Cisco Certified Network Associate (CCNA)

Juan Estrella : Juan@tia.edu
A+, Network+, Security+, CASP, CTT+, MCSE/MCITP, CCNA, CCNP, MOS, Adobe ACE, ITIL, CEH, CISSP
Are you ready for the CCNA course?

Before taking the CCNA course, learners should be familiar with:

- **Basic computer literacy**
  - Being able to identify the parts of a basic computer system and how they interoperate.
- **Basic PC operating system navigation skills**
  - Being able to interact with a computer system, open/close/save files. Familiarity with command line utilities.
- **Basic Internet usage skills**
  - Familiar with working with emails, websites, software downloads and basic security.
- **Basic IP address knowledge**
  - Know how to identify the class of an IP address and the corresponding subnet mask. Know the difference between private and public IP addresses.
- **Good understanding of network fundamentals**
  - Understand the difference between routers, switches, hubs, and other common network devices. Be able to identify the different network media types (Cables/RF).
CCNAv3 Exam 200-125

- 90 minutes
- 50 – 60 Questions
  - Multiple choice
    - Pick one or many answers. No partial credit.
  - Drag & Drop
    - Match objects to a diagram.
  - Simlets
    - Troubleshooting scenarios where you’ll answer four to seven questions using diagnostic commands.
  - Simulators
    - These are hands-on troubleshooting scenarios where you’ll have to identify a misconfiguration, resolve it, verify it, and test it to complete the question.
- Passing score is about 810/1000
- Exam fee is $325 (NOT included in the course)
- Exam will be retired on Feb. 23rd, 2020

The CCNA certification expires 3 years after you pass the exam. You can recertify in 3 years or take another Cisco certification exam at the Network Associate level to extend it by 1 year.
CCNA Exam 200-301

- 120 minutes
- 50 – 60 Questions
  - **Multiple choice**
    - Pick one or many answers. No partial credit.
  - **Drag & Drop**
    - Match objects to a diagram.
  - **Simlets**
    - Troubleshooting scenarios where you’ll answer four to seven questions using diagnostic commands.
  - **Simulators**
    - These are hands-on troubleshooting scenarios where you’ll have to identify a misconfiguration, resolve it, verify it, and test it to complete the question.

- **Passing score is about ??**
- **Exam fee is $325 ( NOT included in the course )**
- Exam will be available **Feb. 24th, 2020**

<table>
<thead>
<tr>
<th>Domain</th>
<th>% of Exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Fundamentals</td>
<td>20%</td>
</tr>
<tr>
<td>Network Access</td>
<td>20%</td>
</tr>
<tr>
<td>IP Connectivity</td>
<td>25%</td>
</tr>
<tr>
<td>IP Services</td>
<td>10%</td>
</tr>
<tr>
<td>Security Fundamentals</td>
<td>15%</td>
</tr>
<tr>
<td>Automation and Programmability</td>
<td>10%</td>
</tr>
</tbody>
</table>

The CCNA certification expires 3 years after you pass the exam. You can maintain your certification by taking any other Cisco exam before your certification expires.
Recommended Textbook

CCNA Routing and Switching Complete Study Guide: Exams 100-105, 200-105, and 200-125

Author: Todd Lammle  
Hardcover: 1136 pages  
Language: English  
ISBN-10: 1119288282  

Available at all major book sellers like Amazon

TIPS:  
• You don’t need to memorize the entire book, but you do need to read it.  
• At the end of every chapter there’s an “Exam Essentials” section where the author notes the most important topics in the chapter, make sure you understand those topics.
Course Outline

Day 1
- Chapter 1: Internetwork
- Chapter 2: Ethernet Networking and Data Encapsulation
- Chapter 3: Introduction to TCP/IP
- Chapter 4: Easy Subnetting
- Chapter 5: VLSMs, Summarization, and Troubleshooting TCP/IP

Day 2
- Chapter 6: Cisco’s Internetworking Operating System (IOS)
- Chapter 7: Managing a Cisco Internetwork
- Chapter 8: IP Routing

Day 3
- Chapter 9: Open Shortest Path First (OSPF)
- Chapter 20: Multi-Area OSPF
- Chapter 19: Enhanced IGRP

Day 4
- Chapter 14: Internet Protocol Version 6 (IPv6)
- Chapter 12: Security
- Chapter 13: Network Address Translation (NAT)

Day 5
- Chapter 10: Layer 2 Switching
- Chapter 15: Enhanced Switched Technologies
- Chapter 16: IP Services
- Chapter 18: Troubleshooting IP, IPv6, and VLANs
- Chapter 21: Wide Area Networks
Recommended Textbook

CCNA Routing and Switching Complete Review Guide: Exam 100-105, Exam 200-105, Exam 200-125

Author: Todd Lammle
Hardcover: 480 pages
Language: English
ISBN-10: 1119288363

Available at all major book sellers like Amazon

Note:
- This book is very concise and does not provide a complete view of all network topics. I only recommend this book for those who have CompTIA Network+ level of experience along with real-world experience.
How to be a successful student

- **Work hard.** You can't be lazy if you want to be successful in life.
- **Be punctual.** Develop the habit of judging time and learn how to arrive where you need to be on time.
- **Read the books.** If you are having reading trouble, talk to the teacher for ways to improve your reading skills. The teacher is not here to read you the book, they’re there to explain the contents.
- **Use common sense.** Realize what the class is about, what the work is supposed to teach you, and how you should handle yourself with the other students.
- **Learn how to schedule your life and time** to balance school work and other parts of your day. You can't do everything, and going out at night, playing videogames, watching television will surely take time away from studying or that deadline.
- **Talk to the teachers.** If you show them you are honest and serious, or need help, they will usually respond well and offer to guide you as needed. But don't think the teacher will just pity you and excuse your poor work.
- **Study throughout the course.** Don't wait until days before the test to study.
- **Remember your educational goals.** If you can't really think of any, talk to a career counselor and learn what kind of education you will really need to get the salary or job you want.
- **Learn that a student and a professor make a team.** Most instructors want exactly what you want: they would like for you to learn and pass your exams. Join forces with your instructor, they are not an enemy, you share the same interests, the same goals - in short, you're teammates.
### The Open System Interconnection (OSI) Model

<table>
<thead>
<tr>
<th>Layer 7</th>
<th>Application</th>
<th>Software protocols ie; http, smtp, pop3, imap, ect…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 6</td>
<td>Presentation</td>
<td>Gets the Data ready for the application layer by providing translation services like compression/decompression and encryption/decryption</td>
</tr>
<tr>
<td>Layer 5</td>
<td>Session</td>
<td>Provides dialog control by allowing multiple streams of data from different sources to be properly combined or synchronized.</td>
</tr>
<tr>
<td>Layer 4</td>
<td>Transport</td>
<td>Handles the End-to-End communication. Can provide reliability by using Connection-Oriented Communication via TCP.</td>
</tr>
<tr>
<td>Layer 3</td>
<td>Network</td>
<td>Routing, IP addresses, ICMP, ARP, Routers and Firewalls</td>
</tr>
<tr>
<td>Layer 2</td>
<td>Data Link</td>
<td>Switches/Bridges, MAC addresses, PPP, HDLC</td>
</tr>
<tr>
<td>Layer 1</td>
<td>Physical</td>
<td>Hubs, Repeaters, Media(Cable/Radio Frequency), RS-232, CSU/DSU, Bits</td>
</tr>
</tbody>
</table>

### Mnemonics

All People Seem To Need Data Processing or Please Do Not Throw Sausage Pizza Away
Connection-Oriented Communication

The **Transport** layer can provide reliable communication by using **Connection-Oriented Communication**.

A **three-way handshake** is required for Connection-Oriented communication.

Once two hosts perform the three-way handshake a **Virtual Circuit** is established.

**Benefits**

- **Acknowledgments**: guarantees that the data that is sent has been received.
- **Flow control**: prevents one host from overflowing the buffers of the other.
- **Windowing**: controls how much information goes from one host to the other.
Network Devices

- **Hub**: is a layer 1 device which can only communicate via broadcasting.
- **Switch/Bridge**: is a layer 2 device that uses MAC addresses to forward frames.
- **Router**: is a layer 3 device used to transmit packets between multiple networks using IP addresses.
- **Firewall**: is a layer 3 IPS commonly used to protect private networks from the public network. They have security features like SPI (Stateful Packet Inspection) and a DMZ (Demilitarized Zone) which are not available on a router.
- **Layer 3 Switch**: is a switch/router, meaning every port is capable of layer 3 and layer 2 functions.
Wireless Network Devices

- **Wireless Access Point (WAP):** is a layer 2 device that allows a wireless host to connect to a wired network.

- **Wireless Controller** can be used to manage multiple access points, usually as part of a mesh network. Can provide additional functionality like rogue access point detection, network heat maps, RADIUS authentication and management frame protection.
Ethernet Networking

Collision Domains, Broadcast Domains, Internetworks, Duplexing, Cabling, and Data Encapsulation
**Ethernet Network**

**Collision Domain:** A network segment where only one host can transmit data at a time.

**Broadcast Domain:** A network segment where one address can be used to transmit data to all hosts.
**Ethernet Networks**

**Internetwork:** is created by a router where each interface is a different broadcast domain that can communicate to each other but can NOT pass broadcast to each other.
CSMA/CD (Carrier-sense multiple access with collision detection) is part of the Ethernet standard and utilized by hubs to manage collisions.

When CSMA/CD senses a collision, it sends out a jam signal that instructs all host on the collision domain to stop transmitting. The jam signal also includes a random number letting the host know when it can transmit again.
Interface Duplexing

- **Half-duplex (send or receive)**
  - One way communication. While you’re sending data you **CANNOT** receive data, while you’re receiving data you **CANNOT** send data.

- **Full-duplex (send and receive)**
  - Two way communication. Sending and receiving data are done simultaneously.

- **Troubleshooting Duplexing**
  - When you mismatch duplexing you will **NOT** experience total failure, instead you’ll experience **inconsistent networking**.
  - The “Late Collision” counter incrementing is a sign of a duplex mismatch.
**Ethernet Cabling**

**Straight-through**
- Used to connect unlike devices, hub to routers, computers to switches, switch to router, etc.

**TIP:** One device has an IP address the other device does not

**Crossover**
- Used to connect like devices, hub to hub, router to router, hub to switch, etc.

**TIP:** Both devices have IP addresses or both devices lack IP addresses.

**Rolled**
- Used to console to a router or switch
- One end is RJ-45 the other end is RS-232 (Serial/DB9)
Data Encapsulation

PDU (Protocol Data Unit): represents a unit of data specified in the protocol of a given layer, which consists of protocol control information and user data.

![Diagram of the OSI model with PDU encapsulation]

Mnemonic:
- Don’t
- Stop
- Pouring
- Free
- Beer

Bits: 01000010 01101001 01101110
Data Encapsulation PDUs and Addresses

<table>
<thead>
<tr>
<th>OSI Layer</th>
<th>PDU</th>
<th>Source Address</th>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport</td>
<td>Segment</td>
<td>Source Port Number</td>
<td>Destination Port Number</td>
</tr>
<tr>
<td>Network</td>
<td>Packet</td>
<td>Source IP Address</td>
<td>Destination IP Address</td>
</tr>
<tr>
<td>Data Link</td>
<td>Frames</td>
<td>Source MAC Address</td>
<td>Destination MAC Address</td>
</tr>
<tr>
<td>Physical</td>
<td>Bits</td>
<td>01100010 01100001 01100011 01101111 0110111</td>
<td>01100010 01100001 01100011 01101111 0110111</td>
</tr>
</tbody>
</table>

- **Segments** encapsulate application data while adding source and destination application **port numbers**.
- **Packets** encapsulate segments while adding source and destination **IP addresses**.
- **Frames** encapsulate packets while adding source and destination **MAC addresses**.
Cisco Three-Layer Hierarchical Model

This model is used to describe the function of the device in the network.

Core (Backbone)
- This is the network backbone and responsible for transporting huge amounts of traffic.

Distribution (Routing)
- This layer's primary functions are routing, filtering and WAN access.

Access (Switching)
- This layer controls user and workgroup access to the resources on the network.
TCP/IP | DoD Model

Application Protocols, Transport Protocols, Network Protocols, IP Addressing
TCP/IP – DoD Model

TCP/IP

- Process/Application
- Host-to-Host
- Internet
- Link or Network Access

OSI

- Application
- Presentation
- Session
- Transport
- Network
- Data-Link
- Physical
Process/Application Protocols

- **Telnet**: Remotely command a host over the network in plain text.
  - TCP Port: 23
- **SSH**: Remotely command a host securely over the network with encryption.
  - TCP Port: 22
- **FTP**: Transfer files, robust with many features
  - TCP Ports: 20, 21
- **TFTP**: Transfer files, lightweight protocol that only supports the transfer of files
  - UDP Port: 69
- **SNMP**: Gather information about the host on a network
  - UDP Port: 161
- **HTTP**: Transmits webpages
  - TCP Port: 80
- **HTTPS**: Securely transmits webpages
  - TCP Port: 443
- **NTP**: Allows a host to synchronize its clock with a server
  - UDP Port: 123
- **DNS**: Resolves names to IP addresses
  - TCP/UDP Port: 53
- **DHCP/BootP**: Gives out configurations to clients on a network
  - UDP Port: 67
- **APIPA**: Assigns a host an IP address if DHCP is unavailable
  - Address range: 169.254.0.0 – 169.254.255.255
Host-to-Host

- **TCP (Transmission Control Protocol)**
  - Reliable
  - Sequenced
  - Connection-oriented
  - Virtual circuit
  - Acknowledgements
  - Windowing flow control

- **UDP (User Datagram Protocol)**
  - Unreliable
  - Unsequenced
  - Connectionless
  - No virtual circuit
  - No acknowledgements
  - No windowing or flow control
Internet Layer

- **IP (Internet Protocol)**
  - Every packet gets an IP header at layer 3 which includes a *Source IP* and a *Destination IP*

- **ICMP (Internet Control Message Protocol)**
  - Returns a status on a unicast. Used by *ping*.
  - Returns a hop count. Used by *traceroute*.

- **ARP (Address Resolution Protocol)**
  - Resolves an IP address to a MAC address via *broadcasting*. Every host maintains their own ARP table, listing IP address to MAC addressing resolutions.
**Binary to Decimal & Binary to Hexadecimal**

**Bit:** 0 or 1  
**Byte:** 8 bits  
**Nibble:** 4 bits  
**Decimal:** 0 – 255  
**Hexadecimal:** 0 – 9, A – F

### Binary to Decimal

<table>
<thead>
<tr>
<th>Binary to Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

### Binary to Hexadecimal

<table>
<thead>
<tr>
<th>Binary to Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

IPv4 Addressing

IPv4 Address: 32 bit address used to communicate at layer 3 of the OSI.
Octet: 8 bit segmentation used for IP addresses.

Classful Networking

<table>
<thead>
<tr>
<th>Class</th>
<th>1st Octet</th>
<th>Number of Host</th>
<th>Subnet Mask</th>
<th>CIDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>1 - 126</td>
<td>16.7 Million</td>
<td>255.0.0.0</td>
<td>/8</td>
</tr>
<tr>
<td>Class B</td>
<td>128 - 191</td>
<td>65 Thousand</td>
<td>255.255.0.0</td>
<td>/16</td>
</tr>
<tr>
<td>Class C</td>
<td>192 - 223</td>
<td>254</td>
<td>255.255.255.0</td>
<td>/24</td>
</tr>
</tbody>
</table>

Non-routable IPv4 addresses (Private/NAT Addresses)

<table>
<thead>
<tr>
<th>Class</th>
<th>First Address</th>
<th>Last Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>10.0.0.0</td>
<td>10.255.255.255</td>
</tr>
<tr>
<td>Class B</td>
<td>172.16.0.0</td>
<td>172.31.255.255</td>
</tr>
<tr>
<td>Class C</td>
<td>192.168.0.0</td>
<td>192.168.255.255</td>
</tr>
</tbody>
</table>
### IPv4 Addressing in bits

<table>
<thead>
<tr>
<th>Class</th>
<th>1st Octet</th>
<th>2nd Octet</th>
<th>3rd Octet</th>
<th>4th Octet</th>
<th>Network bits</th>
<th>Host bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Network</td>
<td>Host</td>
<td>Host</td>
<td>Host</td>
<td>8</td>
<td>24</td>
</tr>
<tr>
<td>B</td>
<td>Network</td>
<td>Network</td>
<td>Host</td>
<td>Host</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>C</td>
<td>Network</td>
<td>Network</td>
<td>Network</td>
<td>Host</td>
<td>24</td>
<td>8</td>
</tr>
</tbody>
</table>

- **Network bits** can be used to figure out your subnet mask.
  - 8 network bits is a mask of
    - 11111111.00000000.00000000.00000000
    - 255.0.0.0
  - 16 network bits is a mask of
    - 11111111.11111111.00000000.00000000
    - 255.255.0.0

- **Host bits** are used to figure out the total number of addresses also known are your block size.
  - 8 host bits equals 256 addresses
    - $2^{\text{host bits}} = \text{block} : 2^8 = 256$
  - 16 host bits equals 65,536
    - $2^{\text{host bits}} = \text{block} : 2^{16} = 65,536
IPv4 Address Types

- **Unicast:** One to one communication
  - **Public IP:** Routable IP addresses assigned by an ISP.
  - **Private IP:** Non-routable IP addresses.
  - **APIPA:** Automatic private IP address self assigned by a host when DHCP is not available.
    - IP address range `169.254.0.0 – 169.254.255.255`

- **Multicast:** One to many communication
  - IPv4 multicast addresses can be identified by the first octet of `224`.

- **Broadcast (Layer 2):** One to everyone communication
  - MAC address `FF:FF:FF:FF:FF:FF`

- **Broadcast (Layer 3):** One to everyone communication
  - IP address `255.255.255.255`

- **Loopback (localhost):** used for the host to communicate with itself.
  - IP address range `127.0.0.0 – 127.255.255.255`
Internetwork Communication

Host A
192.168.100.10/24
aaaa.aaaa.aaaa

Host B
192.168.200.20/24
bbbb.bbbb.bbbb

Source IP
192.168.200.20
Host B

Source MAC
f000.f000.f000
R1 f0/0

Destination IP
192.168.100.10
Host A

Destination MAC
aaaa.aaaa.aaaa
Host A

R1
f0/0
192.168.100.24
f000.f000.f000

f1/1
192.168.200.24
f111.f111.f111

Source IP
192.168.200.20
Host B

Source MAC
bbbb.bbbb.bbbb
Host B

Destination IP
192.168.100.10
Host A

Destination MAC
aaaa.aaaa.aaaa
Host A

Host B
192.168.200.20/24
bbbb.bbbb.bbbb

Host A ← Host B
IPv4 Subnetting

Classless Interdomain Routing (CIDR) and Variable Length Subnet Mask (VLSM)
Subnetting

Subnetting is when we take a **Classful** network and break it up into many smaller **Sub-networks**.

**Benefits of Subnetting**

- **Minimize Broadcasting**
  - Broadcast is contained within its own subnet.

- **Optimized network performance**
  - Less broadcast means less wasted bandwidth.

- **Simplified management**
  - Issues can often be isolated to their own subnets.

- **Security**
  - Host traffic can be filtered using an IP access list.
**Class C Subnetting Example**

**Network:** 192.168.100.0/24

**Requirement:** 4 subnets

\[
2^x = 4 \text{ subnets} \\
2^2 = 4 \text{ subnets}
\]

\[
2^6 = 64 \quad \text{blocks} \quad \text{host}
\]

\[
62 = 64 - 2 \quad \text{network & broadcast}
\]

<table>
<thead>
<tr>
<th>Network ID</th>
<th>First IP</th>
<th>Last IP</th>
<th>Broadcast ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>.0</td>
<td>.1</td>
<td>.62</td>
<td>.63</td>
</tr>
<tr>
<td>.64</td>
<td>.65</td>
<td>.126</td>
<td>.127</td>
</tr>
<tr>
<td>.128</td>
<td>.129</td>
<td>.190</td>
<td>.191</td>
</tr>
<tr>
<td>.192</td>
<td>.193</td>
<td>.254</td>
<td>.255</td>
</tr>
</tbody>
</table>

**IP:** 32

**Network:** 26

**Host:** 6
Subnet Chart

- Start with the details of a **Class C** /24.
- Next, the CIDR increase by 1, the Block decreases by half.
  - 192.168.50.0/24 is a block of 256
  - 192.168.50.0/25 is a block of 128
- Next, take the last octet of the /24 (.0) and add the block size of a /25 (128), this will give you the last octet of the next subnet mask (.128).
  - /24 = 255.255.255.0 + 128
  - /25 = 255.255.255.128 + 64
  - /26 = 255.255.255.192
- Next, double the number of subnets as the CIDR increase by 1.
  - /24 is 1 subnet
  - /25 is 2 subnets

<table>
<thead>
<tr>
<th>CIDR</th>
<th>Block</th>
<th>Subnet Mask</th>
<th># of Subnets</th>
</tr>
</thead>
<tbody>
<tr>
<td>/24</td>
<td>256</td>
<td>255.255.255.0</td>
<td>1</td>
</tr>
<tr>
<td>/25</td>
<td>128</td>
<td>255.255.255.128</td>
<td>2</td>
</tr>
<tr>
<td>/26</td>
<td>64</td>
<td>255.255.255.192</td>
<td>4</td>
</tr>
<tr>
<td>/27</td>
<td>32</td>
<td>255.255.255.224</td>
<td>8</td>
</tr>
<tr>
<td>/28</td>
<td>16</td>
<td>255.255.255.240</td>
<td>16</td>
</tr>
<tr>
<td>/29</td>
<td>8</td>
<td>255.255.255.248</td>
<td>32</td>
</tr>
<tr>
<td>/30</td>
<td>4</td>
<td>255.255.255.252</td>
<td>64</td>
</tr>
<tr>
<td>/31</td>
<td>2</td>
<td>255.255.255.254</td>
<td>128</td>
</tr>
</tbody>
</table>
VLSM Example

<table>
<thead>
<tr>
<th>Departments</th>
<th># of Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>50</td>
</tr>
<tr>
<td>Marketing</td>
<td>15</td>
</tr>
<tr>
<td>Engineering</td>
<td>7</td>
</tr>
<tr>
<td>Link A</td>
<td>2</td>
</tr>
<tr>
<td>Link B</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network ID</th>
<th>Router IP</th>
<th>Subnet Mask</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.50.0/26</td>
<td>192.168.50.1</td>
<td>255.255.255.192</td>
<td>64</td>
</tr>
<tr>
<td>192.168.50.64/27</td>
<td>192.168.50.65</td>
<td>255.255.255.224</td>
<td>32</td>
</tr>
<tr>
<td>192.168.50.96/28</td>
<td>192.168.50.97</td>
<td>255.255.255.240</td>
<td>16</td>
</tr>
<tr>
<td>192.168.50.112/30</td>
<td>192.168.50.113</td>
<td>255.255.255.252</td>
<td>4</td>
</tr>
<tr>
<td>192.168.50.116/30</td>
<td>192.168.50.117</td>
<td>255.255.255.252</td>
<td>4</td>
</tr>
</tbody>
</table>
**VLSM (Variable Length Subnet Masking)**

**VLSM** is a subnetting design that allows you to create subnets of varying sizes, which means you’ll waste fewer addresses.

<table>
<thead>
<tr>
<th>CIDR</th>
<th>VLSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>All subnets are the same size</td>
<td>Subnet size can vary</td>
</tr>
<tr>
<td>Can be very wasteful of IP addresses</td>
<td>Waste fewer IP addresses</td>
</tr>
<tr>
<td>Subnet order doesn’t matter since they’re all the same size</td>
<td>Must be designed in size order from the largest subnet to smallest subnet</td>
</tr>
</tbody>
</table>
Class B Subnetting Example

Network: 172.20.0.0/16 + 2 = /18

Requirement: 4 subnets

Network ID | First IP | Last IP | Broadcast ID
---|---|---|---
.0.0 | .0.1 | .63.254 | .63.255
.64.0 | .64.1 | .127.254 | .127.255
.128.0 | .128.1 | .191.254 | .191.255
.192.0 | .192.1 | .255.254 | .255.255

IP: 32

Network: /18
Host: 14

2^x = 4 subnets
2^2 = 4 subnets

2^{14} = 16,384 - 2 = 16,382

network & broadcast

256 - 192 = 64

Magic number increment

/18
8 bits 11111111 = 255
8 bits 11111111 = 255
8 bits 11000000 = 192
2 bits 00000000 = 0

Increment 40
**Summarization** is done to consolidate the many subnets configured on a router to a single network address. This is done to minimize the size of a neighbor routers routing table.
Class A Subnetting Example

Network: 10.0.0.0/8 + 2 = /10

Requirement: 4 subnets

2^x = 4 subnets
2^2 = 4 subnets

2^22 = 4,194,304 - 2 = 4,194,302 block

Network ID | First IP | Last IP | Broadcast ID
---|---|---|---
.0.0.0 | .0.0.1 | .63.255.254 | .63.255.255
.64.0.0 | .64.0.1 | .127.255.254 | .127.255.255
.128.0.0 | .128.0.1 | .191.255.254 | .191.255.255
.192.0.0 | .192.0.1 | .255.255.254 | .255.255.255
## Subnet Chart

<table>
<thead>
<tr>
<th>CIDR</th>
<th>Block</th>
<th>Subnet Mask</th>
<th>Class C</th>
<th>Class B</th>
<th>Class A</th>
</tr>
</thead>
<tbody>
<tr>
<td>/8</td>
<td>16,777,216</td>
<td>255.0.0.0</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>/9</td>
<td>8,388,608</td>
<td>255.128.0.0</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>/10</td>
<td>4,194,304</td>
<td>255.192.0.0</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>/11</td>
<td>2,097,152</td>
<td>255.224.0.0</td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>/12</td>
<td>1,048,576</td>
<td>255.240.0.0</td>
<td></td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>/13</td>
<td>524,288</td>
<td>255.248.0.0</td>
<td></td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>/14</td>
<td>262,144</td>
<td>255.252.0.0</td>
<td></td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>/15</td>
<td>131,072</td>
<td>255.254.0.0</td>
<td></td>
<td></td>
<td>128</td>
</tr>
<tr>
<td>/16</td>
<td>65,536</td>
<td>255.255.0.0</td>
<td>1</td>
<td></td>
<td>256</td>
</tr>
<tr>
<td>/17</td>
<td>32,768</td>
<td>255.255.128.0</td>
<td>2</td>
<td></td>
<td>512</td>
</tr>
<tr>
<td>/18</td>
<td>16,384</td>
<td>255.255.192.0</td>
<td>4</td>
<td></td>
<td>1,024</td>
</tr>
<tr>
<td>/19</td>
<td>8,192</td>
<td>255.255.224.0</td>
<td>8</td>
<td></td>
<td>2,048</td>
</tr>
<tr>
<td>/20</td>
<td>4,096</td>
<td>255.255.240.0</td>
<td>16</td>
<td></td>
<td>4,096</td>
</tr>
<tr>
<td>/21</td>
<td>2,048</td>
<td>255.255.248.0</td>
<td>32</td>
<td></td>
<td>8,192</td>
</tr>
<tr>
<td>/22</td>
<td>1,024</td>
<td>255.255.252.0</td>
<td>64</td>
<td></td>
<td>16,384</td>
</tr>
<tr>
<td>/23</td>
<td>512</td>
<td>255.255.254.0</td>
<td></td>
<td>128</td>
<td>32,768</td>
</tr>
<tr>
<td>/24</td>
<td>256</td>
<td>255.255.255.0</td>
<td>1</td>
<td></td>
<td>256</td>
</tr>
<tr>
<td>/25</td>
<td>128</td>
<td>255.255.255.128</td>
<td>2</td>
<td>512</td>
<td>131,072</td>
</tr>
<tr>
<td>/26</td>
<td>64</td>
<td>255.255.255.192</td>
<td>4</td>
<td>1,024</td>
<td>262,144</td>
</tr>
<tr>
<td>/27</td>
<td>32</td>
<td>255.255.255.224</td>
<td>8</td>
<td>2,048</td>
<td>524,288</td>
</tr>
<tr>
<td>/28</td>
<td>16</td>
<td>255.255.255.240</td>
<td>16</td>
<td>4,096</td>
<td>1,048,576</td>
</tr>
<tr>
<td>/29</td>
<td>8</td>
<td>255.255.255.248</td>
<td>32</td>
<td>8,192</td>
<td>2,097,152</td>
</tr>
<tr>
<td>/30</td>
<td>4</td>
<td>255.255.255.252</td>
<td>64</td>
<td>16,384</td>
<td>4,194,304</td>
</tr>
<tr>
<td>/31</td>
<td>2</td>
<td>255.255.255.254</td>
<td>128</td>
<td>32,768</td>
<td>8,388,608</td>
</tr>
</tbody>
</table>
Cisco's Internetworking Operating System (IOS)

Access lines, Basic device configurations, Device components, Backup and Restoring, Password Resets
Every Cisco router and switch runs a version of the Cisco IOS.
- The current versions valid for your exam are **IOS 12.4 – 15.3**.

The Cisco IOS is a Command Line (CLI) based operating system which means the only way we’ll configure it is by typing commands into the prompt.

You can access the IOS device via one of these three access lines below.
- **Console Line**
- **Auxiliary Line**
- **Virtual Terminal Lines (Telnet/SSH)**
Access Lines

- **Console Line** is reserved for physical access
  - Connect to a router or switch using a Rolled cable which has a **Serial/RS-232/DB9** connection on one end and a **RJ45** on the other.

- **Auxiliary Line** is usually reserved for Modem access
  - You usually connect a serial **dial-in modem** to access your equipment when there’s no chance of Telnet/SSH access or if physical access is not possible/feasible. Only routers include an Auxiliary port.

- **VTY Line (Telnet/SSH)** is your main access
  - Once configured you use your **Telnet/SSH** client to gain access to the router or switch
Access Modes

Once you log-in to your device there are different three modes which you have access to. Each mode will have its own set of commands for you to le

- **User EXEC**: This mode does not allow you change any configurations. It is the least privileged access.

  User Exec mode prompt
  router>

- **Privileged EXEC**: This mode allows you to query the device but not to change any configurations.

  Privilege Exec mode prompt
  router#

- **Global configuration**: This mode is where all configurations can be made.

  Global Configuration mode prompt
  router(config)#
Exercise 1 – Basic Device Configurations

1. Change Hostname
2. Configure the current time and date
3. Protect privileged mode with a secret
4. Protect Access lines with a password
   - Console line
   - Auxiliary line
   - First 5 VTY lines (telnet/ssh)
5. Enable password encryption
6. Configure an MOTD banner with the hostname and location of the router.
7. Configure any 3 interfaces with any valid IPv4 address
8. Verify all your configurations in the running configurations
9. Save your configurations
10. Verify your saved configurations in the startup configurations
11. Erase your startup-config
12. Verify that your configurations are erased
13. Reboot the router
Basic IOS Commands

Some commands require that you supply parameters, those are indicated by text inside of “< >”

eg; hostname <hostname> means replace <hostname> with the actual name to be set. hostname R1

Enter Privileged mode from User mode

`router> enable`

Return to User mode from Privileged mode

`router# disable`

Enter Global Configuration mode from Privileged mode

`router# configure terminal`

Exit Global Configuration mode return to Privileged mode

`router(config)# exit`

Set a hostname

`router(config)# hostname <hostname>`

Set a banner

`router(config)# banner <type> <delimiter>`
**Basic IOS Commands continued..**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>enable password &lt;password&gt;</code></td>
<td>Protect privilege mode with a password which is clear text by default</td>
</tr>
<tr>
<td><code>enable secret &lt;secret&gt;</code></td>
<td>Protect privilege mode with a secret which is always encrypted</td>
</tr>
<tr>
<td><code>service password-encryption</code></td>
<td>Encrypts all existing passwords and all new passwords.</td>
</tr>
<tr>
<td><code>line &lt;line name&gt; &lt;line number&gt;</code></td>
<td>Enter a line by name and number</td>
</tr>
<tr>
<td><code>password &lt;password&gt;</code></td>
<td>Protect a line with a password</td>
</tr>
<tr>
<td><code>login</code></td>
<td>Enable password prompt at login</td>
</tr>
<tr>
<td><code>transport input &lt;protocol-1&gt; &lt;protocol-2&gt;</code></td>
<td>Set VTY line method (SSH, Telnet)</td>
</tr>
<tr>
<td><code>transport input ssh</code></td>
<td></td>
</tr>
<tr>
<td><code>transport input ssh telnet</code></td>
<td></td>
</tr>
</tbody>
</table>
**Basic IOS Commands continued...**

**Enter an interface by name and number**
```
router(config)# interface <interface name> <interface number>
router(config)# interface fastethernet 0/0
```

**Set an IPv4 address on an interface**
```
router(config-if)# ip address <ip address> <subnet mask>
router(config-if)# ip address 192.168.250.1 255.255.255.0
```

**Set a plain text description on an interface**
```
router(config-if)# description <your description>
router(config-if)# description Link to Accounting in suite 227
```

**Turn the interface on by removing it from shutdown mode (administratively down)**
```
router(config-if)# no shutdown
```
Configure the time

```
router# clock set <hh:mm:ss> <date/month> <month/date> <year>
```

```
router# clock set 15:35:00 31 March 2017
```

```
router# clock set 21:50:00 March 31 2017
```

Save your current configurations to the NVRAM

```
router# copy running-config startup-config
```

Erase your saved configurations from the NVRAM

```
router# erase startup-config
```

Reboot the device

```
router# reload
```
## Basic IOS Show Commands

<table>
<thead>
<tr>
<th>Command Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displays your current configurations</td>
<td>router# show running-config</td>
</tr>
<tr>
<td>Displays your saved configurations</td>
<td>router# show startup-config</td>
</tr>
<tr>
<td>Displays detailed interface information</td>
<td>router# show interface</td>
</tr>
<tr>
<td>Displays detailed layer 3 interface information</td>
<td>router# show ip interface</td>
</tr>
<tr>
<td>Displays basic layer 3 interface information</td>
<td>router# show ip interface brief</td>
</tr>
<tr>
<td>Displays last 10 commands</td>
<td>router# show history</td>
</tr>
<tr>
<td>Displays hardware, IOS and other device information. Same output as POST.</td>
<td>router# show version</td>
</tr>
</tbody>
</table>
## Enhanced editing commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+A</td>
<td>Moves your cursor to the beginning of the line</td>
</tr>
<tr>
<td>Ctrl+E</td>
<td>Moves your cursor to the end of the line</td>
</tr>
<tr>
<td>Esc+B</td>
<td>Moves back one word</td>
</tr>
<tr>
<td>Ctrl+B</td>
<td>Moves back one character</td>
</tr>
<tr>
<td>Ctrl+F</td>
<td>Moves forward one character</td>
</tr>
<tr>
<td>Esc+F</td>
<td>Moves forward one word</td>
</tr>
<tr>
<td>Ctrl+D</td>
<td>Deletes a single character</td>
</tr>
<tr>
<td>Backspace</td>
<td>Deletes a single character</td>
</tr>
<tr>
<td>Ctrl+R</td>
<td>Redisplays a line</td>
</tr>
<tr>
<td>Ctrl+U</td>
<td>Erases a line</td>
</tr>
<tr>
<td>Ctrl+W</td>
<td>Erases a word</td>
</tr>
<tr>
<td>Ctrl+Z</td>
<td>Ends configuration mode and returns to EXEC</td>
</tr>
<tr>
<td>Tab</td>
<td>Finishes typing a command for you</td>
</tr>
</tbody>
</table>

The table lists the enhanced editing commands available on a Cisco IOS device.
Be aware of the functions of each of the following components.

- **ROM (Read only memory)**
  - **Bootstrap**
    - Instructs the router on how to load the IOS
  - **POST (Power On Self Test)**
    - Displays device details while performing a simple hardware test
  - **Mini- IOS**
    - Small operating system used to modify the boot process or troubleshoot the boot process when an IOS device won’t boot.
  - **ROMMON (ROM Monitor)**
    - ROMMON is displayed when in the Mini- IOS to distinguish it from the IOS
  - **Configuration Register**
    - Is a hexadecimal memory address that controls the boot process
Device Components

- **Flash**
  - Cisco IOS (Internetwork Operating System)
    - Stores the IOS in a single “.bin” file.

- **RAM (Random Access Memory)**
  - Running Configurations
    - Live, production settings.

- **NVRAM (Non-Volatile Random Access Memory)**
  - Startup Configurations
    - Saved settings.
Backing up IOS and Configurations

Backing up configurations can be done by either copying the `startup-config` or the `running-config` so be aware of which one they ask for on the test.

- **Backup current settings**
  - `copy running-config tftp`

- **Backup saved settings**
  - `copy startup-config tftp`

- **Restore configurations into operations**
  - `copy tftp running-config`

- **Restore configurations to NVRAM**
  - `copy tftp startup-config`

Backing up your IOS requires you to know the exact name of the file as it’s stored in the flash.

- **Backup your IOS**
  - `copy flash tftp`

- **Restore or upgrade your IOS**
  - `copy tftp flash`
Router Password Reset

1. Interrupt the boot process to access the Mini-IOS. This is done via the console line.
   - Hold **Control** key and press the **Break** key
2. Modify the configuration register address to bypass loading the NVRAM.
   - `confreg 0x2142`
3. Reboot the router from within the Mini-IOS.
   - `reset`
4. Gain privilege access.
   - `enable`
5. Restore the saved configurations to maintain the production environment.
   - `Copy startup-config running-config`
6. Change any password that is not known.
7. Restore the default configuration register address.
   - `config-register 0x2102`
8. Save your current settings and passwords.
   - `Copy running-config startup-config`
IP Routing

Static routing, Default route.
Network Routing Concepts

**RA**
- C – 1
- C – 3
- S – 2 > 3B
- S – 4 > 3B
- S – 5 > 3B
- S – 6 > 3B
- S – 7 > 3B

**RB**
- C – 2
- C – 3
- S – 1 > 3A
- C – 5
- S – 4 > 5C
- S – 6 > 5C
- S – 7 > 5C

**RC**
- C – 4
- C – 5
- S – 1 > 5B
- S – 2 > 5B
- S – 3 > 5B
- C – 7
- S – 6 > 7D

**RD**
- C – 6
- C – 7
- S – 1 > 7C
- S – 2 > 7C
- S – 3 > 7C
- S – 4 > 7C
- S – 5 > 7C
Static Routing Example

R1(config)# ip route 192.168.200.0 255.255.255.0 192.168.50.2

R2(config)# ip route 192.168.100.0 255.255.255.0 192.168.50.1
Static Route verification outputs

R1# show ip interface brief

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>OK? Method</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastEthernet0/0</td>
<td>192.168.100.1</td>
<td>YES manual</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>Serial0/0/0</td>
<td>192.168.50.1</td>
<td>YES manual</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>

R1# show ip route

[OUTPUT CUT]

C 192.168.50.0/24 is directly connected, Serial0/0/0
C 192.168.100.0/24 is directly connected, FastEthernet0/0
S 192.168.200.0/24 [1/0] via 192.168.50.2
R1(config)# ip route 192.168.200.0 255.255.255.0 s0/0/0

R2(config)# ip route 192.168.100.0 255.255.255.0 s0/1/1
Static Route verification outputs exit interface

R1# sh ip interface brief

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>OK?</th>
<th>Method</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastEthernet0/0</td>
<td>192.168.100.1</td>
<td>YES</td>
<td>manual</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>Serial0/0/0</td>
<td>192.168.50.1</td>
<td>YES</td>
<td>manual</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>

R1# sh ip route

[OUTPUT CUT]

C  192.168.50.0/24 is directly connected, Serial0/0/0
C  192.168.100.0/24 is directly connected, FastEthernet0/0
S  192.168.200.0/24 is directly connected, Serial0/0/0
Default Route

The **default route** is also known as the “**gateway of last resort**”, when the router does not have a matching route it will forward the packet using the default route.

```
ip route 0.0.0.0 0.0.0.0 192.168.50.1
or
ip route 0.0.0.0 0.0.0.0 s0/0/0
```

An IP address of **0.0.0.0** means all IP addresses.
A Subnet mask of **0.0.0.0** means all networks.
Static Routes

The "**ip route**" command is used to set a static route.

- **destination network**: the network address of the network you need to route to
- **mask**: the subnet mask of the destination network
- **next hop/exit interface**: the ip address of your neighbor’s interface that leads to the destination network or your routers interface that leads to the destination network
- **administrative distance (AD)**: a metric that lets you set the priority of the route, lower number is better (optional)
- **permanent**: keeps the route in the routing table even if the next hop is not available (optional)

```
ip route [destination] [mask] [next_hop/exit_interface] [AD] [permanent]
ip route 192.168.50.0 255.255.255.0 192.168.51.1 150
ip route 172.16.0.0 255.255.0.0 10.100.100.2 permanent
```
Routing Protocols

Routing Protocol Categories, IGP protocol types
There are two categories of routing protocols

- **Interior Gateway protocols (IGP):** protocols are used to route a **single autonomous system (AS)**
  - Used to route **LAN to LAN**
  - Includes; RIP, IGRP, IS-IS, OSPF, EIGRP

- **Exterior Gateway protocols (EGP):** protocols are used to route between **multiple autonomous systems**
  - Used to route **WAN to WAN**
  - Includes; BGP
There are two types of routing protocols; Link-state and Distance-vector.

- **Distance-vector** is hops based and take the path with the fewest amounts of hops. Limited to a maximum of 15 hops.
  - RIPv1 and RIPv2 are distance-vector protocols.

- **Link-state** is bandwidth based and take the path with the fastest link. Since they don’t measure hops they have an unlimited hop count.
  - OSPF is the only link-state protocol on your test.
R1 needs to get a packet to R5

It’ll take the path with the fewest amount of hops
R1 needs to get a packet to R5

It’ll take the path with the fastest link.
## IGP Routing Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Type</th>
<th>Metric</th>
<th>AD*</th>
<th>Communication</th>
<th>Convergence</th>
<th>Summarization</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIPv1</td>
<td>Distance Vector</td>
<td>Hop count</td>
<td>120</td>
<td>Periodic Broadcast</td>
<td>Slow</td>
<td>Auto Classful only</td>
<td>Bellman-Ford</td>
</tr>
<tr>
<td>RIPv2</td>
<td>Distance Vector</td>
<td>Hop count</td>
<td>120</td>
<td>Periodic Multicast</td>
<td>Slow</td>
<td>Auto &amp; Manual Classless</td>
<td>Bellman-Ford</td>
</tr>
<tr>
<td>OSPF</td>
<td>Link State</td>
<td>Bandwidth</td>
<td>110</td>
<td>Multicast on change via (LSA*)</td>
<td>Fast</td>
<td>Manual Classless</td>
<td>Dijkstra</td>
</tr>
<tr>
<td>EIGRP</td>
<td>Advanced Distance Vector</td>
<td>Bandwidth, hops, reliability and delay</td>
<td>90</td>
<td>Reliable Multicast on change via (RTP*)</td>
<td>Fast</td>
<td>Auto &amp; Manual Classless</td>
<td>Diffusing Update Algorithm (DUAL)</td>
</tr>
</tbody>
</table>

**Key**
- AD* = Administrative Distance
- LSA* = Link State Advertisement
- RTP* = Reliable Transport Protocol
Routing Information Protocol version 2 (RIPv2)

- **RIPv2** is a distance vector routing protocol that uses the Bellman-Ford algorithm to find the path with the fewest amount of hops.

**Main features of RIPv2**
- Periodic full routing table updates.
- Routing loop prevention via Split Horizon.
- Supports VLSM/CIDR discontiguous networks
- Supports Auto and Manual route summarization
- Sends updates via the multicast address of **224.0.0.9**
- Open standard works on any enterprise class router

**RIP timers**
- **Update - 30 seconds**: full routing table updates are sent via a multicast periodically
- **Invalid - 180 seconds**: till a route is marked as invalid
- **Hold-down - 180 seconds**: uptime must be maintain before a route is used
- **Flush - 240 seconds**: till an invalid route is removed from the routing table
RIP Terminology

- **Link**: refers to the network or interfaces between two routers. Routers must have a common link to share routing information.

- **Split Horizon**: is used to prevent routing loops by assuring that updates sent out by a router about its own networks are NOT sent back to that router.

- **Route Poisoning**: is when a route is marked as invalid after 180 seconds of inactivity. This is done by setting the metric of the route to **infinity (16 hops)**. The route is then removed after the 240 second flush timer elapses.
RIPv2

R1(config)# router rip
R1(config-router)# version 2
R1(config-router)# network 172.20.100.0
R1(config-router)# network 10.20.0.0
R1(config-router)# passive-interface f0/0

R2(config)# router rip
R2(config-router)# version 2
R2(config-router)# network 172.20.100.0
R2(config-router)# network 10.30.0.0
R2(config-router)# passive-interface f0/0
## RIP configuration commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>router(config)# router rip</code></td>
<td>Create/Enter RIP process</td>
</tr>
<tr>
<td><code>router(config-router)# version 2</code></td>
<td>Enable RIP version 2</td>
</tr>
<tr>
<td><code>router(config-router)# network &lt;network address&gt;</code></td>
<td>Set network advertisement for your own networks</td>
</tr>
<tr>
<td><code>router(config-router)# passive-interface &lt;interface&gt;</code></td>
<td>Set interface to passive so it does not process routing table updates</td>
</tr>
<tr>
<td><code>router(config-router)# no auto-summary</code></td>
<td>Disable auto-summarization (on by default)</td>
</tr>
</tbody>
</table>
# RIP verification commands

<table>
<thead>
<tr>
<th>Display the routing table (Active routes)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>router# show ip route</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Display RIP configurations, RIP version, network advertisements, passive interfaces, RIP timers.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>router# show ip protocols</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Display RIP database (All routes including redundant ones)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>router# show ip rip database</code></td>
</tr>
</tbody>
</table>
## RIP verification outputs

R1# `show ip route`

```
[OUTPUT CUT]
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
R  10.0.0.0/8 [120/1] via 172.20.100.2, 00:00:09, Serial0/0/0
C   10.20.0.0/16 is directly connected, FastEthernet0/0
C   172.20.0.0/24 is subnetted, 1 subnets
C   172.20.100.0 is directly connected, Serial0/0/0
```

R1# `show ip rip database`

```
10.0.0.0/8    auto-summary
10.0.0.0/8    [1] via 172.20.100.2, 00:00:08, Serial0/0/0
10.20.0.0/16   auto-summary
10.20.0.0/16   directly connected, FastEthernet0/0
172.20.100.0/24   auto-summary
172.20.100.0/24   directly connected, Serial0/0/0
```
RIP verification outputs

R1#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 8 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 2

<table>
<thead>
<tr>
<th>Interface</th>
<th>Send</th>
<th>Recv</th>
<th>Triggered RIP</th>
<th>Key-chain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial0/0/0</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Automatic network summarization is not in effect
Maximum path: 4
Routing for Networks:
  10.0.0.0
  172.20.0.0
Passive Interface(s):
  FastEthernet0/0
Routing Information Sources:
  Gateway         Distance      Last Update
  172.20.100.2         120      00:00:08
Distance: (default is 120)
Open Shortest Path First (OSPF)

- OSPF is a **link-state** routing protocol which works by using the **Dijkstra** algorithm to build its routing table.

**Main features of OSPF**

- Fast convergence because it is event triggered
- Efficient routing table updates because it will Multicast only on change
- Supports VLSM/CIDR discontiguous networks
- No hop count limit
- Open standard works on any enterprise class router
- Hierarchical network design using areas
OSPF Terminology

- **Link**: refers to the network or interfaces between routers.
- **Adjacency**: is the relationship that must be formed between two OSPF routers that allows them to exchange routing table updates. Adjacencies can only be formed by routers that are in the same area.
- **OSPF Area**: grouping of routers which is used to create a hierarchical network design. OSPF routers must be in the same area to form an adjacency.
- **Router ID**: is a unique identifier used by OSPF routers to communicate with each other. It can be set by using the `router-id` command, but if it’s not set the highest IP address on the loopback interface will become your router-id, if the loopback is not set then it’s the highest IP on a physical interface.
OSPF Terminology continued...

- **Designated router**: is determined by the **priority** set on the router (highest number wins). If the priority matches then the **Router-id is the tie breaker** (highest Router-id wins). The designated router (DR) forms adjacencies to all routers so it can send and receive all routing table updates.

- **Backup Designated router**: is there as a backup for the DR. It’s the router with the next highest priority or the next highest Router-ID. It receives all routing table updates but does NOT send them out.

- **Hello Protocol**: used to discover other OSPF routers and maintains the adjacency between them.

- **Link State Advertisement**: is an OSPF data packet that sends out link-state and routing information. Only routers that form an adjacency can exchange link state advertisements (**LSA**)
OSPF Terminology continued……

- **Neighborship database**: list of all OSPF routers. This is a unique feature of OSPF.
- **Topological database**: list of all OSPF paths (routes).
- **Multi-access (Broadcast) / Non-broadcast multi-access (NBMA)**: are types of networks that require a DR and a BDR. This is determined by the type of WAN link you’re connected to.
- **Point-to-Point / Point-to-multipoint**: Do **NOT** require a DR or BDR. This is determined by the type of WAN link you’re connected to.
OSPFv2 IPv4 (Single Area) by Advertising

R2(config) router ospf 1
R2(config-router) router-id 2.2.2.2
R2(config-router) network 172.20.0.160 0.0.0.15 area 0
R2(config-router) network 172.20.0.180 0.0.0.3 area 0

R1(config) router ospf 1
R1(config-router) router-id 1.1.1.1
R1(config-router) network 172.20.0.128 0.0.0.31 area 0
R1(config-router) network 172.20.0.176 0.0.0.3 area 0

Corp(config) router ospf 1
Corp(config-router) router-id 99.99.99.99
Corp(config-router) network 172.20.0.0 0.0.0.127 area 0
Corp(config-router) network 172.20.0.176 0.0.0.3 area 0
Corp(config-router) network 172.20.0.180 0.0.0.3 area 0

R1(config) router ospf 1
R1(config-router) router-id 1.1.1.1
R1(config-router) network 172.20.0.128 0.0.0.31 area 0
R1(config-router) network 172.20.0.176 0.0.0.3 area 0
OSPFv2 IPv4 (Single Area) by Interface

R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# int g0/0
R1(config-if)# ip ospf 1 area 0
R1(config-if)# int s0/0/0
R1(config-if)# ip ospf 1 area 0

Corp(config)# router ospf 1
Corp(config-router)# router-id 99.99.99.99
Corp(config-router)# int g0/0
Corp(config-if)# ip ospf 1 area 0
Corp(config-router)# int s0/0/0
Corp(config-if)# ip ospf 1 area 0
Corp(config-router)# int s0/0/1
Corp(config-if)# ip ospf 1 area 0

R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# int g0/0
R2(config-if)# ip ospf 1 area 0
R2(config-if)# int s0/0/1
R2(config-if)# ip ospf 1 area 0
## OSPF configuration commands

<table>
<thead>
<tr>
<th>Command Description</th>
<th>Command Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create/Enter OSPF process</td>
<td><code>router(config)# router ospf &lt;process-id&gt;</code></td>
</tr>
<tr>
<td>Set Router-id</td>
<td><code>router(config-router)# router-id &lt;actual router-id to be set&gt;</code></td>
</tr>
<tr>
<td>Set network advertisement</td>
<td><code>router(config-router)# network &lt;network address&gt; &lt;wildcard mask&gt; area &lt;area number&gt;</code></td>
</tr>
<tr>
<td>Place an interface into the OSPF process</td>
<td><code>router(config-if)# ip ospf &lt;ospf process-id&gt; area &lt;area number&gt;</code></td>
</tr>
<tr>
<td>Set interface to passive</td>
<td><code>router(config-router)# passive-interface &lt;interface name &amp; number&gt;</code></td>
</tr>
</tbody>
</table>
## OSPF verification commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>router# show ip ospf</code></td>
<td>Display OSPF configurations, like router ID, area information, SPF stats, LSA timers</td>
</tr>
<tr>
<td><code>router# show ip ospf database</code></td>
<td>Display the Link ID of all routers and which neighbor is advertising that router to you.</td>
</tr>
<tr>
<td><code>router# show ip ospf neighbor</code></td>
<td>Display your OSPF neighbors</td>
</tr>
<tr>
<td><code>router# show ip ospf interface</code></td>
<td>Display interfaces IP, Area, Process ID, Router ID, Network type, Cost, Priority, DR/BDR election, and timers.</td>
</tr>
<tr>
<td><code>router# show ip protocols</code></td>
<td>Display OSPF configurations you have set, Process ID, Router ID, Network advertisements.</td>
</tr>
</tbody>
</table>
Corp# show ip route

172.20.0.0/16 is variably subnetted, 8 subnets, 5 masks
C 172.20.0.0/25 is directly connected, GigabitEthernet0/0
L 172.20.0.1/32 is directly connected, GigabitEthernet0/0
O 172.20.0.128/27 [110/65] via 172.20.0.177, 00:03:18, Serial0/0/0
O 172.20.0.160/28 [110/65] via 172.20.0.181, 00:03:08, Serial0/0/1
C 172.20.0.176/30 is directly connected, Serial0/0/0
L 172.20.0.178/32 is directly connected, Serial0/0/0
C 172.20.0.180/30 is directly connected, Serial0/0/1
L 172.20.0.181/32 is directly connected, Serial0/0/1
Corp# 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 99.99.99.99
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Maximum path: 4
Routing for Networks:
   172.20.0.0 0.0.0.31 area 0
   172.20.0.176 0.0.0.3 area 0
   172.20.0.180 0.0.0.3 area 0
Routing Information Sources:
   Gateway         Distance      Last Update
   1.1.1.1              110      00:04:37
   3.3.3.3              110      00:04:26
   99.99.99.99          110      00:04:26
Distance: (default is 110)
Corp# show ip ospf
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x000000
Number of opaque AS LSA 0. Checksum Sum 0x000000
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
Area BACKBONE(0)
Number of interfaces in this area is 3
Area has no authentication
SPF algorithm executed 3 times
Area ranges are
Number of LSA 3. Checksum Sum 0x015d6b
Number of opaque link LSA 0. Checksum Sum 0x000000
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
**OSPF verification outputs**

### Corp# `show ip ospf database`

**OSPF Router with ID (99.99.99.99) (Process ID 1)**

Router Link States (Area 0)

<table>
<thead>
<tr>
<th>Link ID</th>
<th>ADV Router</th>
<th>Age</th>
<th>Seq#</th>
<th>Checksum</th>
<th>Link count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.1</td>
<td>1.1.1.1</td>
<td>458</td>
<td>0x80000003</td>
<td>0x00eb2f</td>
<td>3</td>
</tr>
<tr>
<td>99.99.99.99</td>
<td>99.99.99.99</td>
<td>445</td>
<td>0x80000005</td>
<td>0x001c16</td>
<td>4</td>
</tr>
<tr>
<td>2.2.2.2</td>
<td>2.2.2.2</td>
<td>445</td>
<td>0x80000003</td>
<td>0x00cc0d</td>
<td>3</td>
</tr>
</tbody>
</table>

### Corp# `show ip ospf neighbor`

<table>
<thead>
<tr>
<th>Neighbor ID</th>
<th>Pri</th>
<th>State</th>
<th>Dead Time</th>
<th>Address</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.2.2</td>
<td>0</td>
<td>FULL/</td>
<td>00:00:37</td>
<td>172.20.0.182</td>
<td>Serial0/0/1</td>
</tr>
<tr>
<td>1.1.1.1</td>
<td>0</td>
<td>FULL/</td>
<td>00:00:33</td>
<td>172.20.0.177</td>
<td>Serial0/0/0</td>
</tr>
</tbody>
</table>
Corp# show ip ospf interface s0/0/0
Serial0/0/0 is up, line protocol is up
    Internet address is 172.20.0.178/30, Area 0
    Process ID 1, Router ID 99.99.99.99, Network Type POINT-TO-POINT, Cost: 64
    Transmit Delay is 1 sec, State POINT-TO-POINT, Priority 0
    No designated router on this network
    No backup designated router on this network
    Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
        Hello due in 00:00:05
    Index 1/1, flood queue length 0
    Next 0x0(0)/0x0(0)
    Last flood scan length is 1, maximum is 1
    Last flood scan time is 0 msec, maximum is 0 msec
    Neighbor Count is 1, Adjacent neighbor count is 1
        Adjacent with neighbor 1.1.1.1
    Suppress hello for 0 neighbor(s)
**Single Area OSPF**

**Problems with single Area OSPF**
- When a link goes up or down the routers will then do an LSA flood which will send updates to all the routers in that area.
- It’ll also cause high CPU and Memory overhead.
Multi-Area OSPF

**Backbone Router**: any router with an interface in area 0.

**Area Border Router (ABR)**: a router that has interfaces in multiple areas.

**Internal Router**: a router with interfaces in a single area.

**Autonomous system boundary router (ASBR)**: is a router with at least one interface connected to an external network or different AS.
OSPFv2 IPv4 (Multi-Area)

R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 172.20.0.128 0.0.0.31 area 1
R1(config-router)# network 172.20.0.176 0.0.0.3 area 0

Corp(config)# router ospf 1
Corp(config-router)# router-id 99.99.99.99
Corp(config-router)# network 172.20.0.0 0.0.0.127 area 0
Corp(config-router)# network 172.20.0.176 0.0.0.3 area 0
Corp(config-router)# network 172.20.0.180 0.0.0.3 area 0

R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 172.20.0.160 0.0.0.15 area 2
R2(config-router)# network 172.20.0.180 0.0.0.3 area 0
Enhanced IGRP (EIGRP)

- EIGRP is an advanced distance vector routing protocol.

**Main features of EIGRP**
- Supports IPv4 and IPv6
- Supports VLSM/CIDR discontiguous networks
- Supports Auto and Manual summarization
- Supports equal and unequal load balancing
- Efficient neighbor discovery: these 3 things must happen so you can become neighbors
  - Hello or ACK is received
  - AS numbers have to match
  - Identical metrics (K values are used to “weigh” a metric)
    - K1– Bandwidth, K2-Load, K3-Delay, K4-Reliability and K5-MTU
- Communicates using the **Reliable Transport Protocol (RTP)** thru a reliable multicast.
- Route is selected via the **Diffusing Update algorithm (DUAL)**
EIGRP Terminology

- **Link**: refers to the network or interfaces between routers.
- **Adjacency**: is the relationship that must be formed between two EIGRP routers that allows them to exchange routing table updates. Adjacencies can only be formed by routers that are in the same **AS (Autonomous System)**.
- **Feasible distance (FD)**: metric given to a route by **DUAL**. Route with the lowest FD is considered the best route.
- **Reported/Advertised Distance**: Metric given to a route by your neighbor
EIGRP Terminology continued...

- **Successor**: is the route with the lowest feasible distance (FD) to a given network. If the FD is the same for multiple routes EIGRP will load balance those successors.

- **Feasible Successor**: any route that is not the best route. EIGRP can store up to 32 feasible successors in IOS 15.0 and later.

- **Neighbor table**: List of directly connected EIGRP routers

- **Topology table**: List of all EIGRP Feasible Successor and Successor

- **Split Horizon**: is used to prevent routing loops by assuring that updates sent out by a router about it’s own networks are NOT sent back to that router.
EIGRPv6 IPv4 without Wildcard

R1(config)# router eigrp 10
R1(config-router)# network 172.25.200.0
R1(config-router)# network 172.25.0.0
R1(config-router)# passive-interface f0/0

R2(config)# router eigrp 10
R2(config-router)# network 172.25.200.0
R2(config-router)# network 172.25.1.0
R2(config-router)# passive-interface f0/0
EIGRPv6 IPv4 with Wildcard

**Diagram:**
- **R1**
  - `S0/0/0` 172.25.10.1/30
  - `G0/0` 172.25.100.1/24
- **R2**
  - `S0/0/0` 172.25.10.2/30
  - `G0/0` 172.25.200.1/24
- **S1**
- **S2**

**Commands:**

**R1(config)# router eigrp 100**
- **R1(config-router)# network 172.25.100.0 0.0.0.255**
- **R1(config-router)# network 172.25.10.0 0.0.0.3**
- **R1(config-router)# passive-interface g0/0**

**R2(config)# router eigrp 100**
- **R2(config-router)# network 172.25.200.0 0.0.0.255**
- **R2(config-router)# network 172.25.10.0 0.0.0.3**
- **R2(config-router)# passive-interface g0/0**
## EIGRP configuration commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enter EIGRP configurations</strong></td>
<td><code>Router(config)# router eigrp &lt;autonomous system number&gt;</code></td>
</tr>
<tr>
<td><strong>Set network advertisement</strong></td>
<td><code>Router(config-router)# network &lt;network address&gt;</code></td>
</tr>
<tr>
<td><strong>Disable auto-summary</strong></td>
<td><code>Router(config-router)# no auto-summary</code></td>
</tr>
<tr>
<td><strong>Set interface to passive</strong></td>
<td><code>Router(config-router)# passive-interface &lt;interface name&gt; &lt;interface number&gt;</code></td>
</tr>
<tr>
<td><strong>Set maximum number of feasible successors, default is 4</strong></td>
<td><code>Router(config-router)# maximum-path &lt;number between 1 - 32&gt;</code></td>
</tr>
<tr>
<td><strong>Set maximum number of hops, default is 100</strong></td>
<td><code>Router(config-router)# maximum-hops &lt;number between 1 - 255&gt;</code></td>
</tr>
</tbody>
</table>
EIGRP verification commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Router# show ip eigrp topology</code></td>
<td>Show all EIGRP paths, includes Successors and Feasible Successors</td>
</tr>
<tr>
<td><code>Router# show ip eigrp traffic</code></td>
<td>Show EIGRP packet count, how many packets that are sent and received.</td>
</tr>
<tr>
<td><code>Router# show ip eigrp neighbors</code></td>
<td>Show your EIGRP directly connected neighbors</td>
</tr>
<tr>
<td><code>Router# show ip eigrp interface</code></td>
<td>Show which interfaces are routing with EIGRP</td>
</tr>
<tr>
<td><code>Router# show ip protocols</code></td>
<td>Show EIGRP configurations, AS number, Summarizations, Network advertisements, and K values</td>
</tr>
</tbody>
</table>
EIGRP verification output

R1# show ip protocols
Routing Protocol is "eigrp 100"

[OUTPUT CUT]
EIGRP-IPv4 Protocol for AS(100)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240
  Router-ID: 172.25.10.1
  Topology : 0 (base)
    Active Timer: 3 min
    Distance: internal 90 external 170
    Maximum path: 4
    Maximum hopcount 100
    Maximum metric variance 1
  Automatic Summarization: disabled
  Automatic address summarization:
    Maximum path: 4
  Routing for Networks:
    172.25.100.0/24
    172.25.10.0/30
### EIGRP verification output

#### R1# show ip route

[OUTPUT CUT]

- 172.25.0.0/16 is variably subnetted, 5 subnets, 3 masks
- C 172.25.10.0/30 is directly connected, Serial0/0/0
- L 172.25.10.1/32 is directly connected, Serial0/0/0
- C 172.25.100.0/24 is directly connected, GigabitEthernet0/0
- L 172.25.100.1/32 is directly connected, GigabitEthernet0/0
- D 172.25.200.0/24 [90/2170112] via 172.25.10.2, 00:12:05, Serial0/0/0

#### R1# show ip eigrp topology

**IP-EIGRP Topology Table for AS 100/ID(172.25.100.1)

<table>
<thead>
<tr>
<th>Code</th>
<th>Network Address</th>
<th>Successors</th>
<th>Forwarding Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>172.25.10.0/30</td>
<td>1</td>
<td>2169856</td>
</tr>
<tr>
<td></td>
<td>via Connected, Serial0/0/0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>172.25.100.0/24</td>
<td>1</td>
<td>2816</td>
</tr>
<tr>
<td></td>
<td>via Connected, GigabitEthernet0/0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>172.25.200.0/24</td>
<td>1</td>
<td>2170112</td>
</tr>
<tr>
<td></td>
<td>via 172.25.10.2 (2170112/2816), Serial0/0/0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### EIGRP verification output

**R1# show ip eigrp neighbors**

<table>
<thead>
<tr>
<th>H</th>
<th>Address</th>
<th>Interface</th>
<th>Hold Uptime</th>
<th>SRTT</th>
<th>RTO</th>
<th>Q</th>
<th>Seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>172.25.10.2</td>
<td>Se0/0/0</td>
<td>10</td>
<td>00:15:55</td>
<td>40</td>
<td>1000</td>
<td>0</td>
</tr>
</tbody>
</table>

**R1# show ip eigrp interfaces**

<table>
<thead>
<tr>
<th>Interface</th>
<th>Peers</th>
<th>Xmit Queue</th>
<th>Mean SRTT</th>
<th>Pacing Time</th>
<th>Multicast</th>
<th>Pending Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gig0/0</td>
<td>0</td>
<td>0/0</td>
<td>1236</td>
<td>0/10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Se0/0/0</td>
<td>1</td>
<td>0/0</td>
<td>1236</td>
<td>0/10</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
EIGRP verification output

R1# show ip eigrp traffic
IP-EIGRP Traffic Statistics for process 100
   Hellos sent/received: 459/227
   Updates sent/received: 2/2
   Queries sent/received: 0/0
   Replies sent/received: 0/0
   Acks sent/received: 2/2
   Input queue high water mark 1, 0 drops
   SIA-Queries sent/received: 0/0
   SIA-Replies sent/received: 0/0

R1# show ip eigrp events
Event information for AS 10:
  1  22:24:24.258 Metric set: 172.16.10.0/30 2169856
  2  22:24:24.258 FC sat rdbmet/succmet: 2169856 0
  3  22:24:24.258 FC sat nh/ndbmet: 0.0.0.0 2169856
  4  22:24:24.258 Find FS: 172.16.10.0/30 2169856
  5  22:24:24.258 Metric set: 172.16.10.4/30 2169856
  6  22:24:24.258 FC sat rdbmet/succmet: 2169856 0
  7  22:24:24.258 FC sat nh/ndbmet: 0.0.0.0 2169856
  8  22:24:24.258 Find FS: 172.16.10.4/30 2169856
  9  22:24:24.258 Metric set: 192.168.10.0/24 2172416
 10  22:24:24.258 Route install: 192.168.10.0/24 172.16.10.2
 11  22:24:24.258 Route install: 192.168.10.0/24
Internet Protocol version 6 (IPv6)

IPv6 benefits, IPv6 addressing, IPv6 shortened expression, IPv6 migration techniques, IPv6 routing
IPv6 Benefits

- **More addresses**: An IPv6 address is **128 bits** long which leads to us having $3.4 \times 10^{38}$ number of addresses.
- **IPSec**: is built in as a standard which should lead to better security practices.
- **No Broadcast**: IPv6 eliminates broadcast and replaces it with “Anycast”.
- **Multicast Based**: IPv6 will multicast when it wants to communicate with many host since there is no broadcast.
- **Plug-n-Play**: an IPv6 LAN can be configured without the need of a DHCP server by using a **Link-Local** address.
- **Auto-configuration**: IPv6 can use your **MAC address** as part of your IPv6 address.
IPv6 Addressing

IPv6 addresses are **128 bits**
- **8 fields** separated by colons
- **16 bits** per field
- The first 64 bits are made up of the Global prefix and Subnet-ID and is referred to as the Subnet prefix.
- **The last 64 bits are reserved for the Interface ID**
## IPv6 Shortened Expression

<table>
<thead>
<tr>
<th>Original</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 2001:0db8:3c4d:0012:0000:0000:1234:56ab</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Drop the leading zeros in a group</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 2001:db8:3c4d:12:0000:0000:1234:56ab</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Note 4 zeros with a single zero</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 2001:db8:3c4d:12:0:0:1234:56ab</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Drop groups of 4 zeros and note them with a ::</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 2001:db8:3c4d:12::1234:56ab</td>
</tr>
</tbody>
</table>
IPv6 Manual Addressing

- Just like IPv4 you have to assign an IP address to your interface, but with IPv6 you have a few options.
  - First option is the manual approach.
    ```
    R1(config)# int f0/0
    R1(config-if)# ipv6 address 2001:db8:3c4d:1:0260:d6ff:fe73:1987/64
    ```
  - You can also just enable the Link-local address like this
    ```
    R1(config)# int f0/0
    R1(config-if)# ipv6 enable
    ```
  - To verify you can use one of these commands
    ```
    R1# show ipv6 interface
    R1# show ipv6 interface brief
    ```
IPv6 Stateless Autoconfiguration (eui-64)

- With IPv6 you now have the ability to have the router fill-in the interface-ID portion of the address.
  
  R1(config)# int f0/0
  R1(config-if)# ipv6 address 2001:db8:3c4d:1::/64 eui-64

- The interface will now have an IPv6 address that looks like this.
  
  R1# show ipv6 interface brief
  R1# FastEthernet0/0             [up/up]
  fe80::230:a3ff:fe4d:d701
EUI-64 explained

- EUI-64 uses the interfaces *MAC address* to generate the *interface-id* for the IPv6 address.
- Since the MAC address is only 48 bits and the interface-id requires 64 bits, eui-64 will pad the MAC address with **FFFE** after the first 3 bytes of the MAC address. (3 bytes = 24 bits = 6 characters)
- It also flips the 7\(^{th}\) bit of the MAC address to maintain global uniqueness which is why **902B** become **922B**.

\[
\begin{array}{c}
\text{MAC Address} \\
902B.34A4.E25A
\end{array}
\quad \begin{array}{c}
\text{EUI-64 padding} \\
922B:34FF:FEA4:E25A
\end{array}
\]
IPv6 Autoconfiguration explained.

<table>
<thead>
<tr>
<th>Original MAC</th>
<th>EUI-64 7th bit flipped</th>
<th>Original MAC</th>
<th>EUI-64 7th bit flipped</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 4 2 1 HEX</td>
<td>8 4 2 1 HEX</td>
<td>8 4 2 1 HEX</td>
<td>8 4 2 1 HEX</td>
</tr>
<tr>
<td>1 0 0 1 9</td>
<td>1 0 0 1 9</td>
<td>1 1 1 1 1 F</td>
<td>1 1 1 1 1 F</td>
</tr>
<tr>
<td>0 0 0 0 0</td>
<td>0 0 1 0 2</td>
<td>1 1 1 1 1 F</td>
<td>1 1 0 1 D</td>
</tr>
<tr>
<td>0 0 1 0 2</td>
<td>0 0 1 0 2</td>
<td>0 1 1 1 1 7</td>
<td>0 1 1 1 1 7</td>
</tr>
<tr>
<td>1 0 1 1 B</td>
<td>1 0 1 1 B</td>
<td>1 1 0 0 C</td>
<td>1 1 0 0 C</td>
</tr>
<tr>
<td>0 0 1 1 3</td>
<td>0 0 1 1 3</td>
<td>0 1 1 0 6</td>
<td>0 1 1 0 6</td>
</tr>
<tr>
<td>0 1 0 0 4</td>
<td>0 1 0 0 4</td>
<td>1 0 1 0 A</td>
<td>1 0 1 0 A</td>
</tr>
<tr>
<td>1 0 1 0 A</td>
<td>1 0 1 0 A</td>
<td>1 1 1 1 1 F</td>
<td>1 1 1 1 1 F</td>
</tr>
<tr>
<td>0 1 0 0 4</td>
<td>0 1 0 0 4</td>
<td>1 1 1 1 1 F</td>
<td>1 1 1 1 1 F</td>
</tr>
<tr>
<td>1 1 1 0 E</td>
<td>1 1 1 0 E</td>
<td>0 0 1 1 3</td>
<td>0 0 1 1 3</td>
</tr>
<tr>
<td>0 0 1 0 2</td>
<td>0 0 1 0 2</td>
<td>0 1 0 1 5</td>
<td>0 1 0 1 5</td>
</tr>
<tr>
<td>0 1 0 1 5</td>
<td>0 1 0 1 5</td>
<td>0 0 0 1 1</td>
<td>0 0 0 1 1</td>
</tr>
<tr>
<td>1 0 1 0 A</td>
<td>1 0 1 0 A</td>
<td>1 1 0 1 D</td>
<td>1 1 0 1 D</td>
</tr>
</tbody>
</table>
IPv6 Address Types

- **Unicast**: One to one, packet is sent to a single interface.
  - **Global Unicast Addresses**: publicly routable addresses, they start at `2000::/3`
  - **Unique Local Addresses**: non-publicly routable addresses, they start at `FC00::/7`
  - **Link-local addresses**: replacement for Automatic Private IP Addresses (APIPA), Every IPv6 host will have a link-local address. They start at `FE80::/10`

- **Multicast**: One to many, packet is sent to many interfaces. They begin at `FF00::/8`

- **Anycast**: One to nearest, packet is sent to the nearest interface in terms of routing distance. Replaces broadcast from IPv4.
IPv6 Protocols: DHCPv6

DHCPv6 (Stateful): Same as it was for IPv4 but now IPv6 compatible

- **Client** communicates using UDP port 546
- **Server** communicates using UDP port 547
IPv6 Protocols: ICMPv6

- **ICMPv6**: Replaces ICMP from IPv4 but is no longer its own protocol, it is now an extension of the IPv6 protocol.

- **NDP**: Neighbor Discovery Protocol is part of ICMPv6 and replaces ARP from IPv4.
  - Determines the MAC address of neighbors
  - Router solicitation (RS) `FF02::2`
  - Router advertisements (RA) `FF02::1`
  - Neighbor solicitation (NS)
  - Neighbor advertisement (NA)
  - Duplicate address detection (DAD)
IPv6 Migrating Technique

- **Dual-Stacking**: Is going to be the most common technique because it’s the easiest. You apply an IPv4 and a IPv6 address to each interface.

- **6to4 Tunnel**: The router is configured to tunnel traffic from an IPv6 network to an IPv4 network or vice-versa.

- **NAT-PT**: The router translates a IPv6 packet to an IPv4 packet by translating the protocol stack.
IPv6 Commands

Enable IPv6 routing
```
router(config)# ipv6 unicast-routing
```

Set an IPv6 address to an interface
```
router(config-if)# ipv6 address <ipv6 address>
```

Set an IPv6 address to an interface with the autoconfiguration
```
router(config-if)# ipv6 address <ipv6 prefix> eui-64
```

Show interface basic information
```
router# show ipv6 interface
```

Show interface layer 3 details
```
router# show ipv6 interface brief
```

Show ipv6 routing table
```
router# show ipv6 route
```

Show ipv6 neighbors
```
router# show ipv6 neighbors
```
IPv6 Routing Protocols

- **RIPng**: Distance vector IPv6 routing
  - Essentially the same as RIPv2 except it’s IPv6 compatible.

- **OSPFv3**: Link-State IPv6 routing
  - Essentially the same as OSPFv2 except it’s IPv6 compatible.

- **EIGRPv6**: Advanced Distance vector IPv6 routing
  - EIGRPv6 supports both IPv4 and IPv6 routing.

<table>
<thead>
<tr>
<th>IPv6 Multicast Address</th>
<th>IPv4 Multicast Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ff02::5</td>
<td>224.0.0.5</td>
<td>OSPFv3 All SPF routers</td>
</tr>
<tr>
<td>ff02::6</td>
<td>224.0.0.6</td>
<td>OSPFv3 All DR routers</td>
</tr>
<tr>
<td>ff02::9</td>
<td>224.0.0.9</td>
<td>RIP routers</td>
</tr>
<tr>
<td>ff02::a</td>
<td>224.0.0.10</td>
<td>EIGRP routers</td>
</tr>
</tbody>
</table>
IPv6 Static Routing Example

R1(config)# ipv6 route FC00:10:10:B::/64 FC00:10:10:C:202:16FF:FE7E:1D01

R2(config)# ipv6 route FC00:10:10:A::/64 S0/0/0
# IPv6 configuration commands

**Manually Set IPv6 Address**

```
router(config-if)# ipv6 address <address with prefix length>
router(config-if)# ipv6 address 2001:0db8:3c4d:0012:0000:0000:1234:56ab/64
```

**Set an IPv6 address by using the auto configuration**

```
router(config-if)# ipv6 address <prefix address with prefix length> eui-64
router(config-if)# ipv6 address 2001:0db8:3c4d:0012::/64 eui-64
```

**Enable IPv6 on an interface (Only needed if you’re not going to set another IPv6 address to the interface)**

```
router(config-if)# ipv6 enable
```

**Set an IPv6 Static route**

```
router(config)# ipv6 route <destination> <next-hop | exit interface> <AD> <permanent>
router(config)# ipv6 route 2001:1:1:1::/64 2400:0db8:3c4d:0012:0000:0000:1234:56ab 150
router(config)# ipv6 route 2001:1:1:1::/64 s0/0/0 150
```
IPv6 RIPng Routing Example

**R1 CLI**
- R1(config)# ipv6 unicast-routing
- R1(config)# ipv6 router rip 1
- R1(config-rtr)# passive-interface f0/0
- R1(config)# int f0/0
- R1(config-if)# ipv6 rip 1 enabled
- R1(config)# int s0/0/0
- R1(config-if)# ipv6 rip 1 enabled

**R2 CLI**
- R2(config)# ipv6 unicast-routing
- R2(config)# ipv6 router rip 1
- R2(config-rtr)# passive-interface f0/0
- R2(config)# int f0/0
- R2(config-if)# ipv6 rip 1 enabled
- R2(config)# int s0/0/0
- R2(config-if)# ipv6 rip 1 enabled
## RIPng configuration commands

<table>
<thead>
<tr>
<th>Create/Enter RIPng process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config)# ipv6 router rip &lt;process-id&gt;</td>
</tr>
<tr>
<td>Router(config)# ipv6 router rip 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set an interface to passive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-rtr)# passive-interface &lt;interface&gt;</td>
</tr>
<tr>
<td>Router(config-rtr)# passive-interface f0/0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Place the interface into the RIPng process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-if)# ipv6 rip &lt;process-id&gt; enable</td>
</tr>
<tr>
<td>Router(config-if)# ipv6 rip 1 enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable IPv6 Unicast Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config)# ipv6 unicast-routing</td>
</tr>
</tbody>
</table>
IPv6 OSPFv3 Routing Example

**R1 CLI**
- `R1(config)# ipv6 unicast-routing`
- `R1(config)# ipv6 router ospf 1`
- `R1(config-rtr)# router-id 1.1.1.1`
- `R1(config-rtr)# passive-interface f0/0`
- `R1(config)# int f0/0`
- `R1(config-if)# ipv6 ospf 1 area 0`
- `R1(config)# int s0/0`
- `R1(config-if)# ipv6 ospf 1 area 0`

**R2 CLI**
- `R2(config)# ipv6 unicast-routing`
- `R2(config)# ipv6 router ospf 1`
- `R2(config-rtr)# router-id 2.2.2.2`
- `R2(config-rtr)# passive-interface f0/0`
- `R2(config)# int f0/0`
- `R2(config-if)# ipv6 ospf 1 area 0`
- `R2(config)# int s0/0`
- `R2(config-if)# ipv6 ospf 1 area 0`
### OSPFv3 configuration commands

<table>
<thead>
<tr>
<th>Create/Enter OSPFv3 process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config)# ipv6 router ospf &lt;process-id&gt;</td>
</tr>
<tr>
<td>Router(config)# ipv6 router ospf 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set a Router-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-rtr)# router-id &lt;4 octet router-id&gt;</td>
</tr>
<tr>
<td>Router(config-rtr)# router-id 1.1.1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set an interface to passive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-rtr)# passive-interface &lt;interface&gt;</td>
</tr>
<tr>
<td>Router(config-rtr)# passive-interface f0/0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Place the interface into the OSPFv3 process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-if)# ipv6 ospf &lt;process-id&gt; area &lt;area number&gt;</td>
</tr>
<tr>
<td>Router(config-if)# ipv6 ospf 1 area 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable IPv6 Unicast Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config)# ipv6 unicast-routing</td>
</tr>
</tbody>
</table>
### OSPFv3 verification commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Router# show ipv6 ospf</code></td>
<td>Show OSPF configurations, like router ID, area information, SPF stats, LSA timers</td>
</tr>
<tr>
<td><code>Router# show ipv6 ospf database</code></td>
<td>Show the Link ID of all routers and which neighbor is advertising that router to you.</td>
</tr>
<tr>
<td><code>Router# show ipv6 ospf neighbor</code></td>
<td>Show your OSPF neighbors</td>
</tr>
<tr>
<td><code>Router# show ipv6 ospf interface</code></td>
<td>Show each interfaces IP Address, Area, Process ID, Router ID, Network type, Cost, Priority, DR/BDR, timers.</td>
</tr>
<tr>
<td><code>Router# show ipv6 protocols</code></td>
<td>Show OSPF configurations you have set, Process ID, Router ID, Network advertisements.</td>
</tr>
</tbody>
</table>
IPv6 EIGRPv6 Routing Example

R1 CLI
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 router eigrp 10
R1(config-rtr)# no shutdown
R1(config-rtr)# eigrp router-id 1.1.1.1
R1(config)# int f0/0
R1(config-if)# ipv6 eigrp 10
R1(config)# int s0/0
R1(config-if)# ipv6 eigrp 10

R2 CLI
R2(config)# ipv6 unicast-routing
R2(config)# ipv6 router eigrp 10
R2(config-rtr)# no shutdown
R2(config-rtr)# eigrp router-id 2.2.2.2
R2(config)# int f0/0
R2(config-if)# ipv6 eigrp 10
R2(config)# int s0/0
R2(config-if)# ipv6 eigrp 10
# EIGRPv6 IPv6 configuration commands

<table>
<thead>
<tr>
<th>Create/Enter EIGRPv6 IPv6 process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config)# ipv6 router eigrp &lt;AS&gt;</td>
</tr>
<tr>
<td>Router(config)# ipv6 router eigrp 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable EIGRPv6 IPv6 routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-rtr)# no shutdown</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set a Router-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-rtr)# eigrp router-id &lt;4 octet router-id&gt;</td>
</tr>
<tr>
<td>Router(config-rtr)# eigrp router-id 1.1.1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set an interface to passive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-rtr)# passive-interface &lt;interface&gt;</td>
</tr>
<tr>
<td>Router(config-rtr)# passive-interface f0/0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Place the interface into the EIGRPv6 IPv6 process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-if)# ipv6 eigrp &lt;AS&gt;</td>
</tr>
<tr>
<td>Router(config-if)# ipv6 eigrp 1</td>
</tr>
<tr>
<td><strong>EIGRPv6 IPv6 verification commands</strong></td>
</tr>
<tr>
<td>--------------------------------------</td>
</tr>
</tbody>
</table>
| **Show all EIGRP paths, includes Successors and Feasible Successors**  
Router# `show ipv6 eigrp topology` |
| **Show EIGRP packet count, how many packets that are sent and received.**  
Router# `show ipv6 eigrp traffic` |
| **Show your EIGRP directly connected neighbors**  
Router# `show ipv6 eigrp neighbors` |
| **Show which interfaces are routing with EIGRP**  
Router# `show ipv6 eigrp interface` |
| **Show EIGRP configurations, AS number, Summarizations, Network advertisements, and K values**  
Router# `show ipv6 protocols` |
IPv4 vs IPv6

IPv4 Addressing
- 32 bit address length
- Classful Networks
  - Class A
    - 8 Network bits (1 Network)
    - 24 Host bits (16.7 Million host)
  - Class B
    - 16 Network bits (1 Network)
    - 16 Host bits (65,534 host)
  - Class C
    - 24 Network bits (1 Network)
    - 8 Host bits (254 host)
- Classless Network (VLSM/CIDR)

IPv6 Addressing
- 128 bit address length
- Classful Networks (N/A)
- Classless Networks only
  - 64 bit Prefix-length (1 Network)
    - 48 bit Global Prefix (Globally Unique)
    - 16 bit Subnet-ID (Locally Unique)
  - 64 bit Interface ID (1 Host)
    - 64 bits are always reserved for the host.
Security

IP Access List (Standard & Extended)
Security

- **Access List**: is a list of rules that can control the flow of packets that are coming in or out of an interface.
  - The rules are applied in the order that they are listed.
    - When a packet is received it will be tested against line 1, if it does not match then line 2 and so on until there’s a match.
  - Once a packet matches a line in the list no other rules will apply to that packet.
  - If there is no rule that can be applied to a packet then it’s discarded. This is because of the “**implicit deny**” rule.
Access List Types

- **Standard Access List (numbered between 1 – 99)**
  - Can only filter packets based off the source IP address
  - Best if used to deny or permit packets from an entire network, subnet, or host

- **Extended Access List (numbered between 100 – 199)**
  - Can filter packets based off these conditions
    - **IP address**: source or destination
    - **Protocol**: IP, TCP, UDP
    - **Application Port**: Port numbers like 23 for telnet or 80 for http

- **Named Access List**
  - Can either be a “Standard Named Access List” or an “Extended Named Access List”.
  - The main difference is that a Named Access List is referred to by a “Name” instead of a number.
  - Named access list offer a simpler way to manage access list rules.

- **Dynamic Access List**
  - Allows access to a specific host based off a user authentication process.
  - Very useful for granting limited timed access that will be revoked on its own.
Inbound vs Outbound

- When you write any Access List it then must be applied to the Inbound or Outbound direction of an interface.

- When Sales wants to reach Finance this is the flow of the packets.
  - Sales enters the **f0/0** of R1 and exits the **s0/0/0** of R1 then enters R2 on **s0/0/0** and exits **f0/0** of R2 to enter Finance.
Access List guidelines

- An interface can only have one inbound list and one outbound list.
- You put more specific rules at the top of your list.
- New rules are always added to the bottom of the list.
- You can not remove one line from an access list. (Except for named access list).
- Unless an access list ends with a `permit any` the rule of `implicit deny` will discard the packet.
- You create the access list then apply to the interface, if you do not apply it to anything then it will not filter any packets.
- Access list will not filter traffic that originates from the router.
- Place **Standard** Access list as close to the **destination** as possible.
- Place **Extended** Access list as close to the **source** as possible.
- If you need a rule to apply to all host you can replace any source/destination address with the word “any”. any = 0.0.0.0 255.255.255.255.
Access List Configuration Commands

**Standard list**

R1(config)# access-list <#> <rule> <source>
R1(config)# access-list 10 permit host 192.168.100.100
R1(config)# access-list 10 deny 192.168.100.100 0.0.0.255
R1(config)# access-list 10 permit any

**Extended list**

R1(config)# access-list <#> <rule> <protocol> <source> <destination> eq <port>
R1(config)# access-list 100 deny tcp host 192.168.100.100 host 192.168.200.200 eq 23
R1(config)# access-list 100 permit tcp 192.168.100.0 0.0.0.255 192.168.200.0 0.0.0.255 eq 25
R1(config)# access-list 100 deny tcp host 192.168.100.100 192.168.200.0 0.0.0.255 eq 80
R1(config)# access-list 100 deny tcp any host 192.168.200.200 eq 23

**Apply the ACL to the interface**

R1(config-if)# ip access-group <acl number> <direction>
R1(config-if)# ip access-group 110 out

**Apply the ACL to a line**

R1(config-line)# ip access-class <acl number> <direction>
R1(config-line)# ip access-class 1 in
### Named Access List Configuration Commands

**Create/Enter Named ACL**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>R1(config)# ip access-list &lt;type&gt; &lt;name&gt;</code></td>
<td>Create or enter a named access list.</td>
</tr>
<tr>
<td><code>R1(config)# ip access-list extended control_web</code></td>
<td>For extended access lists.</td>
</tr>
<tr>
<td><code>R1(config)# ip access-list standard control_web</code></td>
<td>For standard access lists.</td>
</tr>
</tbody>
</table>

**Add rule to a Standard Named ACL**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>R1(config-std-nacl)# &lt;rule&gt; &lt;source&gt;</code></td>
<td>Add rules to a standard named access list.</td>
</tr>
<tr>
<td><code>R1(config-std-nacl)# permit 192.168.50.0 0.0.0.255</code></td>
<td>Permit a specific source IP.</td>
</tr>
</tbody>
</table>

**Add rule to a Extended Named ACL**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>R1(config-ext-nacl)# &lt;rule&gt; &lt;protocol&gt; &lt;source&gt; &lt;destination&gt; eq &lt;port&gt;</code></td>
<td>Add rules to an extended named access list.</td>
</tr>
<tr>
<td><code>R1(config-ext-nacl)# permit tcp any 192.168.50.0 0.0.0.255 eq 80</code></td>
<td>Permit a specific TCP protocol and source and destination IP.</td>
</tr>
</tbody>
</table>

**Apply the Named ACL to the interface**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>R1(config-if)# ip access-group &lt;acl name&gt; &lt;direction&gt;</code></td>
<td>Apply a named access list to an interface.</td>
</tr>
<tr>
<td><code>R1(config-if)# ip access-group block_bob out</code></td>
<td>Block Telnet traffic.</td>
</tr>
</tbody>
</table>

**Apply the Named ACL to a line**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>R1(config-line)# ip access-class &lt;acl name&gt; &lt;direction&gt;</code></td>
<td>Apply a named access list to a specific line.</td>
</tr>
<tr>
<td><code>R1(config-line)# ip access-class block_telnet in</code></td>
<td>Block Telnet traffic on a specific line.</td>
</tr>
</tbody>
</table>
Access List Verification Commands

Display all Access List
R1# show access-list

Display all IPv4 Access List
R1# show ip access-list

Verify which IPv4 interfaces have access-list applied
R1# show ip interface
**IPv4 Access List Lab**

**IPv4 Standard ACL**
Create a Standard access list on R1 using the fewest lines possible. The access-list should restrict the Sales subnet from accessing the Accounting subnet, except for host Scott who should still be able to access the Accounting subnet. All other traffic should be allowed.

**IPv4 Extended ACL**
Create an Extended access list using the fewest lines possible. The access-list should prevent all of the subnets on router R1 access to the HTTP server on host WebSrv, except for host Corrigan who should be able to access the HTTP server with his web browser. All other traffic should be allowed.
**Network Address Translation (NAT)**

- NAT translates one IP address to another IP address
  - **Advantages**
    - Conserve public IP addresses
    - Eliminates address overlap events
    - Makes it easier to connect to the internet
    - Eliminates address renumbering if your network changes
  - **Disadvantages**
    - Translation introduces switching path delays
    - Cause loss of end-to-end IP traceability
    - Certain applications will not function with NAT enabled
Static NAT

Translates One to One

R1 CLI
R1(config)# ip nat inside source static 192.168.100.10 64.10.10.10
R1(config)# int f0/0
R1(config-if)# ip nat inside
R1(config)# int s0/0
R1(config-if)# ip nat outside
Dynamic NAT

Translates Many to Many

```
R1 CLI
R1(config)# ip nat pool TIA 79.10.10.33 79.10.10.63 netmask 255.255.255.224
R1(config)# access-list 1 permit 192.168.100.0 0.0.0.255
R1(config)# ip nat inside source list 1 pool TIA
R1(config)# int f0/0
R1(config-if)# ip nat inside
R1(config)# int s0/0
R1(config-if)# ip nat outside
```
Overloading (PAT/PNAT)

Translates Many to One

R1 CLI
R1(config)# ip nat pool MEGAPATH 24.1.10.1 24.1.10.1 netmask 255.255.255.255
R1(config)# access-list 2 permit 192.168.100.0 0.0.0.255
R1(config)# ip nat inside source list 2 pool MEGAPATH overload
R1(config)# int f0/0
R1(config-if)# ip nat inside
R1(config)# int s0/0
R1(config-if)# ip nat outside
NAT Commands

<table>
<thead>
<tr>
<th>Static NAT rule</th>
<th>R1(config)# <code>ip nat &lt;direction&gt; source static &lt;inside local&gt; &lt;inside global&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1(config)# <code>ip nat inside source static 192.168.100.1 24.1.1.10</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAT pool (used for Dynamic and Overloading)</th>
<th>R1(config)# <code>ip nat pool &lt;pool name&gt; &lt;first ip&gt; &lt;last ip&gt; netmask &lt;subnet mask&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1(config)# <code>ip nat pool verizon 26.1.1.1 26.1.1.126 netmask 255.255.255.128</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic NAT rule</th>
<th>R1(config)# <code>ip nat &lt;direction&gt; source &lt;acl number&gt; pool &lt;pool name&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1(config)# <code>ip nat inside source list 1 pool verizon</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Overloading NAT rule</th>
<th>R1(config)# <code>ip nat &lt;direction&gt; source &lt;acl number&gt; pool &lt;pool name&gt; overload</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1(config)# <code>ip nat inside source list 2 pool Verizon overload</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAT interface configuration</th>
<th>R1(config-if)# <code>ip nat &lt;direction&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1(config-if)# <code>ip nat inside</code></td>
</tr>
<tr>
<td></td>
<td>R1(config-if)# <code>ip nat outside</code></td>
</tr>
</tbody>
</table>

| Display active NAT translation       | R1# `show ip nat translation`                                                   |
NAT Names

Overloading

R1

f0/0 192.168.1.1/24

Bob 192.168.1.10/24

Inside Local

Inside Global 64.10.10.1/24

Outside Global 74.125.228.65/24

Web Server 192.168.100.10/24

Outside Local

Inside Local  ➔ Inside Global  ➔ Outside Global  ➔ Outside Local
Basic switch functions, Spanning-Tree Protocol (STP), Virtual LANS (VLANS), Port Security, EtherChannel
Ethernet Switching

- **Store-and-Forward Ethernet Switching**
  - The switch receives the entire frame (store) before it sends the frame (forward).
  - This allows the switch to check the frame for errors by using the FCS (frame check sequence) from the end of the frame.

- **Cut-through switching**
  - The switch starts to forward the frame after it receives the first 6 bytes of the frame. This allows the switch to match the destination MAC address and make a forwarding decision before it received the entire frame.
  - This is done to **reduce latency** but can **lead to errors** since the switch forwards the frame before FCS is received.
Three Switch Functions at Layer 2

- **Address learning**
  - Switches learn MAC addresses and store them in the **CAM** (Content Addressable Memory)

- **Forward/Filter decisions**
  - Switches use the CAM table to decide whether to Forward a frame or Filter it.

- **Loop avoidance**
  - **(STP) Spanning Tree Protocol** is used to avoid causing a loop in your network.
**Switch Address Learning**

**Address learning**: switches learn the source MAC addresses of a frame and store it in a Content Addressable Memory (CAM) filter table also referred to as a MAC address table.

---

**Step 1**

**Step 2**

**Step 3**

**Step 4**

**Step 5**

---

**CAM/MAC forward/filter**

```
S1# show mac-address-table
```

<table>
<thead>
<tr>
<th>Vlan</th>
<th>Mac Address</th>
<th>Type</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0000.0000.aaaa</td>
<td>DYNAMIC</td>
<td>Fa0/1</td>
</tr>
<tr>
<td>1</td>
<td>0000.0000.bbbb</td>
<td>DYNAMIC</td>
<td>Fa0/2</td>
</tr>
</tbody>
</table>
Switch Forward/Filter Decision

**Forward/Filter decisions:** when a frame is received it’s compared to the MAC address table, when a match is found it’s then forwarded out that interface. If there is no match then the frame is flooded.
Spanning Tree Protocol

- Spanning Tree Protocol (STP) achieves its primary objective of preventing network loops on layer 2 network bridges or switches by monitoring the network to track all links and block the redundant ones.

- STP uses the spanning-tree algorithm (STA) to first create a topology database and then search out and disable redundant links.

- With STP running, frames will be forwarded on only premium, STP-chosen links.
Spanning Tree Protocol Versions

- **IEEE 802.1d**: is the original Spanning Tree Protocol (STP) which is considered very **slow**.
- **IEEE 802.1w**: faster version of spanning tree known as Rapid Spanning Tree (RSTP).
- **PVST+**: Cisco’s enhanced version of **802.1d** which allows for **each VLAN** to have its own instance of STP so each VLAN can have their own root bridge.
- **Rapid PVST+**: Cisco’s enhanced version of **802.1w** which allows for each VLAN to have its own instance of STP so each VLAN can have their own root bridge.

**Compatibility note**
- Cisco switches only run PVST+/RPVST+, if a port is connect to a switch that does NOT support PVST+/RPVST+ it will revert that port only to STP.
Spanning Tree Protocol Terms

- **Root bridge**: is responsible for telling all the other switches which ports to use. The switch with the **lowest priority** will become the Root bridge. If priorities match the switch with the **lowest MAC address** will become the Root bridge.

- **Non-root bridges**: all switches that are not the root bridge.

- **BPDU**: is the protocol used by switches to exchange information.

- **Bridge ID**: is how STP identifies a switch on the network. It’s based off the priority set on the switch plus the VLAN-ID. The switch with the **lowest Bridge-ID** becomes the Root bridge.

- **Port cost**: is used to determine the best path to the Root bridge. The higher the bandwidth the lower the cost.

- **Path cost**: when there are multiple links between a switch and the Root bridge a switch will calculate the cost of the path based off the port cost.
Spanning Tree Protocol Ports

<table>
<thead>
<tr>
<th>Port Speed</th>
<th>Port Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Gbps</td>
<td>4</td>
</tr>
<tr>
<td>100 Mbps</td>
<td>19</td>
</tr>
<tr>
<td>10 Mbps</td>
<td>100</td>
</tr>
</tbody>
</table>

Port Speed | Port Cost
---|---
1 Gbps | 4
100 Mbps | 19
10 Mbps | 100
Spanning Tree Protocol Ports

- **Root port:** port with the *lowest cost leading to the root bridge*. It’s used to forward frames to the root bridge.

- **Designated port:** is a port that has a *low cost* leading to certain network segment (*away from the root bridge*) and is allowed to forward frames.

- **Non-designated port:** is a port with a *high cost* so it is not allowed to forward frames.

- **Alternate port (RSTP only):** is a non-designated port marked to become the next *root port*.

- **Backup port (RSTP only):** is a non-designated port marked to become the next *designated port*.
Rapid Spanning Tree Protocol Ports
# STP vs RSTP Port states

<table>
<thead>
<tr>
<th>Description</th>
<th>STP 802.1d</th>
<th>RSTP 802.1w</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port is administratively Down</td>
<td>Disabled</td>
<td>Discarding</td>
</tr>
<tr>
<td>Can NOT send or receive frames, will still process BPDUs</td>
<td>Blocking</td>
<td>Discarding</td>
</tr>
<tr>
<td>Can NOT send or receive frames, will still process BPDUs</td>
<td>Listening</td>
<td>Discarding</td>
</tr>
<tr>
<td>Can <strong>only</strong> learn MAC addresses</td>
<td>Learning</td>
<td>Learning</td>
</tr>
<tr>
<td>Can process frames, BPDUs and can also update the CAM.</td>
<td>Forwarding</td>
<td>Forwarding</td>
</tr>
</tbody>
</table>

RSTP combines the Blocking and Listening port states. This allows a port to move from a blocking state to forwarding to happen in 20 seconds instead of 50 seconds.
PortFast and BPDU Guard

- **PortFast**: This feature is used to eliminate the need for a switchport to transition from blocking to forwarding by keeping the interface in the forwarding state. This should only be enabled on an interface connected to a single host. It is recommended to also enable BPDU guard on a PortFast interface to provide some protection.

- **BPDU Guard**: Once enabled, this feature will bring an interface down when a BPDU is received on that interface. The interface will show its status as “err-disabled” instead of administratively down.
Spanning Tree Configuration Commands

Set Bridge Priority
switch(config)# spanning-tree vlan <vlan number> priority <priority (0 – 61440)>
switch(config)# spanning-tree vlan 1 priority 4096

Enable Rapid Spanning Tree Protocol (pvst, rapid-pvst, mst)
switch(config)# spanning-tree mode <stp mode>
switch(config)# spanning-tree mode rapid-pvst

Set switch as the root bridge
switch(config)# spanning-tree vlan <vlan number> root primary
switch(config)# spanning-tree vlan 1 root primary

Display Spanning Tree configurations
switch# show spanning-tree
## PortFast/BPDU Guard Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable PortFast on an Interface</strong></td>
<td>switch(config-if)# spanning-tree portfast</td>
</tr>
<tr>
<td><strong>Enable PortFast on all ports globally</strong></td>
<td>switch(config)# spanning-tree portfast default</td>
</tr>
<tr>
<td><strong>Enable BPDU guard on an Interface</strong></td>
<td>switch(config-if)# spanning-tree bpduguard</td>
</tr>
<tr>
<td><strong>Enable BPDU guard on all interfaces globally</strong></td>
<td>switch(config)# spanning-tree bpduguard default</td>
</tr>
<tr>
<td><strong>Disable PortFast on an Interface</strong></td>
<td>switch(config-if)# spanning-tree portfast disable</td>
</tr>
<tr>
<td><strong>Disable BPDU guard on an Interface</strong></td>
<td>switch(config-if)# spanning-tree bpduguard disable</td>
</tr>
</tbody>
</table>
VLAN (Virtual LAN)

- VLANs allow you to take a switch and break it up into multiple broadcast domains.

**Benefits**

- **Broadcast control**
  - Each VLAN is its own broadcast domain

- **Security**
  - Can only communicate within your VLAN

- **Flexibility and scalability**
  - VLAN behave as though they’re their own LAN, so what happens in one VLAN does not affect the others.
VLANs Explained

<table>
<thead>
<tr>
<th>Marketing</th>
<th>VLAN 2</th>
<th>172.16.20.0/24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shipping</td>
<td>VLAN 3</td>
<td>172.16.30.0/24</td>
</tr>
<tr>
<td>Engineering</td>
<td>VLAN 4</td>
<td>172.16.40.0/24</td>
</tr>
<tr>
<td>Finance</td>
<td>VLAN 5</td>
<td>172.16.50.0/24</td>
</tr>
<tr>
<td>Management</td>
<td>VLAN 6</td>
<td>172.16.60.0/24</td>
</tr>
<tr>
<td>Sales</td>
<td>VLAN 7</td>
<td>172.16.70.0/24</td>
</tr>
</tbody>
</table>

Provides inter-VLAN communication and WAN services
VLAN Example

S1(config)# vlan 2
S1(config-vlan)# name sales
S1(config)# vlan 3
S1(config-vlan)# name accounting
S1(config)# int f0/1
S1(config-if)# switchport mode access
S1(config-if)# switchport access vlan 2
S1(config)# int f0/2
S1(config-if)# switchport mode access
S1(config-if)# switchport access vlan 3
S1(config)# vlan 20
S1(config-vlan)# name voice
S1(config)# int f0/3
S1(config-if)# switchport mode access
S1(config-if)# switchport access vlan 2
S1(config-if)# switchport voice vlan 20
VLAN Terminology

- **Access Port:** Switchports must be configured as an access port before you can assign an access VLAN or voice VLAN to it.

- **Trunk Port:** is used to transmit VLAN frames between devices. Trunk ports tag frames with the VLAN-ID as they exit the port using one of the protocols below.
  - **Inter-Switch Link (ISL):** Cisco proprietary
  - **IEEE 802.1q:** Open standard

- **Dynamic Port:** is the default port mode. In this mode the switchport will use DTP (Dynamic Trunking Protocol) to automatically switch between being an access port or trunk port.

- **Native VLAN:** VLAN 1 is the default native VLAN on all switches. The native VLAN is also referred to as the *untagged* VLAN because its frames are not modified by the trunk port.
### VLAN Configuration Commands

<table>
<thead>
<tr>
<th>Command Description</th>
<th>Command Details</th>
</tr>
</thead>
</table>
| **Create/Enter VLAN**                                    | `switch(config)# vlan <vlan id>`  
  `switch(config)# vlan 2`  |
| **Name VLAN**                                            | `switch(config-vlan)# name <vlan name>`  
  `switch(config-vlan)# name sales`  |
| **Disable a VLAN**                                       | `switch(config-vlan)# shutdown`  |
| **Alternate method to disable a VLAN**                   | `switch(config)# shutdown vlan <vlan id>`  
  `switch(config)# shutdown vlan 2`  |
| **Set switchport mode (access | trunk | dynamic auto | dynamic desirable)** | `switch(config-if)# switchport mode <port mode>`  
  `switch(config-if)# switchport mode access`  |
| **Assign an access VLAN to an interface**                | `switch(config-if)# switchport access vlan <vlan id>`  
  `switch(config-if)# switchport access vlan 2`  |
### VLAN Configuration Commands (continued)...

**Assign a voice VLAN to an interface**

```
switch(config-if)# switchport voice vlan <vlan id>
switch(config-if)# switchport voice vlan 20
```

**Set allowed VLANs on a trunk port. Actions (add | all | except | remove)**

```
switch(config-if)# switchport trunk allowed vlan <action> <vlan id>
switch(config-if)# switchport trunk allowed vlan add 2
```

**Set native VLAN on a trunk port.**

```
switch(config-if)# switchport trunk native vlan <vlan id>
switch(config-if)# switchport trunk native vlan 99
```

**Configure an IP address to the native VLAN**

```
switch(config)# interface vlan 1
switch(config-if)# ip address 172.16.10.10 255.255.0.0
switch(config-if)# no shutdown
```
### VLAN Verification Commands

<table>
<thead>
<tr>
<th>Command Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display VLAN all information like which interfaces belong to which VLANs.</td>
<td><code>switch# show vlan</code></td>
</tr>
<tr>
<td>Display less VLAN database information</td>
<td><code>switch# show vlan brief</code></td>
</tr>
</tbody>
</table>
| Display single VLAN database information by vlan id                               | `switch# show vlan id <vlan id>`
|                                                                                 | `switch# show vlan id 2` |
| Display single VLAN database information by vlan name                              | `switch# show vlan name <vlan name>`
|                                                                                 | `switch# show vlan name sales` |
| Display interface switchport information (Assigned VLANs, switchport mode, native VLAN) | `switch# show interface <interface> switchport`  
|                                                                                 | `switch# show interface fastethernet 0/1 switchport` |
| Display Trunk ports                                                                | `switch# show interface trunk` |
VLAN Trunking Protocol (VTP)

- **VTP (VLAN trunking protocol):** is used to help manage the VLAN database across all the switches in the network.

- **VTP Modes**
  - **VTP Server:** Shares the VLAN database with the VTP clients. This is the default mode for all switches.
  - **VTP Client:** Synchronizes it’s database with the VTP server. You can NOT configure the VLAN database on the client.
  - **VTP Transparent:** Does not synchronize or share it’s VLAN database with any other switch. If it receives a VTP update it will NOT process it but will let it pass to the next switch so it can still receive the updates.

- **VTP Pruning:** Allows the VTP server to remove VLANs from trunk ports based on the access VLANs assigned to interfaces. This is done to minimize the propagation of flooding.
VTP Configuration Example

S1(config)# vtp mode server
S1(config)# vtp domain tia
S1(config)# vtp password cisco
S3(config)# vtp mode transparent
S2(config)# vtp mode client
S2(config)# vtp domain tia
S2(config)# vtp password cisco

All clients are configured with the same settings. S2 is being used as an example.
### VTP Commands

<table>
<thead>
<tr>
<th>Set VTP Mode (server</th>
<th>client</th>
<th>transparent</th>
<th>off)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch(config)# vtp mode &lt;mode&gt;</code></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# vtp mode client</code></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set VTP domain name</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch(config)# vtp domain &lt;domain name&gt;</code></td>
</tr>
<tr>
<td><code>switch(config)# vtp domain tia.local</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set VTP password</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch(config)# vtp password &lt;password&gt;</code></td>
</tr>
<tr>
<td><code>switch(config)# vtp password cisco</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable VTP pruning on a server</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch(config)# vtp pruning</code></td>
</tr>
</tbody>
</table>

### Show VTP configurations

<table>
<thead>
<tr>
<th>Show VTP password</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch# show vtp password</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Show VTP status</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch# show vtp status</code></td>
</tr>
</tbody>
</table>
Routing Between VLANs with an Interface Per-VLAN

- You can route between VLANs by using a physical interface on the router matched to an *access port* on the switch.
- The router interfaces do not require any configuration beyond the basics (*IP address, no shutdown*)
Routing Between VLANs with a Router on a Stick

**ROAS (Router on a Stick):** is used to route multiple VLANs using one physical port on the router to one physical port on the switch. This accomplished using **Sub-Interfaces** which are separate logical broadcast domains on one physical interface.

```plaintext
S1(config)# int g0/1
S1(config-if)# switchport mode trunk

R1(config)# int g0/0
R1(config-if)# no shutdown
R1(config)# int g0/0.2
R1(config-subif)# encapsulation dot1q 2
R1(config)# int g0/0.3
R1(config-subif)# encapsulation dot1q 3
R1(config)# int g0/0.4
R1(config-subif)# encapsulation dot1q 4
```
Port Security

- **Port Security**: allows you to control which devices and how many devices can be connected to a switch port.

```
switch(config)# int f0/1
switch(config-if)# switchport mode access
switch(config-if)# switchport port-security maximum 1
switch(config-if)# switchport port-security mac-address sticky
```

If any of the port-security rules are violated the switchport will be put into “**err-disabled**” state.
Port Security Configuration Commands

Set switchport mode to access. You can only apply security to access ports.
switch(config-if)# switchport mode access

Set the maximum number of allowed simultaneous MAC address (from 1 – 4097)
switch(config-if)# switchport port-security maximum <number of addresses to allow>

Set the switchport to register MAC addresses automatically
switch(config-if)# switchport port-security mac-address sticky

Manually register MAC address to switchport
switch(config-if)# switchport port-security mac-address <MAC address>

Set port-security violation (shutdown – protect – restrict)
switch(config-if)# switchport port-security violation <violation mode>

Enable port security
switch(config-if)# switchport port-security

Display Port-Security configurations
switch# show port-security
An EtherChannel allows you to combine multiple physical interfaces into one logical interface, which increases performance while maintaining redundancy.

The logical interface is known as a “Port-channel” which operates at the speed of the combined physical interfaces.

To channel interfaces **duplexing, speed, and encapsulation** MUST be the same.

Also known as Port Aggregation (**PAgP**), Link Aggregation (**LACP**), channel bonding, or multi-linking.
# EtherChannel Configuration Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Create/Enter a Port-Channel** | switch(config)# interface port-channel <number>  
switch(config)# interface port-channel 1 |
| **Enter an interface range** | switch(config)# interface range <interface type> <first number> - <last number>  
switch(config)# interface range fastethernet 0/1 - 10 |
| **Set switchport mode to trunk** | switch(config-if-range)# switchport mode trunk |
| **Disables the switchports ability to automatically change modes** | switch(config-if-range)# switchport nonegotiate |
| **Place an interface range into a port-channel** | switch(config-if-range)# channel-group <port-channel number> mode desirable  
switch(config-if-range)# channel-group 1 mode desirable |
| **Display Etherchannel configurations** | switch# sh interface etherchannel |
Wide Area Networks (WAN)

WAN terminology, connection types, and BGP.
WAN Terminology

- **Customer Premise Equipment (CPE):** equipment at your location that connects you to the provider.
- **CSU/DSU:** are Layer 1 interfaces used for T-Series connections. It provides the clocking of the interface.
- **Demarcation Point (DMARC):** where your provider's network physically ends.
- **Local Loop:** the provider's equipment in your area. This is where your DMARC comes from.
- **Central Office (CO):** is a building where your provider's network exists. Also known as the **Point of Presence (POP)** or the "High end".
- **Toll Network:** is when you leave your provider's network to enter another provider's network.
WAN Connection Types

- **Dedicated (Leased lines):** are **Point-to-Point** links that connect your network directly to the provider's network. These links often use HDLC or PPP encapsulation.
  - T1, T3, E1, OC-12, OC-48, and more

- **Circuit Switching:** lines go over the Public Switched Telephone Network (PSTN). These lines are low-cost and also low-bandwidth.
  - Dial-up, ISDN (Integrated Services Digital Network)

- **Packet Switching:** lines pass your traffic through a series of routers. These lines are affordable because they share bandwidth with everyone at the local loop.
  - X.25, ATM (Asynchronous Transfer Mode), Frame-Relays, MPLS (Multi-protocol Label Switching)
WAN Connection Types

**Encapsulation:** Every WAN interface needs to encapsulate to the providers protocol. Be aware of which encapsulation to use depending on the connection.

- **HDLC (High-Level Data Link Control):** is a *layer 2* protocol and is the default encapsulation for serial links.

- **PPP (Point-to-Point Protocol):** is a *layer 2* protocol used for many leased lines since it can be used to create links between different vendors’ equipment. It also supports *authentication* and *multilink* connections.
  - **PPP Authentication types**
    - **PAP (Password Authentication Protocol):** authenticates in plain text so it is not secure.
    - **CHAP (Challenge Handshake Authentication Protocol):** authenticates with encryption so it is secure.

- **PPPoE (Point-to-Point Protocol over Ethernet):** is a *layer 2* protocol that encapsulates PPP frames into Ethernet frames. Commonly used by DSL services.
## WAN Interface Configuration Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>R1(config-if)# encapsulation &lt;type&gt;</code></td>
<td>Set interface encapsulation (hdlc</td>
</tr>
<tr>
<td><code>R1(config-if)# encapsulation ppp</code></td>
<td></td>
</tr>
<tr>
<td><code>R1(config-if)# ppp authentication &lt;mode&gt;</code></td>
<td>Set PPP authentication mode (pap</td>
</tr>
<tr>
<td><code>R1(config-if)# ppp authentication chap</code></td>
<td></td>
</tr>
<tr>
<td><code>R1(config-if)# ppp pap sent-username &lt;username&gt; password &lt;password&gt;</code></td>
<td>Set PAP username and password on an interface</td>
</tr>
<tr>
<td><code>R1(config-if)# ppp pap sent-username cisco password P@ssw0rd</code></td>
<td></td>
</tr>
<tr>
<td><code>R1(config)# username &lt;neighbor hostname&gt; password &lt;password&gt;</code></td>
<td>Set username and password used by CHAP</td>
</tr>
<tr>
<td><code>R1(config)# username R2 password cisco</code></td>
<td></td>
</tr>
</tbody>
</table>
# WAN Multilink Commands

## Create/Enter a Multilink interface

```text
R1(config)# interface multilink <number>
R1(config)# interface multilink 1
```

## Enable multilink support for PPP on the multilink interface

```text
R1(config-if)# ppp multilink
```

## Set multilink group on an interface

```text
R1(config-if)# ppp multilink group <group number>
R1(config-if)# ppp multilink group 1
```

## Display detailed PPP multilink status

```text
R1# show ppp multilink
```

## Display PPP status for each PPP link

```text
R1# show ppp all
```
Border Gateway Protocol (BGP)

- BGP is a *distance vector* EGP protocol used to connect *multiple autonomous systems* to each other.
- eBGP is used to connect the “Customer Edge” to the “Providers Edge”
## Routing Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Type</th>
<th>AD</th>
<th>Usage</th>
<th>Communication</th>
<th>Neighbors</th>
<th>Route exchange</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP</td>
<td>Distance Vector</td>
<td>eBGP: 20 / iBGP: 200</td>
<td>Routes Autonomous systems</td>
<td>Unicast</td>
<td>Specifically configured</td>
<td>Neighbors only</td>
</tr>
<tr>
<td>RIPv2</td>
<td>Distance Vector</td>
<td>120</td>
<td>Routes Networks</td>
<td>Periodic Multicast</td>
<td>Discovered</td>
<td>Neighbors only</td>
</tr>
<tr>
<td>OSPF</td>
<td>Link State</td>
<td>110</td>
<td>Routes Networks</td>
<td>Multicast on change</td>
<td>Discovered / Configured</td>
<td>Adjacent Neighbors</td>
</tr>
<tr>
<td>EIGRP</td>
<td>Advanced Distance Vector</td>
<td>90</td>
<td>Routes Networks</td>
<td>Reliable Multicast on change</td>
<td>Discovered</td>
<td>Adjacent Neighbors</td>
</tr>
</tbody>
</table>
Corp(config)# router bgp 10
Corp(config-router)# neighbor 24.10.10.1 remote-as 1
Corp(config-router)# network 192.168.50.80 mask 255.255.255.252
Corp(config-router)# network 192.168.50.84 mask 255.255.255.252
## eBGP Commands

<table>
<thead>
<tr>
<th>Command Description</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create/Enter eBGP process</td>
<td><code>router(config)# router bgp &lt;as&gt;</code></td>
</tr>
<tr>
<td>eBGP neighbor definition and AS</td>
<td><code>router(config-router)# neighbor &lt;neighbor ip address&gt; remote-as &lt;neighbor as&gt;</code></td>
</tr>
<tr>
<td>Set network advertisement</td>
<td><code>router(config-router)# network &lt;network address&gt; mask &lt;subnet mask&gt;</code></td>
</tr>
<tr>
<td>Displays eBGP table</td>
<td><code>router# show ip bgp</code></td>
</tr>
<tr>
<td>Displays basic configuration for the local router, neighbors IP address, AS number, statistics, versions</td>
<td><code>router# show ip bgp summary</code></td>
</tr>
<tr>
<td>Displays TCP connections that terminate at the router</td>
<td><code>router# show tcp summary</code></td>
</tr>
</tbody>
</table>
LAN and Infrastructure Services

To help manage large and complex networks companies make use of many different services. This section will cover the services listed below.

- QoS (Quality of Service)
- DHCP (Dynamic Host Configuration Protocol)
- NTP (Network Time Protocol)
- CDP (Cisco Discovery Protocol)
- DNS (Domain Name Service)
- FHRP (First Hop Redundancy Protocol)
- SNMP (Simple Network Management Protocol)
- Syslog
Quality of Service (QoS)

- **Quality of Service (QoS):** is used to prioritize different data types so service quality can be maintained throughout the network operations.

**Issues Solved by QoS**
- Delay
- Dropped Packets
- Error
- Jitter
- Out-of-Order Delivery

<table>
<thead>
<tr>
<th>Traffic Types</th>
<th>Data</th>
<th>Voice</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth / Bursty</td>
<td>Smooth</td>
<td></td>
<td>Bursty</td>
</tr>
<tr>
<td>Benign / Greedy</td>
<td>Benign</td>
<td></td>
<td>Greedy</td>
</tr>
<tr>
<td>Drop Insensitive</td>
<td>Drop Sensitive</td>
<td></td>
<td>Drop Sensitive</td>
</tr>
<tr>
<td>Delay insensitive</td>
<td>Delay sensitive</td>
<td></td>
<td>Delay sensitive</td>
</tr>
<tr>
<td>TCP retransmits</td>
<td>UDP priority</td>
<td></td>
<td>UDP priority</td>
</tr>
</tbody>
</table>
DHCP (Dynamic Host Configuration Protocol)

- **DHCP** is used to assign configurations such as IP address, Subnet mask, DNS server, WINS server, TFTP server and others.

Configure device as a DHCP server

Router(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.10
Router(config)# ip dhcp pool Some_Pool_Name
Router(dhcp-config)# network 192.168.10.0 255.255.255.0
Router(dhcp-config)# default-router 192.168.10.1
Router(dhcp-config)# dns-server 4.4.4.4
Router(dhcp-config)# lease 3 12 15
A DHCP Relay is required when there is a single DHCP server that provides configurations to host is many broadcast domains.

Configure device as a DHCP Relay

Router(config)# int f0/1
Router(config-if)# ip helper-address 172.20.100.10
## DHCP Verification Commands

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display DHCP information about which IP addresses were leased and the lease time</td>
<td><code>R1# show ip dhcp binding</code></td>
</tr>
<tr>
<td>Display DHCP pools</td>
<td><code>R1# show ip dhcp pool</code></td>
</tr>
<tr>
<td>Display DHCP server statistics</td>
<td><code>R1# show ip dhcp server statistics</code></td>
</tr>
<tr>
<td>Display DHCP server IP conflicts</td>
<td><code>R1# show ip dhcp conflict</code></td>
</tr>
</tbody>
</table>
DHCP Snooping

- **DHCP Snooping** is a security feature on a switch to make sure a host can only receive configurations from an authorized DHCP server.

- Once enabled only a server connected to a **trusted interface** can reply to a DHCP request, while the **untrusted interfaces** can receive configurations.

- **Dynamic ARP Inspection (DAI)** allows a DHCP Snooping enabled switch to build a “bindings table” which store the mapping of IP addresses to MAC addresses. This can be used to prevent a Man-In-The-Middle (MITM) attack by only allowing traffic that matches to an entry the binding table.

![Diagram of DHCP Snooping](image-url)
NTP (Network Time Protocol): is used to synchronize the time between a client device and a server. NTP is used to support proper authentication and logging. A router can act as an NTP server, NTP client or both.

Configure device as an NTP client so it will synchronize with an NTP server.
R1(config)# ntp server 172.16.10.1 version 4

Configure device as an NTP server so NTP clients can synchronize with it. Stratum level range is from 1 to 15.
R1(config)# ntp master 2

Configure which IP address will be used as the source of the NTP server.
R1(config)# ntp source f0/0

Display NTP client and server relationships.
R1# show ntp associations

Display NTP status such as if the clock has been synchronized. NTP servers will display as “unsynchronized”.
R1# show ntp status
Cisco Discovery Protocol and Link Layer Discovery Protocol

CDP (Cisco Discovery Protocol) is used to identify a directly connected IOS neighbor. LLDP (Link Layer Discovery Protocol) is an open standard equivalent.

- **Enable CDP globally (Enabled by default)**
  
  R1(config)# cdp run

- **Enable LLDP globally (Disabled by default)**
  
  R1(config)# lldp run

- **Display CDP|LLDP status**
  
  R1# sh <cdp|lldp>

- **Display CDP|LLDP connected neighbors detail**
  
  R1# sh <cdp|lldp> neighbors detail

- **Display CDP|LLDP interface details**
  
  R1# sh <cdp|lldp> interface

- **Display CDP|LLDP connected neighbors**
  
  R1# sh <cdp|lldp> neighbors

- **Display CDP|LLDP entry details**
  
  R1# sh <cdp|lldp> entry <hostname>

- **Display CDP|LLDP packet counts**
  
  R1# sh <cdp|lldp> traffic

- **Enable the transmission of LLDP messages**
  
  R1(config-if)# lldp transmit

- **Enable the receiving of LLDP messages**
  
  R1(config-if)# lldp receive
CDP Outputs

S1# sh cdp
Global CDP information:
- Sending CDP packets every 60 seconds
- Sending a holdtime value of 180 seconds
- Sending CDPv2 advertisements is enabled

S1# sh cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone
Device ID Local Intrfce Holdtme Capability Platform Port ID
R1      Fas 0/1   175   R          C2600   Fas 0/0
S2      Fas 0/24  169   S          2950    Fas 0/24
S2      Fas 0/23  169   S          2950    Fas 0/23
S3      Fas 0/22  169   S          2950    Fas 0/24
S1# sh cdp neighbors detail
Device ID: R1
Entry address(es):
   IP address : 192.168.100.1
Platform: cisco C2600, Capabilities: Router
Interface: FastEthernet0/1, Port ID (outgoing port): FastEthernet0/0
Holdtime: 175

Version:
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.2(28), RELEASE SOFTWARE (fc5)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 27-Apr-04 19:01 by miwang

advertisement version: 2
Duplex: full
Resolving Names

**Host Table**: can be used to build your own name resolution table. This table is only useable by the device it was created on.

Configure the host table

Router(config)# ip host R1 10.10.10.1
Router(config)# ip host R2 10.10.10.2

Now you can ping R1 by typing "**ping R1**" and the router will resolve it to 10.10.10.1

**DNS Resolution**: point your device to a DNS server to resolve names.

Configure a DNS server on your device

Router(config)# ip domain-lookup
Router(config)# ip name-server 4.4.4.4
Router(config)# ip domain-name localdomain.com

Now the router will send DNS request to the server 4.4.4.4
Syslog (System Message Logging)

- Using a Syslog server is the best way to store and retrieve messages that normally only appear during a console session.
- The Syslog server will organize messages by timestamp, service, type and severity to make it simpler to use the logged data.

<table>
<thead>
<tr>
<th>Severity Level</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emergency (severity 0)</td>
<td>System is unusable.</td>
</tr>
<tr>
<td>Alert (severity 1)</td>
<td>Immediate action is needed.</td>
</tr>
<tr>
<td>Critical (severity 2)</td>
<td>Critical condition.</td>
</tr>
<tr>
<td>Error (severity 3)</td>
<td>Error condition.</td>
</tr>
<tr>
<td>Warning (severity 4)</td>
<td>Warning condition.</td>
</tr>
<tr>
<td>Notification (severity 5)</td>
<td>Normal but significant condition.</td>
</tr>
<tr>
<td>Information (severity 6)</td>
<td>Normal information message.</td>
</tr>
<tr>
<td>Debugging (severity 7)</td>
<td>Debugging message.</td>
</tr>
</tbody>
</table>

- A severity level can be configured to control which messages are logged. The lower the number the more severe the condition.
- When a higher level is configured it will include all the level below it.

I want to look at the console messages of the SF router from last night.
<table>
<thead>
<tr>
<th>Syslog Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable messages to the console (enabled by default)</strong></td>
</tr>
<tr>
<td>R1(config)# logging console &lt;severity number</td>
</tr>
<tr>
<td>R1(config)# logging console warning</td>
</tr>
<tr>
<td><strong>Enable messages to all logged in users. This is needed for SSH and Telnet sessions. (disabled by default)</strong></td>
</tr>
<tr>
<td>R1(config)# logging monitor &lt;severity number</td>
</tr>
<tr>
<td><strong>Required to enable messages during SSH or Telnet sessions</strong></td>
</tr>
<tr>
<td>R1# terminal monitor</td>
</tr>
<tr>
<td><strong>Enable messages to the RAM (disabled by default)</strong></td>
</tr>
<tr>
<td>R1(config)# logging buffered &lt;severity number</td>
</tr>
<tr>
<td><strong>Enable messages to a Syslog server (disabled by default)</strong></td>
</tr>
<tr>
<td>R1(config)# logging host &lt;IP address</td>
</tr>
<tr>
<td>R1(config)# logging host 192.168.10.10</td>
</tr>
<tr>
<td><strong>Set message level for Syslog</strong></td>
</tr>
<tr>
<td>R1(config)# logging trap &lt;severity number</td>
</tr>
<tr>
<td><strong>Display logging status (get message counts and which logging types are enabled)</strong></td>
</tr>
<tr>
<td>R1# show logging</td>
</tr>
</tbody>
</table>
SNMP (Simple Network Management Protocol)

SNMP is used to help manage devices on a network. It provides the capabilities to request information from a host, have the host automatically send information without a request being made and set configurations on devices.

- **GET**: used to request information
- **SET**: used to configure a host
- **TRAP**: used to have a host automatically send information

<table>
<thead>
<tr>
<th>Features</th>
<th>SNMPv1</th>
<th>SNMPv2c</th>
<th>SNMPv3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communities</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>UDP</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>TCP</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>GET</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>GET Bulk</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>MD5</td>
<td>SHA Authentication</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>DES</td>
<td>DES-256 Encryption</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>
### SNMP Configuration Commands

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set SNMP community for SNMPv1 or SNMPv2c with Read Only access</td>
<td>R1(config)# snmp-server community &lt;community string&gt; RO</td>
</tr>
<tr>
<td></td>
<td>R1(config)# snmp-server community tiaReadOnly RO</td>
</tr>
<tr>
<td>Set SNMP community for SNMPv1 or SNMPv2c with Read and Write access</td>
<td>R1(config)# snmp-server community &lt;community string&gt; RW</td>
</tr>
<tr>
<td></td>
<td>R1(config)# snmp-server community tiaReadWrite RW</td>
</tr>
<tr>
<td>Set SNMP location information</td>
<td>R1(config)# snmp-server location &lt;device location&gt;</td>
</tr>
<tr>
<td></td>
<td>R1(config)# snmp-server location London</td>
</tr>
<tr>
<td>Set SNMP contact information</td>
<td>R1(config)# snmp-server contact &lt;device contact&gt;</td>
</tr>
<tr>
<td></td>
<td>R1(config)# snmp-server contact Juan E</td>
</tr>
<tr>
<td>Set SNMPv2c host to receive traps</td>
<td>R1(config)# snmp-server host &lt;hostname or IP&gt; version &lt;version&gt; &lt;community string&gt;</td>
</tr>
<tr>
<td></td>
<td>R1(config)# snmp-server host 192.168.10.10 version 2c tiaCollector</td>
</tr>
<tr>
<td>Enable the sending of traps to the specified host</td>
<td>R1(config)# snmp-server enable traps</td>
</tr>
</tbody>
</table>
### SNMP Verification Commands

| Displays SNMP community information | R1# show snmp community |
| Displays SNMP contact information   | R1# show snmp contact |
| Displays SNMP location information  | R1# show snmp location |
| Displays SNMP host that were configured to receive traps | R1# show snmp host |
| Displays SNMP status and counters   | R1# show snmp |
| Displays SNMPv3 configured user information | R1# show snmp user |
| Displays SNMPv3 configured user group information | R1# show snmp group |
Client Redundancy Issues

- **First Hop Redundancy Protocols (FHRP)** allows you to build redundancy in your router network by using multiple physical routers.

![Diagram showing two routers (R1 and R2) with VR: 10.0.0.10/24 and 10.0.0.2/24, and a host (Host A: 10.0.0.100/24, Gateway: 10.0.0.10).]

When a **Proxy ARP**–enabled router receives an ARP request for an IP address that it knows isn’t on the same subnet as the requesting host, it will respond with an ARP reply packet to the host.
HSRP (Hot Standby Router Protocol)

- HSRP is a **Cisco proprietary** protocol that can be run on most, but not all, of Cisco’s router and multilayer switch models.
- HSRP supports up to **8 physical** routers
- It defines a standby group, and each standby group that you define includes the following routers:
  - **Active** router (only 1 router is ever active in a group)
  - **Standby** router (up to 7 in a group)
  - **Virtual router** is configured on all routers in the HSRP Group
HSRP Virtual MAC

- Here is an example of what an HSRP MAC address would look like:
  - 0000.0c07.ac0a
- The first 24 bits (0000.0c) are the vendor ID of the address; in the case of HSRP being a Cisco protocol, the ID is assigned to Cisco.
- The next 16 bits (07.ac) is the well-known HSRP ID.
- The last 8 bits (0a) are the only variable bits and represent the HSRP group number that you assign.