is also a full-time computer enthusiast and first class administrator. He has been in the computing industry since the 1980s and, to this day, believes that there is no greater job than teaching.

Domain 1 - The Benefits of Cisco Certification

Since the CCENT and CCNA Cisco certifications are typically how most people begin their journey into Cisco networking, let's take a moment to talk about some of the benefits of obtaining this certification.

1. **Credibility** - Cisco certifications are considered by many to be some of the most real-world applicable certification paths in the industry. Obtaining Cisco certification is no easy feat, so when you do become certified, the certification acronyms you place after your name (such as CCENT, CCNA or CCNP) actually means something to other IT professionals!

2. **Marketability** - Organizations are looking for Cisco-certified individuals! Cisco has structured their partner program in such a way that it *requires* organizations to hire certified individuals to move to higher levels in their partner relationship with Cisco. The higher the partner relationship, the bigger discount on Cisco equipment (along with many other benefits). So, an organization can actually *save* money by hiring you. Nice!

3. **Sense of Accomplishment** - When you take a Cisco certification exam, regardless of the pass or fail mark, you will know that the exam is fair. Cisco does not attempt to mislead you in their exam questions or ask questions that are looking for the "Cisco answer" rather than "how it really works." There's nothing like passing the exam and then thinking to yourself, "Wow. There's no way I could have passed that test without *really* knowing what I was doing."

With that in mind, let's move into the material that will help you get there.

Domain 2 - Network Foundations

The Purpose and Pieces of Networking

When you move into the realm of Cisco networking, you have entered a world of building the roads that makes business possible. Most of the time, users and other network administrators take these roads for granted, just like you take them for granted when you drive a car. You simply assume that the roads will be there and that they'll carry you through to your destination. However, a poorly timed construction project (network maintenance) or unscheduled road closure (network outage) will bring the entire infrastructure crumbling down. The goal of a network is to establish communications throughout an organization. Let's take a look at the core building blocks that make this communication possible:

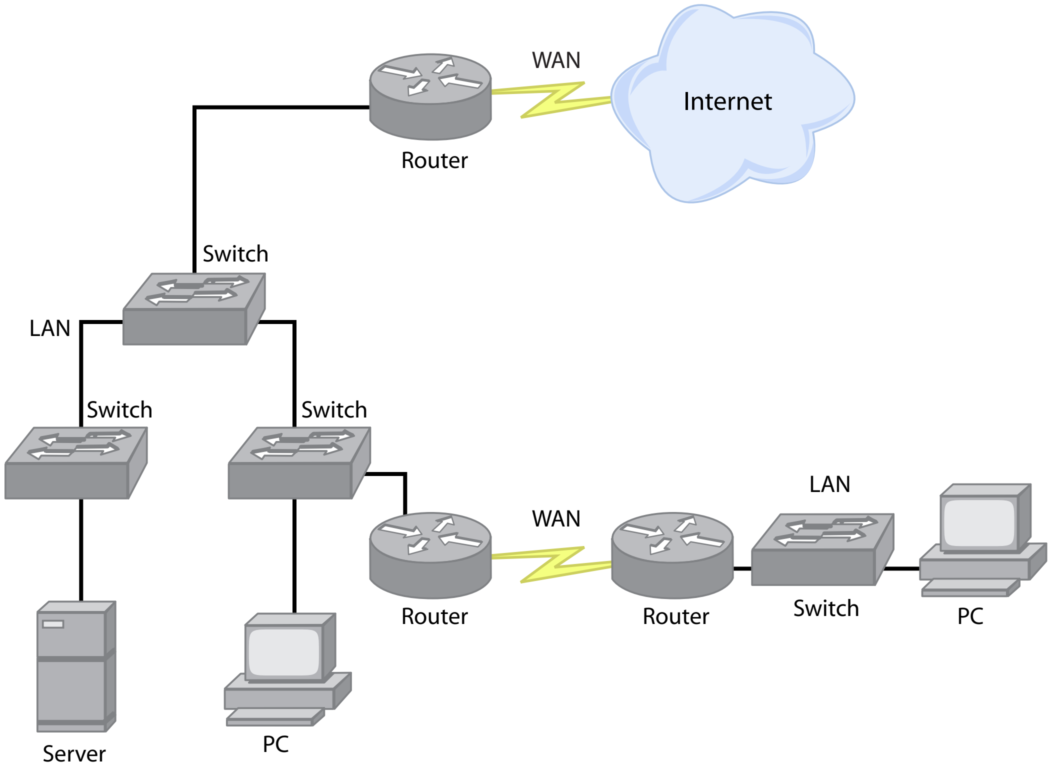
- **Personal Computers (PCs) and Servers** - These devices serve as the endpoints in the network and are responsible for sending and receiving data to and from the network.
- **Network Connections** - You must have a way to attach a device to the network; this building block includes the network interface card (NIC), cabling and connectors.
- **Hubs and Switches** - These devices provide points on which all the end systems of a network can attach.
- **Routers** - Routers connect multiple networks together and find the best way to reach each network.

These components can build a network within a local area (LAN) or across a wide area (WAN). In recent years, the lines between a LAN and WAN have begun to blur, since Wireless and Fiber Optic technology can extend the reach of a LAN much further than older technology ever could. Regardless, the following definitions still stand strong:

- **Local Area Network (LAN)** - A computer network covering a small geographic area such as a home, office or group of buildings.
- **Wide Area Network (WAN)** - A computer network covering a large geographic area such as a city, state, nation or globe.

Interpreting a Network Diagram

The following figure shows the placement of each of the core network components:



Types of Network Communication

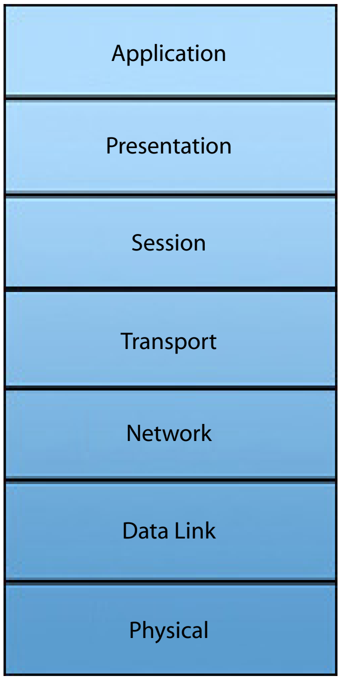
Not that long ago, network communication was solely restricted to data: internal corporate data or external Internet data. Nowadays, the network has evolved to support all types of communication. Organizations have begun to merge their telephone system with the network, creating Voice over IP (VoIP) network traffic. Users have begun to mount video cameras on their computers and in conference rooms to create streaming video traffic. What's more, the network has begun to become so entirely saturated with different application types, network administrators now require a way to divide the different traffic types into application classes, some of which are far more important than others. Because of this, all of Cisco's newer equipment supports Quality of Service (QoS) features allowing you to manage the priorities of data crossing the network. For example, at a major bottleneck in the network (such as the transition from the high-speed LAN to the low-speed WAN), you could set up a system stating that the VoIP traffic is sent first, followed by the streaming video, followed by the business-critical applications and so on.

Using the OSI Model

With all of this network communication occurring, trying to understand the network can become very complex, very quickly. In order to try to make sense of it all, you can use the handy OSI Model. It's a little known fact that the OSI Model was never meant to be just a model that describes network communication. There is also an OSI protocol (technically called the "OSI Networking Suite") that was designed to compete with TCP/IP. We now know the end of the story: TCP/IP wins. However, the OSI Model is still used today as an excellent way to describe and fully understand network communication. You will find that a deep understanding of the OSI Model is critical to your networking success in the Cisco world. This is not one of those models that you learn in order to pass the exam and then never use again. With that foundation, let's begin.

The Layers of the OSI Model

The OSI Model is comprised of 7 layers, each of which describes a specific aspect of network communication:

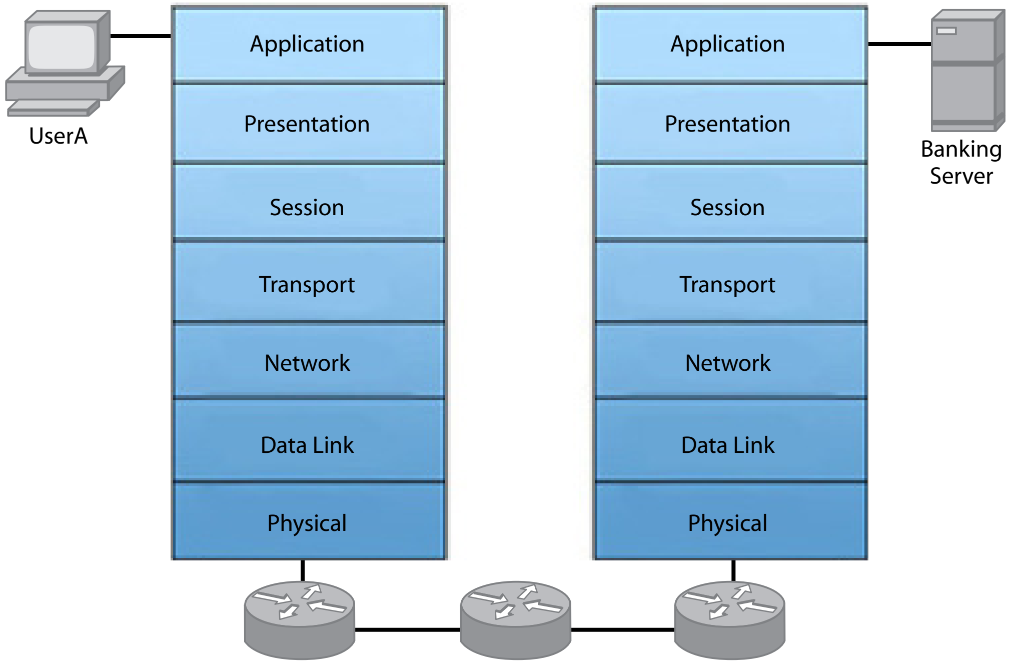


Your first job will be to memorize the layers and their order. There are two handy memorization tips you can use to remember the layers: **A**ll **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing, where each word contains the first letter of the layers from the top-down, or you can use **P**lease **D**o **N**ot **T**hrow **S**ausage **P**izza **A**way, where each word contains the first letter of the layers from the bottom-up. I personally love sausage pizza, so I prefer the latter.

Once you've got the layers down, you now need to know what each of them accomplishes. I'll present this to you in two ways. First, we'll look at the cold, hard facts about each layer, and then we'll look at a practical example of how the OSI Model is used in real-world network communication. So, here are the facts:

- **Application Layer**: This layer interfaces directly with the network-aware application, giving it access to network resources. Without this layer, no user application would be able to get access to the network.
- **Presentation Layer**: Encodes the data being sent or received into a generic format that will be understood by both devices. For example, a web browser might receive data in HTML format or a picture in JPG format, which are generic and well understood standards.
- **Session Layer**: Begins, ends, and manages the sessions between devices.
- **Transport Layer**: Handles the reliability of the connection and logical separation of applications. For example, if a computer is surfing the Internet with a web browser and at the same time listening to Internet-radio, this layer ensures the correct data arrives to the correct application. In addition, this layer handles flow-control (ensuring one side does not send information faster than the other can receive) and data integrity (ensuring the data is not corrupt). The most common Transport Layer protocol is TCP.
- **Network Layer**: Provides logical addressing services allowing a device to dictate the source and destination address used for end-to-end communication. This layer is also responsible for routing the packet from its source to its destination. The most common Network layer protocol is IP.
- **Data Link Layer**: Provides physical addressing services allowing a device to dictate the source and destination address used for local network communication. This layer permits communication between devices connected to the same network. This layer is also responsible for error detection.
- **Physical Layer**: Defines the physical standards used for network communication.

Now that we've seen the facts, let's put them together into a practical example of network communication. On the next page is a network diagram representing a task many people do frequently: using online banking to manage finances. In this case, UserA (shown to the left) has used a web browser to issue a request to transfer $100.00 from his checking account to his savings account. Let's follow this network request step-by-step as it passes through the layers of the OSI Model.



Step 1: The Application Layer (Layer 7)
The user is operating in a web browser. For this example, we'll say he's using Internet Explorer (IE). While the user is interacting with IE, this application doesn't represent function of the OSI Application Layer. The Application Layer is invoked when IE attempts to communicate over the network. The operating system (Microsoft Windows in this case) sees the request and captures it from the application. It then takes the $100.00 transfer request and passes it down to the Presentation Layer.

Step 2: The Presentation Layer (Layer 6)
The job of the Presentation Layer is to take the user data (the $100.00 transfer request) and format it into a generic language understandable by industry standard applications. Here's what that means in English: I mentioned above that the user was using Internet Explorer (IE) to perform the transfer request. However, IE is not the only web browser on the market. The user could have been using Mozilla Firefox, Opera, Apple Safari or Netscape Navigator, just to name a few. Likewise, the web server for online banking could have been running on Microsoft's Internet Information Server (IIS), IBM's Websphere or Apache. How do you ensure that the online bank is able to understand the $100.00 transfer request from the user? What if the user is using Firefox and the web server is running IIS? That's the job of the Presentation Layer. It will format the request in a generic format (such as HTML). It also secures the connection using generic encryption that any standards-compliant web browser is able to support. Once the data has been formatted correctly, it is then passed down to the session layer.

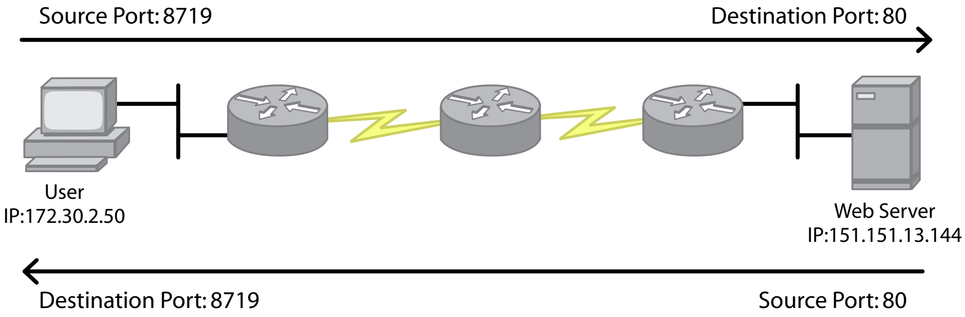
Step 3: The Session Layer (Layer 5)
The Session Layer has a simple function: starting, ending and managing sessions between devices. At any one time, your PC may have numerous network connections going to and from it. Likewise, busy network servers can have hundreds or even thousands of network connections occurring at any one time. The Session Layer is responsible for managing all of these active sessions, as long as the device can keep them all straight. The $100.00 transfer example we are working through already has an active session with the online bank that started when the user first logged into the online banking website. Once the user closes the web browser (or navigates to a different website), the Session Layer will close down the session.

Step 4: The Transport Layer (Layer 4)
If there was an award for being the most important layer of the OSI Model, this layer would win it (in the Cisco world, at least). This layer performs two critical functions: handling the reliability of the session and logically separating applications (this integrates closely with the function of the Session Layer). When the $100.00 transfer request reaches the Transport Layer, the application (IE, in this case) will need to choose whether to send the request reliably (ensuring the other side has received the request) or unreliably (assumes the other side receives the request). Keep in mind that the application chooses this connection type, not the user. In this case, IE is using the HTTP or HTTPS protocol to communicate with the web server. These protocols are considered reliable, which causes the web server to acknowledge (ACK) any information that is received. When the PC sends the $100.00 request, the web server receives it and says, "ACK" back to the PC, which lets the computer know that the request got there.

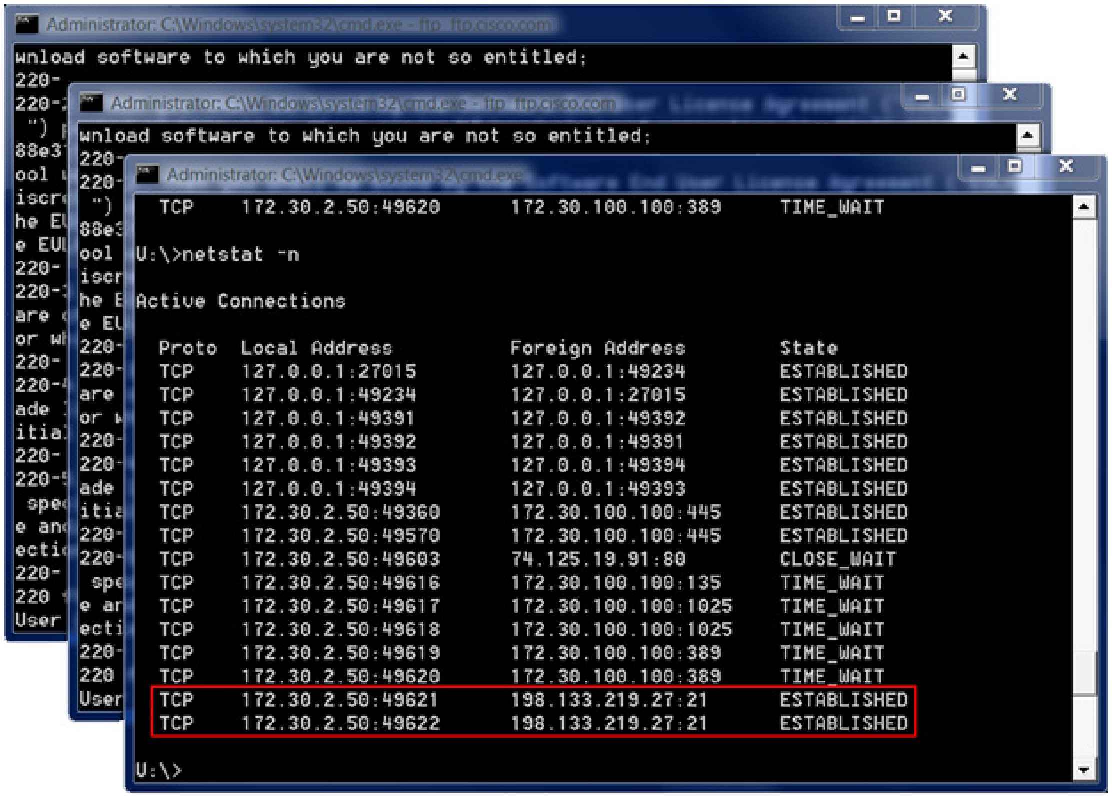
After seeing how reliable communication works, you might wonder why any application would choose to communicate unreliably. Unreliable communication is very useful for any "real-time" application that is sending or receiving data as it actually happens. For example, it would not benefit the communication at all to use a reliable connection for Voice over IP (VoIP) since packets are being sent as a person is speaking. If one or more of the packets are dropped along the path, it would not make sense to send that packet at a later time. The communication has already passed that point. Common applications that use unreliable communication are VoIP, video over IP and online games.

The second job of the Transport Layer is to logically separate applications. In the TCP/IP protocol suite (which will be fully discussed layer), this is done using port numbers. When the user's IE application sends the $100.00 transfer request to the web server, it needs to ensure the *web server* application receives it and not some other application service. For example, the online banking web server might also be an email server and a database server. In order to ensure the web server application gets the request instead of the email server application, the user's IE will send the request to the web server's well known port number. If we are using HTTP to complete the transfer request, the well known port number will be 80. If we are using HTTPS, the well known port number will be 443. This is considered the *destination port number* used for communication. This destination port number directs the network request to the correct server application.

While it is necessary to distinguish the correct server application the user is requesting, the user's IE application must also separate itself from the other network applications that are running on the PC. For example, our user transferring $100.00 might also be listening to online Internet radio, watching stock prices and playing an online game of chess with a person in Russia, all while performing this transfer. The IE application needs a way to ensure that communications coming from the web server end up back in the IE application window rather than being received by the online chess game. This is the job of the *source port number*. When IE attempts to communicate across the network, Microsoft Windows will generate a unique source port number for the application. This source port number is communicated to the web server along with the $100.00 request. When the web server responds to the $100.00 transfer, it will direct the response to the user's source port number, allowing Microsoft Windows to return the data to the requesting IE application window.



We'll talk plenty more about port numbers later on in this guide; however, a very handy way to see the port numbers that are coming to and from your PC is by using the netstat command line utility. Let me show you one more example of these port numbers in action. I am going to open three command-line windows in Microsoft Windows (you can do this by clicking **Start > Run > cmd**). In two of them, I am going to open a File Transfer Protocol (FTP) session to **ftp.cisco.com** (a public Cisco FTP site). In the third, I will enter the **netstat** command.


Administrator: C:\Windows\system32\cmd.exe - ftp ftp.cisco.com
Administrator: C:\Windows\system32\cmd.exe - ftp ftp.cisco.com
Administrator: C:\Windows\system32\cmd.exe

```
wnload software to which you are not so entitled;
```

```
wnload software to which you are not so entitled;
```

```
  TCP    172.30.2.50:49620      172.30.100.100:389     TIME_WAIT

U:\>netstat -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:27015        127.0.0.1:49234        ESTABLISHED
  TCP    127.0.0.1:49234        127.0.0.1:27015        ESTABLISHED
  TCP    127.0.0.1:49391        127.0.0.1:49392        ESTABLISHED
  TCP    127.0.0.1:49392        127.0.0.1:49391        ESTABLISHED
  TCP    127.0.0.1:49393        127.0.0.1:49394        ESTABLISHED
  TCP    127.0.0.1:49394        127.0.0.1:49393        ESTABLISHED
  TCP    172.30.2.50:49360      172.30.100.100:445     ESTABLISHED
  TCP    172.30.2.50:49570      172.30.100.100:445     ESTABLISHED
  TCP    172.30.2.50:49603      74.125.19.91:80        CLOSE_WAIT
  TCP    172.30.2.50:49616      172.30.100.100:135     TIME_WAIT
  TCP    172.30.2.50:49617      172.30.100.100:1025    TIME_WAIT
  TCP    172.30.2.50:49618      172.30.100.100:1025    TIME_WAIT
  TCP    172.30.2.50:49619      172.30.100.100:389     TIME_WAIT
  TCP    172.30.2.50:49620      172.30.100.100:389     TIME_WAIT
  TCP    172.30.2.50:49621      198.133.219.27:21      ESTABLISHED
  TCP    172.30.2.50:49622      198.133.219.27:21      ESTABLISHED

U:\>
```

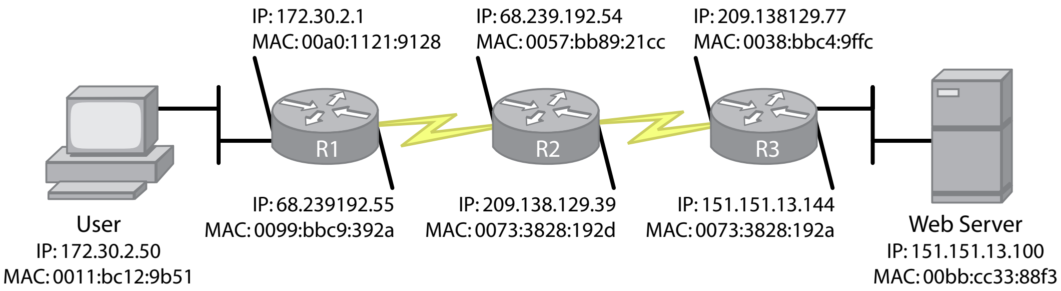
My PC has many connections coming to it from other applications, but I've drawn a box around the two that we need to see. The request is going to the **Foreign Address** of **198.133.219.27:21**, which represents the IP address of **ftp.cisco.com** and the destination port number 21. The request is coming from the IP address of **172.30.2.50** (which is my PC) with a source port number of **49621** and **49622**. These port numbers represent the two command prompt windows with an FTP session open (shown behind the active window).

Step 5: The Network Layer (Layer 3)

After the IE application has chosen a reliable or unreliable connection and has been assigned port numbers, it will now add the logical addressing information stating the data source (where the data is coming from) and destination (where the data is going to). The logical addressing information (most commonly known as IP addresses) represents the end-to-end communication between the two devices. In our example, the source IP address would be the user's PC (172.30.2.50) and the destination IP address would be the web server (151.151.13.144). If I were relating the concept to driving, logical addressing would be similar to saying that I would like to drive from South America to Canada. Yes, this is possible, but there will need to be many stops along the way. That's the job of Layer 2.

Step 6: The Data Link Layer (Layer 2)

After the $100.00 transfer is tagged with the source and destination IP address information, it passes down to the Data Link layer where it will be tagged with the physical addressing information. Physical addresses (referred to as MAC addresses in the realm of Ethernet technology) are used to establish communications between devices plugged into the same network. In our example, the user needs to communicate to the far end web server but needs to use a router to get there. The computer can't change the destination IP address to that of the router (shown in the figure below), because then the router will not know that the data really needs to go to the online banking web server. Instead, the user's PC will put the IP address of the web server as the destination IP address and the MAC address of the router as the destination MAC address. When the router receives the packet, it will realize that it is meant to receive the packet but that the final destination of the packet is not the router but, rather, the web server.



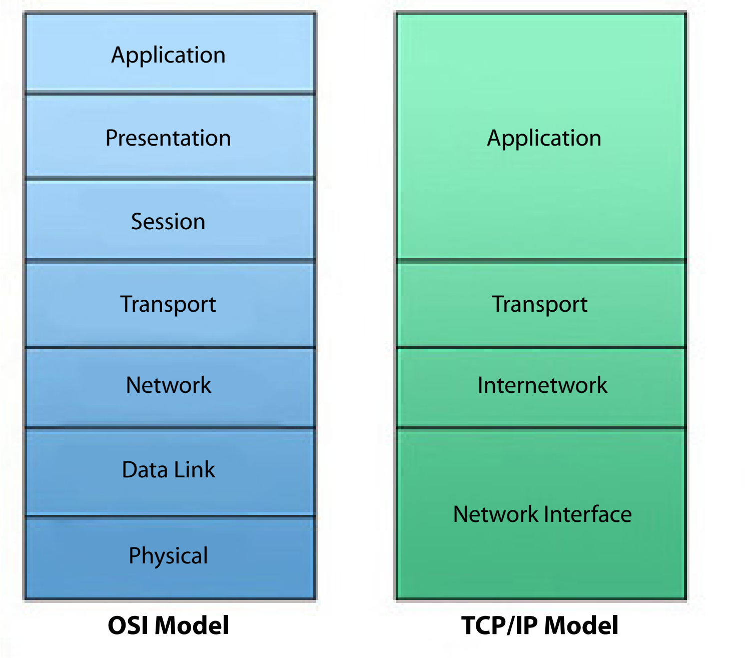
Once the router has received the packet, it will look at the destination IP address and compare it to its routing table, which tells the router where to send the data next. The router sees that it needs to send the packet to the next router in the chain (R2) to reach the web server's network. In order to accomplish this, it removes the original source and destination MAC address that was used previously and replaces it with the source MAC address (**0099:bbc9:392a**) and destination MAC address (**0057:bb89:21cc**) that allows it to reach the next router. When R2 receives the packet, it sees (based on its routing table) that it needs to send the packet to R3 to reach the web server. To accomplish this, it removes the last source and destination MAC address and replaces it with the source MAC address (**0073:3828:192d**) and destination MAC address (**0038:bbc4:9ffc**) that allows it to reach R3. This process continues until the web server finally receives the data.

Step 7: The Physical Layer (Layer 1)

The physical layer is responsible for taking all of this data and translating it into electrical or optical signals that is understandable by the wire. Since the user executing a $100.00 transfer is connected to an Ethernet network, the electrical signal will be formatted in a way that complies to the Ethernet standard.

Comparing the OSI Model and TCP/IP Model

The OSI and TCP/IP protocol suites were developed around the same time, and both of them had a model describing network communication. The OSI model is primarily used because of the amount of detail it provides; however, you should be prepared to encounter both. The graphic below compares both network models.

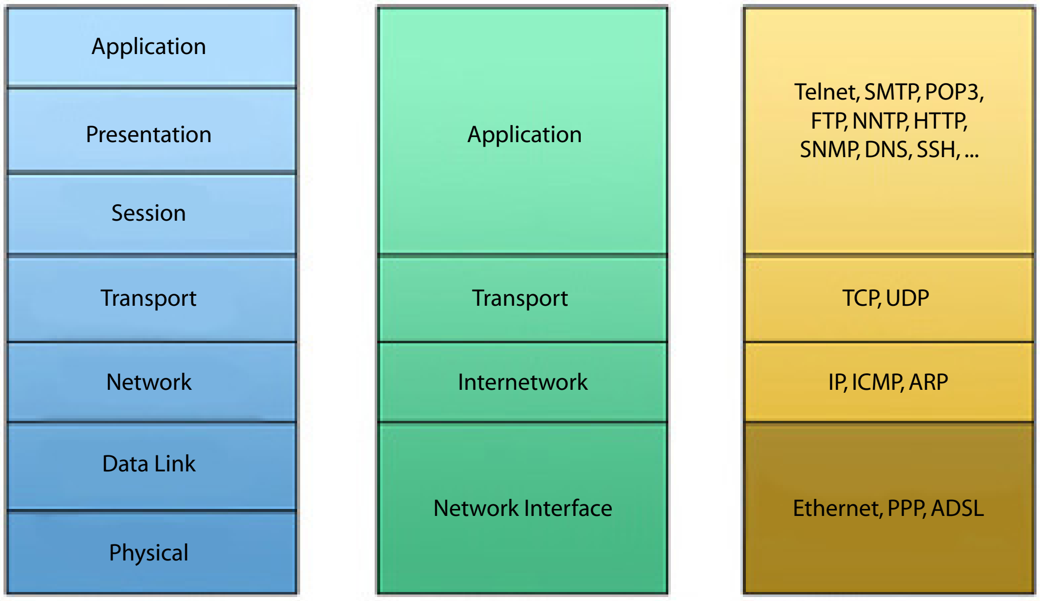


Notice that the TCP/IP model groups the top three layers into a single "Application" layer. This is because these functions typically occur before the data leaves the application itself. Also, because the Data Link and Physical layers of the OSI model are so closely related together, the TCP/IP model groups them into a single "Network Interface" layer.

Understanding TCP/IP Foundations

TCP/IP has become the "fabric of networks" in modern times. This is primarily because the Internet has become such a key part of corporate operations (and day-to-day life for many people). Nowadays, it is extremely rare to find a network in operation that is not running TCP/IP. With that in mind, having a thorough and deep understanding of TCP/IP should be of utmost importance to anyone wanting to properly manage Cisco network operations.

Just like Microsoft Office is not just one application but, rather, a suite of applications, TCP/IP is not just one protocol but a suite of protocols. Only by combining the function of a variety of protocols in the package are we able to have successful network communication. The figure on the next page shows the most common TCP/IP protocols used today along with the functioning OSI and TCP/IP model layer.



OSI Model **TCP/IP Model (DoD Model)** **TCP/IP - Internet Protocol Suite**

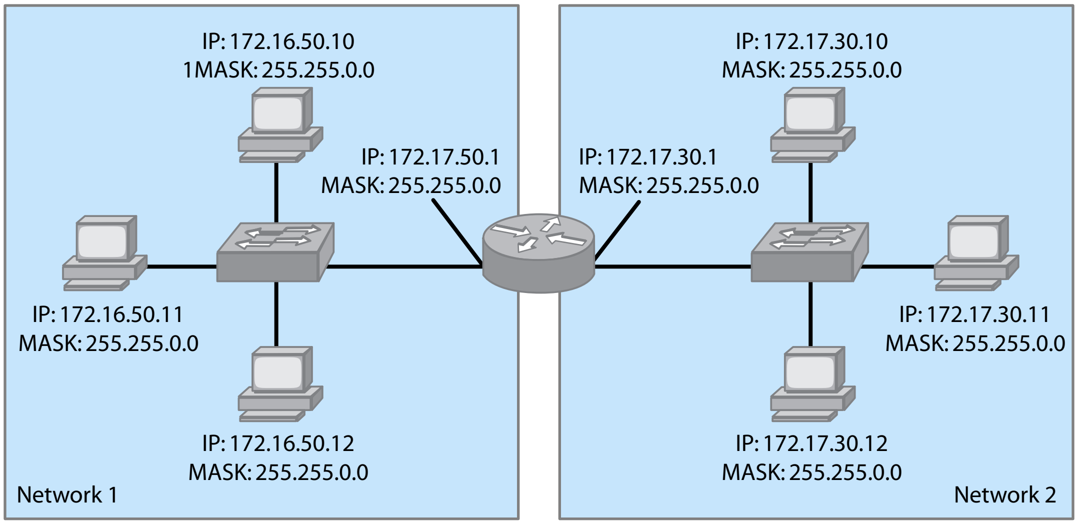
The protocol we will tackle first is IP. Earlier in the document, I mentioned that IP (and the Network layer of the OSI Model) is responsible for addressing in the network. Just by looking at an IP address, you will see four octets divided by periods with some number between 0 and 255 in each one. In addition, you will rarely see an IP address without an accompanying subnet mask:

IP Address: **172.16.172.38**
Subnet Mask: **255.255.0.0**

The IP address and subnet mask give you enough information to understand a network design. The subnet mask divides the IP address into two pieces: the network and the host.

Network	**Host**
172.16.	172.38
255.255.	0.0

In the example shown above, the network is defined as "172.16" and the host is defined as "172.38". All of the devices that are plugged into the same network (not passing through a router) must begin with the same network ID, as shown in the figure below:



Notice that all the PCs to the left of the router begin with 172.16 (defining Network 1), and all the PCs to the right of the router begin with 172.17 (defining Network 2). This helps to demonstrate a couple key points. First, every interface of a router *must* be attached to a unique network. Second, a router both separates and provides communication between different networks. We'll get into the specific functions of routers in just a moment. Let's get back to the foundations of IP addressing.

Understanding IP Addressing Foundations

When the "powers that be" created the TCP/IP protocol suite, they designed five classes of addresses. Three of those classes are designed for use in networks around the world:

	Subnet Mask	First Octet Value	Number of Hosts Per Network
Class A	255.0.0.0	1-126	16,777,214
Class B	255.255.0.0	128-191	65,534
Class C	255.255.255.0	192-223	254

You should be able to look at an IP address and determine what class of address it belongs to:

- 10.35.149.44 = **Class A** Address
- 150.151.33.223 = **Class B** Address
- 200.69.12.1 = **Class C** Address

Whenever you use the default subnet mask with an IP address, you are using **classful addressing**. While using classful addressing is popular with Class C addresses (because the network size is fairly manageable), Class A and B addresses will usually use **classless addressing**. This means that you are using a subnet mask other than the default with these larger classes of addresses. For example, I could put a Class C subnet mask on a Class A address such as 10.25.96.22, 255.255.255.0. At this point, 10.25.96 would represent the network and 22 would represent the host. This is a simple form of a skill known as subnetting, which we will discuss much later in this guide.

Class A and B networks are commonly subnetted because of the large number of hosts they have on each network. Cisco's recommendation for the *maximum* amount of hosts per network is currently at 500. Despite this recommendation, most administrators do not like allowing networks to grow much beyond 200 hosts per network. As you can see, Class A and B networks slightly exceed this maximum with 16,777,214 and 65,534 hosts per network, respectively. The more hosts you have on a network, the more broadcast traffic you must deal with. Broadcast traffic is part of the day-to-day life of a network. It is necessary to ensure proper operation of a network but can become excessive in larger networks and weigh the network down. Too many broadcasts affect both the performance of the network and the performance of the devices attached to the network. This is why the number one purpose of a router is to stop broadcasts. Routers stop broadcast packets from traveling from one network to another.

Reserved IP Addresses

While most IP addresses shown in the Class A, B and C network ranges can be assigned to hosts, there are a few reserved addresses that have specific meaning:

- **Addresses Starting with 127**: Any address that starts with 127 is considered a loopback address and is used for internal testing. For example, you can ping the address 127.0.0.1 to test the PC you are working with. You cannot assign addresses beginning with 127 to a device.

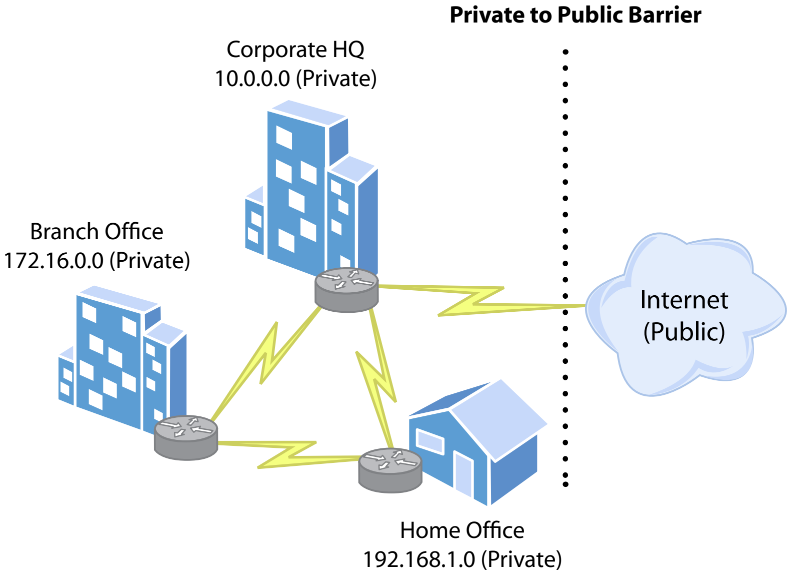
- **Addresses Starting with 169.254**: Any address that starts with 169.254 is considered an autoconfiguration address. These typically occur when a DHCP client is unable to obtain an IP address from a DHCP server. While you can assign addresses in this range to clients, it is not recommended.

- **The First IP address from Each Network Range**: The first IP address from each network is reserved as the Network Identifier (Network ID), which is used in routing processes. For example, 172.16.0.0 is the Network ID for the 172.16.0.0 255.255.0.0 network. You cannot assign this address to any network device.

- **The Last IP address from Each Network Range**: The last IP address from each network is reserved as the Broadcast ID, which is used to send a broadcast packet to all devices in the network. For example, 172.16.255.255 is the Broadcast ID for the 172.16.0.0 255.255.0.0 network. You cannot assign this address to any network device.

Private IP Addresses

Private IP addresses were originally developed to allow an organization to configure a network that used TCP/IP but did not connect to the Internet. Private addresses are functionally the same as public IP addresses (addresses that operate on the Internet); however, every Internet Service Provider (ISP) that allows you to connect to the Internet blocks private addresses from reaching the Internet. This allows an organization to have a full network infrastructure between all of their offices without any fear of users in their organization reaching the Internet or anyone on the Internet reaching users in the organization.



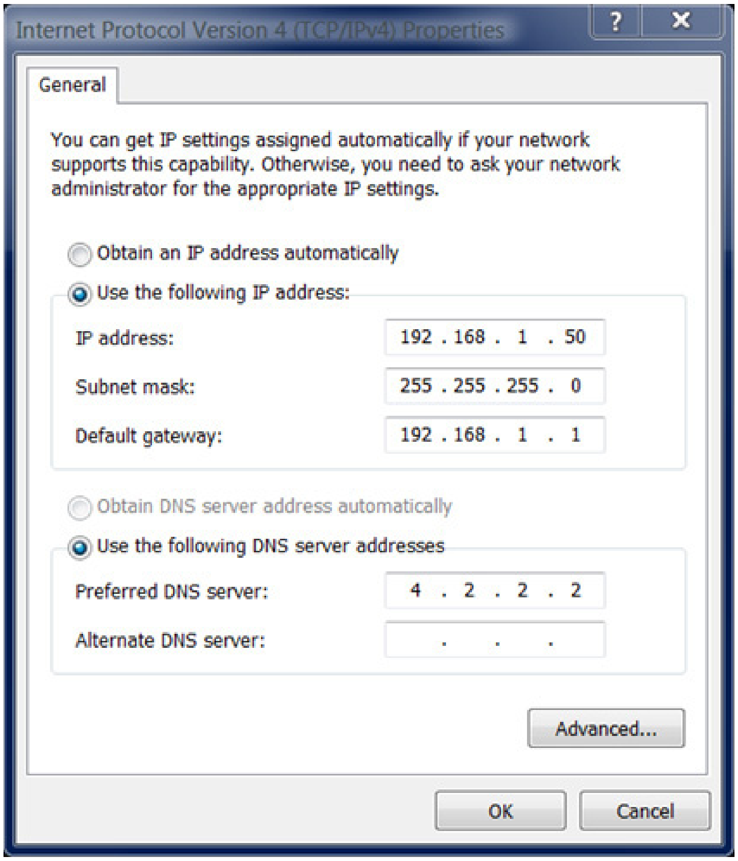
There is one private address range for the Class A, B and C address ranges. They are as follows:

- Class A: **10.0.0.0 to 10.255.255.255**
- Class B: **172.16.0.0 to 172.31.255.255**
- Class C: **192.168.0.0 to 192.168.255.255**

These addresses will function on internal networks, but will not function on the Internet. As a "sneak peek" of technology to come later in this guide, most organizations will use Network Address Translation (NAT) to allow their users to access the Internet despite the use of private addresses on the internal network.

Common IP Services

When building a network infrastructure, you will need to assign the devices on the network IP addresses from the appropriate network ranges. This can be done either statically or dynamically. In Microsoft Windows XP, you can statically assign an address by opening the Control Panel, double-clicking Network Connections, right-clicking the network adapter you would like to configure and selecting Properties. Once the network properties window appears, you can double click the TCP/IP protocol and enter the appropriate IP address information. In Microsoft Vista, you can statically assign an address by opening the Control Panel, double-clicking Network and Sharing Center, clicking on Network Connections on the left, right-clicking on the network adapter you would like to configure and selecting Properties. Once the network properties window appears, you can double click the TCP/IP protocol and enter the appropriate IP address information.



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 1 . 50

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 1 . 1

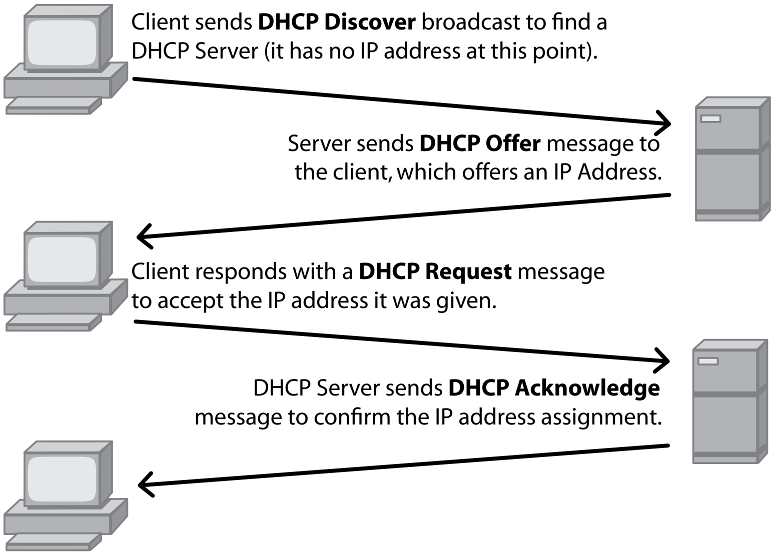
○ Obtain DNS server address automatically

● Use the following DNS server addresses

Preferred DNS server: 4 . 2 . 2 . 2

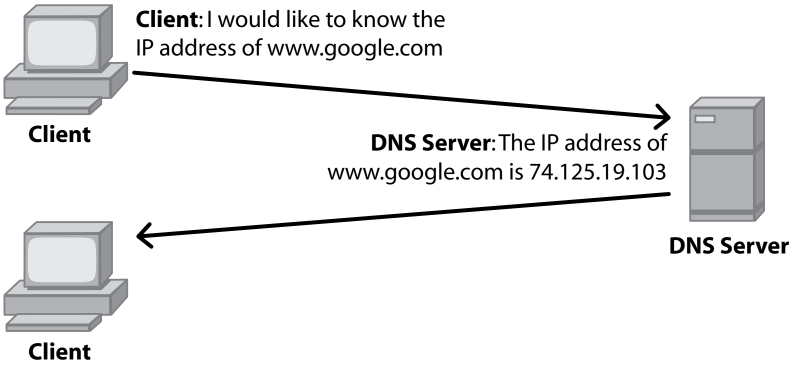
Alternate DNS server: . . .

While statically assigning addresses is absolutely necessary for key network devices such as servers, printers and routers, it can become quite tedious and unmanageable for hundreds, if not *thousands*, of individual PCs. Because of this, some ingenious person created the Dynamic Host Configuration Protocol (DHCP). DHCP allows a network administrator to set up a pool of IP addresses to be handed out by a DHCP Server to the clients. Once that is created, the administrator no longer needs to statically assign IP addresses to each network client. The process of DHCP can be depicted as follows on the page below:


Client sends **DHCP Discover** broadcast to find a DHCP Server (it has no IP address at this point).
Server sends **DHCP Offer** message to the client, which offers an IP Address.
Client responds with a **DHCP Request** message to accept the IP address it was given.
DHCP Server sends **DHCP Acknowledge** message to confirm the IP address assignment.

In addition to sending out IP address information, the DHCP server can assign items such as the subnet mask, default gateway and DNS server information.

Speaking of DNS server information, that topic is the next big IP service that I'd like to cover. The Domain Name Service (DNS) is a system that was created which allows users to remember simple names rather than IP address information. For example, it is much easier to remember www.google.com than 74.125.19.103. In order for a PC to gain DNS name lookup capabilities, it must be configured with the IP address of a DNS server. You can make this assignment either statically (using the same TCP/IP properties window shown previously) or through the DHCP server. Once the client has a DNS server configured, the process is simply as follows:


Client: I would like to know the IP address of www.google.com
Client
DNS Server: The IP address of www.google.com is 74.125.19.103
DNS Server
Client

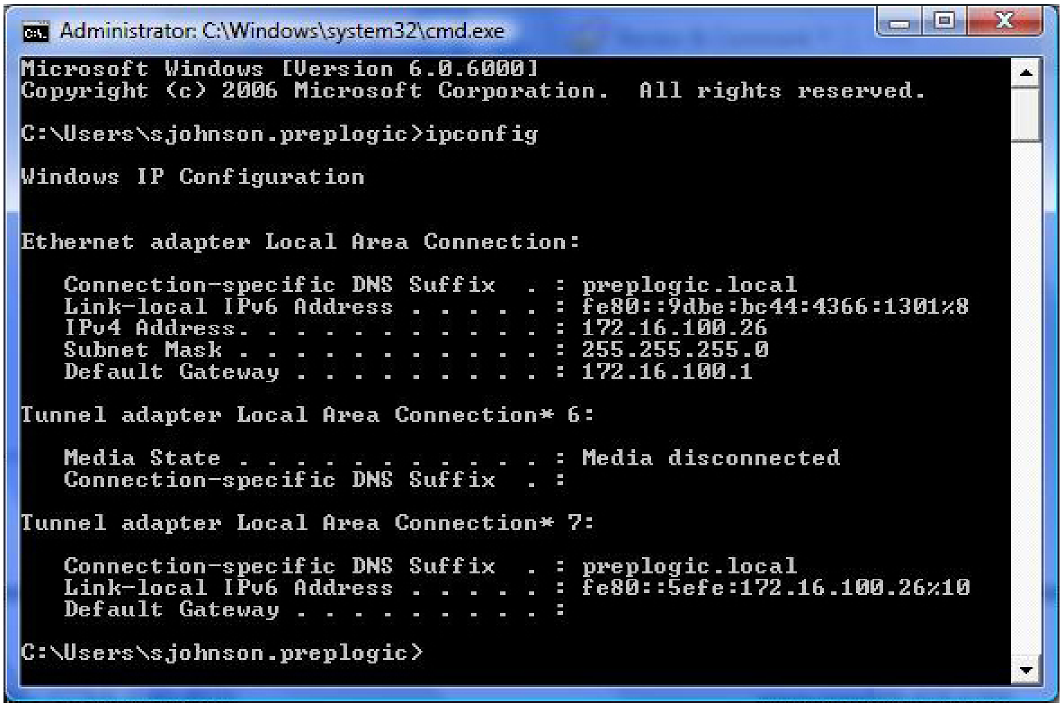
These DNS server lookups occur anytime a client is attempting to access anything using a common name rather than an IP address.

Common Client Tools

While much of the Cisco configuration work involves interacting with routers and switches in the IOS, the majority of your troubleshooting will be done from a network client. Because of this, it is absolutely critical that you understand a few of the common command line test tools.

1. **ipconfig**

The ipconfig command line utility allows you to verify a client's MAC address, IP address, subnet mask, default gateway and DNS server information. To run this utility, simply open a command line and type **ipconfig**. By typing this simple command (shown on the next page), you will be given the IP address, subnet mask and default gateway assigned to the PC.



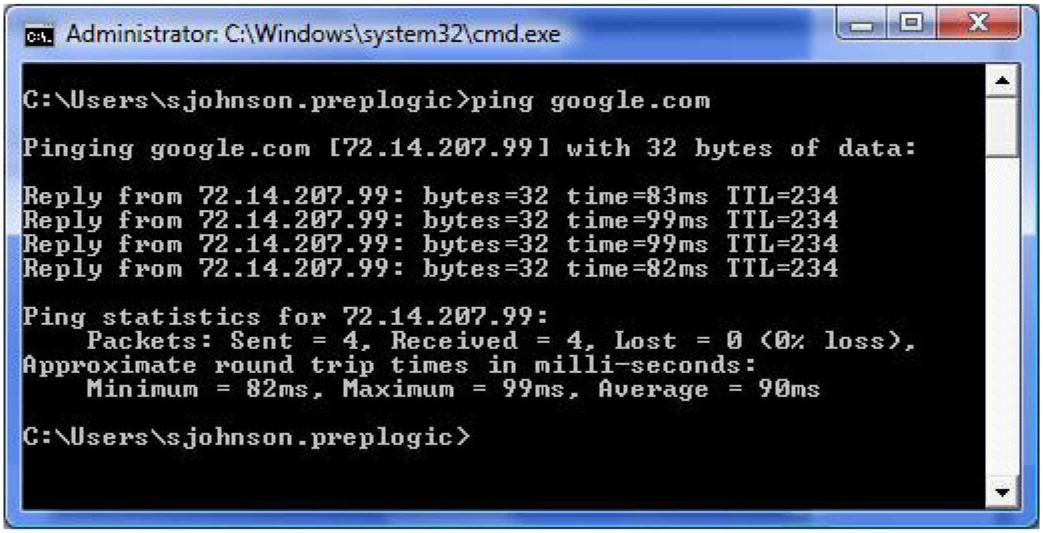
```
Microsoft Windows [Version 6.0.6000]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\sjohnson.preplogic>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : preplogic.local
   Link-local IPv6 Address . . . . . : fe80::9dbe:bc44:4366:1301%8
   IPv4 Address. . . . . . . . . . . : 172.16.100.26
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.100.1

Tunnel adapter Local Area Connection* 6:

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

Tunnel adapter Local Area Connection* 7:

   Connection-specific DNS Suffix  . : preplogic.local
   Link-local IPv6 Address . . . . . : fe80::5efe:172.16.100.26%10
   Default Gateway . . . . . . . . . :

C:\Users\sjohnson.preplogic>
```

The following are common arguments attached to the **ipconfig** command:

- **ipconfig /all** - Using this argument, you will be able to see additional information (such as MAC address and DNS Servers) about the device's TCP/IP configuration.
- **ipconfig /release** - This argument allows you to release an IP address the PC has obtained via DHCP.
- **ipconfig /renew** - This argument allows you to request a new IP address from a DHCP server.
- **ipconfig /flushdns** - This argument allows you to erase any cached DNS entries on the client. For example, after a client resolves www.google.com to 74.125.19.103, it will cache (remember) that mapping for 24 hours. If the IP address of Google changes during that time, it may be necessary to manually flush the cached DNS entries on a client.

2. **ping**

If there were a flathead screwdriver in the network world, the **ping** command would be it. The ping command tests network connectivity to a remote device. Technically, it puts the entire alphabet (a through z) in a packet and sends that packet to whatever device you specify. When the device receives it, it sends it right back. You can then measure the time it took to reach the remote device. To execute the ping command, simply type **ping <hostname or IP address>**. The following is an example of a ping to www.google.com:



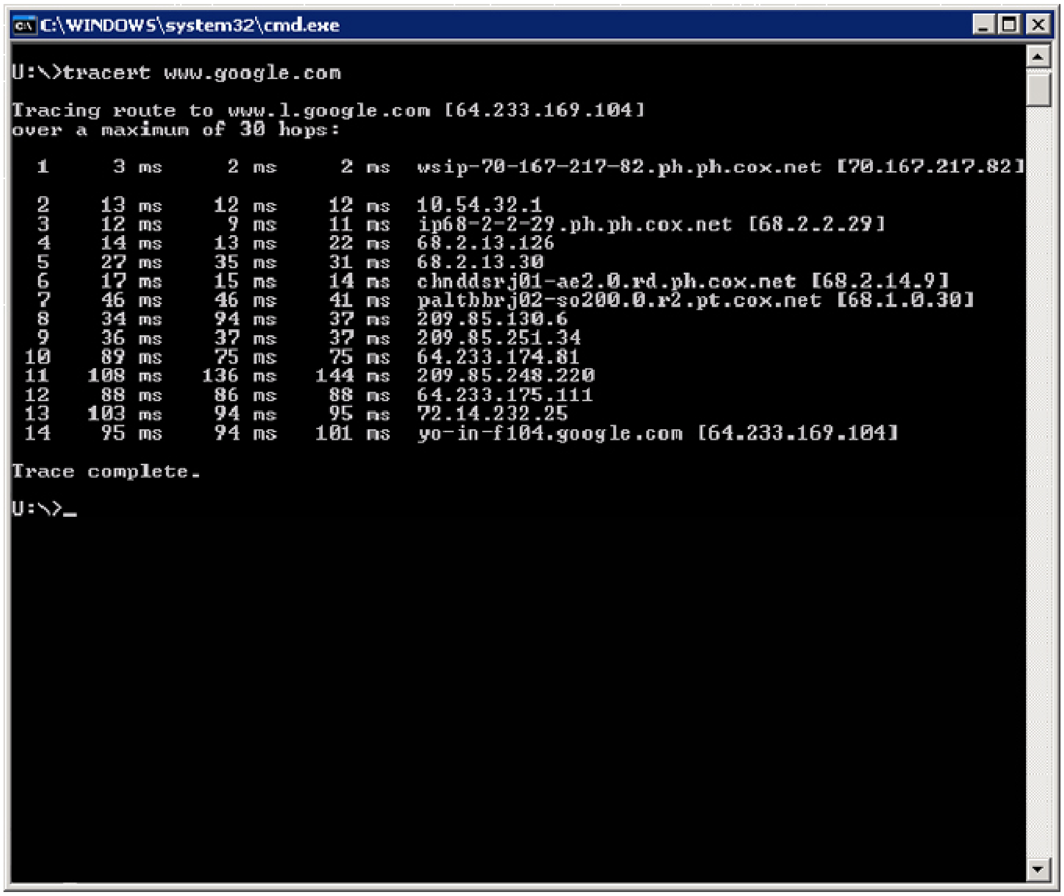
```
C:\Users\sjohnson.preplogic>ping google.com

Pinging google.com [72.14.207.99] with 32 bytes of data:

Reply from 72.14.207.99: bytes=32 time=83ms TTL=234
Reply from 72.14.207.99: bytes=32 time=99ms TTL=234
Reply from 72.14.207.99: bytes=32 time=99ms TTL=234
Reply from 72.14.207.99: bytes=32 time=82ms TTL=234

Ping statistics for 72.14.207.99:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 82ms, Maximum = 99ms, Average = 90ms

C:\Users\sjohnson.preplogic>
```

As you can see, the Microsoft Windows PC will send four ping requests to the destination by default and display the response time (in milliseconds) for each attempt. There are three common arguments used with the **ping** command:

- **ping –t <*hostname or IP address*>** - By using the **–t** flag, the ping will repeat continuously until stopped by using the ^c (Ctrl + c) break string.
- **ping –a <*IP address*>** - The **–a** flag causes ping to perform a reverse DNS lookup on an IP address. For example, issuing the command **ping –a 74.125.19.104** would return the google.com DNS name associated with the IP address.
- **ping –l** <***size in bytes*> <*hostname or IP address*>** - By default, a ping packet is very small. You can increase the size by using the **–l** argument (that is a lowercase L not a 1). For example, you could type **ping –l 1200 www.google.com** to send packets 1200 bytes in size to google.com. This is useful when stress-testing connections or servers.

3. tracert

The **traceroute** command (implemented as **tracert** in Microsoft Windows) is an enhanced version of a ping that shows every router you are passing through on the way to your destination. The syntax is as simple as typing **tracert <*hostname or IP address*>**:



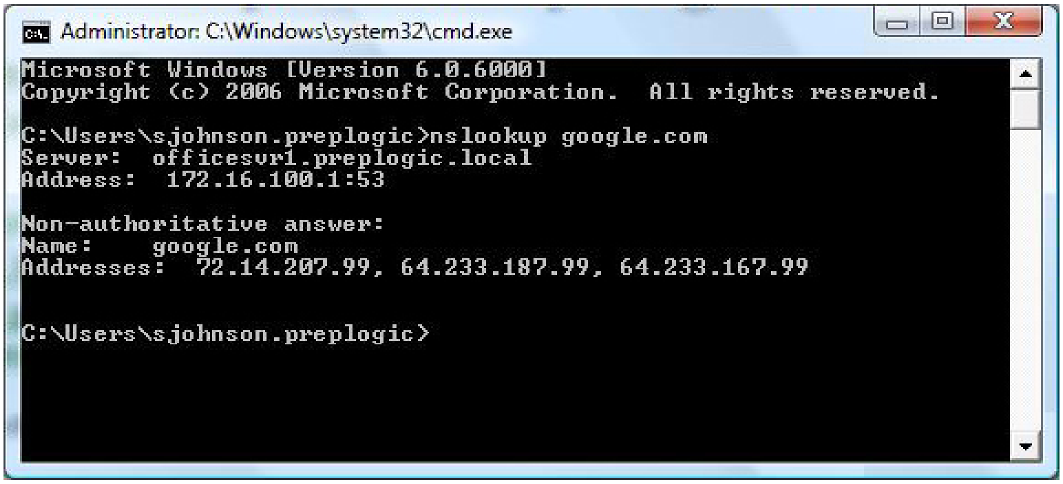
```
U:\>tracert www.google.com

Tracing route to www.l.google.com [64.233.169.104]
over a maximum of 30 hops:

  1     3 ms     2 ms     2 ms  wsip-70-167-217-82.ph.ph.cox.net [70.167.217.82]

  2    13 ms    12 ms    12 ms  10.54.32.1
  3    12 ms     9 ms    11 ms  ip68-2-2-29.ph.ph.cox.net [68.2.2.29]
  4    14 ms    13 ms    22 ms  68.2.13.126
  5    27 ms    35 ms    31 ms  68.2.13.30
  6    17 ms    15 ms    14 ms  chnddsrj01-ae2.0.rd.ph.cox.net [68.2.14.9]
  7    46 ms    46 ms    41 ms  paltbbrj02-so200.0.r2.pt.cox.net [68.1.0.30]
  8    34 ms    94 ms    37 ms  209.85.130.6
  9    36 ms    37 ms    37 ms  209.85.251.34
 10    89 ms    75 ms    75 ms  64.233.174.81
 11   108 ms   136 ms   144 ms  209.85.248.220
 12    88 ms    86 ms    88 ms  64.233.175.111
 13   103 ms    94 ms    95 ms  72.14.232.25
 14    95 ms    94 ms   101 ms  yo-in-f104.google.com [64.233.169.104]

Trace complete.

U:\>_
```

As you can see, the **tracert** command sends three ping requests to each router that it passes through. This allows you to find the bottleneck between the source and the destination. In the example above, 209.85.248.220 was the slowest router in the path between my local PC and www.google.com. There is only one common argument for the **tracert** command:

- **tracert –d <*hostname or IP address*>** - The **–d** argument prevents traceroute from resolving IP addresses to hostnames. For example, ip68-2-2-29.ph.ph.cox.net shown in the traceroute output above is a hostname, 68.2.2.29 is the IP address. This speeds up the traceroute command considerably.

4. **nslookup**

The **nslookup** command allows you to send multiple queries to a DNS server. There are many cases in network troubleshooting where problems originate because a DNS server has incorrect name-to-IP-address mappings in its database. **Nslookup** can help diagnose these issues. The following is an example of using **nslookup** to query a local DNS server:



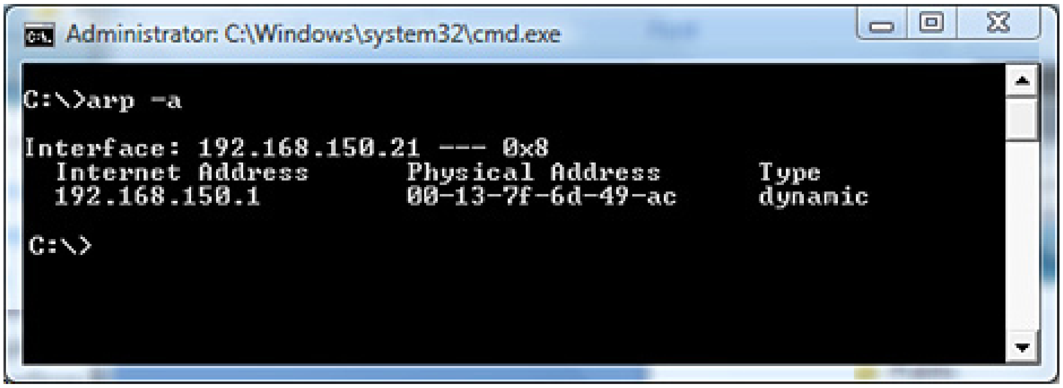
```
Microsoft Windows [Version 6.0.6000]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\sjohnson.preplogic>nslookup google.com
Server:  officesvr1.preplogic.local
Address:  172.16.100.1:53

Non-authoritative answer:
Name:    google.com
Addresses:  72.14.207.99, 64.233.187.99, 64.233.167.99


C:\Users\sjohnson.preplogic>
```

When you enter a domain name to resolve, nslookup will provide the name and IP address of the DNS server resolving the name (adtec.home.local/172.30.100.100 in this case) and the IP address(es) of the domain name you are resolving. Notice that the DNS server provided four IP addresses for www.google.com. This is because Google has a large enough web presence to have redundant servers supporting its domain name. There are many options that can be used with the nslookup command; two of them have common relevance to Cisco technicians. Keep in mind that both of these commands are entered after you have entered the base nslookup command:

- **server <*DNS server name or IP address*> -** The **server** selection option allows you to change DNS servers used for the DNS lookups. For example, in the nslookup output above, I was using the server "adtec.home.local". Perhaps I suspected that the adtec server was returning incorrect information. Using the **server** command, I could redirect my DNS requests to a different server. Tip: the DNS server **4.2.2.2** is a well-known public DNS server.
- **ls <*domain name*>** - There may be times where you want to see all the DNS records associated with a certain domain. For example, google.com contains DNS records for www, mail, images and so on. Typing **ls google.com** can display all these DNS records. Please keep in mind that many DNS servers restrict this command because of the secure information it can display.

5. arp

Whenever a network device attempts to communicate, it will need to have both the Layer 3 (IP address) and Layer 2 (MAC address) of its destination. The ARP command allows you to verify all of the Layer 2 to Layer 3 address mappings (known as Address Resolution Protocol or ARP mappings) a network client has stored in its cache (memory). For example, if the network client 192.168.150.21 attempted to communicate with 192.168.150.1, it would need to send an ARP broadcast to determine the MAC address for 192.168.150.1. Using the **arp** command-line utility, you are able to verify these mappings as shown below:



```
C:\>arp -a

Interface: 192.168.150.21 --- 0x8
  Internet Address      Physical Address      Type
  192.168.150.1         00-13-7f-6d-49-ac     dynamic

C:\>
```

In the example above, the IP address 192.168.150.1 is mapped to the MAC address 0013.7f6d.49ac. The **arp** command is unique in that you cannot issue the command without any arguments. The following are common arguments for use with the **arp** command:

- **arp –a -** The **–a** argument displays all entries currently in the arp table. This command is demonstrated in the previous graphic.
- **arp –d** - The **–d** argument manually deletes entries out of the arp table. By default, Windows will remember IP to MAC address mappings for 10 minutes. In a network where IP addresses are changing (usually due to network maintenance or upgrades), it may be beneficial to flush the arp cache and allow it to dynamically rebuild. Use the **arp –d *** syntax to remove all entries from the arp cache.

Communication Using TCP and UDP

In order to be able to understand and troubleshoot network connections, you should be intimately familiar with network communication using the most common Transport Layer protocols: the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). You might remember from the previous discussion of the OSI Model that TCP provides reliable, acknowledged transport, while UDP provides unreliable transport but faster connections. The following list will provide the "hard-facts" for both protocols.

User Datagram Protocol (UDP)

- **Connectionless**: Does not notify receiving device that information is about to be sent; the UDP application just sends information and assumes it is received.
- **Limited Error Checking**: UDP can optionally include a header checksum, which can be used to test if header corruption occurred when the message was sent.

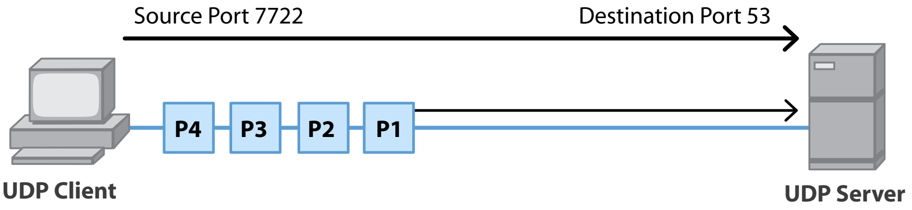
- **No Data Recovery**: The UDP protocol itself cannot recover data if it is lost or contains a bad checksum. If you would like a reliable UDP application, reliability must be coded into the application itself (such is the case with TFTP, a common UDP application).
- **No Sequencing**: UDP does not have the ability to tell what order packets were sent in. Therefore, any data received out of order may be dropped by the receiving device.

Transmission Control Protocol (TCP)

- **Connection Oriented**: Before any data is sent, an active session (connection) is set up between network devices.
- **Error Checking**: TCP always includes a header checksum, which can be used to test if any header corruption occurred when the message was sent.
- **Data Recovery**: After the initial session is created between two network devices, all transmissions are acknowledged (ACKed) to ensure data is not lost. If a message is not acknowledged, it will be retransmitted.
- **Sequencing**: When data crosses a complex TCP/IP network, such as the Internet, there is the potential that data can take multiple paths and end up out of order. TCP uses sequence numbers on all transmitted data to ensure it is able to be placed in the correct order.

Because of its reliability, TCP is by far the most popular protocol for day-to-day application such as web browsing, email and file transfers. UDP is typically used in real-time applications such as VoIP, video over IP and online gaming.

UDP Communication
Understanding UDP network communication is fairly simple since there is no session or error checking involved. The following figure shows UDP communication in action:



In this case, a UDP client is sending four packets to a UDP server. As seen before in our OSI Model discussion, the communication will come from a source port, which designates the client application it came from, and will be directed to a destination port, which designates the server application it would like to access. All four packets are sent without warning and without expecting any acknowledgement.

There are only two common UDP port numbers that you should be able to recognize at the CCNA level:

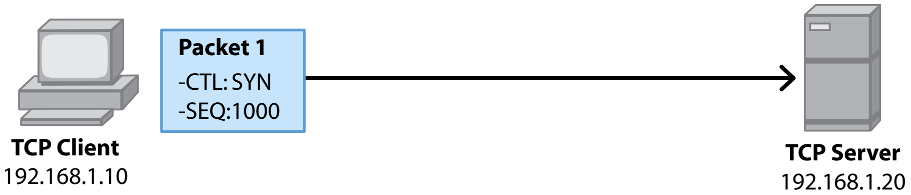
- **UDP Port 53**: DNS Requests (used when a client wants to lookup a DNS name)
- **UDP Port 69**: TFTP File Transfers (used to copy configuration and IOS files to and from a Cisco device)

TCP Communication

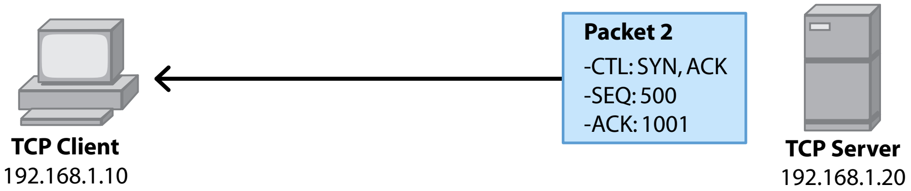
TCP is quite a bit more complex than UDP, simply because of all the reliability mechanisms it includes. The foundation that TCP communication relies on is in building a session. This is accomplished using a process known as a TCP 3-Way Handshake.

TCP Three-Way Handshake

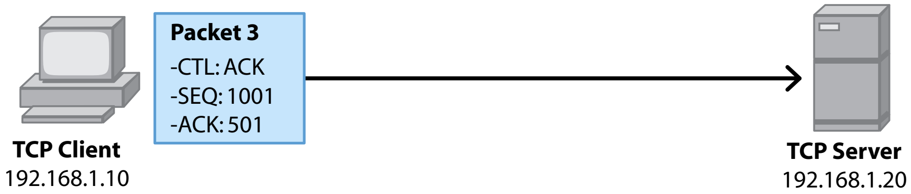
A TCP 3-Way Handshake is *always* the initial communication between two devices. It's very similar to when you first meet someone. You don't run at them initially spewing information (data); instead, you take the time to introduce yourself and find out the identity of this new person. In the figure below, the TCP client (192.168.1.10) has started to communicate with the TCP server (192.168.1.20).



In the initial packet, the client uses two TCP header fields to relay some key information. The first is the Control (CTL) field. The CTL field dictates what type of TCP packet this is. In this case, it is a Synchronization (SYN) message, letting the remote server know the client is beginning communication. The second field is the Sequence Number (SEQ). As data is sent, this sequence number will continue to increase to ensure the data remains in order. This initial packet is telling the server the starting sequence number for the client. In this case it's 1000. Now let's see how the server responds:



Notice that the server has now changed the CTL field to SYN, ACK. That's essentially the server saying, "I received your request to start communication (ACK), so let me start communication with you (SYN)." Since the server is a completely different device than the client, it will have its own starting sequence numbers for communication. In this case, it will begin by starting with SEQ 500. You can also notice a new field in the packet, which is the acknowledgement (ACK) number. Notice that it is 1001, one more than the TCP clients sequence number. This is the server saying, "TCP client, I received your SEQ 1000, so I'll be expecting your *next* sequence number to be 1001." The ACK numbers should always be one more than the last sequence number unless data was lost during transmission. Now, let's take a look at the final message of the TCP 3-Way Handshake.

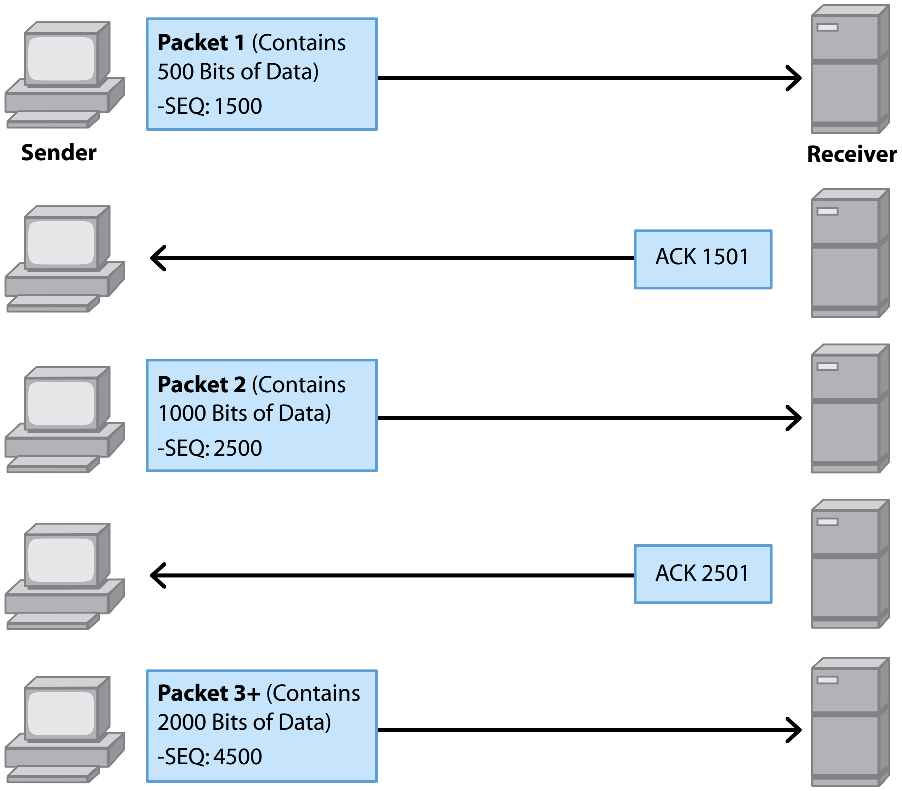


Once the client receives the SYN, ACK packet, it processes it and responds with a final ACK. This is the client saying, "I received your SYN, ACK…let's start talking!" Notice that the client uses SEQ number 1001 to respond to the TCP server, which is what the TCP server was expecting. It also sends ACK 501, which tells the TCP server, "I received your SEQ 500, I'll be expecting 501 next." Once this third packet is received by the TCP server, the session is built and data can be sent.

TCP Windowing

Once the TCP session has been set up, the two devices will begin to transmit data using the SEQ and ACK numbers to keep everything straight. Continuing the example from the TCP Three-Way Handshake, the TCP client might send 1 byte of data to the TCP server. Since the last SEQ number used was 1001, the single byte of data will be sent with SEQ number 1002. When the server acknowledges receipt of the data, it will reply with ACK number 1003. If the client has 1,000,000 bytes (1 Megabyte) of data to send, sending one byte of data at a time is an *extremely* inefficient process to accomplish this. Think of it as having a conversation with a person, but you can only say one word at a time and must wait for each word to be acknowledged before saying another word.

To help with this issue, the idea of TCP windowing was introduced. TCP windowing allows a device to send more and more data while receiving only one ACK. As the devices detect a more reliable connection, the amount they send (known as the window size) increases. The figure below demonstrates this concept.

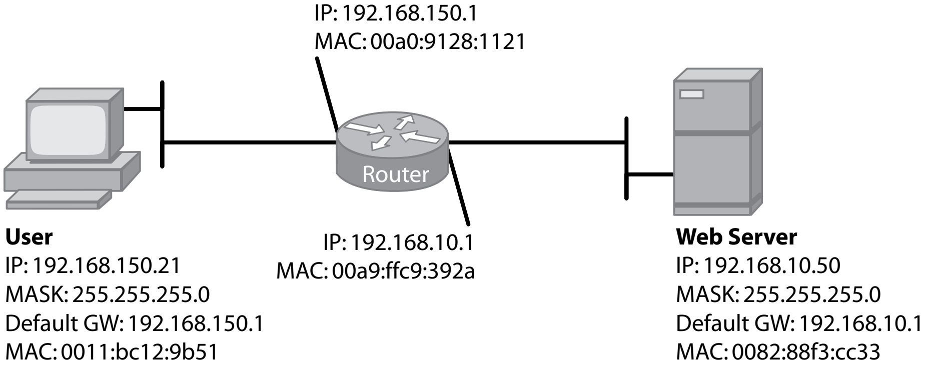

Packet 1 (Contains 500 Bits of Data)
-SEQ: 1500
Sender
Receiver
ACK 1501
Packet 2 (Contains 1000 Bits of Data)
-SEQ: 2500
ACK 2501
Packet 3+ (Contains 2000 Bits of Data)
-SEQ: 4500

Rather than start by sending a single byte of data, most operating systems will begin somewhere around 500 bytes of data. Each time a successful acknowledgement comes back, the sending PC increases its window size by a factor of two, sending 500, then 1000, then 2000 bytes of data. Again, the amount the window size increases is dependent on the operating system, but doubling the window size each time is typical. This process will continue until the sending computer loses some data during the transmission (due to the receiving computer or some network device between the sender and receiver not being able to keep up) or until the receiving computer communicates back that its maximum TCP window size has been reached. There is one last item of note in the concept of TCP sliding windows: you might notice that the final transmission in the previous graphic is shown as **Packet 3+**. This is because most networks have a maximum packet size (known as the Maximum Transmission Unit or MTU) of 1500 bytes. So, for the final transmission, the PC is most likely sending multiple packets for each acknowledgement received.

If you have ever copied a large file to a server, you may have seen the idea of TCP sliding windows in effect. Initially, the amount of time to copy the file is reported as some astronomically large amount, such as 15 hours. As you let a few seconds pass, you notice that the time has decreased dramatically to 4 or 5 hours. After a few more seconds, the copy time estimate has decreased to 2 to 3 hours. You are seeing the effect of TCP sliding windows before your very eyes. As the window size becomes greater, the transmissions become more efficient and the time to copy the data becomes significantly less.

End-to-End Network Communication

After seeing all of these concepts, you may find it helpful to see a complete example of end-to-end network communication between two devices. For this example, we will use the following network diagram:



The user shown on the left of the diagram has opened a web browser to access the web server on the right. In the address bar of the web browser, the user types **http://192.168.10.50**. The following describes the general process of communication:

Phase 1: Initial Network Communication

- The web browser passes the request down to the operating system. Based on the application being used, the operating system realizes this will be TCP-based communication.
- In order to direct the data to the correct application, the operating system tags the packet with the destination port number of **80** (the well-known port number for HTTP). To allow return traffic to the client, the operating system dynamically assigns port **53,422** to the web browser application as the source port number.
- Since this is the initial communication with the web server, the user PC must complete a TCP Three-Way Handshake to build the session. The initial packet sent will be a TCP SYN.

- The TCP SYN data is tagged with the source IP address of **192.168.150.21** (the user) and a destination IP address of **192.168.10.50** (the web server).
- The user's PC is able to determine that the remote web server is on a different network. Because of this, it sends an ARP broadcast to find the MAC address of its default gateway (**192.168.150.1**). The default gateway responds to the ARP message with its MAC address (**00a0:9128:1121**).
- The user's PC adds the source MAC address (**0011:bc12:9b51**) and destination MAC address (**00a0:9128:1121**) as the Layer 2 header of the TCP SYN frame and transmits the data.
- The router receives the frame and checks the destination MAC address. Since it sees itself as the destination MAC address, it begins processing the Network layer (IP address) information.
- Since the router sees 192.168.10.50 as the destination IP address, it realizes this packet is meant to be sent *through* the router rather than *to* the router. The router looks at its routing table and sees it is attached to the 192.168.10.0/24 network.
- The router removes the original source MAC address (**0011:bc12:9b51**) and destination MAC address (**00a0:9128:1121**). Before it can send the data to the web server, it must replace these addresses with MAC address information relevant to the web server's network. The router sends an ARP broadcast to find the MAC address of the web server (**192.168.10.50**). The web server responds to the ARP message with its MAC address (**0082:88f3:cc33**).
- The router places the correct source MAC address (**00a9:ffc9:392a**) and destination MAC address (**0082:88f3:cc33**) onto the TCP SYN frame and sends the data out of the interface where the web server resides.
- The web server receives the TCP SYN frame. Since it sees the destination MAC address as itself, it begins processing the Network layer (IP address) information.
- Since the web server sees 192.168.10.50 as the destination IP address, it realizes this packet is destined for itself and begins processing the Transport layer (in this case, TCP) information.
- As the web server processes the Transport layer information, it realizes this is a TCP SYN packet, which indicates a host is attempting to begin network communication.

Phase 2: Ongoing Network Communication

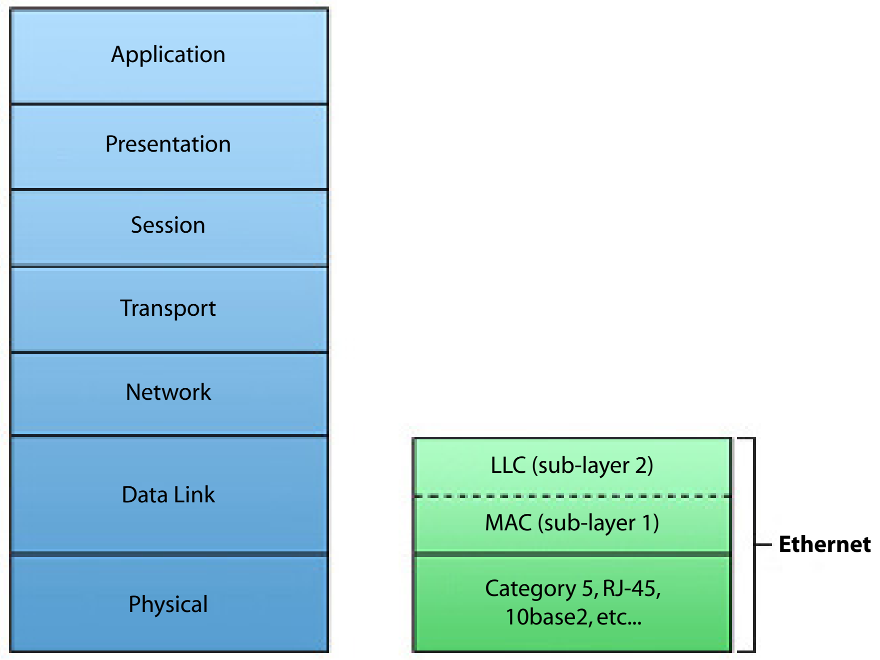
- The web server replies to the TCP SYN message with a SYN ACK message. The SYN ACK message is tagged with a source IP address of **192.168.10.50** (the web server) and a destination IP address of **192.168.150.21** (the user).
- The web server realizes the destination IP address is not on its network and adds its own source MAC address (**0082:88f3:cc33**) and the destination MAC address of the router (**00a9:ffc9:392a**). An ARP broadcast is not necessary since this MAC address information has been cached. The frame is sent to the router.
- The router receives the frame and checks the destination MAC address. Since it sees itself as the destination MAC address, it begins processing the Network layer (IP address) information.
- Since the router sees 192.168.150.21 as the destination IP address, it realizes this packet is meant to be sent *through* the router rather than *to* the router. The router looks at its routing table and sees it is attached to the 192.168.150.0/24 network.

- The router removes the original source MAC address (**0082:88f3:cc33)** and destination MAC address (**00a9:ffc9:392a**). Before it can return the data to the user, it must replace these addresses with MAC address information relevant to the user's network. The router adds its own MAC address as the source (**00a0:9128:1121**) and the user's MAC address (**0011:bc12:9b51**) as the destination. An ARP broadcast is not necessary since this MAC address information has been cached. The frame is sent to the user's PC.
- The user's PC receives the frame and processes it due to its MAC address and IP address being identified in the header. Once the data reaches the Transport layer of the OSI Model, the user's PC sees the SYN ACK message and replies with an ACK. The sequence numbers (SEQ) and acknowledgement numbers (ACK) are now synchronized between the two devices.
- Once the web server receives the ACK, the TCP Three-Way Handshake is complete and the web server begins to send data to respond to the user's request(s).

Ethernet Foundations

Developed in the 1970s, Ethernet has become the fabric of LANs around the world. This initial introduction is designed to "hit the high-points" of Ethernet networking.

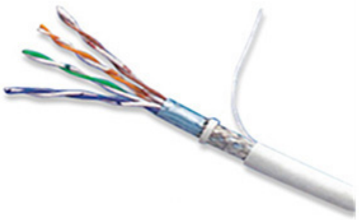
Ethernet is a Physical and Data Link layer standard, when matched to the OSI Model, as shown in the following figure:



In the big picture of networks, you can swap out Ethernet for many other technologies (such as Serial WAN links, Token Ring, Wireless and so on) without changing the functions of TCP/IP. Remember, TCP/IP operates from the Network layer through the Application layer. So, let's work through Ethernet, starting with the Physical layer aspects.

Ethernet: The Physical Layer

While Ethernet has gone through many evolutions of physical cabling, the most important to us today is Unshielded, Twisted-Pair (UTP) cable. This physical cabling standard specifies eight individual wires that are twisted together into a cable and crimped using an RJ-45 tip.

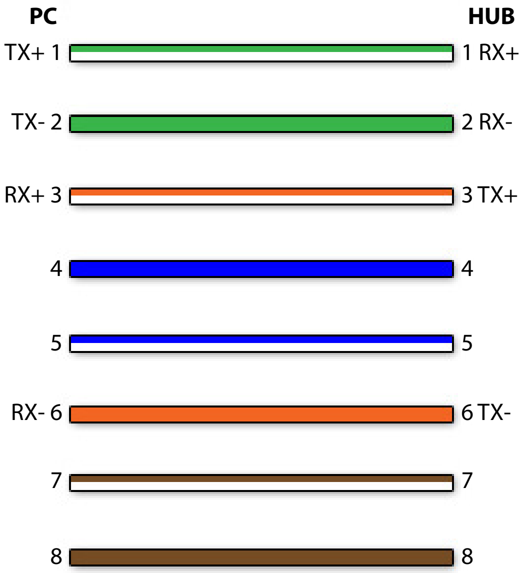


UTP cable without an RJ-45 tip

UTP cable with an RJ-45 tip

The CCENT and CCNA exams are not so much focused on Ethernet cabling standards and Physical layer characteristics as they are cable decision making. Simply put, Cisco wants to know, "Do you know which Ethernet cable to use?" This decision comes down to a choice between Ethernet Straight-Through and Ethernet Crossover cabling.

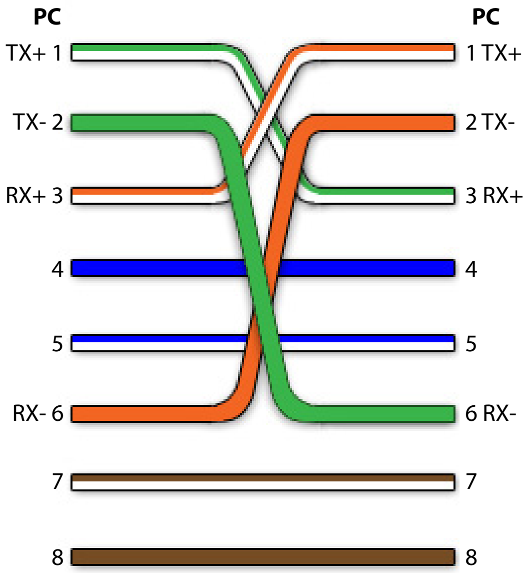
In the LAN environment, there are devices that are designed to naturally connect. For example, a server, PC or router will typically plug directly into a network switch or hub. Because the devices are engineered to connect this way, the network plugs are designed physically differently. A PC will send data on pins 1 and 2 of the Ethernet cable (known as Tx pins). A switch or hub is designed to receive data on pins 1 and 2. Likewise, a PC is designed to receive data on pins 3 and 6 (known as the Rx pins), and a switch or hub is designed to send data on these pins. This is visually demonstrated in the figure below:



Whenever you are making these "natural" connections, you are able to use an Ethernet straight-through cable. An easy way to remember this is that it is used when you are connecting "unlike" devices.
Some examples of these include:

- PC to switch
- Router to switch
- Printer to hub
- PC to hub
- Server to switch

While these connections are the most common in LAN environments, you will also run into cases where you have a need to connect "like" devices. If you attempt to do this using an Ethernet straight-through cable, the devices will not be able to communicate since they are wired to send and receive on the same pins. It would be synonymous to two individuals attempting to speak without listening or listen without speaking. In these cases, you will need to employ an Ethernet crossover cable, which crosses the send and receive pins, as shown in the figure below:



Examples of connections requiring a crossover cable include:

- Switch to switch
- Switch to hub
- PC to PC
- PC to router
- Router to router

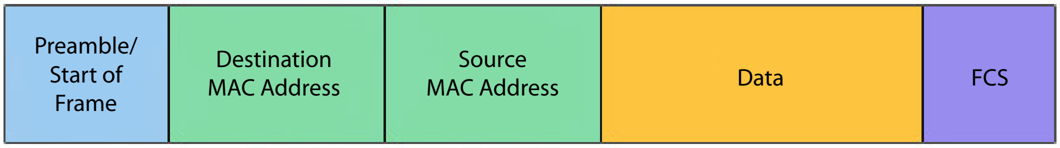
Ethernet: The Data Link Layer

Physically connecting devices using the correct cables is just a small portion of Ethernet communication. The connected devices must now understand what they are saying, which is the job of the Ethernet Data Link layer: correctly formatting electrical signals so they are understandable by each end. To accomplish this, the Ethernet layer is divided into two, smaller sub-layers:

- **Media Access Control (MAC)** - Handles Ethernet frame formatting and addressing
- **Logical Link Control (LLC)** - The connecting layer which allows data to be passed to the correct Network layer protocol

Since the functions of the LLC sub-layer are simpler, let's discuss those first. In today's world, TCP/IP reigns as the network protocol of choice, but it was not always this way. Other protocols such as IPX/SPX, AppleTalk and DecNET were in use in many network environments. The LLC layer of Ethernet provides the capability to select alternate Network layer protocols. Without this layer, Ethernet would be tied to a specific Network layer protocol that could never change unless the whole Ethernet standard was changed.

The MAC sub-layer is primarily responsible for error correction and addressing functions. The error correction comes in through a small check known as the Frame Check Sequence (FCS), which is added on at the end of the frame:



The FCS is the result of a mathematical formula run on the entire frame. This is technically known as a hash. When a PC decides to send data across the network, just before it is sent, it runs a mathematical formula on the entire frame and puts the result of this formula (the hash) in the FCS field at the end. When the data is received, the first thing the receiving device does is to run the same formula on the data and compare the result to the result contained in the FCS. If the results match, the frame is considered good and will be processed. If the results do not match, it means the data in the frame must have become corrupted or maliciously modified (by a network intruder). The frame is considered damaged and will be dropped.

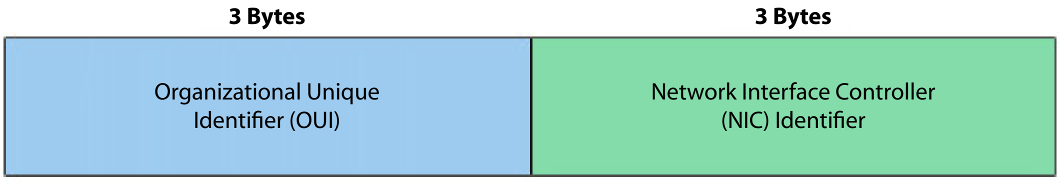
The second major function of the MAC sub-layer is addressing. In the Network Foundations section, we discussed the concept behind the MAC address but never looked specifically at its format. The Ethernet MAC address is 6-bytes long and can be written in a variety of formats, depending on the type of equipment you are working with. The following are examples of the same MAC address written different ways:

- 00-0C-29-9C-F9-F4
- 000C.299C.F9F4
- 000C:299C:F9F4
- 000C299CF9F4

This can sometimes throw off network technicians who are used to the rigidity of the format of an IP address.

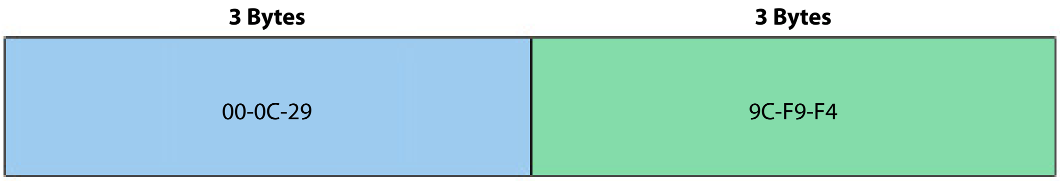
Each MAC address uniquely identifies each network card in the world. With 6-bytes of data (48-bits), it is possible to have 2^{48} or 281,474,976,710,656 possible MAC addresses. Each MAC address is comprised of two pieces: the Organizational Unique Identifier (OUI) and a Network Interface Controller (NIC) identifier.

The OUI is assigned to organizations producing network equipment. For example, if Intel decided to create a new network card, they would apply for an OUI which is assigned to the Intel organization. If Intel were assigned 00-1b-33 as their OUI, they would begin creating network cards starting with this OUI as the MAC address (for example, 00-1b-33-00-00-01, 00-1b-33-00-00-02, and so on).



3 Bytes	3 Bytes
Organizational Unique Identifier (OUI)	Network Interface Controller (NIC) Identifier

MAC Address Format



3 Bytes	3 Bytes
00-0C-29	9C-F9-F4

Example

Ethernet Communication

Ethernet was designed to use a method of transmission known as Carrier Sense, Multiple Access with Collision Detection (CSMA/CD). This standard defines the rules Ethernet must live by when communicating. Based on the Ethernet standard, only one device connected to an Ethernet segment is able to send or receive at a time, otherwise a collision occurs and the data must be re-sent. The rules of CSMA/CD follow this flow:

1. Send the network device package's data into a frame to be sent.
2. The network device listens to the Ethernet wire to see if another device is already transmitting.
3. If a device is transmitting, wait until it finishes. If the line is idle, send the data.

This system works flawlessly unless two network devices happen to be listening to the Ethernet wire at the same time. The more devices you add to the network, the chances of this happening increase. If two devices send data at the same time, the following CSMA/CD procedure occurs:

1. Collision is detected by the sending network devices.
2. A jam signal is transmitted on the Ethernet wire. This signal causes all devices to stop sending.
3. The sending devices set a random retransmit time and send the data again, hoping not to collide.
4. If another collision is detected, the random retransmission timer is continually increased until the data can be sent without colliding.

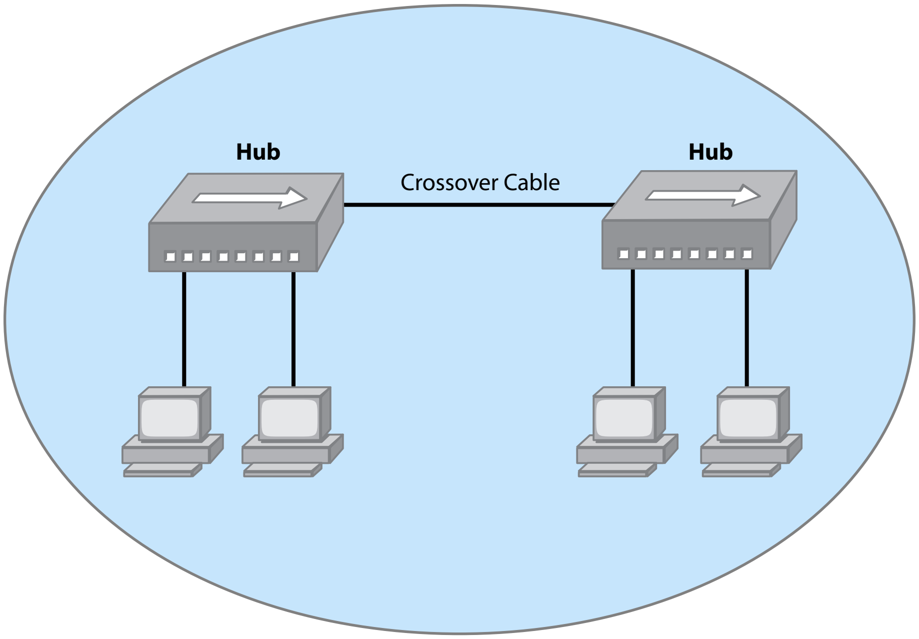
The more collisions you have on a network, the slower your network performs.

Ethernet Network Equipment

From the network client side, anything with an Ethernet-capable NIC is obviously part of the Ethernet network equipment realm; however, as Cisco engineers, we're mostly interested in the network infrastructure realm. In this realm, there are two network devices of concern: the hub and the switch. To have a complete appreciation for these devices, you must understand the concept of collision domains.

Understanding Collision Domains

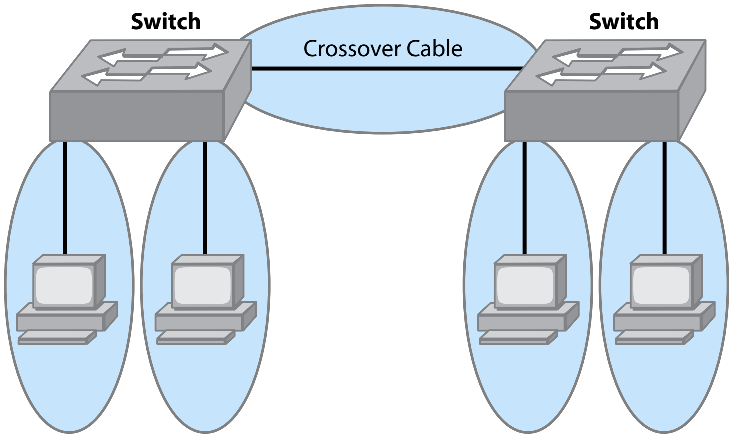
A collision domain represents a shared Ethernet segment where only one device can send or receive at a time. Hubs, which are older network equipment, can only support a single collision domain. This means that no matter how many ports a hub has or how many hubs are connected using crossover cables, there will only be a single collision domain:



1 Collision Domain

This means that only one device connected to either of the two hubs pictured above will be able to send or receive at a time. If more than one device attempts to transmit, a collision will occur and be handled by the rules of CSMA/CD.

This brings us to the major difference between hubs and switches. A network switch isolates each port into its own collision domain:



5 Collision Domains

This allows all four PCs pictured in the previous figure to send *and* receive at the same time (if they are able to run in full-duplex, which we will discuss in a moment). A switch-based network allows a network to be tremendously more efficient than a hub-based environment.

As I just mentioned, switches provide the capability of full-duplex. This allows a network device to send *and* receive at the same time rather than half-duplex, which allows a network device to send *or* receive at the same time. All network equipment is rated as if it were running at half-duplex, so when you use a 100-Mbps NIC, you can actually achieve 200-Mbps of throughput if operating in full-duplex (100-Mbps sending and 100-Mbps receiving at the same time).

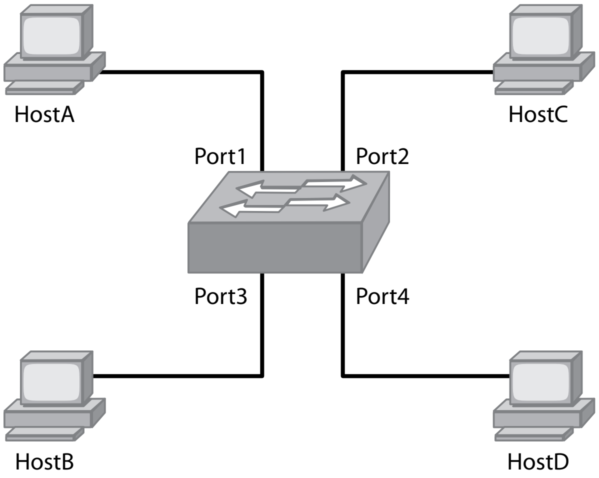
Hubs and Switches, Exposed

In addition to multiple collision domains, switches offer additional benefits over hubs:

- **Dedicated Bandwidth** - Since each host is isolated into their own network segment (collision domain), they are dedicated the full amount of bandwidth the switch port can provide. If you have a 100-Mbps switch, each attached port will receive a full 100-Mbps of bandwidth. If you have a 100-Mbps hub, the bandwidth is divided among the devices needing to transmit data. If four devices have data to send, the bandwidth of the hub would be divided between them.
- **Data Link Layer Intelligence -** Ethernet switches have the ability to learn the MAC addresses of the attached device(s). By building a MAC address table in memory, it can then direct messages out specific ports. For example, if HostA wanted to send data to HostB, the switch could allow the data to only reach HostB rather than sending it to all attached devices. When using hubs (which is a Physical layer device), every message is sent out of all ports regardless of the destination.
- **Speed Mismatches** - Since each of the ports are handled individually, switches are able to handle variable speed ports. For example, you can have a 24-port switch equipped with 22 1-Gbps ports and two 10-Gbps ports. Hubs must have all ports set to equal bandwidth amounts.

Switch MAC Processing

Because it is a Data Link layer device, a switch has the ability to filter traffic based on MAC addresses it can dynamically learn. When a switch initially boots, its MAC address table is completely empty. As network devices transmit data, the switch becomes more intelligent. This process is shown below:



- After initial bootup, the switch MAC address table is empty.
- HostA sends data to HostB. The switch immediately adds the MAC address of HostA to its MAC address table and associates it with Port 1.
- Because the switch does not know the location of HostB, it floods the message from HostA out all ports *except* the port where the message was received.
- HostB sends data back to HostA on Port 3. The switch immediately adds the MAC address of HostB to its MAC address table and associates it with Port 3.
- Since the switch knows the location of HostA, the information is only forwarded out Port 1. HostC and HostD do not receive the information.

Domain 3 - IOS Foundations

IOS Definition

The Cisco Internetwork Operating System (IOS) is the operating system that powers the vast majority of Cisco routers and switches. Learning the operation of this command-line interface is critical to your survival in the Cisco realm. Cisco has designed this command-line interface to be easy to use and navigate once you have learned the foundations. This section is focused on just that: learning the foundations of working with the Cisco IOS. Think of this as being similar to getting a training course on how to work with Microsoft Windows. In the Windows realm, we would look at things such as learning to use the mouse, the Start menu and the Control Panel. In this introductory training, we will focus not so much on how to perform configurations but, rather, just how to get around.

Connecting to the Cisco Switch or Router

When you pull a Cisco switch or router out of the box, it will have little or no configuration. In order to connect to the Cisco switch or router, we must use a specialized Console cable. This type of cable is also known as a rollover cable since the eight Ethernet pins roll over between the ends. Thus, the Ethernet pins of a rollover look like this:

Pin 1 > Pin 8
Pin 2 > Pin 7
Pin 3 > Pin 6
Pin 4 > Pin 5
Pin 5 > Pin 4
Pin 6 > Pin 3
Pin 7 > Pin 2
Pin 8 > Pin 1

This can be useful if you are attempting to create your own console cable. On one end of the console cable, you will need a DB-9 serial port adapter, which connects to the PC. Many of the newer Cisco console cables have these serial adapters built-in:

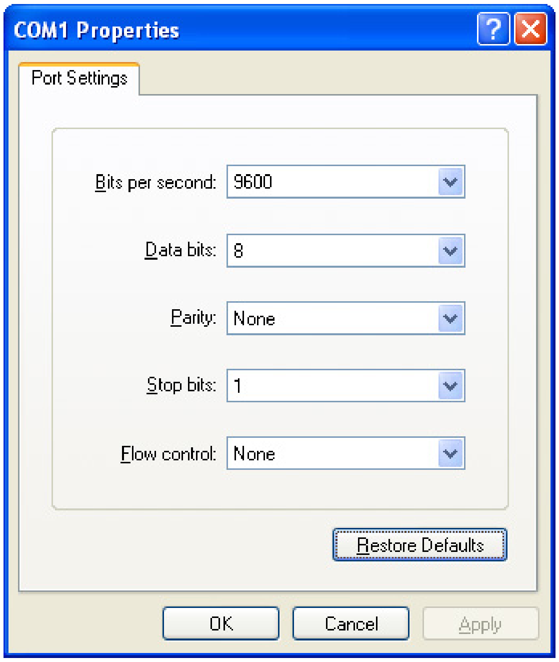
After you have connected one end of the console cable to your PC and the other end to the Cisco switch or router, you will need a terminal emulator program to interact with the Cisco router. The following is a list of common terminal emulators:

- HyperTerm
- Tera Term
- Putty
- Minicom
- SecureCRT

Once you have opened one of these programs, you will need to select your PC COM port for the connection. The COM port MUST be configured with the following settings:

- Baud Rate: **9600**
- Data Bits: **8**
- Parity: **None**
- Stop Bits: **1**
- Flow Control: **None**

The following is an image of the proper setup using HyperTerm:



COM1 Properties

Port Settings

Bits per second: 9600

Data bits: 8

Parity: None

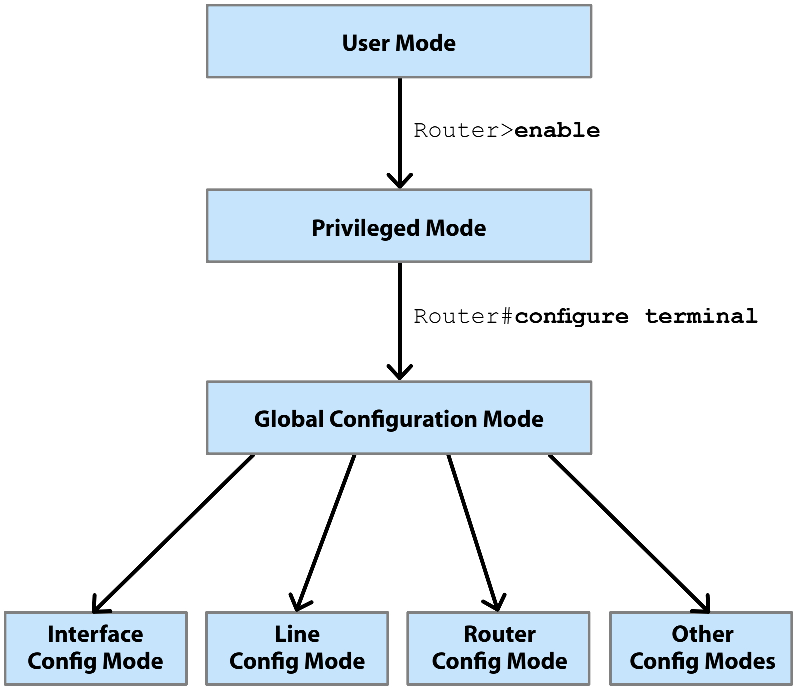
Stop bits: 1

Flow control: None

Once you have configured these settings, you can click OK and press the Enter key a few times in the terminal program. A prompt from the Cisco device should appear.

Understanding the Cisco IOS Modes

When working with the Cisco IOS, understanding the relevance of the mode you are in is almost as important as understanding the command you should type. There are hundreds of different modes, each of which allows you to configure a different aspect of a Cisco device. The modes are always accessed through this general flow:



The following is a brief description of each mode:

- **User Mode** - This is the initial mode you access when you log into the Cisco IOS device. From User Mode, you have extremely limited access to view information such as interface status, router uptime and IOS version information. Commands that could expose any security-related information are disabled in this mode.
- **Privileged Mode** - This mode allows you to view the full configuration of the Cisco IOS device and provides access to troubleshooting and testing utilities such as the **debug** commands. This is the only mode which allows you to save your IOS configuration.
- **Global Configuration Mode** - From this mode, you can change global settings on the Cisco IOS device. Some examples of these settings are the name of the IOS device, a logon banner and the privileged mode password. More importantly, from global configuration mode, you can access any of the sub-configuration modes.
- **Interface/Line/Router/Other Configuration Modes** - After moving into global configuration mode, you are able to access any of the sub-configuration modes. These modes give you an interface to configure specific aspects of the router or switch. For example, you could configure an individual interface by moving into Interface configuration mode.

The IOS prompt always consists of two pieces: the hostname of the device and the mode you are currently accessing. These are formatted as <***hostname***><***mode***>. The following is an example of what each of the modes look like:

- **User Mode** - RouterA> (note the **>** symbol indicates user mode)
- **Privileged Mode** - RouterA# (note the **#** symbol indicates privileged mode)
- **Global Configuration Mode** - RouterA(config)#
- **Interface Configuration Mode -** RouterA(config-if)#

The following syntax example allows you to visualize moving through these modes:

```
RouterA>
RouterA>enable
Password:
RouterA#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RouterA(config)#interface fastethernet 0/1
RouterA(config-if)#end
RouterA#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RouterA(config)#interface fastethernet 0/1
RouterA(config-if)#exit
RouterA(config)#exit
RouterA#
```

Let's discuss a few of the key navigation commands used here:

- **Enable -** The **enable** command moves you from user mode to privileged mode. On a configured Cisco IOS device, this transition is always password protected (as shown in the example). On an unconfigured Cisco IOS device, there will be no password required.
- **Configure Terminal** - The **configure terminal** command moves you from privileged mode to global configuration mode.
- **End / CTRL+Z** - The **end** command and the **ctrl-z** keystroke are assigned an identical function: exit all configuration modes and drop back to privileged mode. No matter how deep you are in the configuration modes, these two commands will always back you out completely.
- **Exit** - The **exit** command moves you back through configuration modes one mode at a time. In the example syntax, you can see that we moved from interface configuration mode back to global configuration mode by typing **exit** the first time. We then moved from global configuration mode back to privileged mode by typing **exit** the second time. If we were to type **exit** a third time (from privileged mode), we would log out of the IOS device.

Cisco IOS Help and Shortcuts

Without a doubt, it is the Cisco IOS help and shortcut systems that made this operating system famous in the networking industry. Thankfully, if you are studying for a certification exam, Cisco has made the help and shortcut systems available in all exam simulation questions. Let's first discuss the help system.

No matter where you are in the Cisco IOS, you can use the question mark key (**?**) to see a list of commands available:

```
RouterA#?
Exec commands:
  <1-99>            Session number to resume
  access-enable     Create a temporary Access-List entry
  access-template   Create a temporary Access-List entry
  archive           manage archive files
  cd                Change current directory
  clear             Reset functions
  clock             Manage the system clock
  cns               CNS agents
  configure         Enter configuration mode
  connect           Open a terminal connection
  copy              Copy from one file to another
  crypto            Encryption related commands.
  …<output omitted>…
```

These commands are always listed alphabetically. The **?** help system stretches beyond just listing commands. You can use it to receive a filtered list of commands by placing the **?** directly following specific letters. For example, typing **c?** produces the following list of commands:

```
RouterA#c?
cd       clear  clock   cns  configure
connect  copy   crypto
```

This represents every command in the current privileged mode that starts with the letter c.

Finally, you can use the **?** to walk through complete command syntax. For example, let's use the **clock** command (shown above) to demonstrate:

```
RouterA#clock ?
  set  Set the time and date

RouterA#clock set ?
  hh:mm:ss  Current Time

RouterA#clock set 13:05:00 ?
  <1-31>  Day of the month
  MONTH   Month of the year

RouterA#clock set 13:05:00 December ?
  <1-31>  Day of the month

RouterA#clock set 13:05:00 December 26 ?
  <1993-2035>  Year

RouterA#clock set 13:05:00 December 26 2010 ?
  <cr>

RouterA#
```

Notice, I used the **?** to prompt the next portion of the command to enter. When using the **?**, you do not need to press the Enter key; the Cisco IOS recognizes the symbol immediately. Because of the vast amount of commands used in the Cisco IOS, the **?** can quickly become your best friend.

Cisco has also engineered an intuitive syntax error system in the IOS. There are three types of messages you will receive if you enter a command incorrectly:

- **Incomplete Command** - This error appears if you have not typed enough information for the IOS device to process the command.

  Example:

  ```
  RouterA#clock set 13:05:00 December 26
  % Incomplete command.
  ```

- **Ambiguous Command** - This error appears if you have not typed enough of a command for it to be uniquely recognized. This often occurs when using shortcuts, which we will discuss in a moment.

  Example:

  ```
  RouterA#cl
  % Ambiguous command:  "cl"
  ```

- **Invalid Input Detected -** This error message appears when you have mistyped or misspelled a command in the IOS. The Cisco IOS will even point out where the mistype occurred, using the ^ symbol.

  Example:

  ```
  RouterA(config)#interfuce fastethernet 0/1
                        ^
  % Invalid input detected at '^' marker.
  ```

Now we can discuss the Cisco IOS shortcuts. The Cisco IOS command parsing system will allow you to type a shortcut for any command, as long as you type enough characters for the command to be uniquely recognized. For example, let's say we wanted to get into global configuration mode on our router:

```
RouterA#c?
cd       clear  clock   cns  configure
connect  copy   crypto

RouterA#conf ?
  confirm            Confirm replacement of running-config with a new config
                     file
  memory             Configure from NV memory
  network            Configure from a TFTP network host
  overwrite-network  Overwrite NV memory from TFTP network host
  replace            Replace the running-config with a new config file
  terminal           Configure from the terminal
  <cr>

RouterA#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
RouterA(config)#
```

Notice, I was able to type **conf** instead of the entire word **configure** since these four letters allow the Cisco IOS to recognize the command uniquely. If I were to just type **con** and press enter, the IOS would return the ambiguous command error message. For the second portion, I was able to type **t** instead of the entire word **terminal** since there were no other commands that began with the letter t.

Using the Tab key can be helpful in shortcut processing. As soon as you have typed enough characters, the IOS will complete the rest of the command for you:

```
RouterA#c  <tab key pressed here, not enough characters>
RouterA#co  <tab key pressed here, not enough characters>
RouterA#con  <tab key pressed here, not enough characters>
RouterA#conf  <tab key pressed here, the IOS completes the command>
RouterA#configure t  <tab key pressed here, the IOS completes the command>
RouterA#configure terminal
```

Finally, the Cisco IOS has some handy navigation commands that can make you more efficient when working in the command-line interface.

Command	Function
Ctrl+P (or up arrow)	Displays the last command entered
Ctrl+N (or down arrow)	Displays previous commands entered
Ctrl+A	Moves the cursor to the beginning of the current line
Ctrl+E	Moves the cursor to the end of the current line
Ctrl+F (or right arrow)	Moves forward one character
Ctrl+B (or left arrow)	Moves backward one character
Esc+F	Moves forward one word
Esc+B	Moves backward one word
Ctrl+R	Re-displays a line (starts a new line, with the same command shown)
Ctrl+U	Erases a line
Ctrl+W	Erases a word
Ctrl+Z	Exits configuration mode, returning you to privileged EXEC mode

I have highlighted the Ctrl commands used most often when working with the IOS. It may seem silly to have Ctrl commands that perform the same functions as the arrow keys on the keyboard; these Ctrl commands exist because some older terminal programs do not allow you to use the arrow keys.

Cisco IOS Optimization

While the base operating environment works well enough to allow you to get the job done, there are a few methods you can use to optimize your terminal environment and experience.

- **terminal history size** <***0-256***> - This privileged mode command allows you to set the number of entered commands the IOS remembers. By default, the IOS remembers the last 10 commands you have entered and can recall them using the up arrow or **ctrl+p** commands.

  Example:

  ```
  RouterA#terminal history size 100
  ```

- **no ip domain-lookup** - This global configuration mode command prevents the Cisco IOS device from trying to lookup names typed in privileged mode. By default, when you type a word not recognized in the Cisco IOS command set in privileged mode, the IOS device assumes you are trying to Telnet to a device by that name. The command processing will hang for about 30 seconds, trying to map the word you typed to an IP address.

```
RouterA#comfigure
Translating “comfigure”...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find
computer address
```

Typing this command in global configuration mode disables this name-lookup feature.

Example:

```
RouterA#configure terminal
RouterA(config)#no ip domain-lookup
```

- **logging synchronous -** This line configuration mode command prevents console messages from interrupting your command input. When working on an IOS device through the console port, status messages will often display and interrupt your command entry.

Example:

```
RouterA#show run
090777: *Jul 24 17:44:41: %CDP-4-DUPLEX_MISMATCH: duplex mismatch
discovered on FastEthernet0/23 (not half duplex), with
AccessServer Ethernet0 (half duplex).ning
```

You may have been trying to type the command **show running-config**, but a status message interrupted your typing. While the command will still work, this can be very hard on your mind. Typing the **logging synchronous** command under line configuration mode (which will be discussed much more in the upcoming material) causes the IOS device to repaint the text you were typing on a new line.

Example:

```
RouterA#configure terminal
RouterA(config)#line console 0
RouterA(config-line)#logging synchronous
RouterA(config-line)#end
RouterA#show run
090782: *Jul 24 17:48:13: %SYS-5-CONFIG_I: Configured from console
by vty0 (172.30.3.105)
RouterA#show running-config
```

- **exec-timeout <*minutes*> <*seconds*> -** This line configuration mode command lets you reconfigure the amount of idle time you can spend before the Cisco IOS device automatically logs you out. This can be useful when initially configuring the IOS device; the default timeout period is 10 minutes.

  Example:

  ```
  RouterA#configure terminal
  RouterA(config)#line console 0
  RouterA(config-line)#exec-timeout 60 0 (configures 1 hour idle timeout)
  ```

Domain 4 - Working with Cisco Switches

Initial Switch Configuration

Unlike Cisco routers, Cisco switches will function fresh out of the box with no configuration. However, without configuring the switch, you will have a very expensive device that performs the same function as a much cheaper, unmanaged switch. The initial configuration of a Cisco switch consists of the following goals:

- Initial boot sequence
- Assigning passwords
- Configuring a hostname and logon banner
- Enabling Secure Shell (SSH)
- Configuring port security (if necessary)
- Optimizing switch ports
- Assigning a Switch IP address
- Verifying and Saving the configuration

We will work through these one at a time.

Initial Boot Sequence

In order to observe the initial boot sequence of a switch, you must first have a console port connection as discussed in the earlier IOS Foundations section. Once you plug in a Cisco switch, it automatically powers on. We can gather some key information by observing the boot sequence, which I will highlight here. I have also cut out much of the not-so-key information from the boot sequence. My comments through the following boot sequence will be highlighted in blue and begin with an exclamation point (!).

```
Xmodem file system is available.
The password-recovery mechanism is enabled.
Initializing Flash...
flashfs[0]: 3 files, 1 directories
flashfs[0]: 0 orphaned files, 0 orphaned directories
flashfs[0]: Total bytes: 15998976
flashfs[0]: Bytes used: 8834560
```

```
flashfs[0]: Bytes available: 7164416
flashfs[0]: flashfs fsck took 16 seconds.
...done Initializing Flash.
Boot Sector Filesystem (bs:) installed, fsid: 3
Loading "flash:/c3550-ipservicesk9-mz.122-40.SE.bin"...###################
##############################################################...<output
omitted>...

File "flash:/c3550-ipservicesk9-mz.122-40.SE.bin" uncompressed and
installed, entry point: 0x3000
executing...
```

! By default, the Cisco IOS is stored in a compressed .bin format in the flash of the switch. During the boot process, the switch decompresses and copies the entire IOS into RAM. This allows the IOS to operate much faster. During this process, pound symbols (#) will buzz across your terminal window.

```
              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

Cisco IOS Software, C3550 Software (C3550-IPSERVICESK9-M), Version
12.2(40)SE, RELEASE SOFTWARE (fc3)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Fri 24-Aug-07 02:15 by myl
Image text-base: 0x00003000, data-base: 0x011209B0
```

! The text above gives us information about the IOS version and feature set that is running on this switch. This is currently running the IOS version 12.2(40)SE with an IP Services feature set. Cisco sells many different versions of the IOS that have different price points. For example, if you run a Voice over IP (VoIP) network, you would want an IP Voice IOS or Enterprise feature set, which has a higher price point than the IP Base feature set all Cisco devices ship with.

```
Initializing flashfs...
flashfs[1]: 3 files, 1 directories
flashfs[1]: 0 orphaned files, 0 orphaned directories
flashfs[1]: Total bytes: 15998976
flashfs[1]: Bytes used: 8834560
flashfs[1]: Bytes available: 7164416
flashfs[1]: flashfs fsck took 9 seconds.
flashfs[1]: Initialization complete.
```

```
...done Initializing flashfs.
POST: CPU Buffer Tests : Begin
POST: CPU Buffer Tests : End, Status Passed
POST: CPU Interface Tests : Begin
POST: CPU Interface Tests : End, Status Passed
...<output omitted>...
```

! During boot, the switch performs a Power-on Self Test (POST) in which all components are tested. The lights on the front of the switch will initially all be lit and will slowly tick down (turning off) as these tests complete.

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

! If you see the above notice, you know that the switch is able to support encryption features. At this point, we are most interested in these features for Secure Shell (or SSH) support, which we will discuss later. If this notice does not appear, your device will only be able to support Telnet access.

```
Cisco WS-C3550-24-PWR (PowerPC) processor (revision B0) with 65526K/8192K
bytes of memory.
```

! The above line represents the amount of memory (RAM) installed in this switch. It currently has 65526 Kilobytes (KB) or 64 Megabytes (MB) installed. The switch partitions this memory into smaller pieces. The 8192K represents a memory partition

```
Processor board ID CAT0711Y14D
Last reset from warm-reset
Running Layer2/3 Switching Image

24 FastEthernet interfaces
2 Gigabit Ethernet interfaces

The password-recovery mechanism is enabled.
384K bytes of flash-simulated NVRAM.
Base ethernet MAC Address: 00:0C:85:4B:EE:80
Motherboard assembly number: 73-8100-06
Power supply part number: 341-0029-01
Motherboard serial number: CAT071105SN
Power supply serial number: DTH0710060E
```

```
Model revision number: B0
Motherboard revision number: A0
Model number: WS-C3550-24PWR-SMI
System serial number: CAT0711Y14D
```

! The information above gives key serial and model number information about the switch which would be needed if you ever called Cisco for technical support or warranty service. This is a Cisco Catalyst 3550 switch (C3550) with 24 ports supporting inline power (24PWR).

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

Would you like to enter the initial configuration dialog? [yes/no]:
```

! The boot process is now complete and the switch is functional. The question above is only asked if the switch has no existing configuration. The Initial Configuration Dialog is a step-by-step process to configure base functionality on the switch. In the real world, nobody uses this step-by-step process, so I would highly suggest answering NO to the above question, as I will do here:

```
Would you like to enter the initial configuration dialog? [yes/no]: n
Would you like to terminate autoinstall? [yes]: y
Press RETURN to get started!
Switch>
```

! We are now in user mode of an unconfigured switch.

Assigning Passwords

When working with Cisco IOS devices, you'll need to get used to the idea of setting multiple levels of passwords. There are two modes to protect: User mode (initial access to the device) and Privileged mode (full administrative access to the device). We'll start by protecting the User mode.

On a Cisco switch, there are only two ways to reach User mode: through the console port and through Telnet/SSH remote access sessions. Use the following syntax to secure both of these portals:

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#line console 0
Switch(config-line)#password cisco
Switch(config-line)#login
Switch(config-line)#exit
Switch(config)#line vty 0 4
Switch(config-line)#password cisco
Switch(config-line)#login
Switch(config-line)#end
Switch#
```

Let’s discuss the key commands:

- **line console 0** - Moves from global configuration mode into line configuration for the console port.
- **line vty 0 4** - Moves from global configuration mode into line configuration for the Virtual Terminal (VTY) lines. These VTY lines receive Telnet and SSH connections. Most Cisco devices allow up to five simultaneous Telnet/SSH connections. Typing **line vty 0 4** configures all five of these ports at the same time.
- **password <*password*> -** Sets the password for the console or VTY lines. In our example, you must now type the password “cisco” to get into user mode from the console or VTY lines.
- **login -** Requires logins to the port. If you enter the **password** command without entering the **login** command, the user will never be prompted for the password, even though you have one set, since logins are not required. Note: the **login** command exists by default under VTY lines but must be entered under the console line.

After performing the previous configuration, the User mode is now protected on your Cisco switch. We now need to protect the transition from User mode to Privileged mode (accomplished by typing the **enable** command). You can use one of two commands to accomplish this:

```
Switch(config)#enable secret cisco
Switch(config)#enable password cisco123
```

Both of these commands accomplish the same objective: requiring a password when a user attempts to move from User to Privileged mode. The difference between them is one of these commands is stored in clear text in the running configuration while the other is encrypted:

```
Switch#show run
Building configuration...

Current configuration : 2258 bytes
!
version 12.2
...<output omitted>...
!
enable secret 5 $1$54DQ$snSsKeDfmgAHUoxp9/s7q0
enable password cisco123
```

Notice the **enable secret** command has been completely encrypted and is not visible on the device. The **enable password** command is displayed in clear text. The **enable password** <***password***> command is the older of the two and is not used unless you are working with older Cisco equipment. If both commands are entered on a Cisco device, the **enable password** is automatically disabled (only the **enable secret** password will function).

Our switch is now password protected.

Configuring a Hostname and Logon Banner

Every Cisco IOS device has a hostname that is used to uniquely identify it from other devices in the network. To assign a hostname, use the following syntax:

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname Fido
Fido(config)#
```

Notice that the prompt immediately changes to Fido. Since I'd rather not have Fido as my hostname for the rest of this guide, I'll remove the hostname using the **no** command

```
Fido(config)#no hostname Fido
Switch(config)#
```

The **no** command can be used to remove any configuration in the Cisco IOS. For example, if I wanted to remove the enable password I previously entered, I could type **no enable password**.

For legal purposes, it's always good to have a logon banner on all Cisco devices. To configure a logon banner (known as a Message of the Day or MOTD banner in the Unix/Cisco realm), use the following syntax:

```
Switch(config)#banner motd &
Enter TEXT message.  End with the character '&'.
**************************************************************
This is a private system. Unauthorized access prohibited.
**************************************************************
&
Switch(config)#
```

The **banner motd <*delimiter*>** command is fairly straightforward. Keep in mind that the delimiter character can be any character you wish. It simply marks the start and end of your logon banner; in my example, I chose to use the ampersand (&) since I did not plan to use that character anywhere in the logon banner.

Enabling Secure Shell (SSH)

By default, every Cisco device supports Telnet access. Telnet has been around since the foundation days of networks when security was not much of an issue. Today, most networks prefer to use SSH since everything sent using the Telnet protocol is sent in clear-text. SSH performs strong encryption on all the data sent or received. Use the following syntax to enable your switch to support SSH:

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#username Jeremy password cisco
Switch(config)#ip domain-name preplogic.com
Switch(config)#crypto key generate rsa
The name for the keys will be: Switch.preplogic.com
```

```
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

Switch(config)#
01:06:57: %SSH-5-ENABLED: SSH 1.99 has been enabled
Switch(config)#line vty 0 4
Switch(config-line)#login local
Switch(config-line)#transport input ssh telnet
```

Let's discuss the key commands used to enable SSH:

- **username <*username*> password <*password*> -** While Telnet requires just a password, SSH connections require both a user account and password. Using this syntax creates a user account on your Cisco switch for SSH access.
- **ip domain-name <*domain*> -** In order to perform SSH, you must use encryption keys (formulas). The switch will generate an encryption key with a specific name; this domain-name is tied to that name. If you glance at the syntax above, you'll see a line that says "The name for the keys will be: Switch.preplogic.com." This is simply an identifier for this encryption key set. You cannot generate encryption keys without first setting a domain name.
- **crypto key generate rsa -** This command actually generates the encryption keys. You'll notice that you are required to enter the size of the "modulus" after entering this command. This is the strength of the encryption formula. The larger the modulus, the stronger the encryption. However, the larger your modulus, the more resources the switch will use to perform the encryption.
- **login local -** Entering this command under the VTY lines instructs them to use the local user database (where we created our user account) to authenticate incoming connections, rather than the simple password we typed under the VTY lines previously.
- **transport input <*protocol(s)*> -** This command instructs the switch as to the specific protocols allowed to access the VTY lines. In the previous syntax example, I chose to allow SSH and Telnet connections. If you only wanted to allow SSH (thereby disallowing Telnet), you could enter **transport input ssh**.

Configuring Port Security (if necessary)

Since switches connect to the end-users of an organization, administrators will often want to secure the switch ports to specific requirements. For example, you might want to restrict your users from installing another switch or hub at their desk and running multiple devices. To do this, you can restrict the switch port to a maximum of one MAC address. Here is a syntax example of a basic switch port security configuration:

```
Switch#conf t
Switch(config)#interface fastethernet 0/10
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security maximum 1
Switch(config-if)#switchport port-security mac-address sticky
Switch(config-if)#switchport port-security violation shutdown
```

Here are the key commands to enable port security:

- **switchport mode access** - Much of the function of this command will be discussed when we talk about VLANs. For now, think of it simply as setting the switch port as an access port which connects to end devices rather than another switch.
- **switchport port-security -** Turns on the port security feature for this port.
- **switchport port-security maximum <*number*> -** Sets the maximum number of allowed MAC addresses on this port. In the previous example, I allowed a maximum of one MAC address.
- **switchport port-security mac-address sticky -** This command is optional. What it does is hardcode the first MAC address it learns (since we limited the maximum MAC addresses to one) as the *only* allowed MAC address on the port. This is useful if you would like to allow only a specific device to access a switch port rather than multiple devices.
- **switchport port-security violation shutdown -** Instructs the switch to shutdown the port if any of the previous criteria are violated.

Optimizing Switch Ports

Switches support a variety of clients connecting with different speed and duplex settings. Because of this, every port on a Cisco switch is set to auto-detect speed and duplex settings. Unfortunately, because of the variety of different network cards, speeds and features that have been created over the years, the auto-detect mechanism occasionally fails. This is a problem on all vendor switches, not just Cisco. If the detection mechanism fails, you might end up with a duplex-mismatch where one side of the connection (the switch or PC) will detect the wrong type of setting. This can cause extremely poor performance or complete port disabling for some of the clients. Because of this, Cisco highly recommends that you hard-code speed and duplex on key connections in your network. Use the following syntax to accomplish this:

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface fastEthernet 0/10
Switch(config-if)#speed 100
Switch(config-if)#duplex full
```

This syntax example sets the port to 100Mbps, full-duplex. You can choose any setting the switch supports, such as 10Mbps, half-duplex or 1000Mbps, full-duplex. Just make sure you hard-code the other side of the connection with the same settings!

You can also add descriptions to key switch ports in your environment by using the **description** command:

```
Switch(config-if)# description Connection to RouterA
```

Finally, as a security practice, Cisco recommends that you shut down any switch port that is not currently in use on your switch. You can accomplish this by accessing the interface(s) you would like to shut down and simply type the shutdown command:

```
Switch(config)#interface fastEthernet 0/10
Switch(config-if)#shutdown
```

Assigning a Switch IP Address

Most of the time, Cisco switches are configured and then locked up in an IT room or wiring closet. It would be much better and faster for you to be able to Telnet or SSH to the switch, rather than walking to that location each time you need to make a configuration change. For this reason, we need to assign the switch an IP address and default gateway. Here is a syntax example:

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface vlan 1
Switch(config-if)#ip address 192.168.10.1 255.255.255.0
Switch(config-if)#no shutdown
Switch(config-if)#exit
Switch(config)#ip default-gateway 192.168.10.1
```

The VLAN 1 interface is considered a virtual interface that is reachable by all ports assigned to VLAN 1 (which all ports are, by default). This concept will be discussed much more when we get into VLANs later in this guide. The **no shutdown** command turns the VLAN interface since it has a shutdown state by default.

Verifying and Saving the Configuration

Finally, let's discuss some commands to verify the configuration of our switch. Nearly all of the verification commands begin with the command **show** and are executed from privileged mode.

Let's start off with the biggest show command of all: **show running-config**

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

Current configuration : 2712 bytes
!
version 12.2
no service pad
```

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Switch
!
no logging console
enable secret 5 $1$54DQ$snSsKeDfmgAHUoxp9/s7q0
enable password cisco123
!
username Jeremy password 0 cisco
no aaa new-model
ip subnet-zero
ip domain-name preplogic.com
!
interface FastEthernet0/1
 switchport mode dynamic desirable
!
interface FastEthernet0/2
 switchport mode dynamic desirable
!
interface FastEthernet0/3
 switchport mode dynamic desirable
!
interface FastEthernet0/4
 switchport mode dynamic desirable
!
interface FastEthernet0/5
 switchport mode dynamic desirable
!
interface FastEthernet0/6
 switchport mode dynamic desirable
!
interface FastEthernet0/7
 switchport mode dynamic desirable
!
interface FastEthernet0/8
 switchport mode dynamic desirable
!
interface FastEthernet0/9
 switchport mode dynamic desirable
```

```
!
interface FastEthernet0/10
 switchport mode access
 switchport port-security
 switchport port-security mac-address sticky
 shutdown
 speed 100
 duplex full
!
!<output omitted>
!
interface Vlan1
 ip address 192.168.10.1 255.255.255.0
!
ip default-gateway 192.168.10.1
!
banner motd ^C
*************************************************************
This is a private system. Unauthorized access prohibited.
*************************************************************
^C
!
line con 0
 password cisco
 login
line vty 0 4
 password cisco
 login local
 transport input telnet ssh
line vty 5 15
 login
!
end
```

This show command shows every single command you've ever typed into the switch. If you want to create a backup of your configuration, copying the output of the **show running-config** command to a text document is a great method.

The second show command of interest is the **show interface** command. This command can be used to verify the statistics of any interface of your switch:

```
Switch#show interfaces fastEthernet 0/12
FastEthernet0/12 is up, line protocol is up (connected)
  Hardware is Fast Ethernet, address is 000c.854b.ee8c
  (bia 000c.854b.ee8c)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, media type is 10/100BaseTX
  input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:06, output hang never
  Last clearing of “show interface” counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 1000 bits/sec, 2 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     32266 packets input, 2310384 bytes, 0 no buffer
     Received 31244 broadcasts (0 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 31244 multicast, 0 pause input
     0 input packets with dribble condition detected
     12489 packets output, 1002817 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

I have highlighted some key information from this output. The first highlighted line shows the status of the interface. "FastEthernet 0/12 is **up**" means that it is physically connected (the interface is physically intact). The "line protocol is **up**" means that the data link connectivity is operational. In this case, the data link connectivity represents Ethernet function. Simply put, that first line says, "I'm physically okay and communicating on this Ethernet network." Understanding how to interpret that first line is a big one for troubleshooting. Here are all the possibilities:

Interface is…	**Line Protocol is…**	**Cause**
Up	Up	Interface is connected and communicating
Up	Down	Interface is physically intact but not communicating on the network (more common on serial connections)
Down	Down	Interface is not physically intact and is not communicating on the network
Administratively Down	Down	Interface has been shutdown by an administrator (by using the **shutdown** command under an interface0

The second piece of information we can see from the **show interface** command is the speed and duplex settings. If the interface has auto-detected the speed and duplex, this allows you to see the settings it negotiated.

Lastly, the **show interface** command gives us plenty of statistics about the interface, including the amount of data sent and received and traffic loads over the last five minutes. This can be very helpful in determining the overall usage of an interface and can be used to detect communication problems.

Now we come to my all-time favorite command on both Cisco switches and routers: **show ip interface brief**.

```
Switch#show ip interface brief
Interface           IP-Address    OK? Method Status               Protocol
Vlan1               192.168.10.1  YES manual up                      up
FastEthernet0/1     unassigned    YES unset  up                      up
FastEthernet0/2     unassigned    YES unset  down                    down
FastEthernet0/3     unassigned    YES unset  down                    down
FastEthernet0/4     unassigned    YES unset  down                    down
FastEthernet0/5     unassigned    YES unset  down                    down
FastEthernet0/6     unassigned    YES unset  down                    down
FastEthernet0/7     unassigned    YES unset  down                    down
FastEthernet0/8     unassigned    YES unset  down                    down
FastEthernet0/9     unassigned    YES unset  down                    down
FastEthernet0/10    unassigned    YES unset  administratively down  down
FastEthernet0/11    unassigned    YES unset  up                      up
FastEthernet0/12    unassigned    YES unset  up                      up
<output omitted>
```

This command gives a "quick view" look at all of the interfaces on the switch along with their associated IP address (if relevant), their Status (Physical layer) and Protocol (Data Link layer). This command is fantastic to get a fast look at switch interface information.

Last, but not least, we need to look at saving the configuration. Every change that we've made to the switch has been stored in the running-config, which is located in RAM. If the switch were to lose power, all of the configurations that we've entered would be lost. Let's first talk about the "CCNA approved" method of saving your configuration, and then I'll show you the shortcut. To save your configuration on a Cisco switch, enter the following:

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

Performing this configuration copies the data from RAM (running-config) to Non-Volatile RAM (NVRAM, or startup-config). The common shortcut used to enter this command is **copy run start**.

The faster method of saving your configuration is to use the **write** command. Just entering **write** (or the **wr** shortcut) from any privileged mode prompt automatically copies the running-configuration to the startup-configuration with no questions asked. While this method is used quite often in Cisco environments, it may or may not work in your CCNA exam simulations.

Domain 5 - Understanding Wireless LANs

Network Connectivity Without Wires

Wireless LANs have redefined the availability of networks and the types of devices that can connect to networks in a short amount of time. In order to properly deploy a wireless LAN (WLAN), you must first understand the following concepts, which we will be discussing in this next section:

- WLAN communication characteristics
- Understanding radio frequency (RF)
- Understanding WLAN standards
- Understanding WLAN security standards
- WLAN implementation best practices

WLAN Communication Characteristics

The advantages of network connectivity without wires have outweighed the disadvantages of wireless network communication. Wireless LANs are not as secure, reliable or as high-speed as wired LAN communication, and yet, it is rare to find a corporate network that does not have some type of wireless network connection available for its employees.

The following characteristics are important to understand about WLAN communication:

- **WLANs are half-duplex**: Currently, full-duplex communication is not yet supported in the wireless arena. A wireless device will be able to send *or* receive at once but can never do both.
- **WLANs use CSMA/CA rather than CSMA/CD**: Since wireless networks do not have the ability to detect a collision, WLANs use a collision avoidance (CA) algorithm that assigns specific timeslots to each client attached to the WLAN. This is important to realize since the more clients you attach to a wireless access point, the slower all devices become.
- **WLANs use radio frequency to communicate**: WLAN network connections are far more susceptible to interference than wired LAN connections. There are also varying regulations in different countries that you will need to consider before deploying a WLAN.

Understanding Radio Frequency (RF)

Because WLANs use RF to communicate, you must have a basic understanding of RF to competently design a wireless network. When setting up a WLAN, a wireless access point (WAP) broadcasts the network RF signal into the air, and clients use this signal to communicate. Clients "tune in" to the same RF to receive and send data.

This RF signal can be absorbed or completely reflected as it strikes different objects. For example, as the RF signal passes through a drywall barrier, the signal will be absorbed and be weaker on the other side. While this is expected, passing through too many absorbing barriers can cause the signal to become too weak to be useful to a network client. There are other materials, such as metal, that can completely reflect the RF signal causing it to be unavailable to a client. For example, you may have a wireless network client in a room with many metal file cabinets. As the client moves around the room, they may completely lose the network RF signal and drop off the network.

Anytime you have a device that broadcasts an RF signal, you must have that device registered with a government entity that manages RF signals. In the United States, this entity is known as the Federal Communications Commission (FCC). Thankfully, the FCC has created three bands of RF that are considered "unlicensed." This means that devices using these bands do not need to be registered. Without unlicensed bands, every cordless phone, microwave oven and WLAN access point would have to go through extensive litigation before it could be used. The three unlicensed bands are:

- **900 MHz**: 902 MHz to 928 MHz
- **2.4 GHz**: 2.400 GHz to 2.483 GHz
- **5 GHz**: 5.150 GHz to 5.350 GHz and 5.725 GHz to 5.825 GHz

Keep in mind that the above RF bands are country specific. While most countries use the same unlicensed bands, it's always best to check with the specific country's RF regulating entity.

When it comes to RF, the higher frequency signals can handle more bandwidth but travel less distance. For example, if you had an RF device using the 900 MHz band, it could transmit further, but handle less traffic than the 2.4 and 5 GHz bands. Because of the low bandwidth amount offered by the 900 MHz band, modern WLANs only use the 2.4 GHz and 5 GHz bands, with the 2.4 GHz band being far more saturated with devices than the 5 GHz band.

Understanding WLAN Standards

Since WLANs represent network standards and RF usage, there are multiple standards organizations that manage WLAN development:

- **International Telecommunication Union - Radiocommunication Sector (ITU-R)**: This organization handles the regulation of the RF aspects of WLAN communication.
- **Institute of Electrical and Electronic Engineers (IEEE)**: This organization handles the development of the 802.11 wireless standards.
- **Wi-Fi Alliance**: This organization certifies WLAN equipment to ensure interoperability between vendors.

There have been three major network standards that have been released since the original wireless network implementations in the late 1990s. The following table gives the "fast facts" on these standards:

	802.11b	**802.11g**	**802.11a**
Frequency	2.4 GHz	2.4 GHz	5 GHz
Non-Overlapping Channels	3	3	12
Maximum Data Rate	11 Mbps	54 Mbps	54 Mbps

At this point, we have only discussed the RF band aspect of these standards. The maximum data rate dictates the maximum speed that each of these standards is able to reach. Keep in mind, this represents the maximum. As the signal becomes weaker (the client moves further away from the WAP), the data rate will decrease. The number of non-overlapping channels represents the number of non-interfering access points you can have in close proximity to each other. The channels will be further discussed in the WLAN best practices section.

Understanding WLAN Security Standards

WLANs have opened corporate networks to a massive security threat. Previously, you would have a physical barrier between your network and potential intruders: if you cannot plug into the network, you cannot access the network. Now, with WLANs, intruders can join the corporate network from a laptop in their vehicle parked just outside the building. These network intruders have a variety of motivations. One may be harmlessly joining the WLAN to get quick access to the Internet, while another may be maliciously probing the network to steal information. Regardless of the motivation, your job is to absolutely prevent unauthorized individuals from joining the WLAN.

A variety of security standards have been released to assist in WLAN network security challenges.

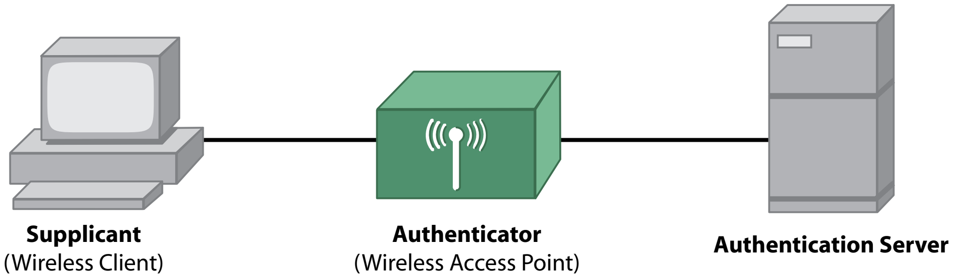
Wired Equivalent Privacy (WEP)

WEP was the initial attempt at wireless security standards. It represented an authentication (verifying identity) and encryption system to secure data sent over open RF airwaves. The security was achieved by entering a static, shared key into both the client and the access point. The static key would be used to generate an encryption algorithm during communication. Unfortunately, WEP was found to have many serious weaknesses requiring improved security methods.

802.1x EAP

The Extensible Authentication Protocol (EAP) was developed in part to address wireless security vulnerabilities. EAP represents a framework of authentication protocols that can be changed to meet specific requirements. EAP is not a protocol that directly secures your network but, rather, a framework that offers a "shell" to plug in your own authentication protocol. For example, Cisco developed LEAP, which was a method that allowed you to have dynamic WEP keys, in response to the weaknesses found in the original WEP standard. With LEAP, you do not enter a static WEP key but, instead, are allowed username/password credentials to securely pass to an authentication server which will generate a new, dynamic WEP key each time the user joins the network. LEAP is just one form of EAP; there are many others.

802.1x is the implementation of EAP on a wired or wireless network. The typical 802.1x architecture is as follows:



With 802.1x, the supplicant passes its authentication credentials to the WAP (the authenticator); however, the WAP does not actually do the authentication. It passes the credentials to the authentication server, which checks the credentials (such as a username and password) against its database. If the credentials match, the server will tell the WAP to generate encryption keys and allow the client access. If the credentials do not match, the client will not be able to access the wireless network.

Using this 802.1x method is fantastic since you can change the method or strength authentication without needing to upgrade your authenticator equipment.

Wi-Fi Protected Access (WPA)

WPA was introduced in response to the many weaknesses found in WEP. This protocol was designed as an interim solution to WEP while a more secure system was developed. The beauty of WPA was the fact that it could use the same wireless hardware as WEP (no need to upgrade all of your preexisting wireless equipment), but it eliminated many of the security vulnerabilities in the network.

Wi-Fi Protected Access 2 (WPA2) or 802.11i

WPA2 is the current final security standard for wireless security. It uses a completely new encryption algorithm, Advanced Encryption Standard (AES), which is far more secure than WPA or WEP ever was. It integrates seamlessly with dynamic key management systems, eliminating the need to enter static keys on all devices.

WLAN Implementation Best Practices

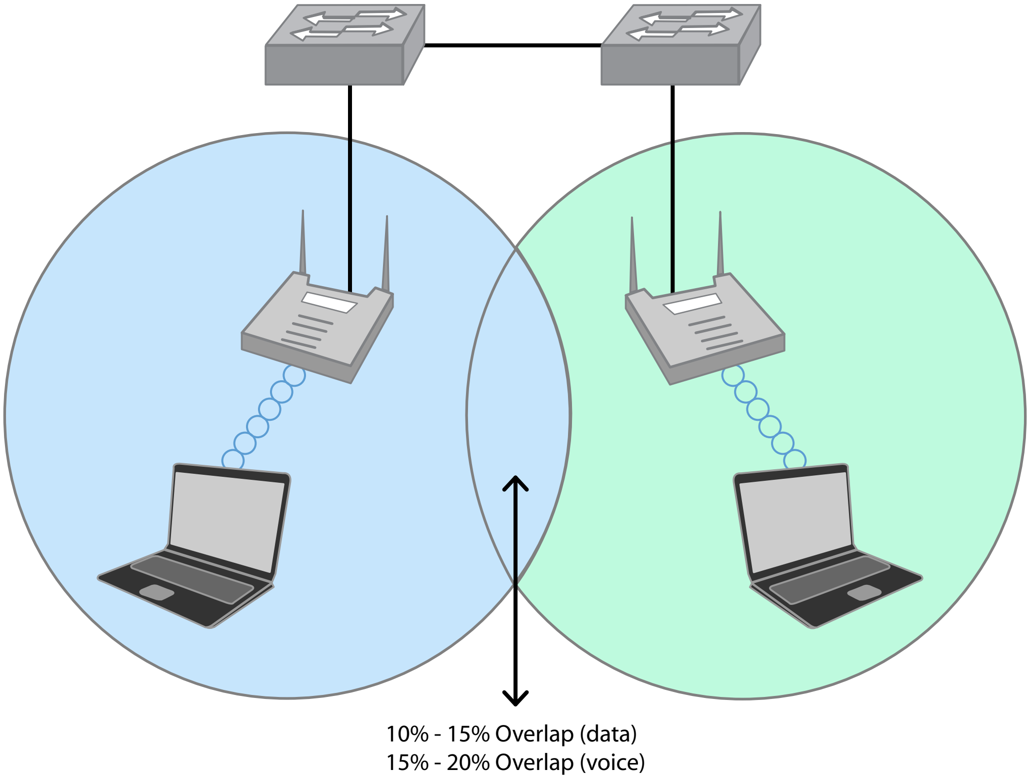
While the CCNA exams do not cover syntax or configuration of Cisco WAPs, Cisco does expect you to have a good idea of wireless design and implantation methods.

When designing a WLAN, there are three methods of WLAN connections that you can use:

- **Independent Basic Service Set (BSS) Ad-Hoc Connections**: WLAN connections between two or more end devices without requiring the use of a wireless access point.
- **Basic Service Set (BSS) Infrastructure Connections**: WLAN connections between one or more clients and a single wireless access point.
- **Extended Service Set (ESS) Infrastructure Connections**: WLAN connections involving two or more BSS systems connected to a common switched network.

Since corporations cover a larger service area, most will use an ESS design.

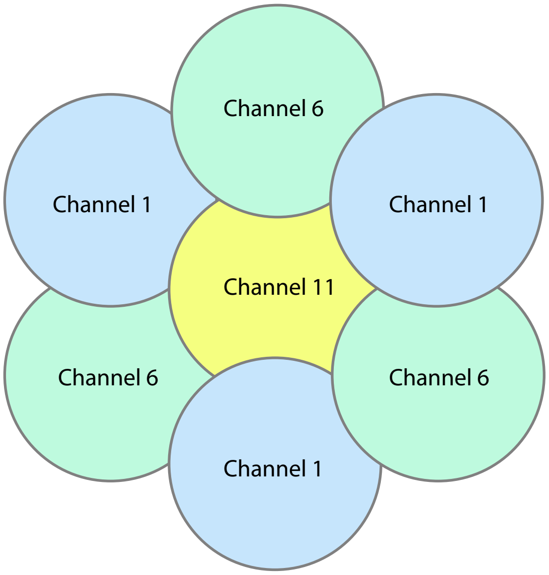
When designing an ESS infrastructure, you should always provide an overlap of wireless coverage cells to allow clients to roam between access points without losing a signal. As shown in the following figure, the cells should have an overlap of 10 to 15 percent for data-only and an overlap of 15-20 percent for VoIP traffic.



When designing overlapping coverage, you must factor in the idea of wireless channels. Wireless access points that offer overlapping signals will interfere with each other if assigned to the same channel. Changing the wireless channel of a WAP assigns it to run on different wireless frequencies. 802.11b and 802.11g support 11 different channels that run on the following frequencies in the 2.4 GHz range:

- **Channel 1** - 2.401 - 2.423 GHz
- **Channel 2** - 2.406 - 2.428 GHz
- **Channel 3** - 2.411 - 2.433 GHz
- **Channel 4** - 2.416 - 2.438 GHz
- **Channel 5** - 2.421- 2.443 GHz
- **Channel 6** - 2.426 - 2.448 GHz
- **Channel 7** - 2.431 - 2.453 GHz
- **Channel 8** - 2.436 - 2.458 GHz
- **Channel 9** - 2.441 - 2.463 GHz
- **Channel 10** - 2.446 - 2.468 GHz
- **Channel 11** - 2.451 - 2.473 GHz

As you can see, many of the channels overlap frequency. For example, if you were to place a WAP using channel 1 (2.401 - 2.423 GHz) next to a WAP using channel 3 (2.411 - 2.433 GHz), the signals would interfere with each other. The only "clean" channels to safely use in close proximity are channels 1, 6, and 11. A typical wireless cell design for a mid- to large-sized organization would look like this:



With 12 non-overlapping channels to use, 802.11a, which runs in the 5 GHz range is much more flexible in terms of network design.

The average 802.11b/g WAP can transmit a usable signal about 300 feet in an "open field" (no obstructions) environment. However, as a client receives a weaker signal, the transmission data rate will decrease (lower speed connections can travel further distances). As an 802.11g client travels away from the WAP, it will negotiate down through the following speeds:

- 54 Mbps
- 48 Mbps
- 36 Mbps
- 24 Mbps
- 18 Mbps
- 12 Mbps
- 9 Mbps
- 6 Mbps

If the downgrade of speed presents a performance issue for your clients, you can place your WAPs closer together.

Depending on the type of WAP you use, the configuration can be performed using a web-based or command-line interface. Regardless of the method used, there are some basic parameters you should set on every WAP:

- **IP address** - Used for remote administration of the WAP
- **Wireless standard(s)** - Choose between running 802.11b, 802.11g, or 802.11a. Many WAPs can run all three standards at the same time.
- **Channel assignment** - Select clean channels for adjacent WAPs. Many WAPs support an auto channel assignment feature, which searches for the cleanest channel.
- **Service Set Identifier (SSID)** - Create one or more SSIDs, which identify the wireless network to clients.
- **Security settings** - Choose encryption and authentication methods you will use for your organization.

The WAPs should be configured in a layered approach. First, test the switch port that you plan to use for the WAP. Ensure it operates correctly for an end PC. Then, install the access point and create an SSID without any security parameters. Join the unsecure SSID with a client and make sure everything works before you add security. Then, add the wireless security and test using the same wireless client.

Most of the wireless troubleshooting you will encounter will be due to interference issues. Since many other devices share the same radio frequencies, finding a clean channel may be a challenge in some environments. Many organizations have chosen to use 802.11a equipment because there are far more clean frequencies to use than 802.11b/g.

Domain 6 - Understanding Routing

Moving Into the Routed World

We've now left the Data Link layer realm of switching for a season and moved into the Network layer realm of routing. Routers are the links that tie multiple networks together. Anytime a device needs to leave the local network to venture into other networks (such as the Internet), a router is necessary. In this upcoming material, we will discuss the following topics:

- Understanding Cisco routers
- The functions of a router
- Understanding TCP/IP routed networks and subnetting

Understanding Cisco Routers

The founders of Cisco are most often credited with the creation of the router (although, this is a highly debated topic). Regardless of who initially created the device, the router is now found in nearly every network today. There are four primary lines of modern Cisco routers that you will deal with at the CCENT and CCNA levels:

Cisco 800 Series

Cisco 1800 Series

Cisco 2800 Series

Cisco 3800 Series

These product lines are typically found in corporate networks. While there are other router lines available, they are usually found in higher-end environments such as Internet Service Providers (ISPs).

The beauty of learning Cisco is the fact that all of these routers have the same IOS. Once you learn to work with one of them, you'll have the foundations for working with all of them. The major differences between the routers are the amount of memory, processing power and interfaces each one is able to support. The higher-end product lines (such as the Cisco 3800 series) support many more interfaces and are able to handle a much larger amount of network traffic than the lower-end product lines (such as the 800 series).

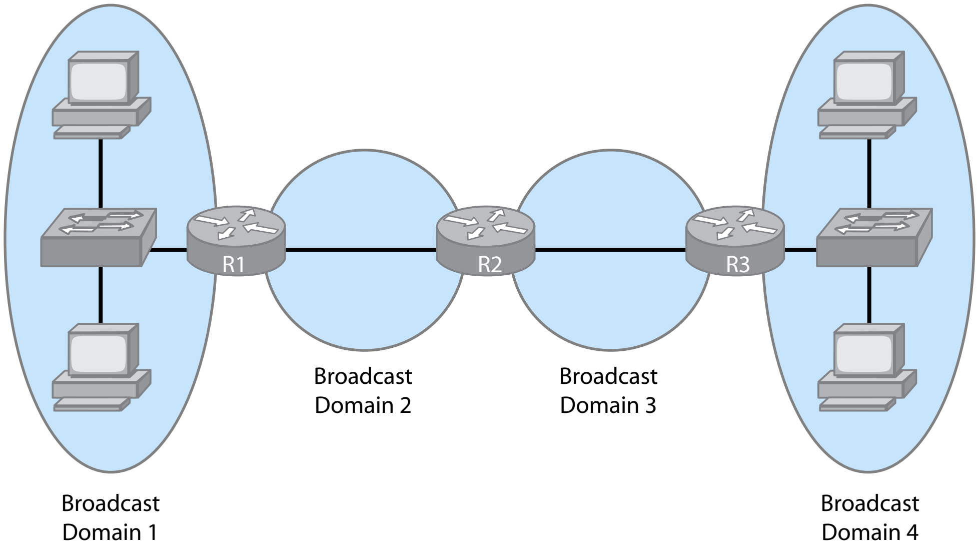
The Functions of a Router

If you were to look at the first item on a router's resume, you would find the following statement:

> My primary goal is to stop broadcast traffic.

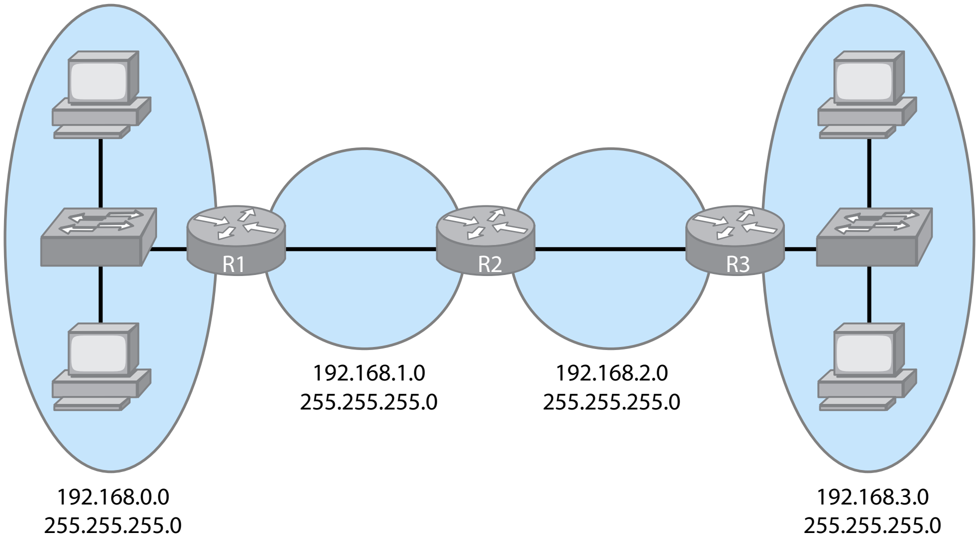
The entire point of breaking up your network using routers is to prevent broadcasts from overwhelming networks all around the world. Routers block and limit broadcasts to the Layer 2 domain.

Effectively, the router creates what is known as a broadcast domain:



The reach of a broadcast domain defines how far a broadcast will go before it is stopped.

While routers stop broadcasts, they do allow unicast (directed, one-to-one communication) between end systems. To accomplish this feat, the router uses a routing table, which lists all the destinations it is able to reach. Let's add IP networks to the previous figure:



If a host from the left side of the network decided it needed to reach a host on the right side of the network, it would send the network traffic to R1, which is its default gateway (for a full discussion of why the host makes this decision, refer back to Domain 2 of this guide). R1 will then look at its routing table, which will look something like this:

R1 Routing Table:
Route 1: Connected - 192.168.0.0/24
Route 2: Connected - 192.168.1.0/24
Route 3: Through R2 - 192.168.2.0/24
Route 4: Through R2 - 192.168.3.0/24

As R1 receives packets for a host in the 192.168.3.0 network, it observes its routing table and realizes the packets need to be sent to R2. Once R2 receives the packets from R1, it looks at its routing table, which will look something like this:

R2 Routing Table:
Route 1: Through R1 - 192.168.0.0/24
Route 2: Connected - 192.168.1.0/24
Route 3: Connected - 192.168.2.0/24
Route 4: Through R3 - 192.168.3.0/24

The process would then continue with R2 sending the packets to R3, who is directly connected to the network where the destination host resides. This is the process that we've all come to know as *routing*. After some base configuration, the routers will know the networks to which they are directly connected. Your job is then to educate these routers about all the networks they are able to reach through other routers. You can do this one of two ways:

- **Static Routing** - A form of routing where you manually enter in each network the router is able to reach and the path to get there. This form of routing is great if you are paid by the hour.
- **Dynamic Routing** - A form of routing where the routers communicate with each other and build the routing tables dynamically. This form of routing is great if you have a salary position.

In most networks, Cisco administrators will use a combination of both styles of routing to accomplish specific goals. So, to review, while routers can accomplish quite a bit, they have two key purposes:

1. To stop broadcast traffic
2. To forward directed data between networks

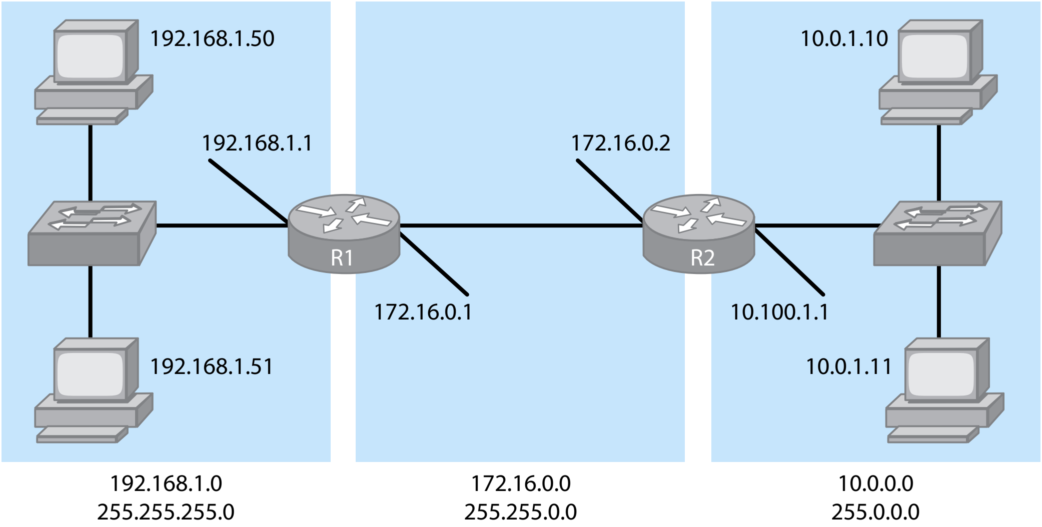
To properly design and support a routed network, you must have a full understanding of how IP networks are created.

Understanding TCP/IP Routed Networks and Subnetting

As discussed in the Domain 2 portion of this guide, the creators of TCP/IP divided the protocol into three major classes of addresses that we can use on our networks today:

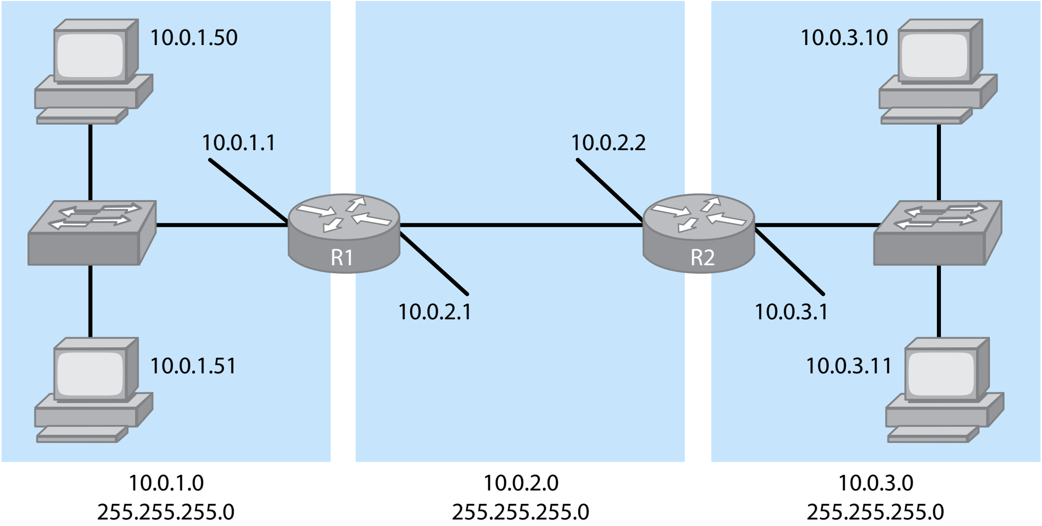
	Subnet Mask	First Octet Value	Number of Hosts Per Network
Class A	255.0.0.0	1-126	16,777,214
Class B	255.255.0.0	128-191	65,534
Class C	255.255.255.0	192-223	254

If we were to use only these default subnet masks in addressing our network, we would be using a **classful** network design. The following figure gives an example of using classful addressing:



Please keep in mind that this network diagram is horrific for many reasons but is primarily used to demonstrate the limitations of classful addressing. If you look at the network on the right, the 10.0.0.0 Class A subnet is in use. This subnet provides more than 16 million addresses and yet, only a few of them are being used. Since the 10.0.0.0 network has been used behind R2, it cannot be used anywhere else in your network.

Today, just about every network in existence uses **classless** addressing. In this form of addressing, the original class of address is only used as a guide. You can take the original subnet mask attached to the address and **subnet** it further down to a more manageable size. For example, I could take the Class A 10.0.0.0 network and apply a Class C subnet mask to it. This basic form of subnetting would provide 65,536 subnets (networks) that I could apply to my organization with 254 hosts per subnet. The following figure gives an example of using this type of classless addressing:



While "easy" subnetting like that shown in the previous example is used most often in real-world corporate environments, all Cisco certification exams expect you to know how to handle difficult subnetting.

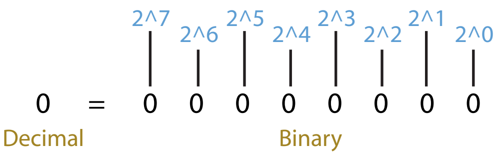
Before we get into subnetting, let me first say that there are probably more methods used to understand and learn subnetting than any other topic in network technology. What I will present in this guide is one form that many have found useful, but if there is another form that you feel more comfortable with, feel free to use it!

The first topic you must master on your way to successful subnetting is converting between decimal and binary. We are used to looking at numbers in decimal form (in **bytes** of information), but network processing looks at numbers in binary form (in **bits** of information). Take, for example, the IP address 216.77.133.249 in its dotted-decimal form (4 bytes) which can be represented in binary form as 11011000.01001101.10000101.11111001 (32 bits).

In order to convert between decimal and binary, you must understand the powers of 2:

2^0 = 1
2^1 = 2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32
2^6 = 64
2^7 = 128

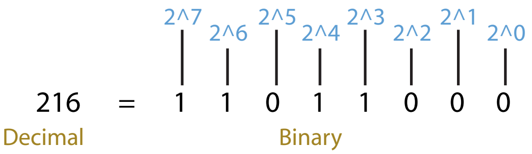
Every byte of information can be broken down into eight bits, where each bit represents a power of two. By flipping a bit from a zero to a one, you enable that power of two:



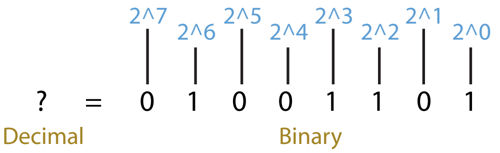
So if you wanted to convert the number 216 to binary, you would use simple subtraction beginning with the largest power of 2:

216 - 127 = 88	(2^7)
88 - 64 = 24	(2^6)
24 - 32 = (can't be done, negative number)	(2^5)
24 - 16 = 8	(2^4)
8 - 8 = 0	(2^3)
0 - 4 = (can't be done, negative number)	(2^2)
0 - 2 = (can't be done, negative number)	(2^1)
0 - 1 = (can't be done, negative number)	(2^0)

Thus, the resulting binary value becomes:



You can use this process to convert any number to binary. You will also want to know how to convert back. For example, if given the binary number 01001101, you should be able to add the respective powers of two back together to get a decimal value:

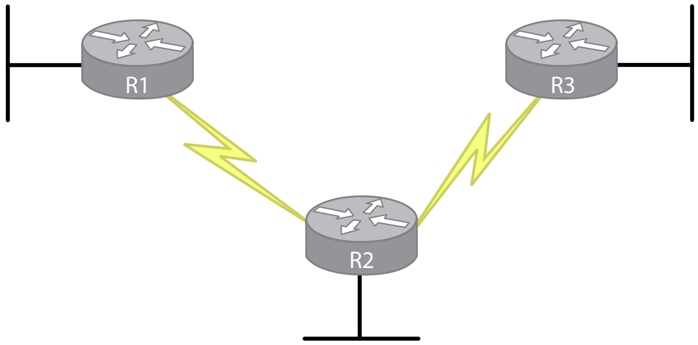


2^6 = 64
2^3 = 8
2^2 = 4
2^0 = 1

64 + 8 + 4 + 1 = 77

Keep in mind that Cisco exams do not allow you to use a calculator! You should practice these decimal-to-binary and binary-to-decimal skills until you can accomplish both conversions rather quickly.

Now that we have the base skill for subnetting mastered, we can move into the process of subnetting itself. The need for, and process of, subnetting is best demonstrated through an example.



Example 1: You are the administrator for the network shown in the prior figure. The organization wishes to use public addressing for all devices in the organization and has been assigned the Class C subnet 200.5.9.0 by their ISP.

The problem with the previous scenario is the fact that the organization has a single, Class C network but has five networks in the organization (3 LANs and 2 WANs). You must break up the single, Class C network into at least five subnetworks. Here's the process:

Step (1) **Determine the number of subnets and convert to binary.**

In this initial step, you must determine the number of networks the organization needs and convert that number to binary. In this case, our organization needed five networks, so:

5 = 00000101

Step (2) **Reserve required bits in a subnet mask and find incremental value.**

In this second step, we need to determine how many bits it takes to reach the number of networks required. Looking at the binary equivalent of five (00000**101**), we can see that it takes three bits (in bold) to get the number five. You cannot get this decimal number with any less than three bits.

Next, to reserve the required bits, we need to look at the original subnet mask in all binary. We were told that this is a Class C subnet mask. If we were to convert the subnet mask to all binary, it would look like this:

255.255.255.0 = 11111111.11111111.11111111.00000000

Up until now, we have been saying that the decimal number "255" represents the network portion of the IP address and the decimal number "0" represents the host. Since we're now working in binary, we need to think the binary bit "1" represents a network bit and the binary bit "0" represents a host bit. The process of subnetting borrows host bits to create more networks (which we need in this case). Since it takes three bits to get the number five, we know that we must add three more network bits (1's) to our subnet mask. The result is as follows:

11111111.11111111.11111111.11100000

Notice that we picked up right where the network "1" bits left off and converted three of the previous host bits into network bits. From this binary number, we can now find what our decimal subnet mask will be for this entire organization. By converting this binary number back to decimal, we get:

11111111.11111111.11111111.11100000 = **255.255.255.224**

Every device in the organization - every router, every switch, every PC - will use this subnet mask in their network configuration. That leads us to the second half of this second step: "find incremental value."

The incremental value is necessary for the third and final step of this subnetting process. The incremental value is the **lowest network bit converted back to a decimal number**. Looking at our subnet mask again:

11111111.11111111.11111111.11(1)00000

We can see that the lowest network bit (which I've put in parenthesis) is **32** as a decimal number.

Step (3) **Use increment to find network ranges.**

The incremental value is used to find the network ranges that we will be using in our organization. All we need to do is add the increment to the original network we were assigned in the same octet as the increment was found. In this case, looking back at the subnet mask:

11111111.11111111.11111111.11100000

(1st octet) (2nd octet) (3rd octet) (4th octet)

We can see that the increment is in the fourth octet. Thus, our math proceeds as follows:

200.5.9.0
200.5.9.32
200.5.9.64
200.5.9.96
200.5.9.128
200.5.9.160
…and so on.

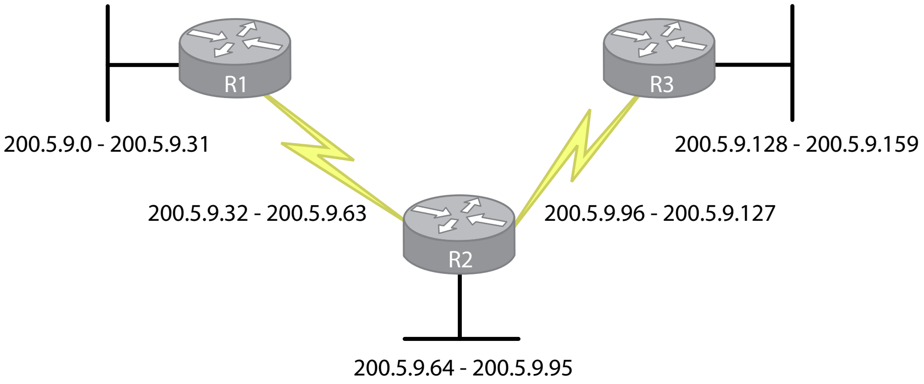
These numbers represent the beginning of each network range. In order to find the complete network range, we need to subtract one from each of the beginning IP addresses to fill in the end of the previous range:

200.5.9.0 - 200.5.9.31
200.5.9.32 - 200.5.9.63
200.5.9.64 - 200.5.9.95
200.5.9.96 - 200.5.9.127
200.5.9.128 - 200.5.9.159
200.5.9.160 - 200.5.9.191
…and so on.

In our original, Class C network of 200.5.9.0 through 200.5.9.255, we cannot use the first IP address, 205.5.9.0 (since it identifies the network) nor could we use the last IP address, 200.5.9.255 (since it is used for sending a broadcast to the entire network). We have now broken our single network into multiple subnetworks. Each one of these subnetworks has a network identifier and a broadcast address, which are unusable. Thus, we could create a table that looks something like this:

Network Range	Network ID	Broadcast ID	Usable Addresses
200.5.9.0 - 200.5.9.31	200.5.9.0	200.5.9.31	200.5.9.1 - 200.5.9.30
200.5.9.32 - 200.5.9.63	200.5.9.32	200.5.9.63	200.5.9.33 - 200.5.9.62
200.5.9.64 - 200.5.9.95	200.5.9.64	200.5.9.95	200.5.9.65 - 200.5.9.94
200.5.9.96 - 200.5.9.127	200.5.9.96	200.5.9.127	200.5.9.97 - 200.5.9.126
200.5.9.128 - 200.5.9.159	200.5.9.128	200.5.9.159	200.5.9.129 - 200.5.9.158
200.5.9.160 - 200.5.9.191	200.5.9.160	200.5.9.191	200.5.9.161 - 200.5.9.190
…and so on			

If we were to assign the address of the network shown in the previous network diagram, it might look like this:



Keep in mind that every device must use the subnet mask 255.255.255.224 if this addressing scheme is going to function.

So, to summarize this example, we have taken a single, Class C network of 200.5.9.0 and broken it into at least five subnets (actually, 8 total subnets) that are able to support 30 hosts per subnet.

Let me show you another example with my explanations shortened:

Example 2: A service provider has given you the Class C network range 209.50.1.0. Your company must break the network into 20 separate subnets.

Step (1) **Determine the number of subnets and convert to binary.**

- In this example, the binary representation of 20 = 00010100.

Step (2) **Reserve required bits in a subnet mask and find incremental value.**

- The binary value of 20 subnets tells us that we need at least 5 network bits to satisfy this requirement (since you cannot get the number 20 with any less than 5 bits - 10100).
- Our original subnet mask is 255.255.255.0 (Class C subnet).
- The full binary representation of the subnet mask is as follows:

  255.255.255.0 = 11111111.1111111.11111111.00000000

- We must "convert" 5 of the client bits (0) to network bits (1) in order to satisfy the requirements:

  New Mask = 11111111.11111111.1111111.11111000

- If we convert the mask back to decimal, we now have the subnet mask that will be used on all the new networks - 255.255.255.248
- Our increment bit is the last possible network bit, converted back to a binary number:

  New Mask = 11111111.11111111.1111111.1111(1)000 - bit with the parenthesis is your increment bit. If you convert this bit to a decimal number, it becomes the number '8'.

Step (3) **Use increment to find network ranges.**

- Start with your given network address, and add your increment to the subnetted octet:

  209.50.1.0
  209.50.1.8
  209.50.1.16
  …etc

- You can now fill in your end ranges, which is the last possible IP address before you start the next range

  209.50.1.0 - 209.50.1.7
  209.50.1.8 - 209.50.1.15
  209.50.1.16 - 209.50.1.23
  …etc

- You can then assign these ranges to your networks! *Remember the first and last address from each range (network / broadcast IP) are unusable.*

Recently, the CCENT and CCNA exams began testing subnetting skills using Class A and Class B examples. Let me walk you through a Class B example, which is nearly identical to Class A.

Example 3: Your company would like to break the Class B private IP address range 172.16.0.0 into 60 different subnets

Step (1) **Determine the number of subnets and convert to binary.**

- In this example, the binary representation of 60 = 00111100

Step (2) **Reserve required bits in a subnet mask and find incremental value.**

- The binary value of 60 subnets tells us that we need at least 6 network bits to satisfy this requirement (since you cannot get the number 60 with any less than 6 bits - 111100).
- Our original subnet mask is 255.255.0.0 (Class B subnet).
- The full binary representation of the subnet mask is as follows:

  255.255.0.0 = 11111111.11111111.00000000.00000000

- We must "convert" 6 of the client bits (0) to network bits (1) in order to satisfy the requirements:

  New Mask = 11111111.11111111.11111100.00000000

- If we convert the mask back to decimal, we now have the subnet mask that will be used on all the new networks - 255.255.252.0
- Our increment bit is the last possible network bit, converted back to a binary number:

  New Mask = 11111111.11111111.11111(1)00.00000000 - bit with the parenthesis is your increment bit. If you convert this bit to a decimal number, it becomes the number '4'.

Step (3) **Use increment to find network ranges.**

- Start with your given network address, and add your increment to the subnetted octet:

  172.16.0.0
  172.16.4.0
  172.16.8.0
  …etc

- You can now fill in your end ranges, which is the last possible IP address before you start the next range

  172.16.0.0 - 172.16.3.255
  172.16.4.0 - 172.16.7.255
  172.16.8.0 - 172.16.11.255
  …etc

- You can then assign these ranges to your networks! Remember the first and last address from each range (network / broadcast IP) are unusable.

These subnetting examples are all you need to solve a network scenario when a specific number of subnets are required. However, there are times when you will need to subnet and are given a specific number of hosts. For example, you might need to break the Class C 209.50.1.0 network into smaller subnets, but the subnets should accommodate at least 50 hosts per network. Let's work through an example using this approach.

Example 4: A service provider has given you the Class C network range 209.50.1.0. Your company must break the network into as many subnets as possible, as long as there are at least 50 clients per network.

Step (1) **Determine the number of clients and convert to binary.**

- In this example, the binary representation of 50 = 00110010

Step (2) **Reserve required bits in a subnet mask and find incremental value.**

- The binary value of 50 clients tells us that we need at least 6 client bits to satisfy this requirement (since you cannot get the number 50 with any less than 6 bits - 110010).
- Our original subnet mask is 255.255.255.0 (Class C subnet).
- The full binary representation of the subnet mask is as follows:

255.255.255.0 = 11111111.11111111.11111111.00000000

We must ensure 6 of the client bits remain client bits (0) in order to satisfy the requirements. When reserving client bits, always reserve from right-to-left (as opposed to reserving from left-to-right, as we do with network bits). All other bits can become network bits:

New Mask = 11111111.11111111.11111111.11 000000 note the 6 client bits that we have saved

- If we convert the mask back to decimal, we now have the subnet mask that will be used on all the new networks - 255.255.255.192
- Our increment bit is the last possible network bit, converted back to a binary number:

New Mask = 11111111.11111111.11111111.1(1)000000 - bit with the parenthesis is your increment bit. If you convert this bit to a decimal number, it becomes the number '64'.

Step (3) **Use increment to find network ranges.**

- Start with your given network address, and add your increment to the subnetted octet:

209.50.1.0
209.50.1.64
209.50.1.128
209.50.1.192

- You can now fill in your end ranges, which is the last possible IP address before you start the next range

  209.50.1.0 - 209.50.1.63
  209.50.1.64 - 209.50.1.127
  209.50.1.128 - 209.50.1.191
  209.50.1.192 - 209.50.1.255

- You can then assign these ranges to your networks! *Remember the first and last address from each range (network / broadcast IP) are unusable.*

Now, let's take a look at an example of subnetting based on a specific number of hosts with a Class B address.

Example 5: Your company would like to break the Class B private IP address range 172.16.0.0 into as many subnets as possible, provided that they can get at least 300 clients per subnet.

Step (1) **Determine the number of clients and convert to binary.**

- Remember, the binary representations of 8 bits (a single octet of an IP address) can only reach 255, but that does not mean we cannot cross octet boundaries when working with Class A or B examples!
- In this example, the binary representation of 300 = 100101100.

Step (2) **Reserve required bits in a subnet mask and find incremental value.**

- The binary value of 300 clients tells us that we need at least 9 client bits to satisfy this requirement (since you cannot get the number 300 with any less than 9 bits - 100101100).
- Our original subnet mask is 255.255.0.0 (Class B subnet).
- The full binary representation of the subnet mask is as follows:

  255.255.0.0 = 11111111.11111111.00000000.00000000

- We must ensure 9 of the client bits (0) remain client bits (save the clients!) in order to satisfy the requirements. All other bits can become network bits:

  New Mask = 11111111.11111111.1111111 0.00000000 note the 9 client bits that we have saved.

- If we convert the mask back to decimal, we now have the subnet mask that will be used on all the new networks - 255.255.254.0
- Our increment bit is the last possible network bit, converted back to a binary number:

  New Mask = 11111111.11111111.111111(1)0.0000000 - bit with the parenthesis is your increment bit. If you convert this bit to a decimal number, it becomes the number '2'.

Step (3) **Use increment to find network ranges.**

- Start with your given network address and add your increment to the subnetted octet:

  172.16.0.0
  172.16.2.0
  172.16.4.0
  etc…

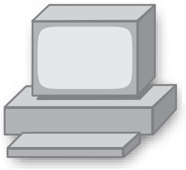
- You can now fill in your end ranges, which is the last possible IP address before you start the next range.

  172.16.0.0 - 172.16.1.255
  172.16.2.0 - 172.16.3.255
  172.16.4.0 - 172.16.5.255
  etc…

- You can then assign these ranges to your networks! Remember the first and last address from each range (network / broadcast IP) are unusable.

Finally, the last style of subnetting requires you to reverse engineer existing subnet problems. For example, you may be troubleshooting and see a specific IP address and subnet mask assignment. You will then need to reverse engineer the problem to find which network range the client, switch or router came from. Let's work through another example.

Example 6: You are troubleshooting the following PC:



IP Address: 192.168.1.58
Subnet Mask: 255.255.255.240

Identify the original range of addresses (the subnet) that this IP address belongs to.

- When reverse engineering a problem, all you need to do is break the subnet mask back into binary and find the increment that was used.

  255.255.255.240 = 11111111.11111111.11111111.11110000

- As before, the last possible network bit is your increment. In this case, the increment is 16.
- Use this increment to find the network ranges until you pass the given IP address:

  192.168.1.0
  192.168.1.16
  192.168.1.32
  192.168.1.48
  192.168.1.64 (passed given IP address 192.168.1.58)

- Now, fill in the end ranges to find the answer to the scenario:

  192.168.1.0 - 192.168.1.15
  192.168.1.16 - 192.168.1.31
  192.168.1.32 - 192.168.1.47
  192.168.1.48 - 192.168.1.63 (*IP address 192.168.1.58 belongs to this range*)

These six examples represent just about every subnetting style that you will encounter on the CCENT and CCNA exams. Before we get into the final method of subnetting, let me add some side information that will help in everything TCP/IP.

1. Subnet masks can be represented in decimal notation or bit notation. Thus far, we have seen them in decimal notation, such as 255.255.255.240. However, you can also write a shorthand version of this subnet mask by using bit notation. Bit notation is simply a forward slash - / followed by the number of network bits (1s) in the subnet mask. For example:

   255.255.255.240 = 11111111.11111111.11111111.11110000

   There are 28 network bits in this subnet mask, so we can write it as a /28.

   Bit notation is usually combined with IP addresses, so writing 192.168.1.0/28 tells you what the network is and what the current subnet mask in use is.

2. There may be times where you are required to know how many hosts can exist on a network when given a certain subnet mask. For example, you may have subnetted a Class B subnet mask (255.255.0.0) to a custom subnet mask of 255.255.254.0. This provides rather large networks (as seen previously in Example 5), but you want to know just how many hosts are allowed on each network. You can find this by using the formula (2 ^ x) - 2, where x represents the number of host bits. For example:

   255.255.254.0 = 11111111.11111111.11111110.00000000

   As you can see, there are 9 host bits (0s) in this subnet mask, so you can use the formula (2^9) - 2 to find that there are 510 valid host IP addresses per network.

3. There may be times where you are required to know how many subnets can exist when given a certain subnet mask. For example, you may have subnetted a Class B subnet mask (255.255.0.0) to a custom subnet mask of 255.255.254.0. You want to know just how many subnets can be created by using this custom subnet mask. You can find this by using the formula (2 ^ x), where x is the number of subnet bits. For example:

   255.255.254.0 = 11111111.11111111.11111110.00000000

   As you can see, there are 7 subnet bits (1s added to the original Class B subnet mask), so you can use the formula (2^7) to find that there are 128 valid subnets.

4. Because the Cisco exam does not allow you to use a calculator, finding large powers of 2 can be time consuming. As a time saving measure, remember that the original values we used for subnetting are indeed powers of two:

    a. 2^0 = 1
    b. 2^1 = 2
    c. 2^2 = 4
    d. 2^3 = 8
    e. 2^4 = 16
    f. 2^5 = 32
    g. 2^6 = 64
    h. 2^7 = 128

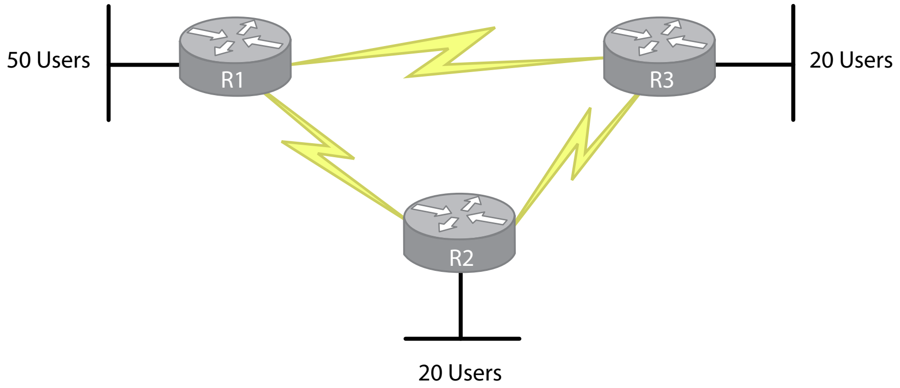
    So, you already know the first 8 powers of two off the top of your head; to find anything larger, just start from 2^7 and keep multiplying by 2 (i.e. 2^8 = 256, 2^9 = 512 and so on).

Now we can move into the final type of subnetting. This type of subnetting does not really introduce any new concepts but, rather, combines multiple subnet problems into one. This is known as Variable Length Subnet Masking (VLSM). With VLSM, you can change subnet masks wherever you want in your organization. In order to use VLSM, you must be using a routing protocol that supports it. While routing protocols will be discussed later in this guide, let me list them here as they relate directly to this topic:

Classful Routing Protocols (No VLSM)	**Classless Routing Protocols (Support VLSM)**
RIPv1	RIPv2
IGRP	OSPF
	IS-IS
	EIGRP
	BGP

Now, let's walk through one more subnetting scenario to explain how VLSM is used.

Example 5: The corporate network for ACME Inc. is shown in the following network diagram. The organization wishes to subnet the Class C address 192.168.100.0/24 to fit their organization. Subnet this Class C network using the most efficient addressing possible.



When approaching this problem, you must remember to begin with the largest subnet first. In this case, the network of 50 users is the largest. So, let's use the skills we've discussed so far to figure this out. I'll be very brief in my descriptions since we've done many examples like this already:

Step (1) **Determine the number of clients and convert to binary.**

- The binary representation of 50 = 00110010.

Step (2) **Reserve required bits in a subnet mask and find incremental value.**

- 50 clients require 6 host bits.
- Our original subnet mask is 255.255.255.0

  255.255.255.0 = 11111111.11111111.11111111.00000000

- We must save 6 host bits to meet the requirements.

  New Mask = 11111111.11111111.11111111.11000000

- Our new mask in a decimal version is **255.255.255.192** or **/26** in bit notation.
- Our increment (lowest network bit) is **64.**

Step (3) **Use increment to find network ranges.**

- Starting with our given network address:

  192.168.100.0
  192.168.100.64

- We can stop after we find a single range since our network diagram shows only one network of 50 users. Let's fill in the end range.

  192.168.100.0 - 192.168.100.63
  192.168.100.64

The single subnet (in bold) is the only large subnet we need. Now, with VLSM, we can move on and find the subnet mask for the next largest subnet: 20 users.

Step (1) **Determine the number of clients and convert to binary.**

- The binary representation of 20 = 00010100.

Step (2) **Reserve required bits in a subnet mask and find incremental value.**

- 20 clients require 5 host bits.
- Our original subnet mask is 255.255.255.0

  255.255.255.0 = 11111111.1111111.11111111.00000000

- We must save 5 host bits to meet the requirements.

  New Mask = 11111111.11111111.11111111.11100000

- Our new mask in a decimal version is **255.255.255.224** or **/27** in bit notation.
- Our increment (lowest network bit) is **32.**

Step (3) **Use increment to find network ranges.**

- Since 192.168.100.0 - 192.168.100.63 are used in our 50-user network, the first two network ranges are not usable. We need to pick up where the last subnet problem left off:

  192.168.100.0 (***in use***)
  192.168.100.32 (***in use***)
  192.168.100.64 (***available***)
  192.168.100.96 (***available***)
  192.168.100.128

- We can stop after we find two ranges since our network diagram shows two networks of 20 users. Let's fill in the end range.

  192.168.100.64 - 192.168.100.95
  192.168.100.96 - 192.168.100.127
  192.168.100.128

The two subnets (in bold) will be used to address the networks of 20 users. Now, we can move on to the final piece of the equation: the WAN links. Each WAN link has two host addresses (one for each router).

Step (1) **Determine the number of clients and convert to binary.**

- The binary representation of 2 = 00000010.

Step (2) **Reserve required bits in a subnet mask and find incremental value.**

- 2 clients require 2 host bits.
- Our original subnet mask is 255.255.255.0

  255.255.255.0 = 11111111.11111111.11111111.00000000

- We must save 2 host bits to meet the requirements.

  New Mask = 11111111.11111111.11111111.11111100

- Our new mask in a decimal version is **255.255.255.252** or **/30** in bit notation.
- Our increment (lowest network bit) is **4.**

Step (3) **Use increment to find network ranges.**

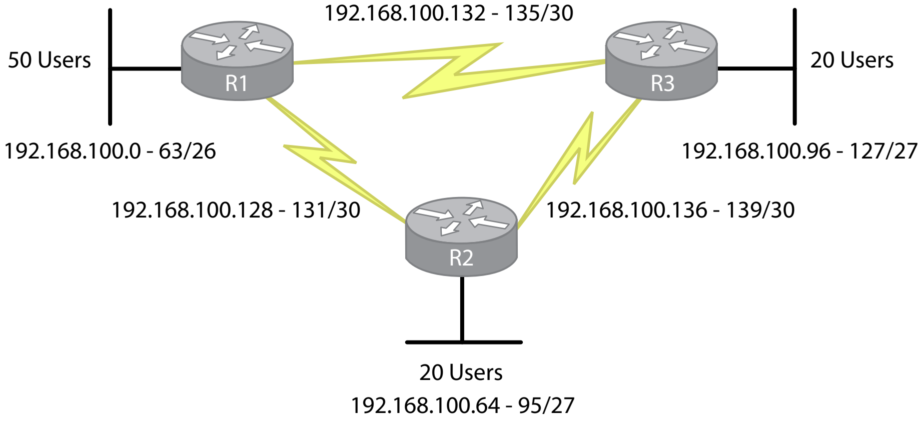
- Since 192.168.100.0 - 192.168.100.127 are used in our 50- and 20-user networks, we need to pick up where the last subnet problem left off.

  192.168.100.128
  192.168.100.132
  192.168.100.136
  192.168.100.140

- We can stop after we find three ranges since our network diagram shows three WAN links. Let's fill in the end range

  192.168.100.128 - 192.168.100.131
  192.168.100.132 - 192.168.100.135
  192.168.100.136 - 192.168.100.139
  192.168.100.128

Now we can assemble all this subnet information into one, big VLSM network diagram:



Beautiful! The last thing I will say to wrap up this section is that subnetting is not a skill you can master overnight. It will take plenty of practice to sharpen your skills. Google can be your best friend for finding subnetting practice.

Domain 7 - Base Configuration of Cisco Routers

Initial Router Configuration

Unlike Cisco switches, Cisco routers will do absolutely nothing until you have configured them appropriately. In this initial configuration section, we will cover the following topics:

- Getting familiar with the router
- Assigning passwords
- Configuring a hostname and logon banner
- Enabling Secure Shell (SSH)
- Configuring router interfaces
- Configuring your router to support Cisco SDM
- Configuring your router as a DHCP server using the SDM
- Managing Cisco routers using Telnet and SSH
- Verifying and saving the router configuration

Getting Familiar with the Router

During the initial router boot process, you will encounter output similar to the Cisco switch. While I won't repeat all that information here, I would like to show you some of the key output during the end of the boot process:

```
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.2(8)T5,  RELEASE SOFTWARE
(fc1)
TAC Support: http://www.cisco.com/tac
Copyright (c) 1986-2002 by cisco Systems, Inc.
Compiled Fri 21-Jun-02 08:50 by ccai
Image text-base: 0x80008074, data-base: 0x80A2BD40

cisco 2620XM (MPC860P) processor (revision 0x100) with 28672K/4096K bytes
of memory.
Processor board ID JAD0711047A (1876098305)
M860 processor: part number 5, mask 2
Bridging software.
X.25 software, Version 3.0.0.
1 FastEthernet/IEEE 802.3 interface(s)
1 Serial network interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read/Write)

         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]:
```

I have bolded the key information from the boot process. Just by watching the router boot, we can observe the following:

- The router model is 2620XM.
- The router has 32Mb of installed memory (28672K + 4096K due to memory partitioning).
- This router is equipped with 1 FastEthernet and 1 Serial interface.
- The router has 32Kb of NVRAM (for the startup-config).
- The router has 16Mb of Flash (for the IOS image).

Just as with the switch, after the initial boot, the router prompts us to enter initial configuration dialog. Because of this question, we can tell that the router does not have any existing configuration. You should always answer "no" to the initial configuration dialog.

Once you have come to a user-mode prompt on the router, it is always best to move directly to privileged mode and enter a command such as **show run** or **show ip interface brief** to become familiar with the exact interfaces of the router:

```
Router>enable
Router#show ip interface brief
Interface        IP-Address  OK?  Method  Status                 Protocol
FastEthernet0/0  unassigned  YES  unset   administratively down  down
Serial0/0        unassigned  YES  unset   administratively down  down
```

From this output, we can determine that the router has two interfaces: FastEthernet 0/0 and Serial 0/0. This would be excellent to document for future reference.

Assigning Passwords

Just as with a switch, you must assign passwords at multiple levels on a Cisco router. Use the following commands to accomplish this:

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line console 0
Router(config-line)#password cisco
Router(config-line)#login
Router(config-line)#exit
Router(config)#line vty 0 4
Router(config-line)#password cisco
Router(config-line)#login
Router(config-line)#end
Router#
```

Let’s discuss the key commands:

- **line console 0** - Moves from global configuration mode into line configuration for the console port.
- **line vty 0 4** - Moves from global configuration mode into line configuration for the Virtual Terminal (VTY) lines. These VTY lines receive Telnet and SSH connections. Most Cisco devices allow up to five simultaneous Telnet/SSH connections. Typing **line vty 0 4** configures all five of these ports at the same time.
- **password <*password*> -** Sets the password for the console or VTY lines. In our example, you must now type the password “cisco” to get into user mode from the console or VTY lines.
- **login -** Requires logins to the port. If you enter the **password** command without entering the **login** command, the user will never be prompted for the password, even though you have one set, since logins are not required. Note: the **login** command exists by default under VTY lines but must be entered under the console line.

After performing the previous configuration, the User mode is now protected on your Cisco router. We now need to protect the transition from User mode to Privileged mode (accomplished by typing the **enable** command). Just as with the switch, this can be accomplished using the **enable secret** or the **enable password** command. Because the **enable password** is stored in the running configuration without encryption, most opt to use just the **enable secret** command, as shown below:

```
Router(config)#enable secret cisco
```

Our Cisco router is now password protected.

Configuring a Hostname and Logon Banner

Every Cisco IOS device has a hostname that is used to uniquely identify it from other devices in the network. To assign a hostname, use the following syntax:

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname Neo
Neo(config)#
```

For legal purposes, it's always good to have a logon banner on all Cisco devices. To configure a logon banner (known as a Message of the Day or MOTD banner in the Unix/Cisco realm), use the following syntax:

```
Neo(config)#banner motd %
Enter TEXT message.  End with the character '%'.
*************************************************************
This is a private system. Unauthorized access prohibited.
*************************************************************
%
Neo(config)#
```

The **banner motd <*delimiter*>** command is fairly straightforward. Keep in mind that the delimiter character can be any character you wish. It simply marks the start and end of your logon banner; in my example, I chose to use the percent (%) since I did not plan to use that character anywhere in the logon banner.

Enabling Secure Shell (SSH)

By default, every Cisco device supports Telnet access. Telnet has been around since the foundation days of networks when security was not much of an issue. Today, most networks prefer to use SSH since everything sent using the Telnet protocol is sent in clear-text. SSH performs strong encryption on all the data sent or received. Use the following syntax to enable your switch to support SSH:

```
Neo#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Neo(config)#username Jeremy password cisco
Neo(config)#ip domain-name preplogic.com
Neo(config)#crypto key generate rsa
The name for the keys will be: Neo.preplogic.com
```

```
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

Neo(config)#
01:06:57: %SSH-5-ENABLED: SSH 1.99 has been enabled
Neo(config)#line vty 0 4
Neo(config-line)#login local
Neo(config-line)#transport input ssh telnet
```

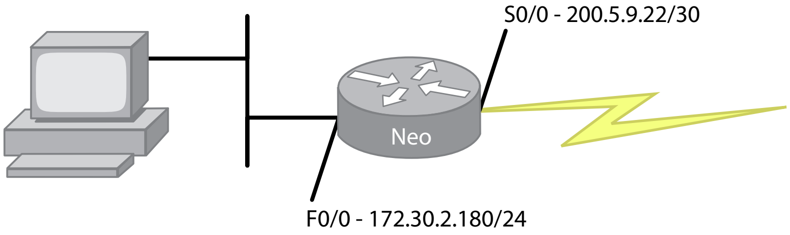
Because this is identical to the switch configuration for SSH, refer back to the section entitled "Working with Cisco Switches" for a deeper description of these commands.

Configuring Router Interfaces

Now that we have the basic security functions and remote access capabilities set up on the router, we can turn our attention to configuring the interfaces. Each interface of a router:

- Is shutdown by default (administratively down)
- Is set to auto-negotiate speed and duplex
- Must be assigned an IP address on a different network than other interfaces on the router
- Creates a separate broadcast domain

For our configuration, we have been given the following network diagram:



To configure this scenario, move to global configuration mode and enter the following commands:

```
Neo(config)#interface fa0/0
Neo(config-if)#ip address 172.30.2.180 255.255.255.0
Neo(config-if)#speed 100
Neo(config-if)#duplex full
Neo(config-if)#no shutdown
Neo(config-if)#exit
Neo(config)#interface s0/0
Neo(config-if)#ip address 200.5.9.22 255.255.255.252
Neo(config-if)#no shutdown
```

For this scenario, I am assuming the switch the Neo router is connected to is also set to a hardcoded speed of 100Mbps and full duplex.

Configuring Your Router to Support Cisco SDM

Once you have assigned IP addresses to the interfaces of the router and enabled them by using the **no shutdown** command, you can now begin configuring the router remotely. The Cisco CCENT certification was the first time Cisco began introducing the Security Device Manager (SDM) into mainstream certifications. The Cisco SDM is a graphic user interface (GUI) you can use to configure common functions on your Cisco router. Before you can use the SDM, two things have to happen:

1. Your router must be configured to support the SDM.
2. You must download the Cisco SDM.

Let's start off by configuring our Cisco router to support the SDM GUI:

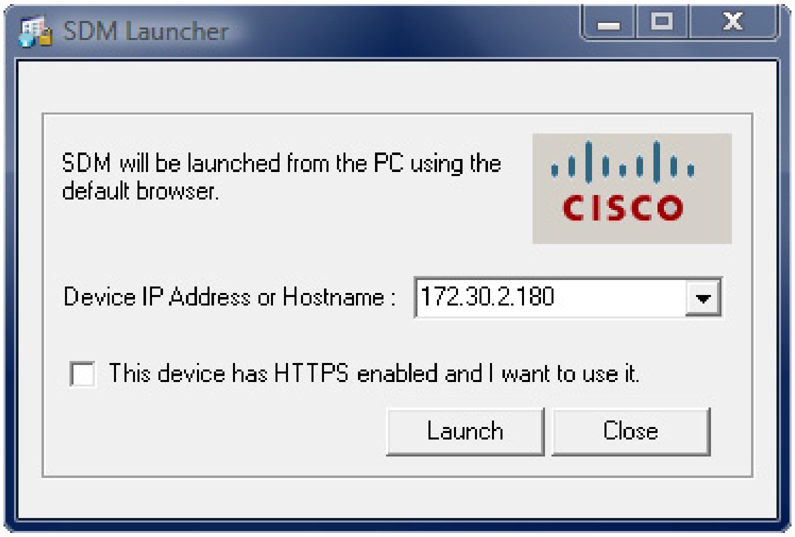
```
Neo(config)#ip http server
Neo(config)#ip http authentication local
Neo(config)#username Jeremy secret cisco
Neo(config)#username Jeremy privilege 15
Neo(config)#line vty 0 4
Neo(config-line)#privilege level 15
Neo(config-line)#login local
```

Let's discuss the key commands:

- **ip http server -** Enables the HTTP server functionality on your router (the SDM is accessed through a web interface). Note: you can also enter the command **ip http secure-server** to enable HTTPS encrypted functionality when using the SDM.
- **ip http authentication local -** Instructs the router to use the local user database when authenticating users attempting to access the SDM interface.
- **username Jeremy secret cisco -** Adds the user account "Jeremy" with an encrypted password of "cisco" to the local user database used for authentication with the SDM.
- **username Jeremy privilege level 15 -** Gives the user account "Jeremy" full, privileged access to the router. This type of user account is necessary for the SDM.

- **privilege level 15** - Gives instant privileged-level access to user accounts that successfully authenticate through the VTY ports of the router. This is necessary since the SDM accesses the router through the VTY ports and entering commands while you are doing the configuration through the SDM GUI.
- **login local -** Instructs the VTY lines to use the local user database on the router for authenticating users rather than the password entered under the VTY lines.

Your router is now ready to be accessed using the SDM graphic interface. To download this software, you must have a Cisco CCO account with the necessary privileges. After the software has been downloaded and installed, you can double-click the Cisco SDM to access the router of your choice:

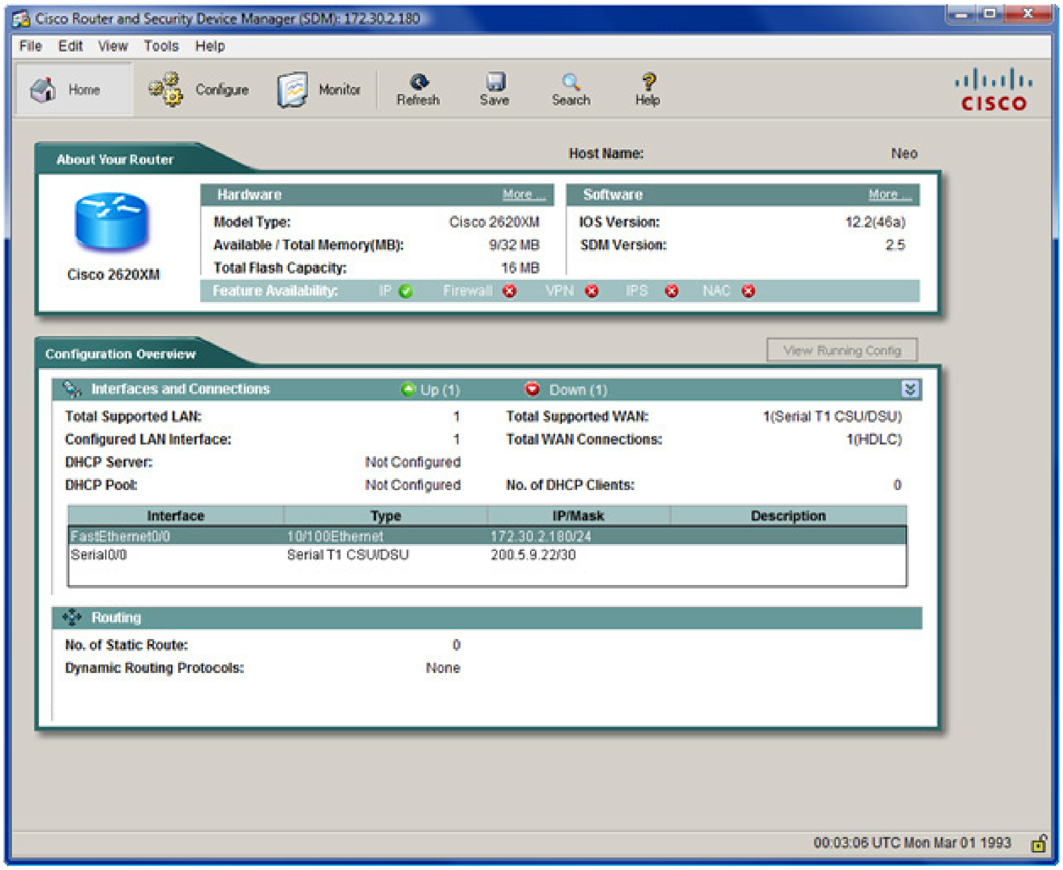


SDM will be launched from the PC using the default browser.

Device IP Address or Hostname : 172.30.2.180

□ This device has HTTPS enabled and I want to use it.

Once you click the **Launch** button, you will be prompted for the authentication credentials you entered in the previous router configuration. In this case, I entered the username of **Jeremy** and a password of **cisco**. The Cisco SDM main window then opens.



Host Name: Neo

Cisco 2620XM

Hardware	
Model Type:	Cisco 2620XM
Available / Total Memory(MB):	9/32 MB
Total Flash Capacity:	16 MB

Software	
IOS Version:	12.2(46a)
SDM Version:	2.5

Feature Availability: IP ✓ Firewall ✗ VPN ✗ IPS ✗ NAC ✗

Interfaces and Connections ● Up (1) ● Down (1)

Total Supported LAN:	1	Total Supported WAN:	1(Serial T1 CSU/DSU)
Configured LAN Interface:	1	Total WAN Connections:	1(HDLC)
DHCP Server:	Not Configured		
DHCP Pool:	Not Configured	No. of DHCP Clients:	0

Interface	Type	IP/Mask	Description
FastEthernet0/0	10/100Ethernet	172.30.2.180/24	
Serial0/0	Serial T1 CSU/DSU	200.5.9.22/30	

Routing

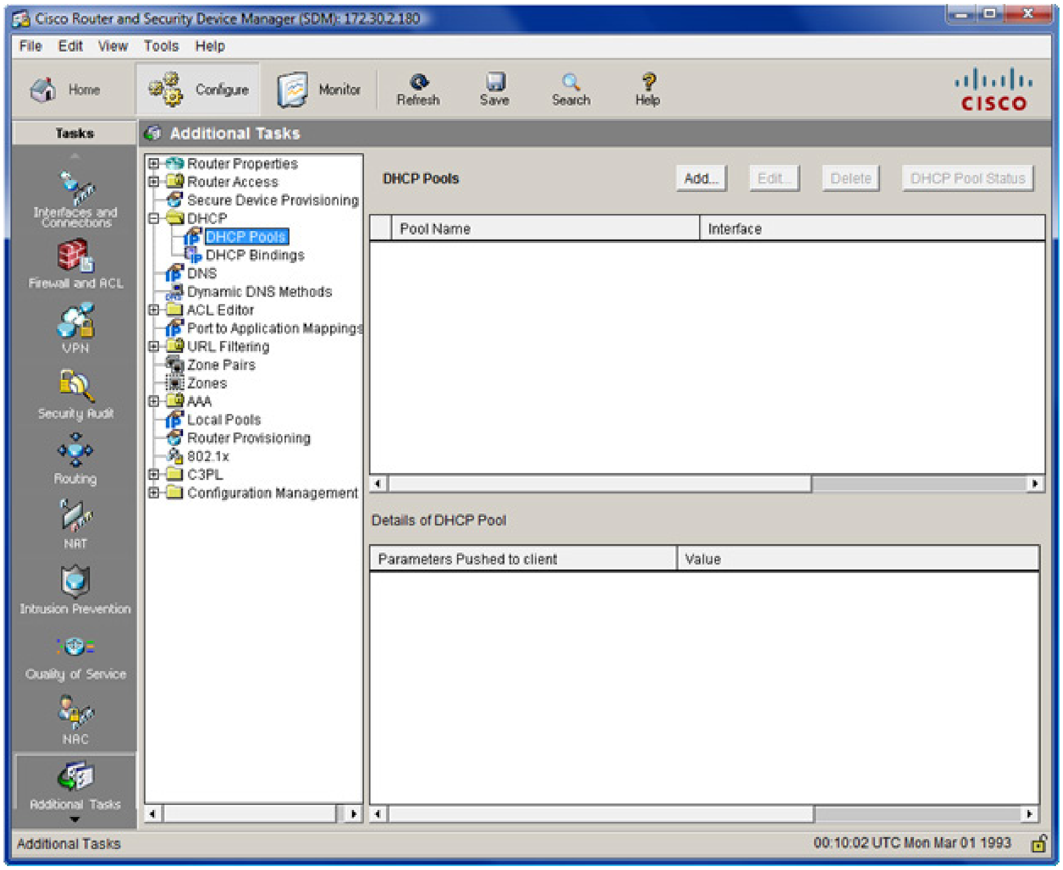
No. of Static Route:	0
Dynamic Routing Protocols:	None

00:03:06 UTC Mon Mar 01 1993

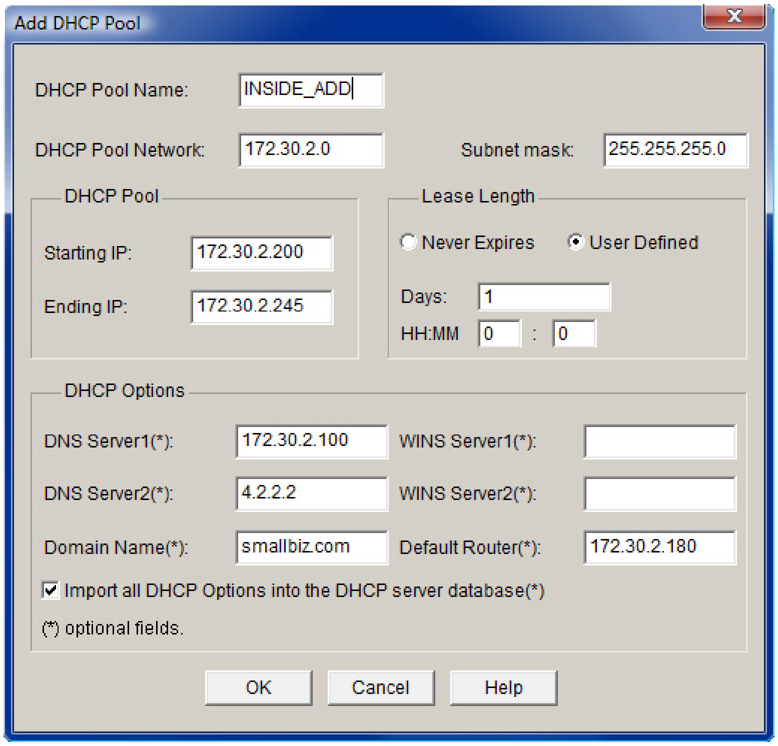
From this initial window, you can get an overview of your router's configuration and operational status.

Configuring Your Router as a DHCP Server Using Cisco SDM

Many small business environments do not have a dedicated server on their network to run as a DHCP server. In this case, you can configure your Cisco router as a DHCP server using the command-line interface or the Cisco SDM. The CCENT and CCNA exams expect you to be able to use the SDM GUI to perform this configuration. To accomplish this, open the SDM and click the **Configure** button at the top. Once you arrive at the configuration screen, scroll down and select the **Additional Tasks** configuration icon from the left side of the window. From the Additional Tasks menu that opens, expand the DHCP folder and select **DHCP Pools**. All of this can be seen in the following figure:



Once you have arrived here, click the **Add** button at the top of the screen. In the new window that appears, you can configure the scope of DHCP addresses you would like to hand out to fit your specific network. I have configured a sample in the following figure:



Add DHCP Pool

DHCP Pool Name: INSIDE_ADD

DHCP Pool Network: 172.30.2.0 Subnet mask: 255.255.255.0

DHCP Pool

Starting IP: 172.30.2.200

Ending IP: 172.30.2.245

Lease Length

○ Never Expires ● User Defined

Days: 1

HH:MM 0 : 0

DHCP Options

DNS Server1(*): 172.30.2.100 WINS Server1(*):

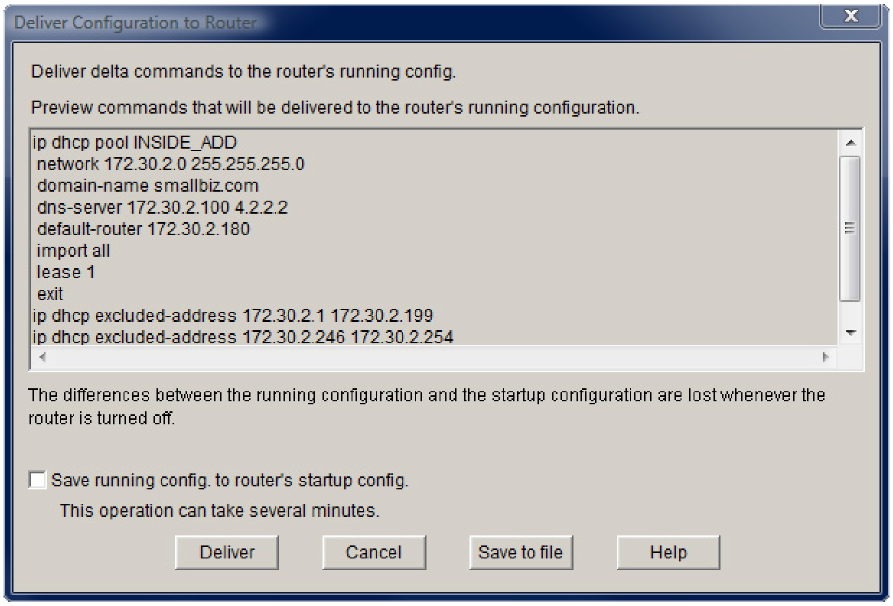
DNS Server2(*): 4.2.2.2 WINS Server2(*):

Domain Name(*): smallbiz.com Default Router(*): 172.30.2.180

☑ Import all DHCP Options into the DHCP server database(*)

(*) optional fields.

Once you have configured the DHCP scope, you can click the **OK** button. Depending on the options you chose in the Cisco SDM window, you can have the router generate a preview of the commands it will be sending to the Cisco router. This is not only a way to verify the configuration but a powerful method of learning the command-line syntax to accomplish many of the features the SDM GUI performs in the click of a button.



Deliver Configuration to Router

Deliver delta commands to the router's running config.

Preview commands that will be delivered to the router's running configuration.

```
ip dhcp pool INSIDE_ADD
 network 172.30.2.0 255.255.255.0
 domain-name smallbiz.com
 dns-server 172.30.2.100 4.2.2.2
 default-router 172.30.2.180
 import all
 lease 1
 exit
ip dhcp excluded-address 172.30.2.1 172.30.2.199
ip dhcp excluded-address 172.30.2.246 172.30.2.254
```

The differences between the running configuration and the startup configuration are lost whenever the router is turned off.

☐ Save running config. to router's startup config.

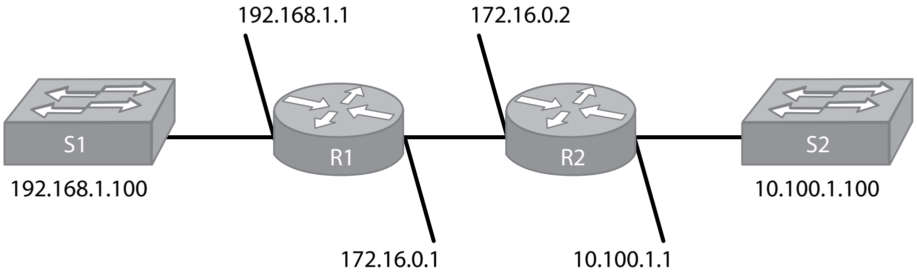
This operation can take several minutes.

Managing Cisco Routers Using Telnet and SSH

Connecting to Cisco routers using the console cable is rarely done since they are typically in a frigid IT room in an inconvenient location. Most of the time, you will access these devices through Telnet or SSH.

- **Telnet** - Convenient, widely available, very insecure protocol (all data sent in clear-text).
- **SSH** - Not-as-convenient or widely available, but very secure protocol (all data encrypted).

While we have already discussed the configuration of a Cisco router to support these remote management protocols, we have not discussed their use on a Cisco router. Say we had the following network configuration:



You have telnetted to R1 from your laptop PC and would like to access R2. You can accomplish this by entering the following command from privileged mode:

```
R1#telnet 172.16.0.2
Trying 172.16.0.2 … Open

User Access Verification
Password:
R2>
```

You have now opened a Telnet session from R1 to R2. From here, you can make any changes to R2 that are necessary. Once you are finished with the telnet session, you can simply type **exit** from privileged mode to close it out. However, there are many times when you may want to temporarily suspend a telnet session to do some work on a previous device. For example, you may be accessing the R2 router and need to get back to R1 without completely closing the current telnet session. To accomplish this, use the following keystroke: <**ctrl + shift + 6>**, followed by **x**. For example:

```
R2>enable
R2#configure terminal
R2(config)#(<ctrl + shift + 6>, x pressed here)
R1#
```

As soon as you type this suspend keystroke, you are immediately taken back to the previous router. You can always verify what open sessions you have by typing the command **show sessions**:

```
R1#show sessions
Conn Host             Address           Byte     Idle     Conn Name
*  1 172.16.0.2       172.16.0.2        0        0        172.16.0.2
```

R1 currently has a single open session to R2 (172.16.0.2). From here, you could telnet to other locations from R1 or resume your connection to R2. To resume the connection, simply type the command **resume <*connection number*>** from privileged mode. In this case, the connection number (shown as "Conn" in the show sessions output) is 1.

You can also enter the **show users** command to verify if anyone is telnetted into your router. For example, if you were to enter the **show users** command on R2, you would get the following output:

```
R2#sh users
Line            User       Host(s)        Idle          Location
*  0 con 0                 idle           00:00:00
  11 vty 0                 idle           00:02:32      172.16.0.1
```

This verifies that R1 (172.16.0.1) is currently accessing R2 on the VTY 0 port. So, to summarize:

- **show sessions** - verifies telnet sessions coming *from* your router.
- **show users** - verifies telnet sessions coming *to* your router.

Viewing and Saving the Router Configuration

The two primary verification commands you'll want to know at this point are the **show running-config** and the **show interface** commands.

- show running-config - contains every command entered into the configuration of the Cisco router.
- show interfaces - verifies key information about a specific interface of the router.

The following is an example of the show interfaces command on a Cisco router:

```
Neo#show interfaces fastethernet 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 000c.31e8.99a0 (bia 000c.31e8.99a0)
  Internet address is 172.30.2.180/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
```

```
Last input 00:00:14, output 00:00:03, output hang never
Last clearing of “show interface” counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
   2485 packets input, 176204 bytes
   Received 283 broadcasts, 0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
   0 watchdog
   0 input packets with dribble condition detected
   2604 packets output, 212149 bytes, 0 underruns
   0 output errors, 0 collisions, 2 interface resets
   0 babbles, 0 late collision, 0 deferred
   0 lost carrier, 0 no carrier
   0 output buffer failures, 0 output buffers swapped out
```

The key information from this output is bolded above and discussed below:

- **FastEthernet0/0 is up, line protocol is up** - Verifies the Physical and Data Link layer operation of the interface.
- **Address is 000c.31e8.99a0** - Displays the MAC address of the interface.
- **Full-duplex, 100Mb/s** - Verifies the duplex and speed settings of the interface.
- **Internet address is 172.30.2.180/24** - Displays the IP address of the interface.

Furthermore, the packet statistics shown at the bottom of the output can aid in troubleshooting poorly performing interfaces.

Domain 8 - Understanding and Configuring Routing

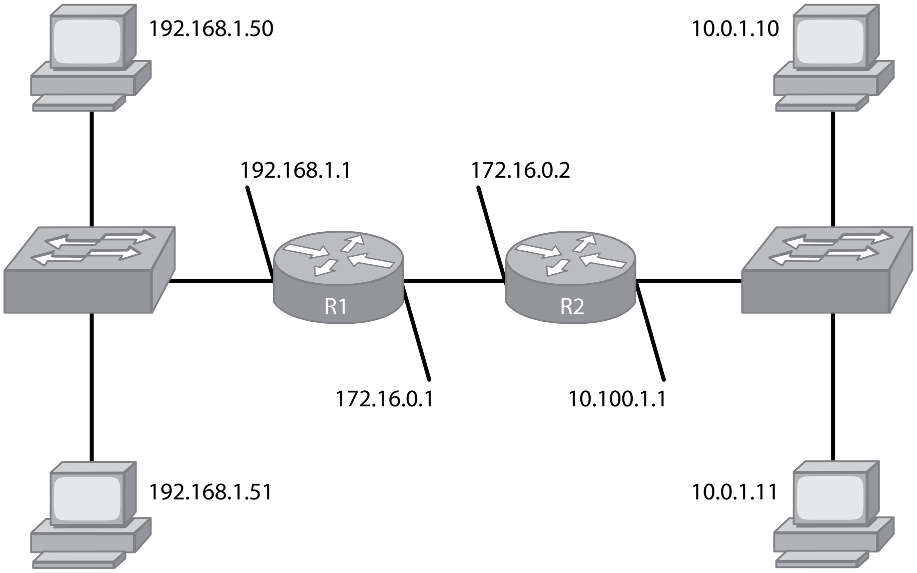
Letting a Router Route

Up until this point, we have put a base configuration on our routers. Thus, our routers have passwords, logon banners and IP addresses; however, at this point, our routers are indeed "routers," but they are not "routing." That is, they are not sending traffic to the destinations where they belong. In the next sections, we will discuss the following topics:

- Understanding the routing operation of routers
- Understanding and configuring static routing
- Understanding the classes of dynamic routing protocols
- Understanding and configuring dynamic routing with RIP
- Understanding and configuring dynamic routing with EIGRP
- Understanding and configuring dynamic routing with OSPF

Understanding the Routing Operation of Routers

As we discussed in the Network Fundamentals section of this guide, the goal of routers is to move unicast packets through the network to their destination. The router with a base configuration immediately encounters a problem with this purpose since it only knows about networks to which it is directly connected. For example, suppose you managed the following network:



After you placed a base configuration (passwords, IP addresses, etc…) on R1, it would be able to successfully reach only directly-connected networks (192.168.1.0/24 and 172.16.0.0/24, assuming Class C subnet masks). R1 could not reach the network behind R2 (10.100.1.0/24). Likewise, if R2 had this same base configuration, it could not reach the 192.168.1.0/24 network behind R1. In order to fix this dilemma, we have to use some form of routing.

In the Cisco realm, there are two forms of routing that exist:

- **Static Routing** - A form of routing which requires administrators to manually enter the destination network and the path to reach that destination.
- **Dynamic Routing** - A form of routing that allows the routers to communicate and exchange network information.

Most networks will use some combination of static and dynamic routing to accomplish their goals.

Understanding and Configuring Static Routing

Static routing is a powerful method of routing that allows a network administrator to manually enter a route into the routing table. To enter a static route, use this general syntax:

```
Router(config)#ip route <destination net> <subnet mask> <next hop
IP address>
```

Let's put this general syntax to practice through an example. Looking at the network diagram on the previous page, we could create a fully routed network through the following two commands:

```
R1(config)#ip route 10.100.1.0 255.255.255.0 172.16.0.2
```

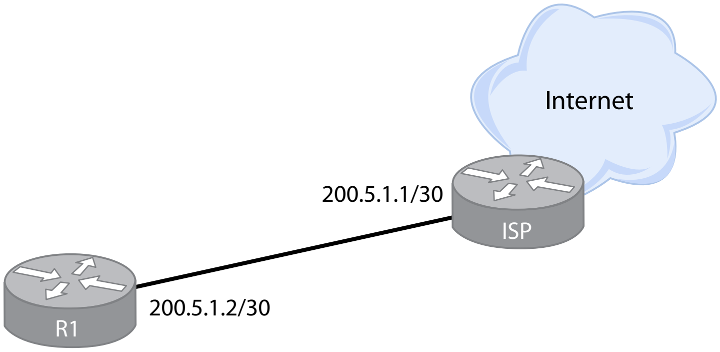
```
R2(config)#ip route 192.168.1.0 255.255.255.0 172.16.0.1
```

Notice first that we are adding two separate routes on two separate routers. We tell R1, "to reach the 10.00.1.0/24 network, go to the IP address 172.16.0.2 (which is R2)." We tell R2, "to reach the 192.168.1.0 network, go to the IP address 172.16.0.1 (which is R1)."

Static routing is one of the simplest methods you can use to configure routing on small networks. However, when the number of networks grow in your organization, static routing can become quite inefficient.

Regardless of how large your organization grows, static routing is almost always used for one specific configuration in your network: the default route. Default routes are typically used to reach the devices on the Internet. Because the Internet routing table is over 100MB in size, most routers simply don't have enough memory to handle it. Likewise, a complex understanding of the Border Gateway Protocol (BGP, the routing protocol of the Internet) is required to manage a router with the Internet routing table.

To demonstrate the configuration of a default route, take the following scenario:



Your router, R1 on the left, needs to be able to route traffic to the internet. To accomplish this, you can enter the following configuration:

```
R1(config)#ip route 0.0.0.0 0.0.0.0 200.5.1.1
```

This statement tells R1 to send all traffic that does not have a more specific destination in the routing table off to the ISP router.

Understanding the Classes of Dynamic Routing Protocols

As your organization grows, your natural configuration tendencies will move from static routing to dynamic routing. Routing protocols are designed to exchange network information between routers, allowing them to build a complete routing table of your network. There are many different routing protocols that exist today, each offering different advantages and disadvantages. Before you can select a routing protocol, you must understand their evaluation criteria.

- Will the routing protocol be used on an ***interior*** or ***exterior*** basis?
    - Interior gateway protocols (IGPs) work inside the network of an organization.
    - Exterior routing protocols (EGPs) work on the Internet, connecting organizations together.
- Will the routing protocol be ***distance vector***, ***link state*** or ***hybrid***?
    - Distance vector routing protocols send the entire routing table on a specific time interval. In the case of RIP, this interval is once every 30 seconds. Changes to the routing table replicate to all routers in the network. These routing protocols are easy to configure.
    - Link state routing protocols send route updates only when changes occur to the routing table. In a well-designed network, updates do not need to be sent to every router and can be constrained to the area where the network change occurred. These routing protocols take more technical knowledge to configure accurately.
    - Hybrid routing protocols try to combine the best features of distance vector and link state routing protocols into a single routing protocol. Unfortunately, hybrid routing protocols are proprietary.

- Will the routing protocol be ***classful*** or ***classless***?
    - Classful routing protocols do not send subnet mask information in routing updates. If R1 were running a classful routing protocol and sending a routing update about the 10.1.1.0/24 network to R2, it would only send an update containing "10.1.1.0" (no /24 mask attached). Because of this, all routers must use the same subnet mask for the network.
    - Classless routing protocols send subnet mask information in routing updates. Using the same scenario, R1 would advertise the 10.1.1.0/24 network to R2, so R2 is not left to guess on the subnet mask.

While it would be nice to custom-pick our own criteria to create our own routing protocol, we must choose from one of the following options (at the CCNA level):

- **RIPv1 -** Interior, distance vector, classful
- **RIPv2 -** Interior, distance vector, classless
- **EIGRP -** Interior, hybrid, classless
- **OSPF -** Interior, link state, classless

The last item you should know before we get into the routing protocols themselves is the idea behind administrative distance (AD). AD represents the *believability* of a routing protocol. Each routing protocol is assigned an AD number; the lower that number, the more believable the routing protocol becomes. This way, a router is able to choose one route over another. For example, if R1 received two routing updates, one from RIP and the other from OSPF, about the 10.1.1.0/24 network, the router would need a way to choose one update over the other. The following table is what routers use to decide:

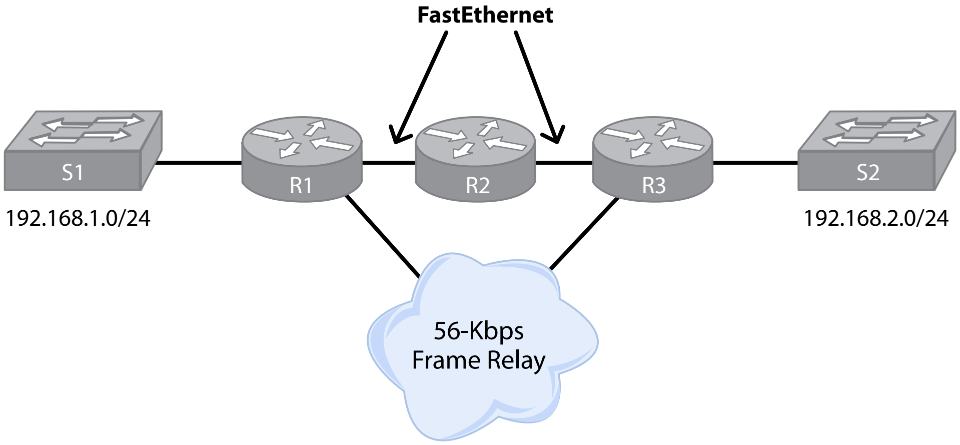
Default Administrative Distances on Cisco Routers	
Connected Interface	0
Static Route to Next Hop Address	1
EIGRP	90
OSPF	110
RIPv1, RIPv2	120

In the case we just discussed, R1 would choose the OSPF protocol over the RIP protocol since it has a lower administrative distance (110 vs. 120).

Understanding and Configuring Dynamic Routing with RIP

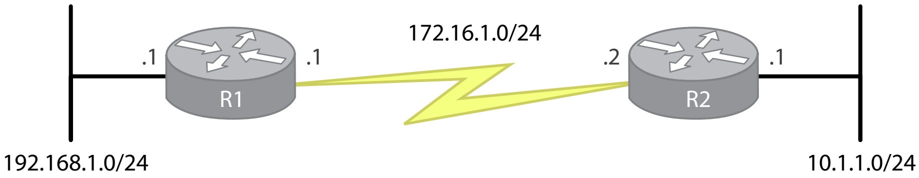
The RIP routing protocol was developed back in the 1970s and will seemingly be around forever. It is by far the simplest and most widely supported routing protocol in the world, but it lacks many features that the larger networks of today require. The following facts are relevant regarding RIP:

- RIP uses hop count as a metric. Each router that traffic passes through is considered a hop. If R1 was attempting to reach the 192.168.2.0/24 in the following figure, it would always choose the 56-Kbps Frame Relay circuit if configured with the RIP protocol.



- RIP supports networks up to 15 hops away. A network that is 16 hops away is considered unreachable.
- RIPv1 sends broadcast messages to neighboring routers with route update information every 30 seconds; RIPv2 uses multicast to send updates every 30 seconds.
- RIPv2 (over RIPv1) is the primary focus of the CCENT and CCNA exams.

To demonstrate the configuration of RIP, take the following scenario:



By default, R1 is able to reach the 192.168.1.0/24 and 172.16.1.0/24 networks since they are directly connected; it is not able to reach the 10.1.1.0/24 network. Likewise, R2 is not able to reach the 192.168.1.0/24 network. If you were to configure RIPv2 to solve this scenario, you could use the following syntax:

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 192.168.1.0
R1(config-router)#network 172.16.0.0

R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 10.0.0.0
R2(config-router)#network 172.16.0.0
```

That's all there is to it! I want to call your attention to the **network** statements in this syntax. These network commands accomplish two things:

1. Enables RIP to run on any interface belonging to that network.
2. Enables RIP to advertise these networks.

So, if you look at R1, RIP will begin running on the 192.168.1.1 interface, since this is part of the 192.168.1.0 network. R1 will also advertise the 192.168.1.0 network to its neighbors. Pay special attention to the next network command: **network 172.16.0.0**. First off, this allows RIP to run on the 172.16.1.1 interface and allows RIP to advertise the 172.16.1.0 network. A common question is, "Why didn't you enter the command **network 172.16.1.0** instead of **network 172.16.0.0**?" This is because the original design of RIP was in a classful sense. *When entering network statements with RIP you MUST enter the original class of address*. Since 172.16.1.0/24 was originally a Class B address, we enter it as 172.16.0.0.

The same thing can be seen for the 10.1.1.0/24 network on R2; the network statement is entered as 10.0.0.0 since this network was originally a Class A address. If you make a mistake on this on a real router, the router will fix the network statement for you (adjust it back to the original class of address). If you make a mistake like this on the CCENT or CCNA exam, you will lose points!

The RIP protocol can be verified through the use of the following three commands:

- **show ip protocols -** verifies the status of all routing protocols active on the router.
- **show ip route -** verifies the current entries in the routing table. RIP should begin populating the routing table after configuration.
- **debug ip rip -** allows you to see RIP updates as they are sent and received.

The following RIP output is generated from configuring the routers in the prior configuration example:

```
R1#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 7 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    Ethernet0             2     2
    Serial1               2     2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.1.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.1.2      120           00:00:26
  Distance: (default is 120)
```

Based on the output from the **show ip protocols** command, you are able to determine how often updates are being sent, the interfaces running RIP and which version of RIP they are running, the networks you are routing, and the neighboring routers on the segment.

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

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
C    172.16.1.0 is directly connected, Serial1
R    10.0.0.0/8 [120/1] via 172.16.1.2, 00:00:05, Serial1
C    192.168.1.0/24 is directly connected, Ethernet0
```

The above routing table verifies that RIP is correctly advertising the networks. Notice that the 10.1.1.0/24 network is advertised as 10.0.0.0/8. This is due to a feature known as auto-summarization. This feature attempts to shrink routing tables by summarizing networks back to their classful boundaries. Most people disable this feature by entering the command **no auto-summary** under the RIP routing process.

```
R1#debug ip rip
RIP protocol debugging is on
00:35:09: RIP: received v2 update from 172.16.1.2 on Serial1
00:35:09:      10.0.0.0/8 via 0.0.0.0 in 1 hops
00:35:15: RIP: sending v2 update to 224.0.0.9 via Ethernet0 (192.168.1.1)
00:35:15: RIP: build update entries
00:35:15:       10.0.0.0/8 via 0.0.0.0, metric 2, tag 0
00:35:15:       172.16.0.0/16 via 0.0.0.0, metric 1, tag 0
00:35:15: RIP: sending v2 update to 224.0.0.9 via Serial1 (172.16.1.1)
00:35:15: RIP: build update entries
00:35:15:       192.168.1.0/24 via 0.0.0.0, metric 1, tag 0
```

After executing the debug command above, we can see that the RIP process received an update about the 10.0.0.0/8 network and then sent an update about the 192.168.1.0/24 network to the neighboring router.

Understanding and Configuring Dynamic Routing with EIGRP

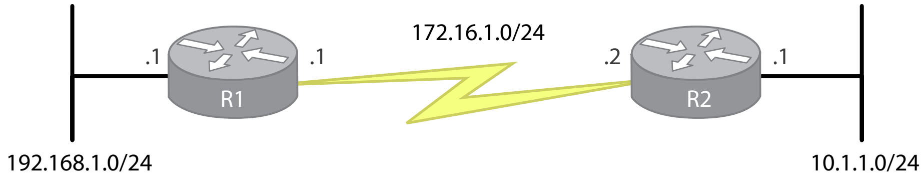
The Enhanced Interior Gateway Routing Protocol (EIGRP) is one of the simplest routing protocols to deploy, and yet it provides enough features to address just about any type of corporate network environment. EIGRP is a Cisco proprietary protocol that was designed to provide the simplistic configurations of distance vector routing protocols along with the advanced features of link state routing protocols. Because of this, it is considered a hybrid routing protocol (not truly fitting in either the distance vector or link state categories). The following is a list of unique features that EIGRP brings to the table:

- **Fast, Efficient Routing Algorithm** - EIGRP uses the Diffusing Update Algorithm (DUAL), which allows it to quickly identify backup paths in the network should a primary route fail. In addition, DUAL is far more processor and memory efficient than the Shortest Path First (SPF) algorithm, which powers the OSPF and IS-IS link state routing protocols.
- **Routing for multiple network-layer protocols** - While not as huge of a feature as it used to be, EIGRP can route for other network layer protocols such as IPX, IPv6 and Appletalk.
- **Auto or manual route summarization** - EIGRP can increase routing table efficiency by summarizing multiple, more specific routing table entries into a broader, summarized route entry. This can be done automatically (through the auto-summarization feature) or manually at any point in the network.
- **Unequal load balancing** - While all other routing protocols can only load balance over equal cost links, EIGRP can load balance accurately over unequal cost links. For example, if an organization had a 3 Mbps and a 1.5 Mbps link connecting between offices, EIGRP could accurately use both links at the same time, sending double the amount of data over the 3 Mbps link.
- **Multicast support** - All EIGRP routing information is exchanged between neighbors using multicast, which is more efficient than using broadcast or multiple unicast messages.
- **Sophisticated metric** - EIGRP uses the configured bandwidth and delay on interfaces to find the best way around the network. Other routing protocols only use a single metric, such as Hop Count (RIP) or Cost (OSPF).

EIGRP supports three memory-resident tables:

- **Neighbor table** - Contains a list of all neighbor relationships an EIGRP router has formed with other routers.
- **Topology table** - Contains a list of *all* routes that exist in the network. The primary routes are identified as *successor* routes, the backup routes are identified as *feasible successor* routes.
- **Routing table** - Contains a list of the *best* routes in the network. These routes were identified as successor routes in the topology table.

To demonstrate the configuration of EIGRP, we will use a network diagram identical to the RIP configuration. This will help show some of the differences between the EIGRP setup and the RIP setup.



The initial configuration of EIGRP looks nearly identical to the initial configuration of RIP:

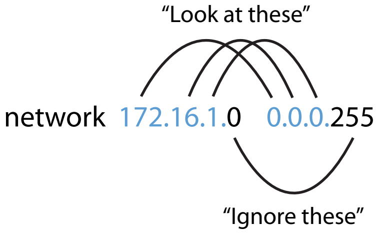
```
R1(config)#router eigrp 90
R1(config-router)#network 192.168.1.0
R1(config-router)#network 172.16.0.0

R2(config)#router eigrp 90
R2(config-router)#network 10.0.0.0
R2(config-router)#network 172.16.0.0
```

The initial, subtle difference you may notice is the number following the **router eigrp** command. This number represents the autonomous system. *In order for routers to exchange routing information, they must be a part of the same autonomous system.* In this case, I chose the number 90 for the autonomous system number.

3. The network commands work exactly the same as RIP. This is because Cisco wanted to make EIGRP as simple as the RIP protocol to configure. However, in order to support some of the advanced features of other routing protocols, Cisco also gives you the opportunity to use *wildcard masks* (also known as *inverse masks*) with your network statements. Wildcard masks give you the ability to specify exactly what interfaces you would like to use with the EIGRP routing protocol. For example, on both R1 and R2, we entered the command **network 172.16.0.0**. This starts the EIGRP process on any interface originally belonging to the class B 172.16.0.0 network. In our case, the WAN links were identified as these interfaces. However, perhaps there were other interfaces on R1 and R2 (not shown in the diagram) that were assigned a 172.16 address that we did not want to run EIGRP. We can use the wildcard mask to be more specific. For example, we could enter the following network statements:

```
R1(config-router)#network 172.16.1.0 0.0.0.255
R2(config-router)#network 172.16.1.0 0.0.0.255
```

This instructs R1 and R2 to run EIGRP on any interface that starts with 172.16.1 rather than just any interface that starts with 172.16. This allows us to be more specific. In general, wherever you see a zero in a wildcard mask, it means "look at these numbers." Wherever you see 255 (or binary 1s) in a wildcard mask, it means "ignore these numbers." For example:



This network statement says "run EIGRP on any interface having an IP address that begins with 172.16.1… but I don't care what is in the last octet of the IP address." So, R1 looks at its interfaces and sees a 172.16.1.1 IP address which matches the filter. R2 looks at its interfaces and sees a 172.16.2.1 IP address which matches the filter. We could have even been more specific with the wildcard mask statements:

```
R1(config-router)#network 172.16.1.1 0.0.0.0
R2(config-router)#network 172.16.1.2 0.0.0.0
```

These types of wildcard masks are commonly used to identify *exactly* the interface you would like to run EIGRP. Finally, we could have been extremely broad:

```
R1(config-router)#network 172.0.0.0 0.255.255.255
R2(config-router)#network 172.0.0.0 0.255.255.255
```

This would start EIGRP on any interface assigned an IP address beginning with 172. A network statement such as **network 0.0.0.0 255.255.255.255** would start EIGRP on all interfaces of a router. This is not suggested since it may cause interfaces to begin running EIGRP before they are completely configured. Wildcard masks will be seen again in both the OSPF routing protocol and access-lists.

After we have enabled EIGRP in the previous network, we can verify its operation by viewing the routing table:

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

Gateway of last resort is not set
```

```
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
D    172.16.0.0/16 is a summary, 00:00:24, Null0
C    172.16.1.0/24 is directly connected, Serial1
D    10.0.0.0/8 [90/2195456] via 172.16.1.2, 00:00:07, Serial1
C    192.168.1.0/24 is directly connected, Ethernet0
```

The routing table on R1 verifies the 10.0.0.0/8 network advertised by R2. It is received this way because of the auto-summarization features that are on with EIGRP. The auto-summarization feature enables itself anytime one class of network is advertised over a different classful network. In our example, the 10.1.1.0/24 network (by default, a Class A network) was advertised over the 172.16.1.0/24 network (a different class of network), so EIGRP summarized it back to the classful boundary of 10.0.0.0/8. If the link between R1 and R2 would have been a subnet of the 10.0.0.0/8 network, auto-summarization would not have engaged.

Anytime you see something with "auto" in its name on a Cisco device, you should immediately think "I auto-not use this." It is generally considered a good practice to disable auto-summarization and put manual summary routes where you deem necessary. To disable auto-summarization, simply go to both R1 and R2 and enter the following command from router configuration mode:

```
R1(config-router)#no auto-summary
R2(config-router)#no auto-summary
```

Once you do this, you can verify the routing table again:

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

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Serial1
     10.0.0.0/24 is subnetted, 1 subnets
D       10.1.1.0 [90/2195456] via 172.16.1.2, 00:00:36, Serial1
C    192.168.1.0/24 is directly connected, Ethernet0
```

As you can see, the 10.0.0.0/8 entry has been replaced by the more specific 10.1.1.0/24 entry.

In addition to the **show ip route** command, you can use the following commands to verify the operation of EIGRP:

- **show ip protocols -** verifies the configured parameters and current state of EIGRP.
- **show ip eigrp neighbors -** verifies the currently formed neighbors also running EIGRP and their status.
- **show ip eigrp topology -** displays the contents of the EIGRP topology table.

The following are examples of these show commands on the previous topology we configured:

```
R1#show ip protocols
Routing Protocol is “eigrp 90”
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 90
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.1.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)         90      00:13:51
    172.16.1.2            90      00:06:27
  Distance: internal 90 external 170
```

Based on the previous output, you are able to verify the network statements configured, the EIGRP autonomous system number and the neighbors (routing information sources) EIGRP has formed.

```
R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 90
H   Address     Interface   Hold Uptime   SRTT   RTO  Q    Seq Type
                            (sec)         (ms)        Cnt  Num
0   172.16.1.2  Se1         11 00:07:53   28     200  0    7
```

This output represents the EIGRP neighbor table. While much can be gleaned from this output, the primary focus is on the neighbor address, the interface to which the neighbor is connected and the Hold timer. The Hold timer represents how long until the neighbor is considered dead. Since EIGRP sends Hello packets once every 5 seconds and neighbors are considered dead if they miss three consecutive Hellos, the Hold timer will fluctuate between 10 - 15 seconds. Only if a Hello packet is missed will the Hold timer drop below 10 seconds.

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(90)/ID(192.168.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.1.0/24, 1 successors, FD is 2195456
         via 172.16.1.2 (2195456/281600), Serial1
P 192.168.1.0/24, 1 successors, FD is 281600
         via Connected, Ethernet0
P 172.16.1.0/24, 1 successors, FD is 2169856
         via Connected, Serial1
```

The previous output represents the EIGRP topology table. All routes are marked (P)assive, which is good. If the route was marked as (A)ctive, it would indicate the router is actively trying to find a replacement route since the primary route had failed.

Finally, EIGRP also supports secure authentication of all neighbors. This prevents a rogue (intrusive, typically configured by a hacker) or invalid router from forming a neighbor relationship and infecting your routing table with invalid routes. To configure authentication, you must first create a key chain in global configuration mode:

```
R1(config)#key chain EIGRP_AUTH
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string $up3r$3cret
R1(config-keychain-key)#accept-lifetime 8:00:00 Jan 1 2009 8:00:00 Jun 1
2009
R1(config-keychain-key)#send-lifetime 8:00:00 Jan 1 2009 8:00:00 Jun 1
2009
R1(config-keychain-key)#exit
R1(config-keychain)#key 2
R1(config-keychain-key)#key-string n3w$up3r$3cret
R1(config-keychain-key)#accept-lifetime 8:00:00 Jun 1 2009 8:00:00 Dec 31
2009
R1(config-keychain-key)#send-lifetime 8:00:00 Jun 1 2009 8:00:00 Dec 31
2009
```

The previous key chain configuration would need to be replicated on R2. The power of these key chains is that they allow the EIGRP authentication passwords to change on a timed interval without intervention by you as the administrator. After you have created the key chain with any number of keys, you need to apply it to the interface where the EIGRP neighbors are located:

```
R1(config)#int s1
R1(config-if)#ip authentication mode eigrp 90 md5
R1(config-if)#ip authentication key-chain eigrp 90 EIGRP_AUTH
```

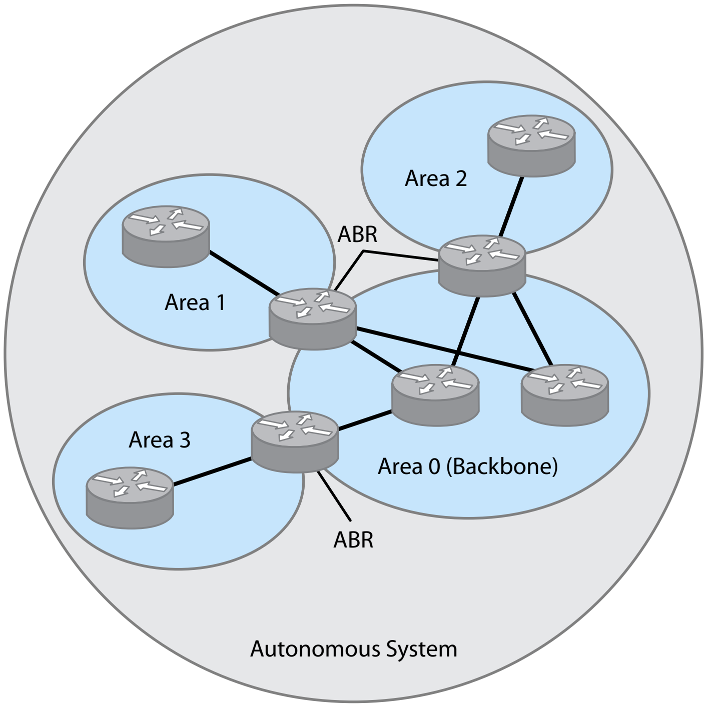
The first command applied above enables EIGRP authentication using MD5 hashing (a very secure way of exchanging passwords). The second command links in the EIGRP_AUTH key-chain we created in global configuration mode. This configuration would need to be mirrored on R2 in order for the authentication to work successfully. Keep in mind that the EIGRP neighbor relationship will stop between R1 and R2 (potentially causing a network outage) until authentication is applied to both routers.

Understanding and Configuring Dynamic Routing with OSPF

The Open Shortest Path First (OSPF) routing protocol is by far the most popular routing protocol used in mid-sized to large businesses. That is because it supports the features that most businesses need while being a non-proprietary, industry standard protocol (unlike EIGRP) able to work on any router platform.

OSPF functions very similarly to EIGRP in that it forms neighbor relationships by using the Hello protocol. By default, Hello messages are sent once every ten seconds to verify that the neighboring router is still online.

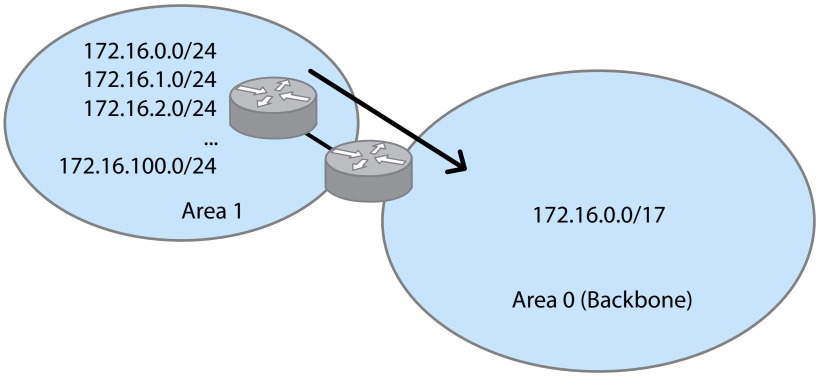
Most of the complexity of OSPF comes in understanding the terminology. The following diagram represents a network design in OSPF:



In OSPF, you must plan your network around the idea of an area. Routers within an area all have exactly the same routing information. As the network grows, the amount of routing information that your routers need to maintain can become excessive, causing all the routers to run slowly. At this point, you can break your network into multiple areas and use summary routes (identical concept to summarization in EIGRP) to limit the amount of data in the routing tables of your routers. For example, using the previous figure, imagine that there were 100 different routers in Area 1 that were using 172.16 addresses:

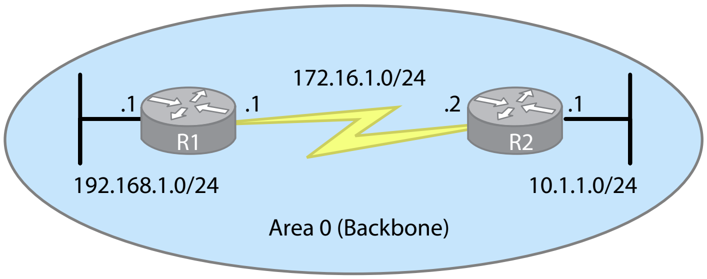
172.16.0.0/24
172.16.1.0/24
172.16.2.0/24
…
172.16.99.0/24
172.16.100.0/24

This would give quite a few entries in the routing table. This is where the summarization function of the Area Border Router (ABR) comes in. OSPF ABRs have the unique capability of summarization between areas. No other router within the OSPF system can perform summarization. In our example, we could implement summarization like this:



By using the summary route 172.16.0.0/17, we encompass the IP addresses 172.16.0.0 - 172.16.127.255, most of which are represented in Area 1. For all the routers in the backbone area (and other OSPF areas), the massive network in Area 1 is summed up in a single route table entry. While the ability to perform route summarization is not currently a CCENT or CCNA topic, it undoubtedly will be soon. Even though an in-depth understanding of route summarization is not required, the CCNA exam does require you to understand the reasoning behind OSPF area design.

Because the CCNA exam focuses on single-area OSPF configuration, we can use our previous topology to demonstrate the configuration of OSPF:



The initial configuration of OSPF is as follows:

```
R1(config)#router ospf 1
R1(config-router)#router-id 1.1.1.1
R1(config-router)#network 172.16.1.0 0.0.0.255 area 0
R1(config-router)#network 192.168.1.0 0.0.0.255 area 0
R2(config)#router ospf 1
R2(config-router)#router-id 2.2.2.2
R2(config-router)#network 172.16.1.0 0.0.0.255 area 0
R2(config-router)#network 10.1.1.0 0.0.0.255 area 0
```

While this is very similar to the EIGRP configuration, there are a few subtle differences. First, the number following the **router ospf** command represents a process-id. This is just a number to identify the OSPF process on the router; it does not need to be the same on every router (though it usually is).

Second, we see the new **router-id** command. This command is used to set the "name" of the router to the OSPF process. This name is advertised in the Hello packets and is used for the more advanced configurations which are part of the CCNP track. If you do not manually set the router-id, it will default to the highest IP address on the router (with loopback interfaces beating physical interfaces). It's always best to hardcode this under the OSPF process; it can be any IP address you want, as long as it's unique in the OSPF network.

Finally, the **network** command works nearly the same as the network command in EIGRP. The two subtle differences are the *absolute requirement* of using wildcard masks (the network command will not work without a wildcard mask). You must also specify the OSPF area to which the network belongs. At the CCNA level, you can expect only OSPF single-area configurations.

Once you have this basic configuration implemented, OSPF is running. We can verify the configuration using similar commands:

```
R2#show ip protocols
Routing Protocol is “ospf 1”
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 2.2.2.2
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.0 0.0.0.255 area 0
    172.16.1.0 0.0.0.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    1.1.1.1         110           1d22h
    2.2.2.2         110           1d22h
  Distance: (default is 110)
```

Based on the output of the **show ip protocols** command, we are able to see that OSPF is running, the networks and areas it is advertising and the routing information sources (neighboring routers).

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

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Serial1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet0
O    192.168.1.0/24 [110/74] via 172.16.1.1, 1d22h, Serial1
```

As you can see from the routing table, routes are now being learned by the (O)SPF protocol.

```
R2#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           1   FULL/  -        00:00:35    172.16.1.1      Serial1
```

Finally, the **show ip ospf neighbor** command is perhaps the handiest troubleshooting command. There are many things that can prevent an OSPF neighbor relationship from forming. The following is a brief list of the most common issues:

- **Hello or Dead Timer Mismatch -** If one neighboring router sends Hello messages on a different timer (10 seconds is the default), neighbor relationships will not form. Dead timers must also match between neighbors.
- **Area Mismatch -** Neighbors cannot form if they are configured in different areas.
- **Password Mismatch -** If you are using authentication with OSPF, neighbors must have the same password.

Lastly, enabling authentication in OSPF is simple:

```
R1(config)#interface serial 1
R1(config-if)#ip ospf authentication message-digest
R1(config-if)#ip ospf authentication-key password

R2(config)#interface serial 1
R2(config-if)#ip ospf authentication message-digest
R2(config-if)#ip ospf authentication-key password
```

As long as the two routers have the same authentication-key (password, in this case), the OSPF neighbor relationship will form.

Domain 9 - VLANs, Trunks and STP

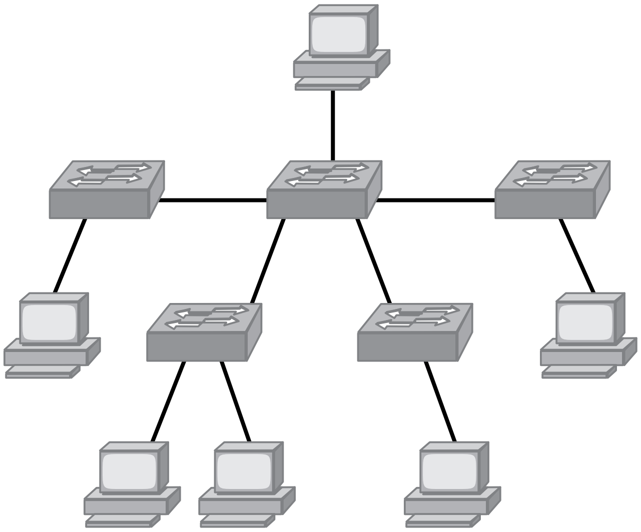
Advanced Switch Configurations

Networks have evolved from a decade ago and advanced switch configurations have become commonplace in today's corporate environments. In this section, we will discuss the following topics:

- Understanding the use of VLANs and Trunks
- Understanding VTP
- Configuration of Trunks, VTP and VLANs
- Implementing routing between VLANs
- Understanding the Spanning Tree Protocol
- Configuring and optimizing STP

Understanding the use of VLANs and Trunks

VLANs can prove quite useful in many network environments but are *most* useful in large, campus networks. A campus network is any network that has a large number of devices connected to the same LAN. For example, a college campus has many buildings, typically connected with high speed cabling that contain a large number of network devices. A high-rise building in a downtown metropolitan area might have many devices connected to the same LAN. Regardless, as the number of devices on the same network grows, so do the network management issues:



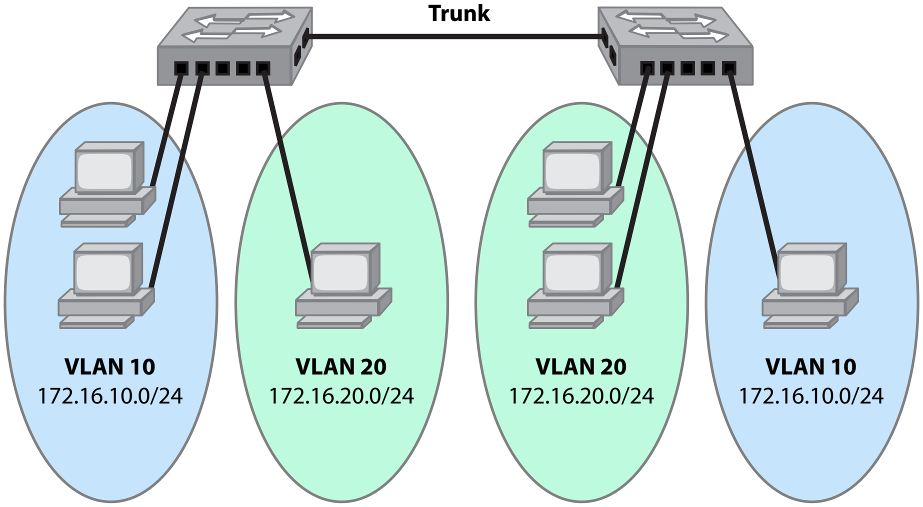
In a network environment like that pictured in the previous figure, a network administrator runs into the following challenges:

- **Unbounded Broadcast Domains -** As devices are added, the number of broadcasts increases significantly. This causes more processor load for all devices connected to the network.
- **Security Vulnerabilities -** Layer 2 security measures are extremely limited when it comes to preventing hosts on the same LAN from fully accessing each other.
- **Unknown MAC addresses** - The switches continually learn new MAC addresses as old entries age out simply because of the size of the network. This further increases the amount of broadcasts on the network since switches will flood packets that have unknown MAC addresses.
- **Management difficulties** - A flat, switched network has very limited management and monitoring support. Isolating and fixing problems in this type of network can be very time consuming.

VLANs can help with these issues. VLANs have the ability to divide the network into separate domains. The best description of a VLAN that I've ever seen is this:

A VLAN = A Broadcast Domain = A Subnet

VLANs divide your network into separate broadcast domains. A broadcast within a VLAN stays in a VLAN. Each VLAN also has its own subnet assignment. A visual representation of VLANs looks something like this:



In the figure above, there are two VLANs: VLAN 10 (the blue VLAN) and VLAN 20 (the green VLAN). When a user in VLAN 10 sends a broadcast, it will remain in VLAN 10. The same idea is true for a broadcast sent in VLAN 20. Each VLAN is assigned its own subnet. VLAN 10 devices use IP addresses from the 172.16.10.0/24 subnet. VLAN 20 devices use IP addresses from the 172.16.20.0/24 subnet.

Notice also the link between the switches. In order for broadcasts from the green and blue VLANs to reach devices connected to both switches, the switches must forward traffic from both VLANs to each other. This special type of port is known in the Cisco realm as a *trunk port*. A trunk port forwards all VLAN traffic between switches. Other switch vendors refer to these ports as *tagged ports*. They call them tagged ports for a good reason: in order for switches to send each other frames from multiple VLANs, each frame must be "tagged" with a VLAN identifier.

The protocol that performs this "tagging" is known as *802.1Q*. It is an industry standard tagging protocol, so you could have a Cisco switch connected to some other vendor switch and still use VLANs between them. The 802.1Q tags are always removed before sending the data to a connected PC (the end devices do not have any concept of VLANs; VLANs are just a managed switch configuration).

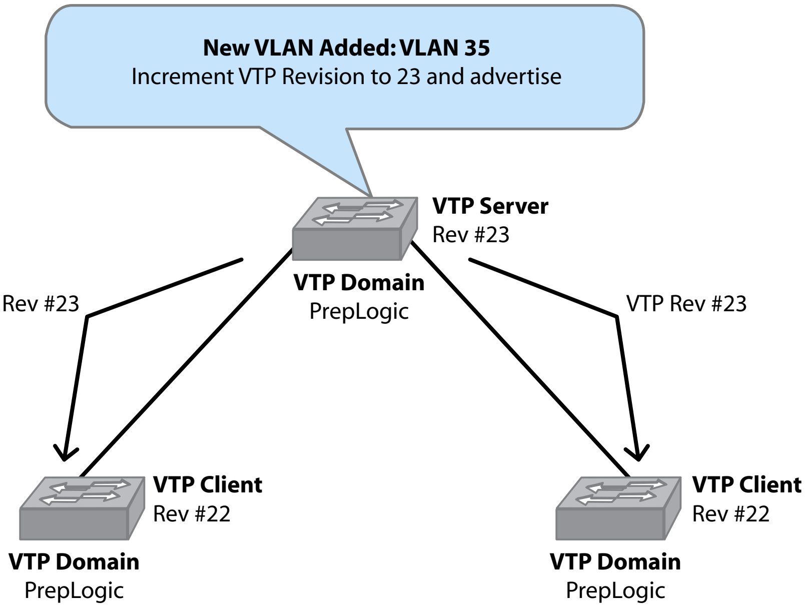
Understanding VTP

Because VLANs are a network-wide configuration, Cisco created a proprietary protocol known as the VLAN Trunking Protocol (VTP). This is the most horrific name Cisco could have chosen since VTP is not a trunking protocol at all but, rather, a replication protocol that works over trunk links. The goal of VTP is to replicate VLANs that you create from one switch to another so you don't have to visit (and configure) every switch in your network anytime a VLAN needs to be added or removed.

The concept behind VTP is simple: perform additions and deletions of VLANs from a switch in your network. That switch will then replicate the changes to all the other switches in your network which will then perform the same configurations. The following are some key facts you'll want to know about VTP:

- In order for switches to exchange VTP information, they must be in the same VTP domain. The VTP domain name is defined by you on each switch in the network. The VTP domain name is also case sensitive.
- There are three VTP modes:
    - **VTP Server** - Every switch is a VTP Server by default. These switches have the authority to create, delete and modify VLAN information. They then replicate that information to the other switches in the VTP Domain. It is best to only have one VTP Server in your switched network.
    - **VTP Client** - By converting a switch to VTP Client mode, it is no longer able to make any changes to the VLANs in the network. It can only receive and apply updates from the VTP Server.
    - **VTP Transparent** - VTP Transparent mode switches do not participate at all in VTP. By changing a switch to this mode, you effectively disable VTP. VLANs can still be created on the switch, but they do not replicate anywhere else in the network.

Since you can have multiple VTP Servers in the network, changes to the VLAN information is tracked by a VTP Revision Number. Each time the VLAN database is modified (by adding or deleting a VLAN), the switch will increment its VTP Revision Number. The new revision is then advertised out to other switches which replace their VLAN database with the new revision.



- VTP and VLAN information is not stored in NVRAM but, rather, in a file in flash called **vlan.dat**. Therefore, if you want to completely clear a switch's configuration, you must do three things:
    - Enter **erase startup-config** (or **write erase**) to erase the switch's configuration in NVRAM.
    - Enter **delete flash:vlan.dat** to delete the VLAN database.
    - Reload the switch (by typing **reload**).
- A brand new switch with no VTP configuration will automatically become a part of the first VTP Domain it hears about through VTP advertisements.

Configuration of Trunks, VTP and VLANs

In order to deploy VLANs in a Cisco switched environment, you must perform the configuration of Trunks, VTP and VLANs (preferably in that order). By default, every switchport on a Cisco switch is set to **switchport mode dynamic desirable** (this is the exact command that is entered under the switchport). This means that each interface will try to negotiate with the end device to either become an *access port* (if a PC or some other end device is attached) or a *trunk port* (if another managed switch is attached). This mode should be changed immediately since it poses many security holes. Technically-savvy end users could attach managed switches to a port in the network and automatically convert the port to a trunk port. This would mean that they would be able to get on any VLAN in your organization. To hardcode access and trunk ports, use the following commands:

```
Switch(config)#interface range fa0/1-22
Switch(config-if-range)#switchport mode access
```

The above syntax assumes you would want the first 22 ports of your switch assigned as access ports, which connect to end devices. These ports will never become trunks.

```
Switch(config)#interface range fa0/23-24
Switch(config-if-range)#switchport trunk encapsulation dot1q
Switch(config-if-range)#switchport mode trunk
```

The above syntax assumes you would want port 23 and 24 of your switch to become trunk ports. The first command (**switchport trunk encapsulation dot1q**) may not be necessary on newer switches since support for the ISL trunking protocol (which was Cisco proprietary) is being phased out. On these newer switches, you would simply enter the **switchport mode trunk** command since only one encapsulation type is supported. Once you have configured your trunk ports, you can verify their operation by typing the command **show interfaces <*interface*> switchport** as shown below:

```
Switch#show interfaces fa0/23 switchport
Name: Fa0/23
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
```

The key output to verify your trunk configuration is bolded above. The "Administrative" sections dictate the configuration of the interface. This could be set to "dynamic desirable," "trunk" or "access." The "Operational" sections dictate what the interface is currently using. For example, you might have a "dynamic desirable" administrative mode, but the operational mode is "access" if an end device connects to the port.

Once you have configured your trunk ports, you can then set up VTP.

```
Switch(config)#vtp mode <server/client/transparent>
Switch(config)#vtp domain <domain name>
Switch(config)#vtp password <password>
```

Cisco recommends that you configure a single VTP Server in your network and set the rest of the switches to be VTP Clients. VTP passwords can also be assigned to further protect the VTP domain. To verify your VTP configuration, use the command **show vtp status**:

```
Switch#show vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 12
VTP Operating Mode              : Server
VTP Domain Name                 : PREPLOGIC
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0xAA 0xFB 0xB4 0x2C 0x10 0x18 0x62 0xD9
Configuration last modified by 172.30.1.1 at 1-8-08 15:37:24
```

From the prior output, you can verify that this switch is configured as a VTP Server in the VTP domain PREPLOGIC (all other switches must use this domain name with the correct case). We can also verify that no (zero) revisions (which are additions/deletions) have been made to the VLAN database since this switch became a VTP Server since the "Configuration Revision" is set to 0.

Lastly, we need to create VLANs and assign ports to those VLANs. The following configuration creates three VLANs

```
Switch(config)#vlan 10
Switch(config-vlan)#name SALES
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name MARKETING
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#name ENGINEERING
```

The name on the VLAN is optional, but extremely helpful when you are trying to identify each VLAN. Now that the VLANs are created, you can assign ports to each VLAN:

```
Switch(config)#interface range fa0/1-5
Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#exit
Switch(config)#interface range fa0/6-10
Switch(config-if-range)#switchport access vlan 20
Switch(config-if-range)#exit
Switch(config)#interface range fa0/11-15
Switch(config-if-range)#switchport access vlan 30
```

You can verify the VLAN assignment by using the command **show vlan**. I prefer to follow that command with the ***brief*** variable to shorten the output:

```
Switch#show vlan brief

VLAN Name             Status    Ports
---- --------------   --------- -------------------------
1    default          active    Fa0/16, Fa0/17, Fa0/18,Fa0/19, Fa0/20,
                                Fa0/21, Fa0/22,Gi0/1, Gi0/2
10   SALES            active    Fa0/1, Fa0/2, Fa0/3, Fa0/4, Fa0/5
20   MARKETING        active    Fa0/6, Fa0/7, Fa0/8, Fa0/9, Fa0/10
30   ENGINEERING      active    Fa0/11, Fa0/12, Fa0/13,Fa0/14, Fa0/15
1002 fddi-default     act/unsup
1003 trcrf-default    act/unsup
1004 fddinet-default  act/unsup
1005 trbrf-default    act/unsup
```

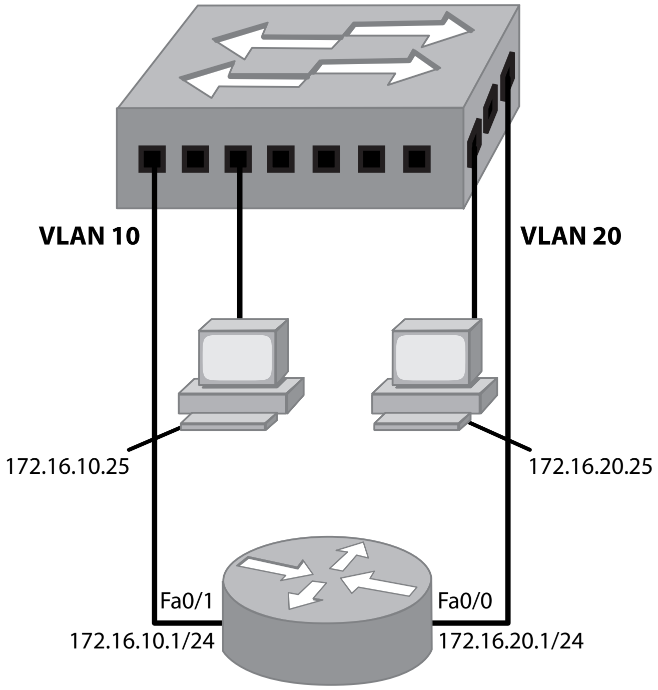
Notice from this output that ports Fa0/23 and Fa0/24 are missing. This is because they have been configured as trunk ports, which are not part of any one VLAN.

We have now created VLANs and assigned the ports. The ports that are assigned to separate VLANs can no longer access each other.

Implementing Routing Between VLANs

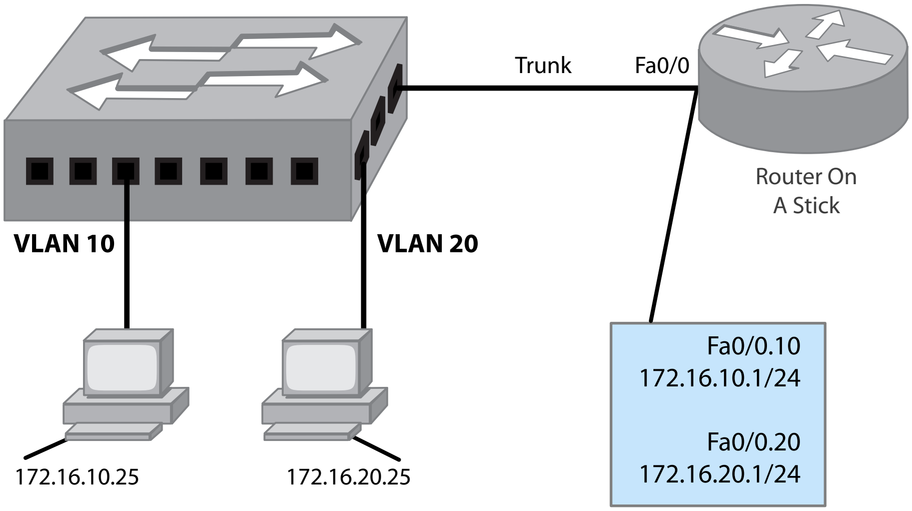
Using VLANs to segment your network can be useful to control broadcast traffic and implement security boundaries. However, allowing absolutely NO access between VLANs is never very beneficial. To fix this, we need to implement inter-VLAN routing. At the CCNA level, there are two ways we can make this happen: attach a unique router port to each VLAN or implement a router on a stick.

A physical diagram of the first of these solutions looks like this:



If the left half of the switch in the previous figure were assigned to VLAN 10 and the right half were assigned to VLAN 20, the router would need one port for each VLAN. If the clients needed to get between the VLANs, they would go through the router (which would be configured as their default gateway).

The problem with this solution is scalability. As the network grows, more VLANs will be added, and you will eventually run out of router ports, not to mention that you are using a separate port for the router interface in each VLAN. A more practical solution is the router on a stick:



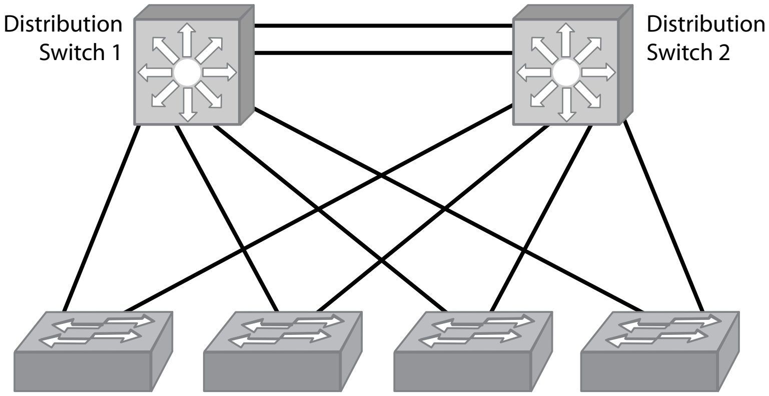
The router on a stick solution requires you to configure a trunk port from the switch to the router. Since a trunk port forwards *all* VLAN traffic, we can then configure sub-interfaces on the router to respond to each of the VLANs. The following would be a router on a stick configuration for the VLANs we previously created (10, 20 and 30):

```
Router(config)#interface fa0/0.10
Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#ip address 172.16.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface fa0/0.20
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 172.16.20.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface fa0/0.30
Router(config-subif)#encapsulation dot1q 30
Router(config-subif)#ip address 172.16.30.1 255.255.255.0
Router(config-subif)#exit
```

The previous configuration assumes we assigned the subnets 172.16.10.0/24, 172.16.20.0/24 and 172.16.30.0/24 to VLANs 10, 20 and 30, respectively. You can create literally hundreds (perhaps even thousands) of sub-interfaces on a router. The **encapsulation dot1q <*vlan*>** command applied to each sub-interface assigns a VLAN number to each sub-interface. This is how the sub-interface knows to which VLAN it should respond. The router will not allow you to assign an IP address to the sub-interface without typing this command first. Clients in each VLAN should have the sub-interface IP address set up as their default gateway.

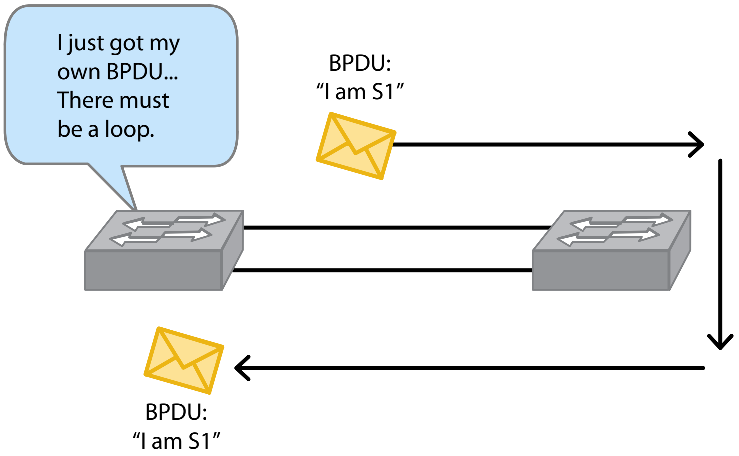
Understanding the Spanning Tree Protocol

Moving into the Spanning Tree Protocol (STP) is a complete shift of topics from VLANs. While it is still related to the switching realm, STP is designed solely to prevent loops in a switched network. For example, take the following switch topology:



The key word in this switch diagram is *redundancy*. Redundancy is good! If either distribution switch were to fail, the access switches would have a backup. However, without STP, redundancy in the switch world can be devastating. Remember, switches forward broadcast traffic out of all ports. If a device were to send a broadcast in this environment, the broadcast would end up looping through the entire network thousands of times every second. This would cause a complete network outage until all the redundant links were removed. Thus, we have discovered the goal of STP: disable the redundant links until they are necessary.

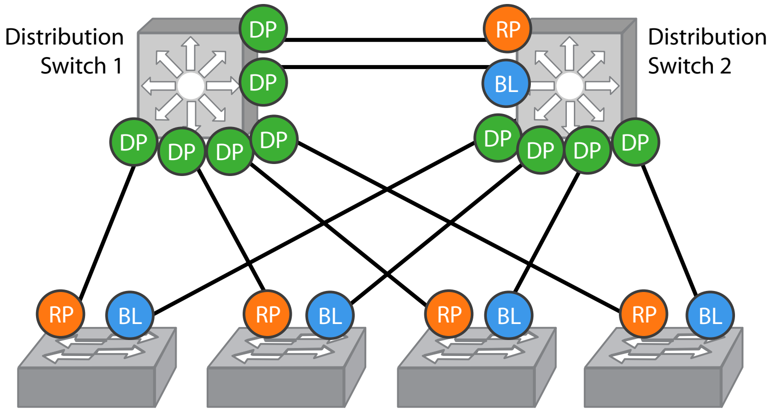
While the goal may seem like a simple task, the size of today's networks has made STP quite complex. STP is designed to allow only one active path at a time. To accomplish this, switches exchange Bridge Protocol Data Unit (BPDU) messages. This is the language of love between the switches. Since BPDU messages are sent out of all ports, it allows the switches to learn where the loops exist:



The BPDUs not only identify loops but act as election ballots. Inside the BPDU "envelope" (aka packet) are two switch IDs:

- **Switch Priority** - A value between 0 and 61440. By default, this value is 32768.
- **Switch MAC Address -** The MAC address assigned to the switch.

These two IDs are combined into one number called the **Bridge ID** and advertised to all the other switches in the network. For example, a Bridge ID might look like 32768.000c.854b.ee80. Once all the switches have exchanged BPDU packets, the switch with the lowest Bridge ID will be elected as the **STP Root Bridge**. Once the STP Root Bridge has been elected, all switches in the network will attempt to find the best path to reach the root bridge and block the redundant paths. Using the initial STP diagram, this is what the STP results would become, assuming Distribution Switch 1 was elected as the Root Bridge:



As you can see, STP uses different port identifiers to dictate the status of each port. These identifiers can be understood as the following:

- **Designated Port (DP) -** A port that is forwarding. The STP Root Bridge will always have all ports set as DPs. STP also requires that there be one DP per segment (only one side of the link is blocked to eliminate redundant paths).
- **Root Port (RP) -** A port that is used to reach the Root Bridge. This port will be kept in the forwarding state.
- **Blocked Port (BL) -** A port that is disabled to eliminate loops in the network.

The switches shown in the previous figure found the most cost-efficient path (based on link speed) to reach the Root Bridge and then blocked the redundant paths.

The STP protocol was created quite some time ago when a few minutes of downtime was not as critical as it is in today's networks. The STP standard has the following port transition process when an Ethernet cable is connected to a switch:

- **Listening (15 seconds)** - During this time, the switch is listening for and sending BPDUs. It is not forwarding any other traffic.
- **Learning (15 seconds)** - During this time, the switch is learning the MAC addresses on the ports. It is not forwarding traffic during this time.
- **Forwarding** - The port is forwarding traffic.

This means that any new Ethernet cable that is attached to a switch will take at least 30 seconds before it is able to send traffic. In addition, every switch has a **Blocking** max-age timer. This adds up to an additional 20 seconds to activate a currently blocked (BL) port should the primary path fail. These timers cause two problems. First, modern PCs are able to boot faster than the 30 second timer, leaving them temporarily without network connectivity. Second, major outages in the network occur anytime STP needs to failover to a new port. To address these problems, we can use the following two solutions:

- **PortFast -** PortFast disables the Listening and Learning timers on any port configured in the "access" mode (**switchport mode access**). This allows ports connected to end devices to immediately assume the Forwarding state.
- **Rapid STP -** A new edition of STP recently released allows the switches to remember and immediately use "Alternate" ports (alternate is a new RSTP port definition). This allows non-root backup ports to immediately failover should the primary port go down.

Configuring and Optimizing STP

Every Cisco switch runs STP by default. The main configuration you need to do is in choosing the Root Bridge in your network. Because of the way switches are manufactured (with lower MAC addresses being used first), the *oldest* switch in your network will typically be elected as the Root Bridge. This is usually the worst possible switch to be used. To configure STP, access the switch that is most core (central) to your organization and enter the following command:

```
CoreSwitch(config)#spanning-tree vlan 1-4094 root primary
```

This command sets the switch as the STP Root Bridge for all VLANs by lowering the STP Priority to 24576. You can verify the status of this by using the following show command:

```
CoreSwitch#show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
             Address     000c.854b.ee80
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    24577  (priority 24576 sys-id-ext 1)
             Address     000c.854b.ee80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------------
Fa0/7               Desg FWD 19        128.7    P2p
```

Cisco switches run a version of STP known as Per-VLAN STP (PVST). This runs an instance of STP on each VLAN. PVST adds the VLAN number to the STP Priority, which is why the above output shows the priority of the Root Bridge as 24577 rather than 24576. Once you have set the priority on the Root Bridge, all the other switches in the network will find the best way to reach the core switch (most central to the network) and block the redundant paths.

Your second step in configuring STP is to enable PortFast on ports connected to end devices. To accomplish this, you can use the following syntax:

```
Switch(config)#interface range fastEthernet 0/1-22
Switch(config-if-range)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION

%Portfast will be configured in 22 interfaces due to the range command
 but will only have effect when the interfaces are in a non-trunking
 mode.
Switch(config-if-range)#
```

In the previous example, ports 1-22 on the switch were set in PortFast mode. Take a look at the warning message given by the switch. This warning message highlights the fact that PortFast should *only* be used on ports connecting to end-devices. Using this feature on ports connected to other switches can cause loops in the network.

Finally, STP can be optimized by moving to the Rapid STP (RSTP) version of STP. However, in order to enjoy the fast failover times supported by RSTP, you must be running it on all switches in your environment. This may mean you need to upgrade IOS versions or even switch hardware. The command to enable RSTP is as follows (this must be executed on all switches in your network):

```
Switch(config)#spanning-tree mode rapid-pvst
```

Domain 10 - ACLs, NAT and IPv6

Advanced IP Operations

When most people hear the term "Access List," immediately, everything security-oriented comes to mind. While security is one very common use of Access Control Lists (ACLs), they have seemingly endless other uses on Cisco devices. One of these uses is the implementation of NAT within an organization. In this next section, we will discuss the following topics:

- Understanding the use of ACLs
- Configuring and applying Standard ACLs
- Configuring and applying Extended ACLs
- Using Named ACLs
- Verifying ACLs
- Understanding NAT implementations and configurations
- Understanding and implementing IPv6

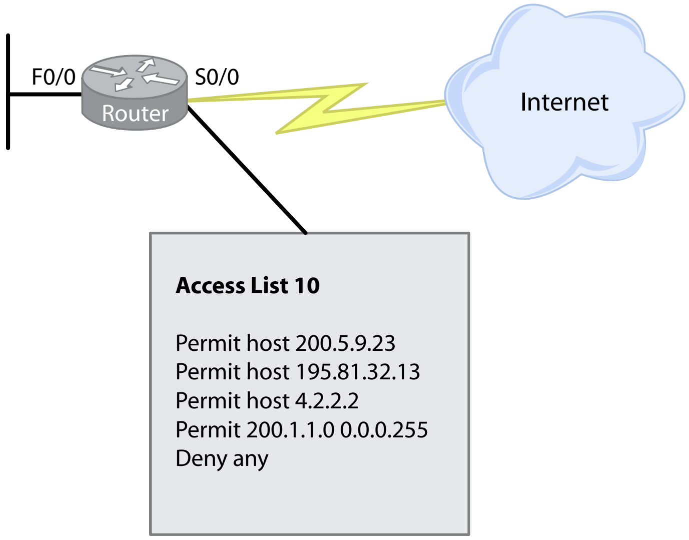
Understanding the use of ACLs

The simplest description of an access list is, "A list of permit and deny statements that identify traffic." How the access list is applied dictates how the identified traffic is treated. For example:

- **ACLs applied for security -** Permit statements dictate the traffic that is allowed through the router; deny statements dictate the traffic that is dropped by the router.
- **ACLs applied for NAT -** Permit statements dictate the traffic which will be translated by NAT; deny statements dictate the traffic which will not be translated by NAT.
- **ACLs applied for QoS -** Permit statements dictate traffic that will receive priority; deny statements dictate traffic that will not get priority.
- **ACLs applied for VPNs -** Permit statements dictate traffic allowed to cross the VPN; deny statements dictate traffic not allowed to cross the VPN.

This list could go on and on through the many features supported by the Cisco IOS. The point is to understand that ACLs are not just used for security-related configurations. While ACLs can be applied to many different aspects of a router, the CCNA exam focuses on security and NAT.

In general, an access control list is just that: a list. It is an ordered list of statements that the router reads from the top-down. For example, imagine that we had the following list filtering inbound traffic to the S0/0 interface of our router:



Access List 10

Permit host 200.5.9.23
Permit host 195.81.32.13
Permit host 4.2.2.2
Permit 200.1.1.0 0.0.0.255
Deny any

As traffic is sent into your router, it looks at the access list in order, comparing each incoming packet to the list of statements. Once it finds a match, it stops processing the list. For example, if the router in the figure above received a packet from the host 4.2.2.2, it would pass the first two lines in the access list since they do not match the IP address. Once it hits the third line in the access list, it would register a match, permit the packet and stop processing. The "Deny any" statement at the end of the access list would only be reached if the packet did not match any of the first four statements.

That also brings us to focus on the fourth statement. Access lists use **wildcard masks**. For a full discussion on this functionality, refer to the OSPF portion of this guide. The fourth statement in the access list permits any IP addresses that start with 200.1.1; the last octet is not inspected.

Finally, the "Deny any" statement in the access list demonstrates a key point about *all* access lists: if a packet is not permitted somewhere by an explicit statement (entered by an administrator) in the access list, it will be denied. This is known as the implicit deny rule. The final rule of every access list is an implicit deny.

Access lists come in two major flavors: Standard and Extended.

- **Standard Access List -** Permits or denies traffic based on the source IP address only
- **Extended Access List -** Permits or denies traffic based on the source or destination IP address, protocol (such as TCP, UDP, etc…), port number information, time of day and many other criteria

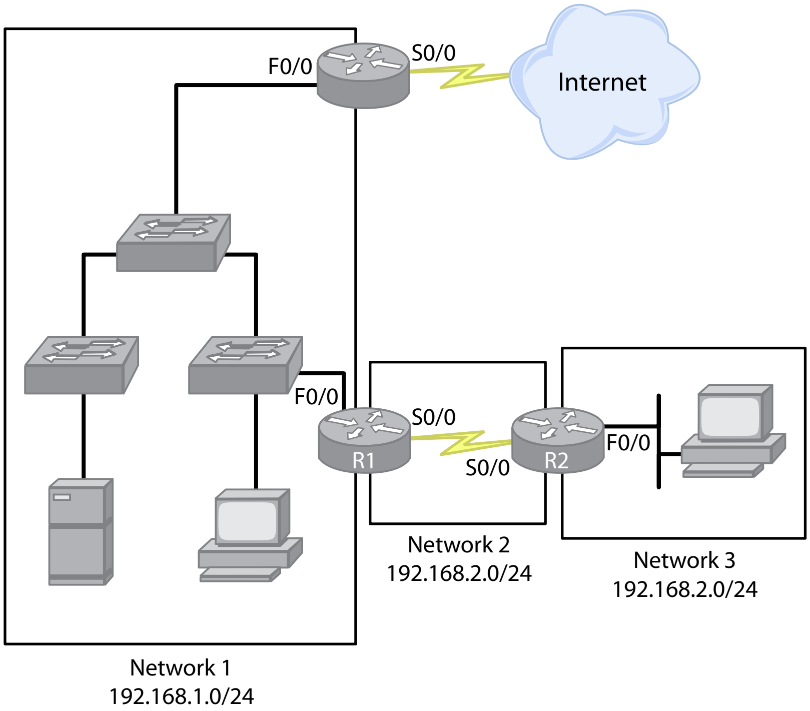
Because of their flexibility, extended access lists are more popular; however, standard access lists are still quite common. When you create an access list on a router, the number of access lists that you create dictates the type of access list:

```
Router(config)#access-list ?
  <1-99>              IP standard access list
  <100-199>           IP extended access list
  <1100-1199>         Extended 48-bit MAC address access list
  <1300-1999>         IP standard access list (expanded range)
  <200-299>           Protocol type-code access list
  <2000-2699>         IP extended access list (expanded range)
  <700-799>           48-bit MAC address access list
  dynamic-extended    Extend the dynamic ACL abolute timer
  rate-limit          Simple rate-limit specific access list
```

As you can see from the previous output, creating an access list identified by numbers 1-99 or 1300–1999 will create a standard access list. Creating an access list identified by numbers 100-199 or 2000-2699 will create an extended access list.

Configuring and Applying Standard ACLs

As mentioned previously, standard ACLs are able to permit or deny traffic based on source IP address information only. To demonstrate the configuration and application of a standard ACL, we will use the following network scenario:



The company represented in the previous figure would like to prevent Network 3 from accessing the Internet. The following syntax will create the list itself:

```
R1(config)#access-list 1 deny 192.168.2.0 0.0.0.255
R1(config)#access-list 1 permit any
```

The above syntax creates access list #1, which is a standard access list (1-99). The first statement denies the 192.168.2.0 subnet access and the second statement permits everyone else. The second statement is necessary because the access list would end up denying all traffic since there is an implicit *deny all* at the end of every access list (it does not need to be added by you). Now, the access list needs to be applied:

```
R1(config)#interface serial 0/0
R1(config-if)#ip access-group 1 out
```

Pay special attention to the router where this configuration was entered and the interface where the command was applied. If we would have applied the standard access list on R2 or R3, it may have denied Network 3 from accessing networks other than just the Internet. This brings us to a fundamental rule of standard ACLs: *always apply standard ACLs closer to the destination*.

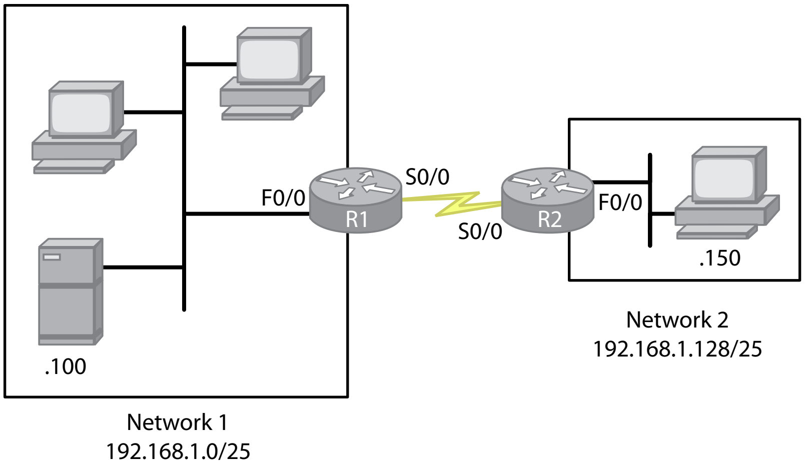
It is also a common practice to apply standard ACLs to the VTY lines of a router. By doing this, you can limit Telnet and SSH access to a specific set of IP addresses. The following is an example of this application:

```
R1(config)#access-list 2 permit 10.1.1.0 0.0.0.255
R1(config)#line vty 0 4
R1(config-line)#access-class 2 in
```

The previous syntax allows only IP addresses from the 10.1.1.0/24 subnet to Telnet or SSH into the router for management purposes.

Configuring and Applying Extended ACLs

Extended ACLs are far more flexible than standard ACLs since they provide a number of additional filtering criteria. To demonstrate the configuration of an extended ACL, consider the following network scenario:



The access list in this scenario must meet the following requirements:

- The host on Network 2 is able to access the server on Network 1 using HTTP, HTTPS and FTP.
- The host on Network 2 should not have any other access to the server on Network 1.
- The host on Network 2 should be allowed access to any other destination.
- Access list statements should be as specific as possible.

The access list to satisfy the above scenario would be as follows:

```
R2(config)#access-list 100 permit tcp host 192.168.1.150 host
192.168.1.100 eq 80
R2(config)#access-list 100 permit tcp host 192.168.1.150 host
192.168.1.100 eq 443
R2(config)#access-list 100 permit tcp host 192.168.1.150 host
192.168.1.100 eq 21
R2(config)#access-list 100 deny ip host 192.168.1.150 host 192.168.1.100
R2(config)#access-list 100 permit ip 192.168.1.128 0.0.0.127 any
```

The first three lines of access list 100 are focused on permitting the host on Network 2 to access the server on Network 1 using HTTP (TCP port 80), HTTPS (TCP port 443) and FTP (TCP port 21). These ports are placed at the end of the access list command since they are destination port numbers. The fourth line of the access list denies the host on Network 1 any other access to the server. Keep in mind that this is accomplished by denying the IP protocol rather than TCP. The final line permits the hosts on Network 2 to access any other destination. The wildcard mask used to permit Network 2 might look a little odd until you look at the subnet mask used by Network 2. The current subnet mask is a /25 in bit notation or 255.255.255.128 in decimal form. To find the wildcard (inverse) mask for a given network, simply subtract the decimal form of the subnet mask from all 255s:

```
  255.255.255.255
- 255.255.255.128
-----------------------
       0.0.0.127
```

Now, the access list needs to be applied:

```
R2(config)#interface fa0/0
R2(config-if)#ip access-group 100 in
```

Again, notice the router and interface to which we applied the access list. As the hosts on Network 2 are coming into the fa0/0, the access list will filter their access. We could have applied this same access list on R2 S0/0, outbound; R1 S0/0, inbound; or R1 Fa0/0, outbound with the same effect. However, any of these locations would require the Network 2 traffic to pass further through the network before it could be dropped. This is not as efficient as R2, Fa0/0 inbound. This leads us to the rule of thumb for extended ACLs: *extended ACLs should always be applied closer to the source*. This is exactly flipped from the rule of standard ACLs. Since the extended ACL can specify destination information, you can be very precise in your restrictions.

Using Named ACLs

More recent IOS versions support a new type of ACL: a named ACL. These ACLs can be standard or extended in nature, but they have the advantage of using a logical name and being editable by sequence numbers. Here's an example:

```
R1(config)#ip access-list standard INTERNET_FILTER
R1(config-std-nacl)#10 permit host 10.5.9.2
R1(config-std-nacl)#20 permit host 192.168.1.59
R1(config-std-nacl)#30 permit host 172.30.100.100
R1(config-std-nacl)#40 permit 192.168.0.0 0.0.0.255
```

Notice that each permit statement has a sequence number in front of it (10, 20, 30 and 40). This allows you to come back and insert lines between the existing statements in the ACL (simply by using sequence numbers such as 15 or 23). You can also remove individual lines from the ACL by using the command **no *<sequence number>*** from the "nacl" (named ACL) configuration mode. These access lists are applied in the same fashion as the numbered access lists.

Verifying ACLs

The best command to verify ACLs is the **show access-list <*number/name*>** command. For example:

```
Router#show access-list INTERNET_FILTER
Extended IP access list INTERNET_FILTER
    10 permit tcp any any eq smtp (1649888 matches)
    20 permit tcp any eq 1723 any (674 matches)
    30 permit icmp any any (5183897 matches)
    40 permit gre any any (6122251 matches)
    50 permit tcp any any eq 1723 (363581 matches)
    60 permit udp any any eq snmp (1341499 matches)
    70 permit tcp any any established (89855906 matches)
    80 permit tcp any host 68.214.31.189 eq domain (1083 matches)
    90 permit udp any host 68.214.31.189 eq domain (15038897 matches)
```

This is an excellent command because it verifies the number of matches (match = packet) that have hit each statement of the access list since the last restart of the router.

You can also verify the access list statements by using the **show run** command. The example below filters the show run output to only include lines containing "access-list"

```
Router#show run | include access-list
access-list 100 deny   ip 192.168.100.0 0.0.0.255 192.168.131.0 0.0.0.255
access-list 100 deny   ip 192.168.121.0 0.0.0.255 192.168.131.0 0.0.0.255
access-list 100 deny   ip 192.168.112.0 0.0.0.255 192.168.131.0 0.0.0.255
access-list 100 deny   ip 192.168.111.0 0.0.0.255 192.168.131.0 0.0.0.255
access-list 100 deny   ip 192.168.101.0 0.0.0.255 192.168.131.0 0.0.0.255
access-list 100 permit ip 192.168.0.0 0.0.255.255 any
```

You can also use the **show run** command to verify the interfaces that have an access list applied to them:

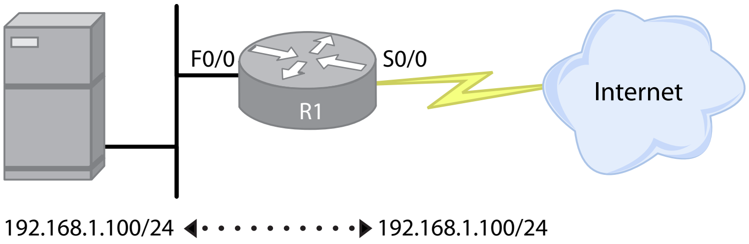
```
Router#show run interface s0/0
Building configuration...

Current configuration : 318 bytes
!
interface Serial0/0/0
 description FW_OUTSIDE
 ip address 68.214.31.189 255.255.255.0
 ip access-group INTERNET_FILTER in
 no ip redirects
 no ip unreachables
```

Understanding NAT Implementations and Configurations

Network Address Translation (NAT) will always be known as the "savior of the IPv4 Internet." In the 1990s, the Internet grew so quickly that there were more devices requiring Internet access than public Internet IP addresses were available. This is where NAT came to the rescue, implementing a feature known as Port Address Translation (PAT) to allow many hosts to share a single public IP address. While this is perhaps the most common and widely-known form of NAT, many other forms exist:

- **Static NAT** - Defines single static translations from one IP address to another
- **Dynamic NAT** - Defines a translation of one pool of IP addresses to another
- **NAT Overload** - Defines a translation of multiple private IP addresses to one or more public IP addresses

Static NAT allows you to use private IP addressing for your internal servers and yet make them available on the public Internet. In the network diagram below, the internal server 192.168.1.100 is mapped to the public IP address 200.50.63.122. Anytime this server accesses the Internet, it is seen as this public IP address. Likewise, anytime someone on the Internet accesses this public IP address, they will translate to this internal server. Static NAT like this should always be filtered using an extended ACL to allow only the necessary ports through to the server.



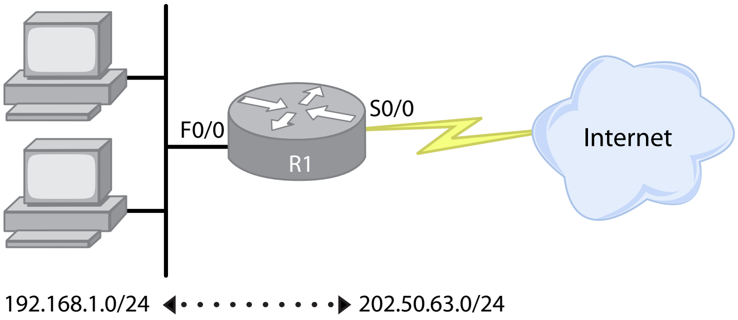
To perform the static NAT configuration show above, you can use the following syntax:

```
R1(config)#ip nat inside source static 192.168.1.100 202.50.63.122
R1(config)#interface fa0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface s0/0
R1(config-if)#ip address 198.53.12.221 255.255.255.0
R1(config-if)#ip nat outside
```

The English translation of the first line of this syntax says, "I want to NAT (**ip nat**) from the inside of my network to the outside (**inside**). The source of this translation will be what I statically define (**source static**). The internal IP address of this translation will be **192.168.1.100**. The external IP address of this translation will be **202.50.63.122**."

After this static translation is in place, we then identify the inside and outside interfaces of the router to the NAT process by using the **ip nat inside** and **ip nat outside** syntax. I purposely used an example where the external IP address of the router was something completely different than the IP address being mapped. I did this to prove the point that the IP address being mapped does not need to be assigned to your router in any way. However, the service provider must be configured to forward traffic to that IP address in the direction of your router (you would accomplish this by purchasing IP addresses from the service provider).

Dynamic NAT allows you to translate a pool of IP addresses. In the diagram below, the group of internal (private) IP addresses is being translated to external (public) IP addresses.



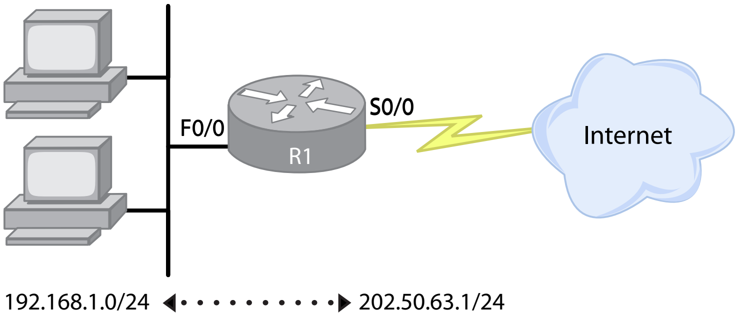
Keep in mind that this is a 1:1 mapping; this is not an example of many hosts sharing a single IP address (known as NAT Overload or PAT). This would keep you from creating more than 200 static translations to map each private IP address to a public IP address. The following configuration would accomplish this:

```
R1(config)#access-list 50 permit 192.168.1.0 0.0.0.255
R1(config)#ip nat pool EXTERNALS 202.50.63.1 202.50.63.254 netmask
255.255.255.0
R1(config)#ip nat inside source list 50 pool EXTERNALS
R1(config)#interface fa0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
```

```
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface s0/0
R1(config-if)#ip address 198.53.12.221 255.255.255.0
R1(config-if)#ip nat outside
```

Notice the changes in the configuration since the static NAT example. Initially, a standard ACL is created which is used to identify (permit) internal IP addresses to be translated. Second, a NAT pool is created which defines the external addresses to be used in the translation. Finally, the third statement puts the two address definitions in action, translating the source IP addresses defined in access list 50 to the external IP addresses defined in the NAT pool EXTERNALS. This form of NAT is least commonly used.

NAT Overload (also known as PAT) allows many private IP addresses to share one or more public IP addresses. In the diagram below, the entire private network (192.168.1.0/24) is sharing the public IP address 202.50.63.1 when accessing the internet.



This is accomplished by using unique source port numbers. If you think back to the discussion of TCP and UDP in the early part of this guide (Network Foundations), the network protocol always generates a random source port number when it sends traffic to a destination port number. NAT Overload uses this source port number to make the external translation unique. Inside the router, a NAT translation table is generated resembling this format:

```
Pro Inside global        Inside local          Outside global
tcp 202.50.63.1:4014     192.168.1.10:4014     206.188.7.218:5190
tcp 202.50.63.1:4446     192.168.1.103:4446    13.37.74.73:80
tcp 202.50.63.1:4447     192.168.1.53:4447     13.37.74.73:80
tcp 202.50.63.1:4473     192.168.1.221:4473    81.68.86.231:443
tcp 202.50.63.1:4474     192.168.1.63:4474     64.245.209.49:443
tcp 202.50.63.1:4475     192.168.1.9:4475      64.245.209.31:443
```

This table is from a real router, but I have modified it slightly for simplicity. The router is translating from **Inside Local** addresses (internal, private IP addresses) to an **Inside Global** (external, public IP address). The Internet host being accessed by the internal user is seen as the **Outside Global** address column.

The following configuration can be used to implement NAT Overload:

```
R1(config)#access-list 50 permit 192.168.1.0 0.0.0.255
R1(config)#ip nat pool EXTERNAL 202.50.63.1 202.50.63.1 netmask
255.255.255.0
R1(config)#ip nat inside source list 50 pool EXTERNAL overload
R1(config)#interface fa0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface s0/0
R1(config-if)#ip address 198.53.12.221 255.255.255.0
R1(config-if)#ip nat outside
```

Notice that this configuration is nearly identical to that of dynamic NAT. The only difference is that the NAT pool consists of only a single IP address and the **ip nat inside source** command is concluded with the keyword **overload**. This instructs the router to use PAT with whatever IP addresses are in the NAT pool. This example shows only a single IP address in the NAT pool being used; however, you can use multiple IP addresses in a larger network which may exhaust the number of ports available on a single IP address.

There are two primary verification commands you can use to ensure NAT is working properly. They are **show ip nat translations** and **show ip nat statistics**.

```
Router#show ip nat translations
Pro Inside global     Inside local       Outside local      Outside global
tcp 202.50.63.1:4014  192.168.1.10:4014  206.188.7.218:80   206.188.7.218:80
tcp 202.50.63.1:4446  192.168.1.13:4446  13.37.74.73:80     13.37.74.73:80
tcp 202.50.63.1:4447  192.168.1.53:4447  13.37.74.73:80     13.37.74.73:80
tcp 202.50.63.1:4473  192.168.1.22:4473  81.68.86.231:80    81.68.86.231:80
tcp 202.50.63.1:4474  192.168.1.63:4474  64.245.209.49:80   64.245.209.49:80
tcp 202.50.63.1:4475  192.168.1.9:4475   64.245.209.31:80   64.245.209.31:80
```

The output above is the same command we issued earlier when discussing the properties of NAT overload. The additional column is the **Outside local** addresses. This is how the internal hosts *see* the external IP addresses. This column will typically mirror the Outside global addresses unless you have implemented the more sophisticated *destination NAT* translations. Destination NAT is not discussed at the CCNA level.

```
Router#show ip nat statistics

Total translations: 2 (0 static, 2 dynamic; 0 extended)
Outside interfaces: Serial0/0
Inside interfaces: FastEthernet 0/0
Hits: 135  Misses: 5
Expired translations: 2
Dynamic mappings:
-- Inside Source
```

```
access-list 50 pool EXTERNAL refcount 2
 pool EXTERNAL: netmask 255.255.255.0
        start 202.50.63.1 end 202.50.63.1
        type generic, total addresses 1, allocated 1 (100%), misses 0
```

This command allows you to see the incoming and outgoing interfaces configured for NAT, along with the number of hits and misses. A hit is when an additional packet is received for a NAT translation already in the table. A miss is when a packet is received that requires a new NAT translation to be made.

Understanding and Implementing IPv6

TCP/IPv6 has been around for a number of years, ready to be deployed worldwide…however, only recently have corporations begun to implement it. NAT has sustained the IPv4 world quite well for quite some time; there has been no business reason to make the upgrade. While the "killer reason" to jump to IPv6 has yet to be found, corporations have started to make the slow move to this new protocol. This is primarily due to government entities moving to IPv6 (especially in areas outside the USA). The motivation to move is more, "Since everyone else is moving, I guess we'll move too" rather than, "IPv6 provides so many great benefits that we should move!"

IPv6 does provide a number of fantastic features; however, very few of them can actually translate into increased revenue for businesses. The number one feature of IPv6 is the increased address space. We have moved from 32-bit addressing in IPv4 to 128-bit hexadecimal addressing in IPv6. The following is an example of an IPv6 address:

Sample IPv6 address: 2001:0050:0300:0000:0000:0ab4:1e2b:98aa

The new IPv6 addresses have eight, 16-bit octets of four hexadecimal characters each. Because these addresses are quite cumbersome, the following two rules were created to shorten them:

- **Rule 1:** Groups of consecutive zeros can be represented with a double colon (::); however, the double colon can only be used once in each address.
    - Ex: 2001:0050:0300::0ab4:1e2b:98aa
- **Rule 2**: Leading zeros in the address can be dropped.
    - Ex: 2001:50:300::ab4:1e2b:98aa

While this still leaves the address quite a bit longer than an IPv4 address, it's slightly more manageable after these two rules are applied.

Because the number of addresses available in IPv6 will be so large, each network device may have multiple IP address assignments. These assignments can fall into three categories of addresses:

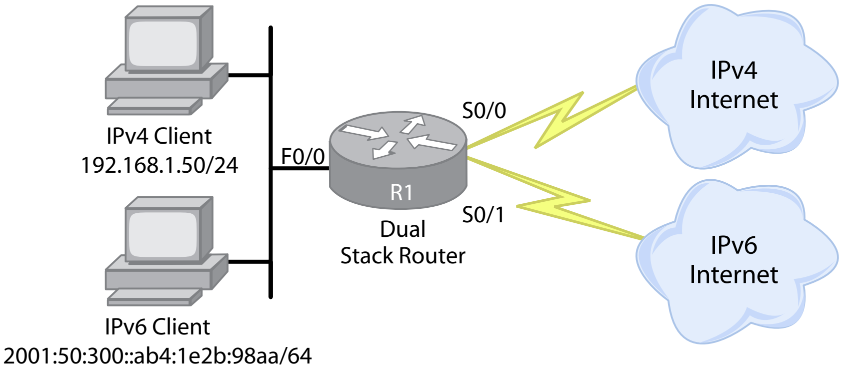
- **Link Local Addresses** - These addresses are very similar to the 169.254.x.x/16 address space of IPv4. They are addresses generated by the local device that allows them to communicate on the local (layer 2) network.
- **Unique / Site Local Addresss** - These addresses are similar to the private address ranges of IPv4. They are used to communicate within an organization. Originally these were called "Site Local Addresses" but more recently have been changed to "Unique Local Addresses."

- **Global Addresses** - These addresses are similar to the public address ranges of IPv4. They will be routable on the new Internet which some have termed "Internet2." \

Along with the new address types, there are new types of communication in IPv6:

- **Unicast -** Just like IPv4, one-to-one communication.
- **Multicast -** One-to-group communication. This messaging type has replaced broadcasts (there are no broadcasts in IPv6).
- **Anycast -** One-to-closest communication. Allows multiple devices to be assigned the same IP address. When this address is requested by a client, the closest device will respond.

The changeover to IPv6 will not be a "drop dead date." That is, we're not going to set a date where everyone must be running IPv6 or be cut off from the world. Rather, it will be a slow transition of routers running both IPv4 and IPv6. Both IPv4 and IPv6 Internet connections will still be made available for quite some time in the future. The following diagram shows how many organizations will slowly migrate to the new IPv6:



Before we look at the configuration of the dual stack router, note the addressing of the IPv6 client. The client has the address 2001:50:300::ab4:1e2b:98aa/64. The /64 subnet mask means the first half of the address represents the network (remember, each octet in these new IPv6 addresses are 16 bits each) and the second half of the address represents the host on the network. Since this is a shortened address, here are the full representations:

- **IPv6 Network Portion**: 2001:0050:0300:0000
- **IPv6 Host Portion**: 0000:0ab4:1e2b:98aa

Now, the router configuration for the LAN interface:

```
R1(config)#ipv6 unicast-routing
R1(config)#interface fa0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#ipv6 address 2001:0050:0300::1/64
```

As of right now, the CCNA exam just requires you to know the foundations of IPv6. You can expect this topic to grow on all Cisco exams as the IPv6 protocol becomes more widespread.

Domain 11 - WAN Connections

Inter-Office Links

One of the major purposes of routers is to tie your offices together. The router gives you access to some type of WAN connection that will bridge your offices together. The type of WAN connection you choose will offer different advantages and disadvantages. The goal of this next section is to discuss the following WAN-related topics:

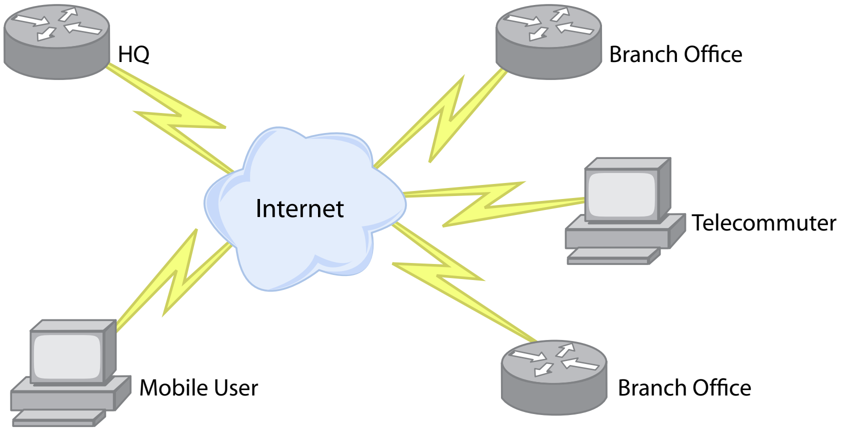
- Understanding VPN connections
- Understanding and configuring Leased Line WAN connections
- Understanding and configuring Frame Relay WAN connections

Understanding VPN Connections

The goal of a Virtual Private Network (VPN) is simple: transmit data between devices securely using a public network as a transport. How it accomplishes this feat is actually quite complex. Regardless of their complexity, VPN connections are becoming more and more popular around the world as a way of connecting remote hosts and offices together. VPNs offer the following two major benefits:

- **Low Cost -** It is much cheaper to purchase an Internet connection at each of your office locations rather than dedicated WAN links. The Internet can then be used not only for normal Internet access but as a backbone for interoffice communication.
- **Availability and Scalability -** Internet connections are widely available. Because of this, your VPN is able to scale to encompass multiple offices, telecommuters and home office users with very little increase in cost.

Typically, anytime you see a VPN network diagram, the core of the focus will be the Internet:



This diagram leads us to understand the three major types of VPN connections:

- **Site-to-Site -** VPN linking two offices together. Routers or firewall equipment handle VPN connections rather than end users.
- **Remote Access -** VPN linking remote user to the corporate network. The end-user usually handles the VPN connection through some installed software.
- **SSL / Web VPN -** Newest type of VPN which allows you to tunnel traffic through an existing SSL connection which is typically initiated through a web browser. SSL/Web VPN connections are commonly categorized as a new style of remote access VPN.

As you might imagine, sending corporate data over the Internet requires plenty of security considerations. Because of this, all Cisco VPN connections use a protocol known as IPSec. Like TCP/IP, IPSec is actually a suite of protocols which focuses on securing the connection. The IPSec protocol addresses each of these security arenas:

- **Authentication -** This portion of the IPSec protocol verifies the source of each packet. This is typically done through pre-shared keys (a simple, but not as secure approach) or certificates (a more complex, more secure approach).
- **Encryption -** IPSec handles scrambling the data before it is sent over the Internet so it is not understandable to anyone but the intended receiver. Common encryption algorithms used over VPN connections are DES, 3DES and AES (listed in order of weakest to strongest).
- **Data Integrity -** IPSec needs to ensure data does not change between the sending and receiving devices. For this, the MD5 (weaker) and SHA-1 (stronger) hashing algorithms are used.
- **Anti-Replay Protection** - IPSec uses sequence numbers to ensure data is not duplicated or sent at a later time or date. If duplicate or late data is received, IPSec will block it.

Understanding and Configuring Leased Line WAN Connections

While leased line connections are still one of the most prevalent in the industry, there is a slow shift to newer technologies. Leased lines are notorious for their reliability, stability and price tag. It is usually the absorbent price tag that causes IT managers to look elsewhere.

A leased line connection is a private, point-to-point link between offices. The private links consist of one or more DS0 channels. Each DS0 is the equivalent of 64 Kbps. Once you reach 24 DS0s, you will achieve a T1 speed (1.544 Mbps). While physical implementations of leased lines will vary, you will always need to choose between two data link WAN protocols:

- **High-Level Data Link Control (HDLC) -** All Cisco routers use the HDLC protocol by default on Serial interfaces. Cisco's version of HDLC is proprietary, so it is useful only when connecting between two Cisco routers. HDLC has very little configuration and features associated with it.
- **Point-to-Point Protocol (PPP) -** PPP is the industry standard protocol for many types of WAN connections including leased lines. You can use this to connect a Cisco router to any other brand of router. PPP supports many more features than HDLC.

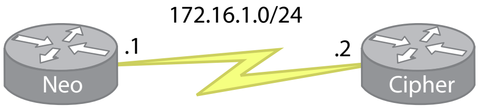
The configuration of HDLC is not worth mentioning since…there is none! Cisco routers run HDLC by default on all Serial interfaces, so the WAN connection operates simply by plugging each router into the leased line connection and assigning IP addresses. You can verify your router is running the HDLC protocol by using the **show interfaces** command:

```
Router#show interfaces serial 1
Serial1 is up, line protocol is up
  Hardware is HD64570
  Internet address is 172.16.1.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:03, output 00:00:05, output hang never
  Last clearing of “show interface” counters 00:00:07
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/3/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     2 packets input, 88 bytes, 0 no buffer
     Received 2 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     2 packets output, 90 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

The PPP protocol supports the following four features beyond basic connectivity:

- **Authentication** - Verifies that the remote device that is connecting provides the necessary authentication credentials.
- **Compression** - Compacts the data being sent to save WAN bandwidth.
- **Callback** - Disconnects the calling device and calls them back at a pre-defined number.
- **Multilink** - Binds multiple physical connections into a single logical link. For example, multilink can combine the bandwidth from two T1 lines into a single, 3 Mbps connection.

The CCNA exam focuses solely on the authentication aspects of PPP. I will use the following network diagram to demonstrate the configuration of PPP authentication:



The initial configuration of PPP is simple:

```
Neo(config)#interface S1
Neo(config-if)#encapsulation ppp

Cipher(config)#interface S1
Cipher(config-if)#encapsulation ppp
```

That's it! We've now converted from HDLC to PPP. Before we add authentication to the mix, we need to highlight the two types:

- **Password Authentication Protocol (PAP)** - PAP was the initial authentication protocol used with PPP. It sends the username and password in clear-text, which is quite dangerous. PAP is rarely used in today's modern environments.
- **Challenge Handshake Authentication Protocol (CHAP) -** CHAP uses a hashed password mechanism, which prevents passwords from being easily discovered between end devices. Most PPP deployments will use CHAP authentication.

To configure CHAP authentication for the previous network scenario, use the following syntax:

```
Neo(config)#username Cipher password cisco
Neo(config)#interface serial 1
Neo(config-if)#ppp authentication chap

Cipher(config)#username Neo password cisco
Cipher(config)#interface serial 1
Cipher(config-if)#ppp authentication chap
```

The previous syntax is known as a two-way CHAP authentication since each router authenticates the other. The first command of the syntax creates a user account for the opposite router. The Neo router expects another router to provide the username Cipher (Cisco routers use their hostname as their username, by default) and a password of "cisco". The Cipher router expects another router to provide the username Neo and a password of "cisco". When performing two-way CHAP authentication, the passwords *must be the same* between the two routers. This is because of the way CHAP handles the password hashing.

Once we have configured PPP, we can verify its operation by using the same **show interfaces** command:

```
Neo#show interfaces serial 1
Serial1 is up, line protocol is up
  Hardware is HD64570
  Internet address is 172.16.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  LCP Open
  Open: IPCP, CDPCP
  Last input 00:00:01, output 00:00:01, output hang never
  Last clearing of “show interface” counters 00:11:40
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     195 packets input, 14191 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     187 packets output, 13210 bytes, 0 underruns
     0 output errors, 0 collisions, 15 interface resets
     0 output buffer failures, 0 output buffers swapped out
     29 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

The highlighted line “LCP Open” shows that the PPP Link Control Protocol (LCP) is operational. LCP is responsible for negotiating all PPP features, including authentication. Below that, we can see that the IP Control Protocol (IPCP) and CDP Control Protocol (CDPCP) are allowing these protocols to work over the PPP link. These are both known as PPP Network Control Protocols (NCPs).

Authentication is one of the common troubleshooting areas for PPP. The best command to troubleshoot authentication issues is **debug ppp authentication**.

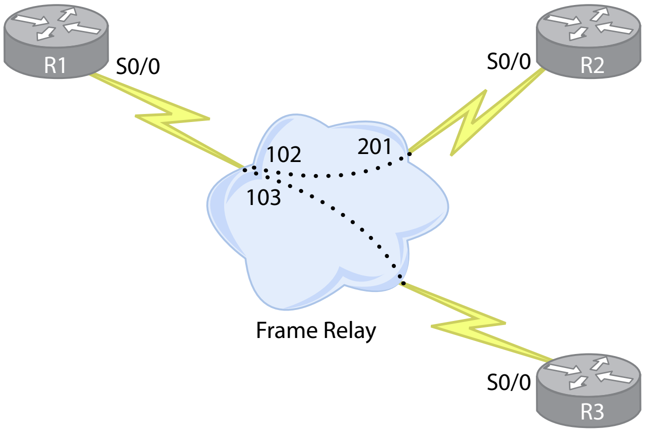
```
Neo(config-if)#do debug ppp authentication
PPP authentication debugging is on
Neo(config-if)#no shutdown
00:26:38: Se1 PPP: Using default call direction
00:26:38: Se1 PPP: Treating connection as a dedicated line
00:26:38: %LINK-3-UPDOWN: Interface Serial1, changed state to up
00:26:38: Se1 CHAP: O CHALLENGE id 2 len 24 from "Neo"
00:26:38: Se1 CHAP: I CHALLENGE id 10 len 27 from "Cipher"
00:26:38: Se1 CHAP: O RESPONSE id 10 len 24 from "Neo"
00:26:38: Se1 CHAP: I RESPONSE id 2 len 27 from "Cipher"
00:26:38: Se1 CHAP: O SUCCESS id 2 len 4
00:26:38: Se1 CHAP: I SUCCESS id 10 len 4
```

As you can see, the three phases of CHAP are Challenge, Response and a Success/Failure message. In this case, our authentication succeeded.

Understanding and Configuring Frame Relay WAN Connections

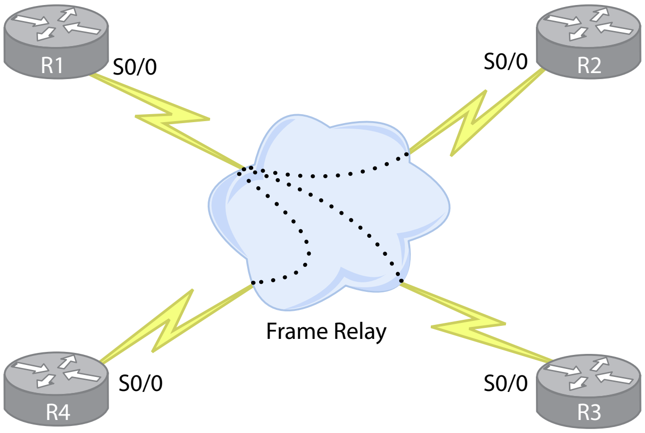
Frame Relay is one of the most common *Packet Switched* WAN connections in existence. Frame Relay allows you to get a good amount of bandwidth for a decent price. This is accomplished by creating a shared bandwidth cloud on the part of the service provider. If you aren't using your bandwidth, chances are, someone else is!

Half of the challenge in understanding the world of Frame Relay is in understanding the terminology. Take the following sample network diagram:

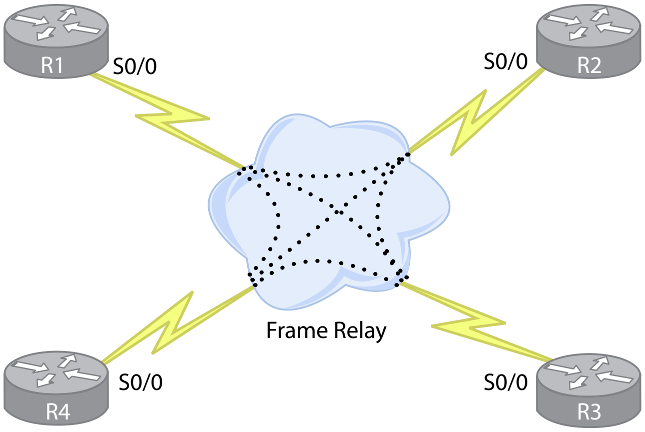


- **Permanent Virtual Circuit (PVC) -** The dotted lines through the Frame Relay cloud represent PVCs. These are circuits established through the service provider's network that links your locations together. Each PVC has a recurring monthly cost based on the amount of bandwidth desired.
- **Data Link Connection Identifier (DLCI) -** The numbers in the cloud represent DLCI numbers. These are the data link-layer addressing that Frame Relay uses.
- **Committed Information Rate (CIR) -** The CIR is the lowest average speed the service provider commits to give you. Many service providers will allow you to burst above your CIR if the bandwidth is available.
- **Local Access Rate (LAR) -** The maximum physical speed your connection supports to the Frame Relay service provider.
- **Local Management Interface (LMI) -** The protocol used between you and the service provider to manage the Frame Relay connections.
- **Inverse Arp (InARP) -** A "backwards" ARP message that attempts to determine a device's IP address based on its data link layer address.

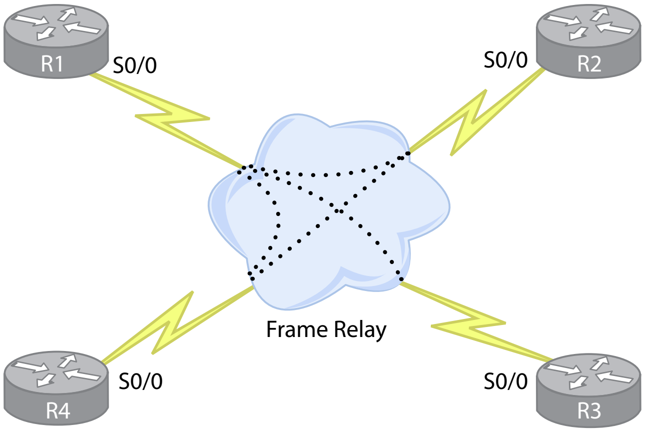
Now that the terms are on the table, here's how Frame Relay works. When you sign up for a Frame Relay connection, you will dictate how many offices will be linked together through the Frame Relay network. You will then need to choose the type of Frame Relay topology you would like to assemble. Here are three styles:



- **Hub and Spoke -** All Frame Relay circuits come in through one central router (the hub). This is the cheapest, but least redundant way to deploy Frame Relay.

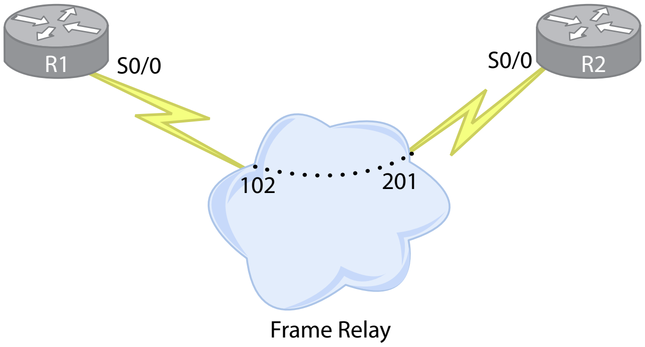


- **Full Mesh -** Every site has a PVC directly to each of the other sites. This is the most redundant configuration but is expensive.



- **Partial Mesh -** Critical sites have multiple connections to each other; non-critical sites have limited PVC connections.

Once you have selected the number of PVCs and their locations, the service provider will assign you DLCI numbers. These DLCIs are the addressing used to connect between offices. The concept of DLCIs can be somewhat confusing since the addressing works backwards when compared to what we are used to seeing. For example, take the following PVC between R1 and R2:

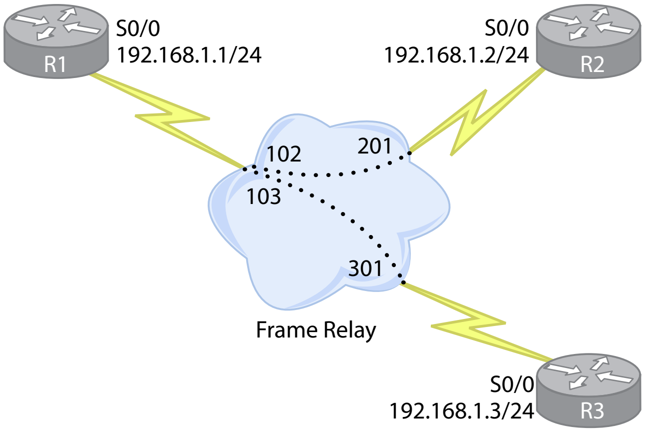


We would assume that if R1 was sending to R2, it would come from the source DLCI 102 and transmit to the destination DLCI 201. Rather, Frame Relay works exactly the opposite: R1 sends to a destination DLCI 102 which is transmitted through the Frame Relay service provider and exits the R2 WAN connection on DLCI 201. I often compare this to flying to a location from an airport. You might leave out of gate B5, but arrive in gate C13. The address "changes" in mid-air, as you are flying through the service provider cloud.

For this reason, people often refer to DLCI numbers as "local DLCIs." This is because the router will transmit data to its local DLCI number which the service provider will translate to the remote DLCI information when the data exits the cloud.

To configure Frame Relay, you first need to plan your sub-interface strategy. Sub-interfaces are often used in Frame Relay since a single physical interface can connect to many destinations. You have two choices in Frame Relay sub-interfaces:

- **Point-to-point -** Point-to-point sub-interfaces are useful when each PVC is to be designated as a separate IP subnet. This design makes the Frame Relay cloud feel like a group of leased line circuits between locations.
- **Multipoint -** Multipoint sub-interfaces are useful when all PVCs share the same IP subnet. This design makes the Frame Relay cloud feel like an Ethernet switch. Be careful - multipoint designs cause issues with many routing protocols, especially distance vector routing protocols which use Split Horizon loop prevention.



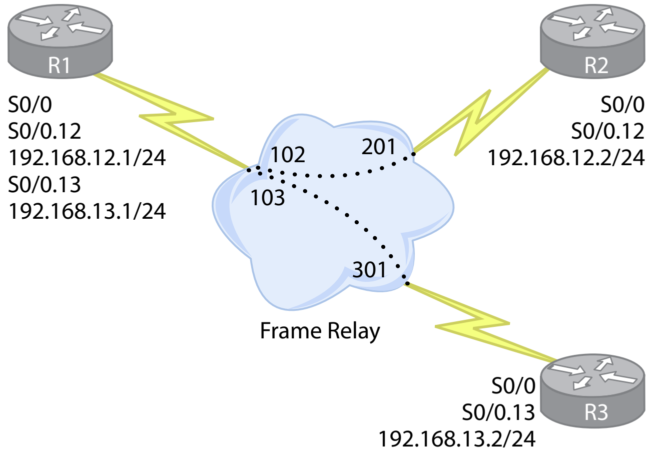
This design is something that you choose - it is not dictated by the Frame Relay service provider. I'll show both designs using the same Frame Relay topology:

Multipoint Configuration

The previous figure illustrates a Frame Relay multipoint configuration. You can tell this because all routers are sharing the same IP subnet. To configure this design, use the following syntax:

```
R1(config)#interface serial 0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#encapsulation frame-relay
R1(config-if)#frame-relay map ip 192.168.1.2 102 broadcast
R1(config-if)#frame-relay map ip 192.168.1.3 103 broadcast

R2(config)#interface serial 0/0
R2(config-if)#ip address 192.168.1.2 255.255.255.0
R2(config-if)#encapsulation frame-relay
R2(config-if)#frame-relay map ip 192.168.1.1 201 broadcast
R2(config-if)#frame-relay map ip 192.168.1.3 201 broadcast

R3(config)#interface serial 0/0
R3(config-if)#ip address 192.168.1.3 255.255.255.0
R3(config-if)#encapsulation frame-relay
R3(config-if)#frame-relay map ip 192.168.1.1 301 broadcast
R3(config-if)#frame-relay map ip 192.168.1.2 301 broadcast
```

The key command in a multipoint configuration is the **frame-relay map** command. This commands maps a remote IP address to the local DLCI number used to reach the remote IP address. Take the following command on R1: **frame-relay map ip 192.168.1.2 102 broadcast**. This command says "to reach the remote IP address 192.168.1.2, use DLCI 102…and allow broadcasts to be sent on this circuit." The **broadcast** keyword allows routing protocol updates to be sent by R1 to R2, in this case. If you left the **broadcast** keyword off, routing protocols would not operate over the Frame Relay cloud.



Point-to-Point Configuration

This figure demonstrates a point-to-point configuration. As you can see, each PVC is assigned its own IP subnet. The sub-interface numbers you choose can be anything you want.

```
R1(config)#interface serial 0/0
R1(config-if)#encapsulation frame-relay
R1(config-if)#exit
R1(config)#interface serial 0/0.12 point-to-point
R1(config-if)#ip address 192.168.12.1 255.255.255.0
R1(config-if)#frame-relay interface-dlci 102
R1(config-if)#exit
R1(config)#interface serial 0/0.13 point-to-point
R1(config-if)#ip address 192.168.13.1 255.255.255.0
R1(config-if)#frame-relay interface-dlci 103

R2(config)#interface serial 0/0
R2(config-if)#encapsulation frame-relay
R2(config-if)#exit
R2(config)#interface serial 0/0.12 point-to-point
R2(config-if)#ip address 192.168.12.2 255.255.255.0
R2(config-if)#frame-relay interface-dlci 201

R3(config)#interface serial 0/0
R3(config-if)#encapsulation frame-relay
R3(config-if)#exit
R3(config)#interface serial 0/0.12 point-to-point
R3(config-if)#ip address 192.168.13.2 255.255.255.0
R3(config-if)#frame-relay interface-dlci 301
```

While it looks like more configuration, the point-to-point Frame Relay design is more optimal than a multipoint since it does not have any issues with routing protocols.

You can verify the Frame Relay configuration by using the following three commands:

- **show frame-relay lmi -** Verifies the LMI signaling between you and the service provider. The LMI should be auto-detected by any router manufactured within the last decade; however, if you have an exceptionally old router, you may need to hardcode the LMI type under the interface configuration mode by typing **frame-relay lmi-type <*type*>.** The type should match whatever the service provider is using; your choices are **cisco**, **ansi**, or **q933a**.
- **show frame-relay map** - Verifies the DLCI-to-IP address mappings.
- **show frame-relay pvc** - Displays statistics (packets sent/received, bytes sent/received, etc…) about each PVC established through the Frame Relay service provider.

Domain 12 - Managing Cisco Devices

CDP, Remote Access and IOS File Systems

There are some topics that apply to all Cisco network devices alike. In this section, we will discuss the following common aspects:

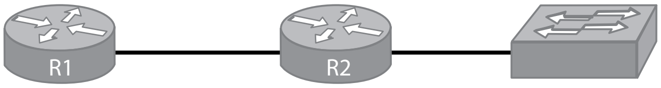
- Understanding the Cisco Discovery Protocol
- Managing remote access between Cisco devices
- Understanding and using the IOS file system

Understanding the Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is a simple protocol that allows you to see other, directly-connected Cisco network devices. For example, you might be accessing a switch in an organization. Unfortunately, the organization does not have an up-to-date, accurate network diagram. By typing the command **show cdp neighbors**, you can get a view of the Cisco devices that are connected to this switch. CDP provides the following information about each device:

- Hostname
- IP address
- Local & remote port connections
- Device model
- IOS version

This can be very handy when attempting to build a network diagram (it sure beats tracing cables). To demonstrate CDP, take the following network diagram:



Based on this network diagram, we do not know what ports are connected to each router, what IP address information each device has or even what model of equipment is in use. To demonstrate CDP, I'll connect to the console port of R1:

```
R1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID    Local Intrfce    Holdtme    Capability  Platform  Port ID
R2           Ser 1            176        R           2520      Ser 1
```

By typing the **show cdp neighbors** command, I am able to see that R1 has a connection to R2 from its Serial 1 interface (local interface). The remote router is a 2520 (platform). R1 is connected to R2's Serial 1 interface (port ID). If we want more information, we can add the **detail** argument:

```
R1#show cdp neighbors detail
-------------------------
Device ID: R2
Entry address(es):
  IP address: 172.16.1.2
Platform: cisco 2520,  Capabilities: Router
Interface: Serial1,  Port ID (outgoing port): Serial1
Holdtime : 144 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-JS-L), Version 12.2(10a), RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-2002 by cisco Systems, Inc.
Compiled Tue 21-May-02 10:55 by pwade

advertisement version: 2
```

From this output, we are able to determine that R2 has the IP address 172.16.1.2 and is running the 12.2(10a) version of the IOS.

After filling this information in on our diagram, we can then telnet to R2 and execute the same command:

```
R1#telnet 172.16.1.2
Trying 172.16.1.2 ... Open

R2>show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID   Local Intrfce   Holdtme    Capability  Platform  Port ID
Switch      Eth 0            171       S I         Cisco WS-CFas 0/4
R1          Ser 1            129       R           2500      Ser 1
```

From the CDP neighbor table on R2 we are able to see the information about the switch connection along with the mirrored information about R1. If we wanted, we could follow this command up with the **detail** keyword to get the specific IP address and IOS version information.

Because CDP does provide sensitive network information, some organizations choose to turn it off. There are two ways to accomplish this:

- **Globally: no cdp run -** Typing the command **no cdp run** from global configuration mode disables CDP on the entire device.
- **Per-Interface: no cdp enable** - Typing the command **no cdp enable** under interface configuration mode prevents CDP from running on just one interface.

Managing Remote Access Between Cisco Devices

Moving between Cisco devices using the Telnet and SSH protocols is quite common. However, once you telnet to one too many devices, it's easy to get lost. Cisco has provided many ways to manage your remote sessions as you move around the network.

As we saw from the previous CDP section, it's easy to telnet between Cisco devices. All you need to do is type **telnet <*remote IP address*>** from a user or privileged mode prompt. Unbelievably, the **telnet** keyword is optional. You can just type the remote device IP address and the IOS will assume you are trying to telnet.

Once you have accessed the remote device, there are two ways to get back: ending your telnet session or suspending your telnet session. To end your telnet session, simply type the **exit** command from a user or privileged mode prompt:

```
R1#telnet 172.16.1.2
Trying 172.16.1.2 ... Open

R2>exit

[Connection to 172.16.1.2 closed by foreign host]

R1#
```

Suspending your telnet session is quite handy when you are jumping back and forth between devices often. The suspend command is quite tricky. Press **ctrl + shift + 6** at the same time, let go, then type **x** on the keyboard:

```
R1#telnet 172.16.1.2
Trying 172.16.1.2 ... Open

R2>    --------- <ctrl + shift + 6> then x entered here ---------
R1#
```

Notice from the previous syntax that pressing the **ctrl + shift + 6,** then **x** combination immediately moves us back to R1. The telnet session is still active, but it has been suspended for the time being. If I would like to see the telnet sessions open from my router, I can do so with the **show sessions** command:

```
R1#show sessions
Conn Host                Address             Byte  Idle    Conn Name
*  1 172.16.1.2          172.16.1.2          0     1       172.16.1.2
```

R1 shows a single session (connection 1) open to R2. If I want to resume the telnet session, I can type **resume 1** (where 1 is the connection number shown in the first column):

```
R1#resume 1
[Resuming connection 1 to 172.16.1.2 ... ]

R2>
```

If you wanted to create multiple telnet sessions from R1 to other devices, you could quickly jump around between them using these skills.

Understanding and Using the IOS File System

Every Cisco device has miniature file system that it uses to manage access to many key components. You should be familiar with the following file system components:

- **Read-Only Memory (ROM) -** The read-only memory contains the power of self-test (POST) and ROM monitor, which is a small operating system used for IOS and password recovery should you accidentally delete the IOS or forget your password.
- **Flash -** The flash contains the IOS in compressed format. When the router boots, it decompresses the IOS into RAM.
- **Non-Volatile RAM (NVRAM) -** The NVRAM is a small piece of non-volatile memory (doesn't go away when the device reboots) used to store the startup configuration.
- **Random Access Memory (RAM) -** The high-speed, volatile memory component used to hold the IOS and running configuration while the device is running.

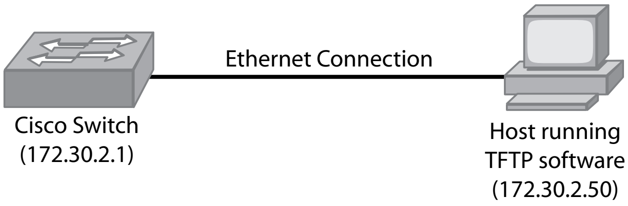
By this point, you are familiar with the method used to save your configuration: **copy running-config startup-config**. This transfers the contents of the running configuration in RAM into NVRAM. The **copy** command can be used to copy between other memory components as well. For example, we could copy the running configuration into flash by typing the following:

```
CAT3550#copy running-config flash:
Destination filename [running-config]? running-backup

8161 bytes copied in 1.484 secs (5499 bytes/sec)
CAT3550#show flash

Directory of flash:/

   2  -rwx 4246296   Mar 1 1993 01:02:29 +00:00   c3550-i5q3l2-mz.121-
22.EA1.bin
   4  -rwx    8161   Sep 9 1993 20:13:06 +00:00   running-backup
   5  -rwx     976   Apr 17 1993 17:48:36 +00:00  vlan.dat
 360  -rwx       0   Mar 1 1993 05:12:23 +00:00   env_vars
 361  -rwx      75   Mar 1 1993 05:12:23 +00:00   system_env_vars
 359  -rwx    8198   Aug 12 1993 18:54:12 +00:00  config.text

15998976 bytes total (3417088 bytes free)
```

Notice the highlighted line from the **show flash** output: we have now stored a backup of our running configuration in flash!

There will be times when you want to transfer files to and from your Cisco devices. This is typically used for IOS upgrades or backup. Moving files between PCs and Cisco devices requires the use of a TFTP server. TFTP server software is widely and freely available on the Internet. The best TFTP software to date is TFTPd32.

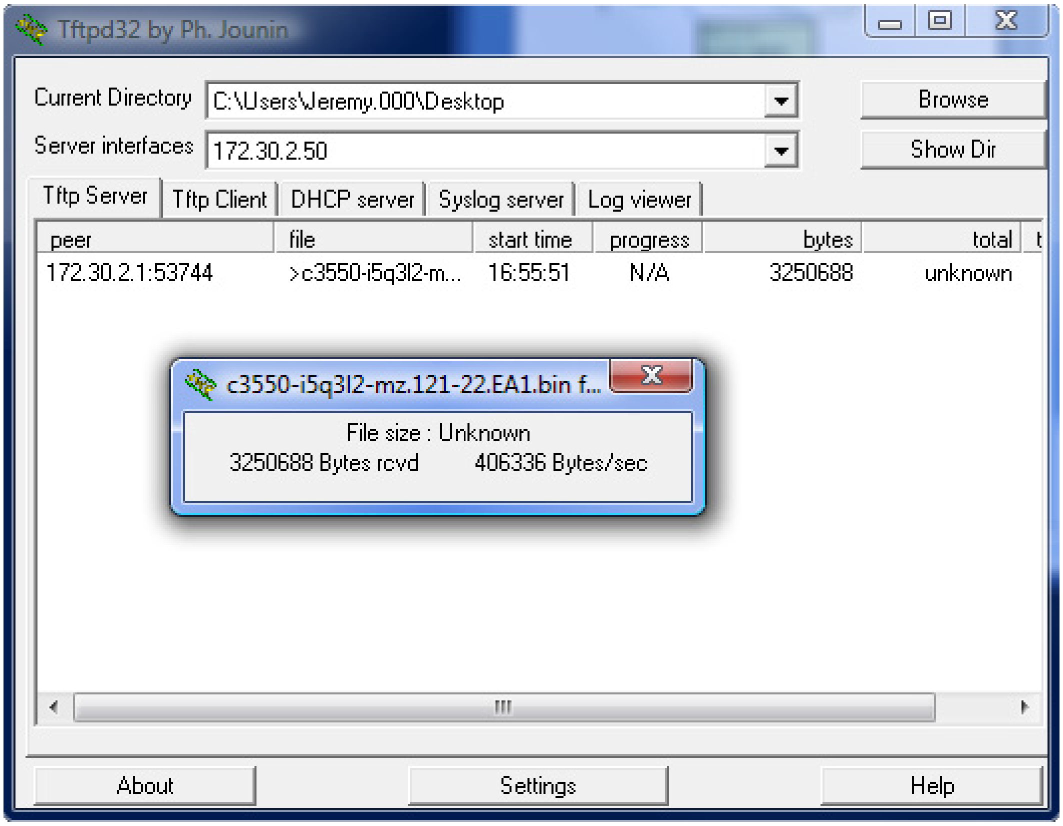
Once you have downloaded and installed TFTP software on your PC, moving files is easy. Take the following network diagram:



While this diagram shows the devices directly connected, this is not a requirement. As long as each device is able to reach the other's IP address, TFTP will work just fine. After you have opened the TFTP software on the host, you can back up the Cisco IOS by using the following syntax:

```
CAT3550#show flash

Directory of flash:/

   2  -rwx 4246296   Mar 1 1993 01:02:29 +00:00  c3550-i5q3l2-mz.121-
22.EA1.bin
   4  -rwx    8161   Sep 9 1993 20:13:06 +00:00  running-backup
   5  -rwx     976  Apr 17 1993 17:48:36 +00:00  vlan.dat
 360  -rwx       0   Mar 1 1993 05:12:23 +00:00  env_vars
 361  -rwx      75   Mar 1 1993 05:12:23 +00:00  system_env_vars
 359  -rwx    8198  Aug 12 1993 18:54:12 +00:00  config.text

15998976 bytes total (3417088 bytes free)
CAT3550#copy flash: tftp:
Source filename []? c3550-i5q3l2-mz.121-22.EA1.bin
Address or name of remote host []? 172.30.2.50
Destination filename [c3550-i5q3l2-mz.121-22.EA1.bin]?
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
4246296 bytes copied in 10.652 secs (398638 bytes/sec)
```

The **show flash** command was executed to retrieve the filename of the IOS we wanted to back up. The exclamation points will begin appearing as the file is being copied. You can also verify the copy progress from the TFTP server window:



Current Directory C:\Users\Jeremy.000\Desktop

Server interfaces 172.30.2.50

peer	file	start time	progress	bytes	total	t
172.30.2.1:53744	>c3550-i5q3l2-m...	16:55:51	N/A	3250688	unknown	

c3550-i5q3l2-mz.121-22.EA1.bin f...

File size : Unknown

3250688 Bytes rcvd 406336 Bytes/sec

The **copy** command can also be used to upgrade or restore the IOS onto a device just by reversing the command: **copy tftp: flash:**.

Practice Questions

Chapter 1 Describe How a Network Works

1. Which of the following pieces of information does a BPDU contain?
Select the three best answers.

□ A. IP address of the switch.
□ B. The hostname of the switch.
□ C. The Bridge ID of the sending switch.
□ D. The spanning tree path cost to the root bridge.
□ E. Hello, forward-delay, and max-age protocol timers.

2. What is the main function of the OSI physical layer?
Select the best answer.

○ A. It puts signals on the wire.
○ B. It provides media access and frame format.
○ C. It provides guaranteed delivery and application communication.
○ D. It provides addressing and routing.
○ E. It negotiates data format and representation.

3. Which of the following statements are correct about network segmentation with bridges?
Choose TWO.

□ A. Bridges act as intelligent network devices, forwarding traffic based on layer 3 addresses.
□ B. Bridges are layer 2 "filters" that keep local traffic local, and forward all other traffic.
□ C. Bridges allow open traffic flow, and perform no segmentation.
□ D. When a destination address is not known to a bridge, it "floods" the traffic to all segments.

4. Which statements about half-and full-duplex are true?
Choose THREE.

□ A. Half-duplex transmissions are prone to collisions.
□ B. All network devices support both transmission modes.
□ C. Hub-based networks must use half-duplex mode in order to detect collisions.
□ D. Full-duplex links have their collision detect circuits disabled.
□ E. Half-duplex mode allows 100% efficiency over Ethernet.

5. When one host transmits data across a network to another host, information is processed through the OSI stack. Align the OSI layers in the correct order in which a destination host processes network traffic with the first layer processed on the top.

Session	
Network	
Presentation	
Transport	
Application	
Physical	
Data Link	

Chapter 2 Configure, Verify and Troubleshoot a Switch with VLANs and Interswitch Communications

1. What information is included in BPDUs sent by bridges and switches as part of their STP operations?
Select the four best answers.

□ A. Values for the hello, forward delay, and max-age protocol timers
□ B. The bridge ID of the sending switch
□ C. The spanning-tree path cost to the root
□ D. Its DNS name
□ E. The ID of the root bridge

2. What might be the reason for a switch port being in STP blocking state?
Select the best answer.

○ A. The port (interface) hardware is faulty.
○ B. Too many collisions have occurred on the segment to which the port is connected.
○ C. The port is part of a FastEtherchannel.
○ D. A data-link loop has been detected in the Internetwork.
○ E. The port has detected a duplicate MAC address in the Internetwork.

3. Your backup server is experiencing poor network performance issues, and your nightly backups are not completing. The server is connected via a 100MB Ethernet to the switch, and you suspect there is a duplex mismatch. What commands would you use to troubleshoot the issue?
Select THREE.

❑ A. show interfaces
❑ B. show interface counters
❑ C. show log
❑ D. show log history
❑ E. show interfaces status

Chapter 3 Implement an IP Addressing Scheme and IP Services to Meet Network Requirements in a Medium-size Enterprise Branch Office Network

1. You want to manually assign a router ID to an OSPF router. Which commands can you use to influence the router ID selection?
Select the three best answers.

❑ A. router ospf 100
❑ B. area 0 virtual-link
❑ C. interface serial 0 ip address 192.168.1.1
❑ D. interface loopback 0 ip address 192.168.1.1
❑ E. router-id

2. You are designing an internetwork. Due to the variety of technologies and connectivity methods in use, you need to select routing protocols that support unequal cost path load balancing. Which protocols will you select?
Select the two best answers.

❑ A. RIP v.1
❑ B. RIP v.2
❑ C. IGRP
❑ D. EIGRP
❑ E. OSPF

3. You are configuring OSPF on a Cisco router. Interface FastEthernet 0/0 belongs to the 192.168.1.0/24 IP subnet. Your router has multiple interfaces. You want only this interface to be included in the OSPF routing process. Which command would you issue?
Select the best answer.

❍ A. network 192.168.1.0 255.255.255.0 area 0
❍ B. network 192.168.1.0 255.255.255.255 area 0
❍ C. network 192.168.1.0 0.0.0.255 area 0
❍ D. interface 192.168.1.0 area 0
❍ E. area 192.168.1.0

4. Users are complaining that your company's website has been extremely slow all day, and in several cases, connections are timing out. You have checked all the servers, and they seem fine, but the load on your router's external interface is abnormally high. You suspect a denial of service attack using spoofed ICMP traffic that is bringing a barrage of echo replies to your network. What single entry in your ACL would provide the most information to trace the attack when applied inbound to the outside interface?
Select the best answer.

○ A. access-list 167 permit icmp any any echo-reply
○ B. access-list 167 permit icmp any any echo-reply log-input
○ C. access-list 67 permit any log
○ D. access-list 67 permit icmp any any echo-reply

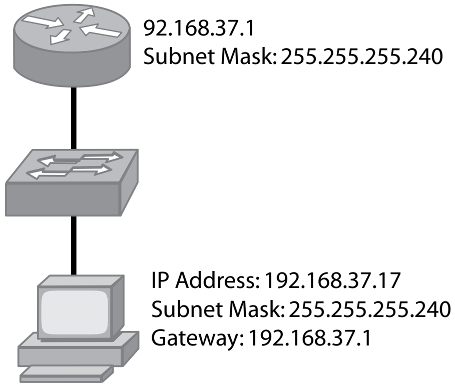
5. A previous consultant designed the following access list for a client. Which of the below statements are true concerning this access list?
access-list 10 permit 10.1.1.1
access-list 10 deny 10.1.1.0 0.0.0.255
access-list 10 permit 192.168.0.0 0.0.255.255
Choose THREE.

□ A. This is an extended access list.
□ B. Traffic from the host 10.1.1.1 will be denied.
□ C. Traffic from 172.16.1.1 will be denied.
□ D. There is an implicit deny at the end of the access list.
□ E. This is a standard IP access list.

6. A user is complaining that they cannot access anything beyond the local LAN network. They are unable to ping the gateway IP address of 192.168.37.1. What is the problem?
Select the best answer.

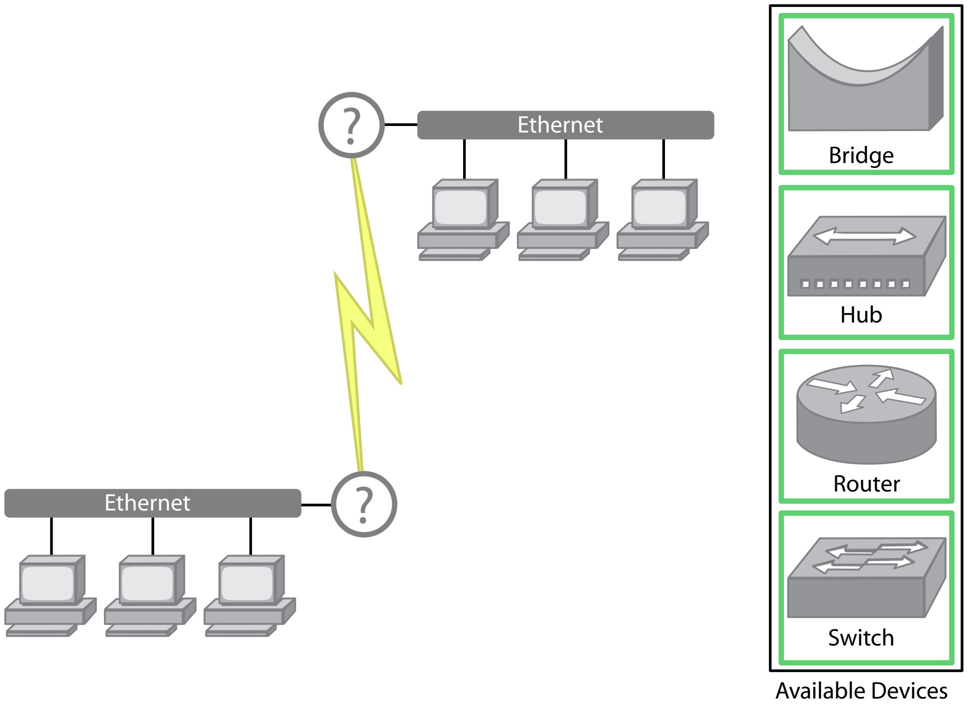
○ A. The subnet mask of the host does not match the gateway.
○ B. The IP address of the host is not on the same subnet as the gateway.
○ C. The default gateway of the host incorrect
○ D. The gateway IP address should be configured on the switch, not the router.

Exhibit(s):


92.168.37.1
Subnet Mask: 255.255.255.240
IP Address: 192.168.37.17
Subnet Mask: 255.255.255.240
Gateway: 192.168.37.1

Chapter 4 Configure, Verify, and Troubleshoot Basic Router Operation and Routing on Cisco Devices

1. You are designing an internetwork. You need to connect two sites (two LANs) using a leased line. Which of the following devices will you use in each of the two positions indicated by a question mark in the diagram below?



2. What is the purpose of NVRAM in Cisco routers?
Select the best answer.

○ A. NVRAM is used to store IOS images.
○ B. NVRAM is used as a fast packet buffer.
○ C. NVRAM is used to store device configuration.
○ D. NVRAM is used to store backup images in case a primary failure occurs.
○ E. NVRAM is used as a fast cache for instructions that need processing.

3. You want to be able to download an image file over the network. Your FTP server requires a username and password. Which command or set of commands can you use to specify authentication information for your FTP connections?
Select the best answer.

○ A. Cisco does not support authenticated FTP sessions.
○ B. copy ftp flash /username:user1 /password:cisco
○ C. copy ftp flash /authentication:user1 /password:cisco
○ D. ip ftp username user1 ip ftp password cisco
○ E. ftp-server username user1 ftp-password cisco

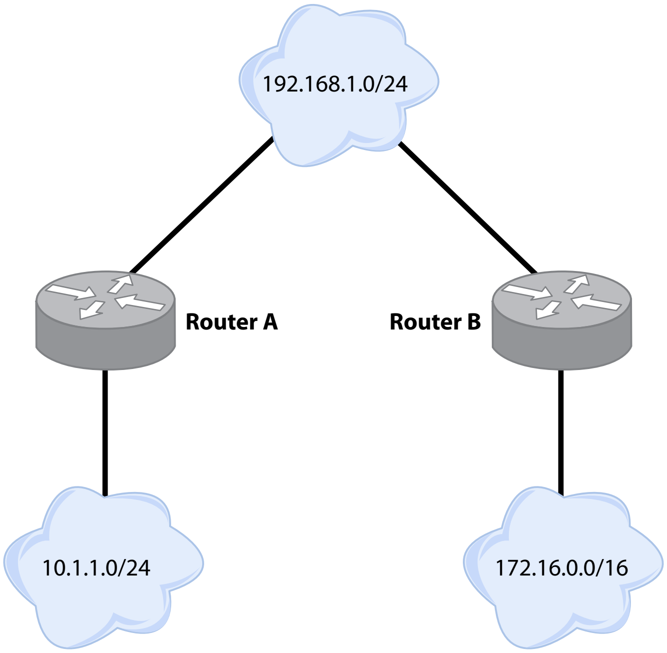
4. You are designing an access list to filter UDP traffic from port 500 on any host to any port on host 172.16.1.1. Which command(s) will you use?
Select the best answer.

❍ A. access-list 101 deny udp any eq 500 host 172.16.1.1
❍ B. access-list 102 deny udp host 172.16.1.1 eq 500 any
❍ C. access-list 103 deny udp host 172.16.1.1 any eq 500
❍ D. access-list 105 deny udp any any eq 500 access-list 105 deny udp host 172.16.1.1 any
❍ E. access-list 105 deny udp any eq 500 any access-list 105 permit udp 172.16.1.1 any

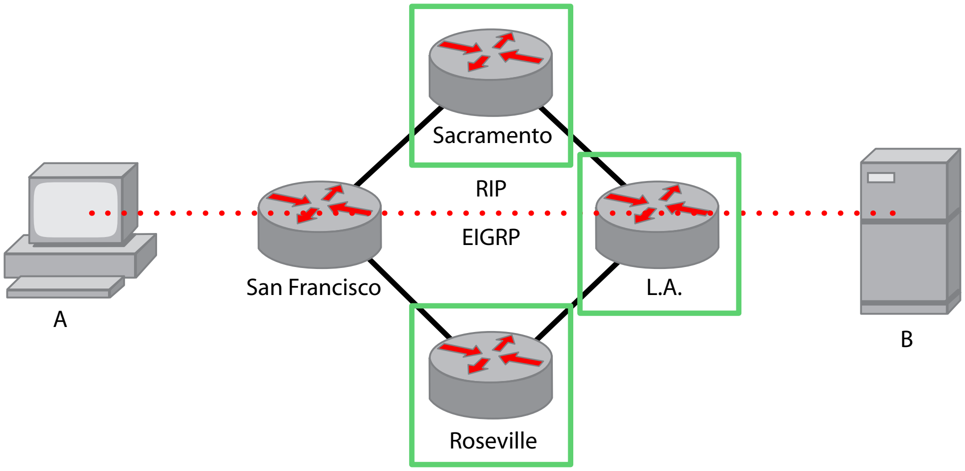
5. You have two routers within your network, and you need to enable RIP as your routing protocol. The routers are connected by the 192.168.1.0/24 network. Router A also has an interface in the 10.1.1.0/24 network. Router B has an interface in the 172.16.0.0/16 network. What are all the commands necessary to enable RIP on router B, and ensure all connected networks are reachable by router A?
Select the best answer.

❍ A. router rip
❍ B. router rip network 192.168.1.0
❍ C. enable router rip network 10.0.0.0 network 192.168.1.0
❍ D. router rip network 172.16.0.0 network 192.168.1.0

Exhibit(s):



6. Host A is sending packets destined for LA to its default Gateway in San Francisco. The San Francisco router has both EIGRP and RIP enabled. What is the next hop for packets destined for LA? (Note: All EIGRP routes are internal, and the LA and SF routers are running both protocols.)



Chapter 5 Explain and Select the Appropriate Administrative Tasks Required for a WLAN

1. Which of the following are key components to the 802.11 Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) Model?
Choose all that apply.

❑ A. Carrier Sense
❑ B. DCF
❑ C. Acknowledgement frames
❑ D. Request to Send/Clear to Send
❑ E. Frame fragmentation

2. In CSMA/CA, which component uses the Network Allocation Vector (NAV) to determine the state of the wireless medium?
Select the best answer.

○ A. Acknowledgement frames
○ B. Carrier Sense
○ C. Frame fragmentation
○ D. Request to Send/Clear to Send

Chapter 6 Identify Security Threats to a Network and Describe General Methods to Mitigate those Threats

1. Examine the following excerpt from a Cisco device's running configuration:
enable secret 5 $1$PhXB$ZF1hptFe6PADLVC/EGN6N/ What does the "5" mean?
Select the best answer.

○ A. That the following characters are to be interpreted as plain text.
○ B. The following characters are to be interpreted as having simple encryption.
○ C. The following characters should be interpreted as being hashed with MD5.
○ D. It means nothing.

2. Which commands can be used to reenable a switch interface when there is a MAC security violation?
Choose TWO.

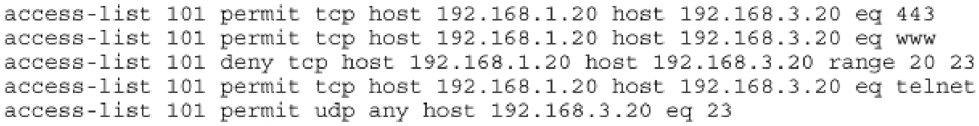
□ A. Switch(config-if)#No shutdown
□ B. Switch(config)# errdisable recovery cause psecure-violation
□ C. Switch(config-if)# errdisable recovery cause psecure-violation
□ D. Switch(config)#disable switchport security

Chapter 7 Implement, Verify, and Troubleshoot NAT and ACLs in a Medium-Size Enterprise Branch Office Network

1. Your boss asked you to limit access from H1 to H2. An ACL was created and applied outbound on R1 fa0/1 to allow only certain access. Which TCP and UDP ports are permitted by ACL 101?
Select the best THREE answers.

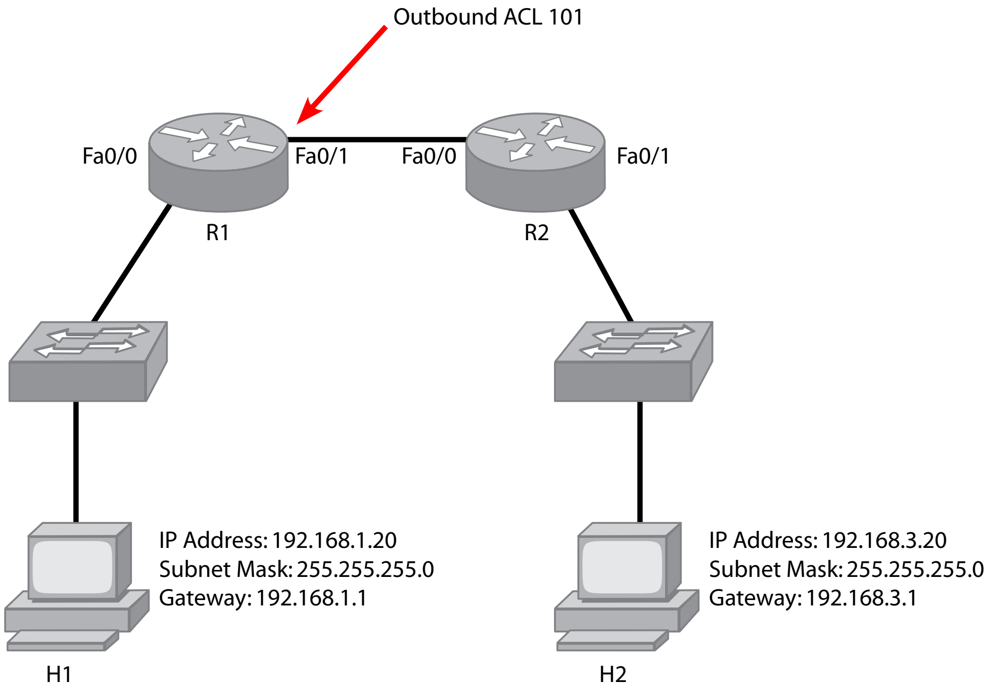
□ A. H1 will be able to access H2 using www
□ B. H1 will be able to access H2 using Telnet
□ C. H1 will be able to access H2 using UDP 23
□ D. H1 will be able to access H2 using HTTPS.

Exhibit(s):



```
access-list 101 permit tcp host 192.168.1.20 host 192.168.3.20 eq 443
access-list 101 permit tcp host 192.168.1.20 host 192.168.3.20 eq www
access-list 101 deny tcp host 192.168.1.20 host 192.168.3.20 range 20 23
access-list 101 permit tcp host 192.168.1.20 host 192.168.3.20 eq telnet
access-list 101 permit udp any host 192.168.3.20 eq 23
```

Exhibit(s):



Chapter 8 Implement and Verify WAN Links

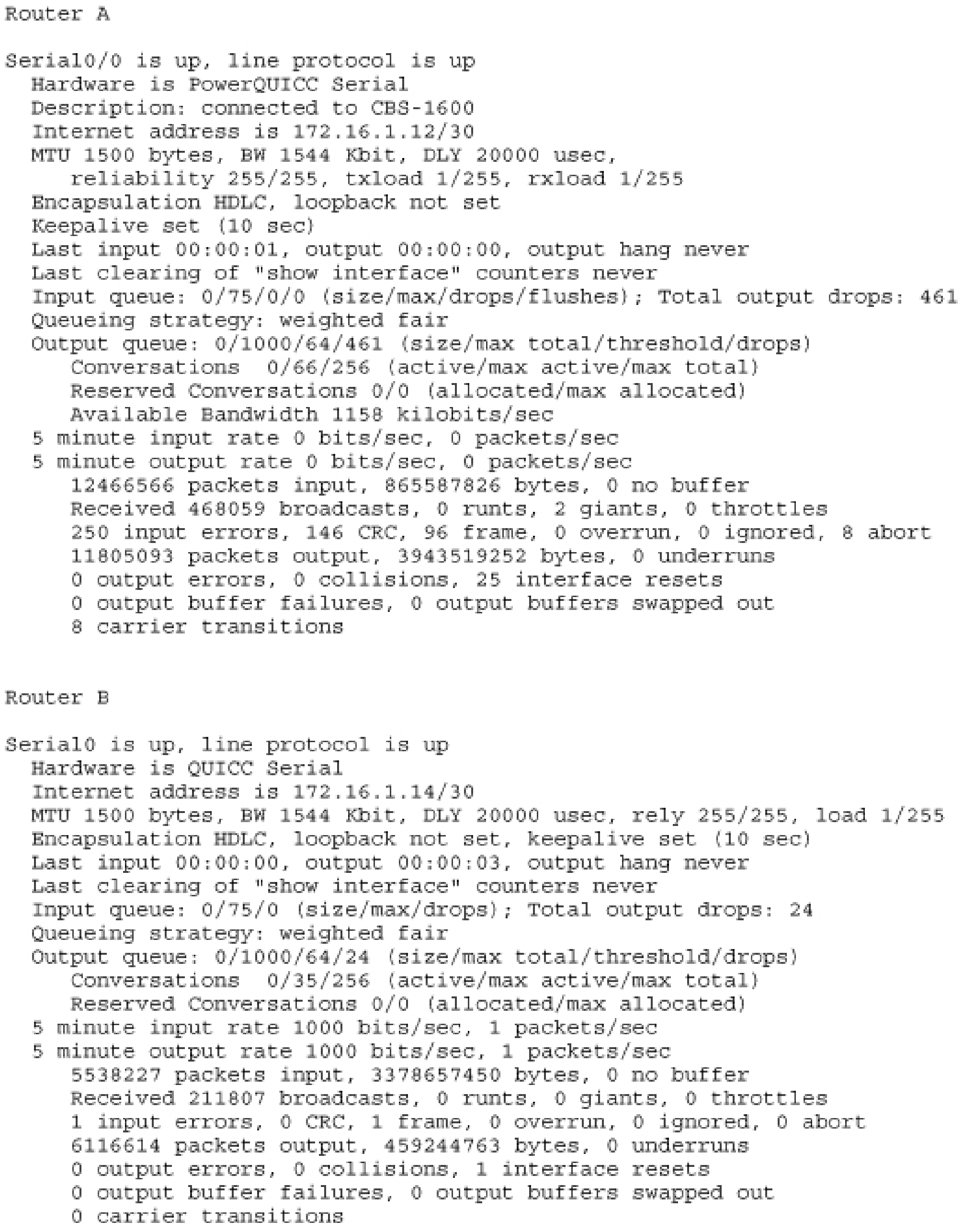
1. You are designing an internetwork. You plan to use ISDN. Which are some of the common features of ISDN that you can include in your design?
Select the three best answers.

❑ A. ISDN can provide voice and video capabilities.
❑ B. A single ISDN line can provide 6 Mbps to a subscriber.
❑ C. ISDN provides mesh connectivity.
❑ D. ISDN can be used for cost effective remote access.
❑ E. ISDN can be used for dial backup.

2. You are attempting to enable one of your WAN links between HQ and one of your branches and you cannot ping the branch. You have "show interfaces" output for each of the serial interfaces, shown within the exhibit. What is the problem?
Select the best answer.

❍ A. The encapsulation types do not match.
❍ B. Router A is utilizing Serial 0/0 and Router B is utilizing Serial 0. These must match in order for the link to work.
❍ C. The IP addresses are not in the same network.
❍ D. Router A and Router B are utilizing different hardware types, PowerQUICC and QUICC

Exhibit(s):



```
Router A

Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Description: connected to CBS-1600
  Internet address is 172.16.1.12/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:01, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 461
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/461 (size/max total/threshold/drops)
     Conversations  0/66/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     12466566 packets input, 865587826 bytes, 0 no buffer
     Received 468059 broadcasts, 0 runts, 2 giants, 0 throttles
     250 input errors, 146 CRC, 96 frame, 0 overrun, 0 ignored, 8 abort
     11805093 packets output, 3943519252 bytes, 0 underruns
     0 output errors, 0 collisions, 25 interface resets
     0 output buffer failures, 0 output buffers swapped out
     8 carrier transitions


Router B

Serial0 is up, line protocol is up
  Hardware is QUICC Serial
  Internet address is 172.16.1.14/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 00:00:00, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 24
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/24 (size/max total/threshold/drops)
     Conversations  0/35/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
     5538227 packets input, 3378657450 bytes, 0 no buffer
     Received 211807 broadcasts, 0 runts, 0 giants, 0 throttles
     1 input errors, 0 CRC, 1 frame, 0 overrun, 0 ignored, 0 abort
     6116614 packets output, 459244763 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Answers and Explanations

Chapter 1

1. Answers: C, D, E

Explanation A. Incorrect. BPDUs don't carry information about IP addresses. In fact, the 802.1D standard does not require switches to have IP addresses, as they are level 2 devices.

Explanation B. Incorrect. BPDUs don't carry information about hostnames. In fact, the 802.1D standard does not require switches to have hostnames, as they are level 2 devices.

Explanation C. Correct. Each bridge that takes part in the Spanning Tree Process must have a Bridge ID. This parameter plays an important role in the process of Root bridge selection. This bridge ID is often derived from the switch MAC address.

Explanation D. Correct. The Spanning Tree Protocol tries to select an optimal loop-free topology. The shortest distance to the root switch is calculated for each switch based on the path cost.

Explanation E. Correct. BPDUs are exchanged periodically between switches. It is important that switches know the communication parameters of their neighbors to detect line and device failures. Each BPDU carries the configuration of the sending switch.

2. Answer: A

Explanation A. Correct. The physical layer defines the physical processes and specifications for activating, maintaining, and deactivating the communications channel between communicating devices. Voltage levels, timing, data rates, maximum transmission distances, etc. are defined by the physical layer.

Explanation B. Incorrect. The data-link layer has two sublayers defined by IEEE: Logical Link Control (LLC) and Media Access Control (MAC). MAC is responsible for media access and is media dependant. LLC describes frame transmission mechanisms, and may provide either reliable, or best-effort delivery.

Explanation C. Incorrect. The transport layer is used for addressing applications by means of TCP/UDP ports. The TCP protocol is also able to provide guaranteed delivery. In addition, the transport layer segments data.

Explanation D. Incorrect. The network layer provides addressing. IP addressing is an example of network layer addressing. Other functions performed by the network layer are routing (in accordance with addressing) and packet fragmentation (takes into consideration the MTU).

Explanation E. Incorrect. The presentation layer deals with data representation. If two devices use different data formats (such as ASCII table versus EBCDIC0), they will be unable to communicate.

3. Answers: B, D

Explanation A. Incorrect. Bridges function at layer 2, forwarding frames to all ports.

Explanation B. Correct. Bridges are intelligent network filters that build forwarding tables to distinguish between local and remote destinations.

Explanation C. Incorrect. Bridges are intelligent devices that segment the network through filtering.

Explanation D. Correct. When a bridge does not have an address within its table, it sends the traffic to all ports.

4. Answers: A, C, D

Explanation A. Correct. Half-duplex transmissions rely on nodes to detect collisions and perform retransmission, and therefore are susceptible to collisions.

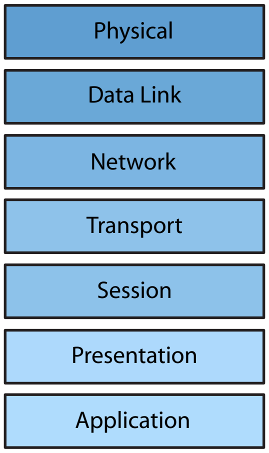
Explanation B. Incorrect. Hubs require end devices to run at half-duplex.

Explanation C. Correct. Hubs rely on network nodes to detect collisions and retransmit.

Explanation D. Correct. On switched networks, full-duplex transmission allows transmissions in both directions, and provides point-to-point transmission. Because there is a "dedicated" connection between end nodes, collision detection is not required.

Explanation E. Incorrect. Half-duplex can attain around 50-60% of the bandwidth of a link due to collisions and retransmissions.

5. Answer:



Explanation: The sequence in which a host receiving traffic will process data is: Physical, Data Link, Network, Transport, Session, Presentation, and Application. A sending host would process in the reverse order.

Chapter 2

1. Answers: A, B, C, E

Explanation A. Correct. This important information allows switches to discover each other on the network, as well as to detect when a switch/bridge failure occurs.

Explanation B. Correct. This information is important because each switch/bridge needs to have a unique way to identify neighbor devices and their knowledge of the STP topology.

Explanation C. Correct. Each switch/bridge informs its neighbor about its proximity to the root bridge/switch. Based on this information, other switches (from lower levels) will calculate per port distance to the root bridge to determine which ports must be disabled.

Explanation D. Incorrect. Switches don't care about DNS names, and don't send this kind of information in BPDUs. DNS operates on layers 5-7 (from session to application layer).

Explanation E. Correct. Each switch needs to inform its neighbors about the switch that it considers a root bridge in the STP topology.

2. Answer: D

Explanation A. Incorrect. This is not a reason for STP to put a port into blocking state. The STP protocol does not manage faulty hardware conditions.

Explanation B. Incorrect. The STP protocol does not manage collisions, and it won't change port state even if very high collision conditions are encountered.

Explanation C. Incorrect. In fact, FastEtherchannel (or just EtherChannel) is a way to avoid ports being put into blocking state, in case you have more than one interconnect between switches.

Explanation D. Correct. The main job of the STP protocol is to find data-link loops within the Internetwork, and eliminate these loops to avoid various problems that can be caused by virtue of multiple paths between end nodes. Redundant paths are put into blocking state.

Explanation E. Incorrect. A duplicate MAC address, although unlikely, is a serious problem within an Internetwork. However, the Spanning Tree Protocol does not handle such faulty conditions.

3. Answers: A, C, E

Explanation A. Correct. This command is the first place to look for performance issues. You will see the speed and duplex (negotiated), and also the error counts on the interface. Duplex errors will result in a high number of errors. If the duplex mismatch is causing the circuit to flap repeatedly, the interface may go into error-disable mode. The error disable status will show up on the "show interfaces" command.

Explanation B. Incorrect. This command will show packet counts for all the interfaces. Although this command is useful for analyzing interface traffic, it will not provide helpful information for troubleshooting duplex issues.

Explanation C. Correct. The "show log" command will display logging information when configured correctly. Duplex mismatches will be logged for the problem interface, as will interface flapping. If the duplex mismatch is causing the circuit to flap repeatedly, the interface may go into error-disable mode. This will show up in the logs.

Explanation D. Incorrect. The "show log history" command displays statistics about the log: messages logged, dropped, received, etc. The "show log" command would be far more useful when troubleshooting duplex mismatches.

Explanation E. Correct. This command will show all the interfaces, their status, VLAN, negotiated duplex, and speed and type.

Chapter 3

1. Answers: C, D, E

Explanation A. Incorrect. This command enables the OSPF process and assigns it a prcoess ID (100 in this particular case). This command does not allow you to influence the selection of a router ID.

Explanation B. Incorrect. This command is used to restore connectivity to a backbone area when no physical connectivity to it exists from a particular area. This command does not influence the selection of a router ID.

Explanation C. Correct. By default, OSPF selects a configured loopback interface as its router ID.
This loopback interface will be the one with the highest IP address. If no loopback interface exists, the IP address on one of the other router interfaces will be selected.

Explanation D. Correct. By default, OSPF selects a configured loopback interface as its router ID. If no loopback interfaces are configured, OSPF will select an interface of any kind. The selected interface will be the one with the highest IP address.

Explanation E. Correct. By using the router-id command in router configuration mode for OSPF, you can manually assign a router ID to a router. After the issuance of the command, you need to restart the OSPF process to force changes.

2. Answers: C, D

Explanation A. Incorrect. RIP v.1 only supports equal cost path load balancing on 4 ports by default. However, RIP v.1 can be configured to support up to 6 equal paths for load balancing.

Explanation B. Incorrect. RIP v.2 only supports equal cost path load balancing on 4 ports by default. However, RIP v.2 can be configured to support up to 6 equal paths for load balancing.

Explanation C. Correct. Both IGRP and EIGRP support unequal cost path load balancing. In other words, a router configured for some of these routing protocols is able to install more than a single "best" path to the destination.

Explanation D. Correct. Both IGRP and EIGRP support unequal cost path load balancing. In other words, a router configured for some of these routing protocols is able to install more than a single "best" path to the destination.

Explanation E. Incorrect. OPSF only supports equal cost path load balancing, over 4 paths by default, with the possibility to configure 6 equal cost paths.

3. Answer: C

Explanation A. Incorrect. The OSPF network command uses wildcard bits, rather than subnet mask. Thus, to specify a match in the first three octets, one should specify
0.0.0.255 instead of 255.255.255.0

Explanation B. Incorrect. The OSPF network command uses wildcard bits, rather than subnet mask. Thus, to specify an exact match, one should specify 0.0.0.0 instead of 255.255.255.255.

Explanation C. Correct. The network command assigns a network (the first command parameter (192.168.1.0)), and specifies the wildcard bits (the second command parameter (0.0.0.255 -exact match)) to an area (the third command parameter (0 ¬Backbone)).

Explanation D. Incorrect. This command does not exist in Cisco IOS.

Explanation E. Incorrect. This command does not exist in Cisco IOS.

4. Answer: B

Explanation A. Incorrect. This entry is correct to identify the traffic, but would not log any of the traffic information. Without turning on the logging functionality, there is no way to identify the source of the DOS attack.

Explanation B. Correct. This IP extended access list identifies the correct traffic, and will log information into the buffer for examination.

Explanation C. Incorrect. This IP standard access list will not achieve the desired result.

Explanation D. Incorrect. You cannot specify protocol or packet type with an IP standard access list.

5. Answers: C, D, E

Explanation A. Incorrect. Standard IP access lists are numbered from 1-99; extended IP access lists are from 100-199.

Explanation B. Incorrect. All IP traffic will be permitted from this host.

Explanation C. Correct. This host is not specifically identified within the access list, and the implicit deny will be applied.

Explanation D. Correct. All access lists have an implicit deny at the end.

Explanation E. Correct. Standard access lists are numbered from 1-99; extended access lists are from 100-199.

6. Answer: B

Explanation A. Incorrect -The subnet mask is the same as the mask of the router gateway.

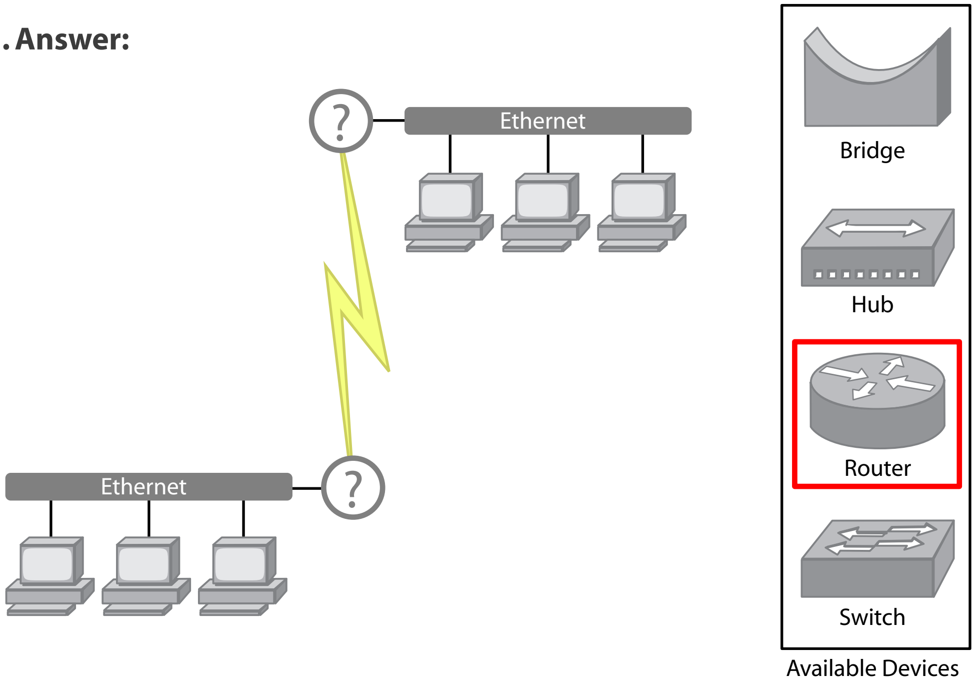
Explanation B. Correct. The IP addressed configured on the host is on a different subnet than the gateway. The IP address range of the 192.168.37.0/28 is 192.168.37.1 to 192.168.37.14

Explanation C. Incorrect -The default gateway is properly configured for 192.168.37.1 -just like the router interface.

Explanation D. Incorrect -The switch is a layer 2 device which does not need to be configured with a layer 3 IP address. The router is the correct device to have an IP address assigned to it for use by hosts as the gateway.

Chapter 4

1. Answer:



Explanation: Routers are used to connect to the WAN. In this particular example, two routers are used to interconnect the two sites using a leased line.

2. Answer: C

Explanation A. This is incorrect. Flash memory is used for storing IOS images. NVRAM is used for storing device configuration files.

Explanation B. This is incorrect. RAM is used as a fast access buffer in Cisco routers and switches. NVRAM is used to store configuration files.

Explanation C. Correct. NVRAM is a type of memory, which due to its relatively low power consumption can be used as a non-volatile device configuration storage.

Explanation D. This is incorrect. NVRAM is used to store configuration files. Images normally get stored in flash, and if there is room for more than one image, a couple of IOS images can be stored in flash.

Explanation E. Incorrect. NVRAM is used to only store configuration information, and is not suitable for being a fast cache.

3. Answer: D

Explanation A. Incorrect. Authenticated FTP sessions for file transfers are supported by Cisco routers and switches. Authentication information needs to be specified in global configuration mode.

Explanation B. Incorrect. This parameter of the copy command does not exist in Cisco IOS.

Explanation C. Incorrect. This parameter of the copy command does not exist in Cisco IOS.

Explanation D. Correct. You need to use the above two commands in global configuration mode to setup a default username and password to be used by the router when performing FTP file transfers. Note that this username and password are used by the router for all FTP file transfers, and not just per session.

Explanation E. Incorrect. The above commands do not exist in Cisco IOS.

4. Answer: A

Explanation A. This is the correct syntax. This access list will deny UDP traffic on port 500 from any host to any UDP port on host 172.16.1.1.

Explanation B. Incorrect. The above access list will deny UDP traffic from port 500 from the host 172.16.1.1 to any UDP port on any host, which is not the requirement of this question.

Explanation C. Incorrect. The above access list will deny UDP traffic from host 172.16.1.1 to any host on UDP port 500, which is not the requirement of this question.

Explanation D. Incorrect. The above access list will block any traffic to UDP port 500 to any host.
In addition, this access list will deny traffic sourced from 172.16.1.1 to any host. This is not the requirement of this question.

Explanation E. Incorrect. The above access list will deny any UDP traffic to port 500, and it will allow host 172.16.1.1 to communicate with any other host using the udp protocol. This is not the requirement of this question.

5. Answer: D

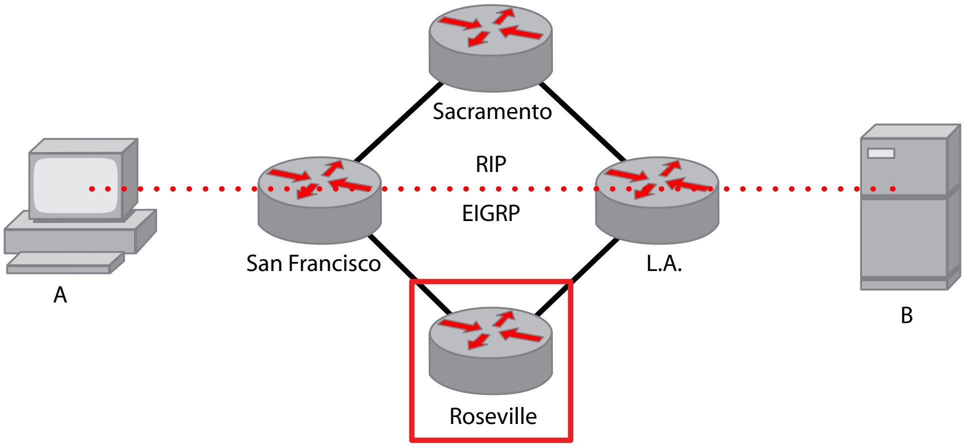
Explanation A. Incorrect. This command would just activate the routing protocol.

Explanation B. Incorrect. In order to ensure that router A learns all the connected networks on router B, you must use the network command for each interface, and directly connnected interfaces.

Explanation C. Incorrect. "enable router rip" is not a correct command.

Explanation D. Correct. The "router rip" command activates RIP as the routing protocol. The "network" command then activates RIP processing for interfaces associated with the networks. In order for router B to propagate and receive information, both the 192.168.1.0 and 172.16.0.0 network statements must be present.

6. Answer:



Explanation: The next hop en route to LA will be the Roseville router. If a router participates in several routing protocols and the destination traffic has separate paths using the different routing protocols, it will utilize the administrative distance of the routing protocol as a tiebreaker to determine the next hop path. The route with the lowest administrative distance will win. In this case, EIGRP routes have an admin distance of 90, where RIP is 120.

Chapter 5

1. Answers: A, B, C, D

Explanation A. Correct. Before a wireless station transmits, it must sense whether the medium is in use. There are two methods for determining this:
-Checking the Physical Layer, or PHY, to see if a carrier is present
-Using the Network Allocation Vector (NAV), which is a transmission timer

Explanation B. Correct. The DCF (distributed coordination function) is an IEEE access mechanism for controlling wireless medium access. DCF prevents two stations that sense a lull in traffic from sending information at the same time through the use of a random backoff timer.

Explanation C. Correct. Acknowledgement frames notify a sending station that a transmission has been received, and receive special treatment in the wireless world. These frames are not subject to any of the backoff timers, and their timely delivery helps avoid needless retransmissions.

Explanation D. Correct. RTS/CTS are special control frames that are used when communicating with an access point, and provide a "two-way" handshake method of transmission control.

Explanation E. Incorrect. Frame fragmentation is part of the 802.11 medium access control, but it is not directly a part of CSMA/CA. It allows for the breakup of frames into smaller pieces to reduce media contention.

2. Answer: B

Explanation A. Incorrect. Acknowledgement frames notify a sending station that a transmission has been received, and receive special treatment in the wireless world. These frames are not subject to any of the backoff timers, and their timely delivery helps avoid needless retransmissions.

Explanation B. Correct. Before a wireless station transmits, it must sense whether the medium is in use. There are two methods for determining this:
-Checking the Physical Layer, or PHY, to see if a carrier is present
-Using the Network Allocation Vector (NAV), which is a transmission timer

Explanation C. Incorrect. Frame fragmentation is part of the 802.11 medium access control, but it is not directly a part of CSMA/CA. It allows for the breakup of frames into smaller pieces to reduce media contention.

Explanation D. Incorrect. RTS/CTS are special control frames that are used when communicating with an access point, and provide a "two-way" handshake method of transmission control.

1. Answer: C

Explanation A. Incorrect. Plain text passwords are type "0" passwords. If this was plain text, the entry would read: enable secret 0 $1$PhXB$ZF1hptFe6PADLVC/EGN6N/

Explanation B. Incorrect. Passwords that can be decrypted are tagged with a "7" in the cisco config. There are several public utilities designed to "crack" these passwords.

Explanation C. Correct. Cisco tags its passwords with a 0, 5, or 7 to identify the type of password. In this case, the password is a type 5, or MD5 encrypted password, which cannot be cracked.

Explanation D. Incorrect. This is a password type identifier.

2. Answers: A, B

Explanation A. Correct. This interface configuration command will always manually reset an interface, resetting the status on port security.

Explanation B. Correct. This global configuration command will reset all port security violations, and enable the ports.

Explanation C. Incorrect. This command is set at the global configuration prompt. If you are in the interface configuration mode, utilize the "no shutdown" command to reset the port.

Explanation D. Incorrect. This is not an IOS command. You will need to utilize either of the above commands to reenable the switchport.

Chapter 7

1. Answers: A, C, D

Explanation A. Correct -The ACL allows H1 to access TCP port 80 (written as www) to host H2.

Explanation B. Incorrect -The deny TCP blocks access on TCP ports 20-23. Access-lists are read by the router in a top down manner so the deny statement is read first and blocks the permit statement below it.

Explanation C. Correct -The ACL allows access from H1 to H2 on UDP port 23. The deny statement above does not affect this rule because it is UDP as opposed to TCP.

Explanation D. Correct -The ACL does permits H1 to access H2 via HTTPS (TCP 443).

Chapter 8

1. Answers: A, D, E

Explanation A. Correct. As ISDN provides integrated services; voice and video are part of the ISDN standard. Since ISDN uses baseband communication, its behavior such as speed, packet drop, and delay are easily predictable, which is a huge plus for voice and video.

Explanation B. Incorrect. DSL technologies allow for speeds up to a couple Mbps, or tens of Mbps in some cases. These technologies use existing telephone lines and complex algorithms for broadband communications.

Explanation C. Incorrect. ISDN provides point-to-point connectivity, as there are only two parties in an ISDN connection: called and called parties. Other technologies, such as Frame Relay, provide for mesh connectivity.

Explanation D. Correct. One of the most important applications of ISDN is for high density dial-up access, used by ISPs and enterprise remote access solutions. The bandwidth of ISDN dial-up connections makes this protocol preferred in comparison to legacy analog lines.

Explanation E. Correct. Pretty similar to remote access -ISDN can be used for cost effective remote access, including dial backup.

2. Answer: C

Explanation A. Incorrect. The encapsulation types match on both routers. They are using HDLC. If this was the problem, the line protocol would be down.

Explanation B. Incorrect. Cisco numbers serial interfaces based on primary and subinterfaces. 0/0 presents hardware that usually has multiple serial interfaces on a card. A single-digit interface usually indicates a fixed interface. This does not matter for WAN connections, and interface numbers do not need to match.

Explanation C. Correct. This WAN link is set up using a /30 network with two IPs. 172.16.1.12/30 is the network, and 172.16.1.15 is the broadcast address. The interfaces should be IP'd with 172.16.1.13 and 14. Router A is utilizing the network address, and this would cause issues.

Explanation D. Incorrect. Cisco is constantly creating new hardware types to provide faster performance with improved signal quality, but this would not cause the stated issue.