www.networkershome.com

CCNA

200 - 301

Lab Workbook

Table of Content..... Page No.

Module 1 – Basic Networking

Module 2 – Router Basics Labs

Module 3 – Basic IPv4 Labs

Module 4 – Router Management

Module 5 – Basic Routing

Module 6 – RIP Labs

Module 7 – EIGRP

Module 8 – OSPF

Module 9 – BGP

Module 10 – IPv6

Module 11 – Ipv4 and Ipv6 Access List for Traffic Filtering

Module 12 – WAN Technologies

MODULE 1

BASIC NETWORKING

Chapter 1 – IPv4 Addressing

Understanding IPv4 Addresses

This module introduces you to the basics of IP addressing and prepares you to create an IP addressing plan for your network. This guide is a concise reference on IP addressing best practices, including:

- The basic concepts of IP addressing
- The IP addressing plan used in the Cisco Smart Business Architecture (SBA) Foundation lab network
- The steps you should follow to create your own IP Addressing Plan
- How to maintain your IP space as your network evolves

IP Addressing Overview

An IP address is an address used to uniquely identify a device on an IP network.

IP version 4 (IPv4) addresses, which uniquely identify a device on an IP network, are 32 bits in length and are typically communicated in a format known as dotted decimal. The 32 binary bits are:

- Divided into a network portion and host portion
- Broken into four octets (1 octet = 8 bits). Each octet can be converted to binary.

Consider this IP address, which is presented in dotted decimal:

10.10.16.1

The address breaks down into the following octets:

- **10**
- **10**
- **16**
- **1**

The value in each octet ranges from:

7	6	5	4	3	2	1	0	**Decimal**
128	64	32	16	8	4	2	1	**Value =**
0	0	0	0	0	0	0	0	**0**
1	1	1	1	1	1	1	1	**255**

Binary to Decimal Conversion Here is how binary octets convert to decimal:

The right most bit, or least significant bit, of an octet holds a value of 20. The bit just to the left of that holds a value of 21. This continues until the left most bit, or most significant bit, which holds a value of 27. So if all binary bits are a one, the decimal equivalent would be 255 as shown here:

1	1	1	1	1	1	1	0	**Decimal**
128	64	32	16	8	4	2	1	**Value =**
128+	64+	32+	16+	8+	4+	2+	1	**255**

Here is a sample octet conversion when not all of the bits are set to 1.

0	1	0	0	0	0	0	0	**Decimal**
128	64	32	16	8	4	2	1	**Value =**
0	64+	0	0	0	0	0	1	**65**

And this is sample shows an IP address represented in both binary and decimal

10. 1. 23. 19	**(Decimal)**
00001010.00000001.00010111.00010011	**Binary)**

These octets are broken down to provide an addressing scheme that can accommodate large and small networks.

There are five different classes of networks, A to E.

(Note to the instructors: Please explain in details with examples the binary to decimal and vice versa conversion methods in the class.)

Chapter 2 – IPv4 Address Detailed

IPv4 Address in Detail

An IP address is a 32-bit number written in dotted decimal notation: four 8-bit fields (octets) converted from binary to decimal numbers, separated by dots. The first part of an IP address identifies the network on which the host resides, while the second part identifies the particular host on the given network. The network number field is called the network prefix. All hosts on a given network share the same network prefix but must have a unique host number. In classful IP, the class of the address determines the boundary between the network prefix and the host number.

Octet is collection of 8 binary bits: 1 28 64 32 16 8 4 2 1	(Bitwise Decimal Value, generated by 2 power bit order)
7 6 5 4 3 2 1 0	(Bit order)
	Decimal Value
[0] [0] [0] [0] [0] [0] [0] [0] =	0
[1] [1] [1] [1] [1] [1] [1] [1] =	255

Classes of IPv4 Addressing:

IP addresses are split up into several different categories, including Class A, B, C, D (Multicast), and E (Reserved). Address classes are defined, in part, based on the number of bits that make up the network portion of the address, and in turn, on how many are left for the definition of individual host addresses.

How the network ID and host ID are different for each class of IP addresses?

Let us discuss the Classful IPv4 address in details.

Class A:

In a Class A address, the first 8 bits are the network portion, the next 24 bits are for the network manager to divide into subnets and hosts as he/she sees fit. Class A addresses are used for networks that have more than 65,536 hosts (actually, up to 16777214 hosts!).

Class	**Ist Octet**	Start	**End**	**Default Subnet**	**CIDR**
A	0XXXXXXX	0.0.0.0	127.255.255.255	255.0.0.0	/8

Class B

In a Class B address, the first 16 bits are the network portion, 16 bits are for local subnets and hosts. Class B addresses is used for networks that have between 256 and 65534 hosts.

Class	**Ist Octet**	**Start**	**End**	**Default Subnet**	**CIDR**
B	10XXXXXX	128.0.0.0	191.255.255.255	255.255.0.0	/16

Class C:

In a Class C address, the first 24 bits are the network portion, 8 bits is for local subnets and hosts perfect for networks with less than 254 hosts.

Class	Ist Octet	Start	End	Default Subnet	CIDR
C	110XXXXX	192.0.0.0	223.255.255.255	255.255.255.0	/24

Class D:

In class D, all four octets are network portion so leaving no host portion. All class D addresses have been reserved for multicast

Class	Ist Octet	Start	End	Default Subnet	CIDR
D	1110XXXX	224.0.0.0	239.255.255.255	-	-

Class E:

This is an experimental class and is not used for common networks. Reserved for future use, or Research and Development Purposes.

Class	Ist Octet	Start	End	Default Subne	CIDR
E	1111XXXX	240.0.0.0	254.255.255.255	-	-

(Where X=0 or 1)

Reserved and Available IP Addresses:

Class	Address or Range	Status
A	0.0.0.0 1.0.0.0 to 126.0.0.0 127.0.0.0	Reserved (Default Routing) Available Reserved (Loopback IP Addresses)
B	128.0.0.0 to 191.254.0.0	Available
C	192.0.0.0 192.0.1.0 to 223.254.254.0 223 .255.255.0	Reserved (Not Allocated) Available Reserved (Not Allocated)
D	224.0.0.0 to 239.255.255.255	Multicast group addresses
E	240.0.0.0 to 254.255.255.255 255.255.255.255	Reserved General Broadcast

Network ID:

It represent the Network Part of an IP Address and cannot assign to any device as IP Address.

Host ID:

It represents the Host Part of an IP Address can assign host address in any device in conjugation with Network ID.

Broadcast Address:

A broadcast address is a logical address at which all devices connected to a multiple-access communications network are enabled to receive datagrams. A message sent to a broadcast address is typically received by all network-attached hosts, rather than by a specific host. The broadcast address for an IPv4 host can be obtained by performing a bitwise OR operation between the bit complement of the subnet mask and the host's IP address. Example: For broadcasting a packet to an entire IPv4 subnet using the private IP address space 10.16.0.0/12, which has the subnet mask 255.240.0.0, the broadcast address is 10.16.0.0 Logical OR 0.15.255.255 = 10.31.255.255.

A special definition exists for the IP broadcast address 255.255.255.255. It is the broadcast address of the zero network or 0.0.0.0, which in Internet Protocol standards stands for this network, i.e. the local network. Transmission to this address is limited by definition, in that it is never forwarded by the routers connecting the local network to other networks.

NOTE: An IP datagram will never have broadcast as Source IP Address. It can only be Destination IP Address.

Multicast:

These addresses are used to identify multicast groups. They should only be used as destination addresses, never as source addresses.

224.0.0.0 to 239.255.255.255

Subnet Mask:

Represents the network bits reserved for an IP Address to determine its Network ID. By convention, the bits for the Network ID are all set to 1. It's called a mask because it can be used to identify the subnet to which an IP address belongs by performing a bitwise AND operation on the mask and the IP address. The result is the network ID or subnetwork address:

For Example	IP Address	Binary Representation
IP Address	150.215.017.009	10010110.11010111.00010001.00001001
Subnet Mask	255.255.240.000	11111111.11111111.11110000.00000000
Subnet Address	150.215.016.000	10010110.11010111.00010000.00000000

The subnet address, therefore, is 150.215.016.000.

Truth Table for AND Operation

[0] [1] = [0]

[1] [0] = [0]

[1] [1] = [1]

Public and Private IP Address:

Public Address: Internet known addresses. These addresses are unique to an organization. These addresses can be used as source address and destination address for an IP Datagram.

Class	Range
Class A:	1.0.0.0 - 9.0.0.0, 11.0.0.0 - 126.0.0.0 /8
Class B:	128.0.0.0 - 171.0.0.0 to 172.15.0.0, 172.32.0.0 - 191.255.0.0 /16
Class C:	192.0.0.0 - 192.167.0.0, 192.169.0.0 - 192.255.255.0 /24

Private IP Address:

These addresses are reserved for local use in home and enterprise environments and are not public address space. These addresses might not be unique, and there is no formal address registration. Packets with these addresses in the source or destination fields are not intended to be routed on the public Internet but are intended to be routed within the enterprise or organization.

Class	Range
Class A:	10.0.0.0 /8
Class B:	172.16.0.0 - 172.31.0.0/16 - (172.16.0.0/12)
Class C:	192.168.0.0 - 192.168.255.0/24 - (192.168.0.0/16)

Special IP address:

A. Documentation:

These addresses are used in examples and documentation. They should never be source or destination addresses.

1. 192.0.2.0/24
2. 198.51.100.0/24
3. 203.0.113.0/24

B. Benchmarking

These addresses are reserved for use in documentation. They should not be used as source or destination addresses.

1.198.18.0.0/15

C. APIPA - Automatic Private IP Addressing:

1. 169.254.0.0 /24

A feature of Microsoft Windows, APIPA is a DHCP failover mechanism for local networks. With APIPA, DHCP clients can obtain IP addresses when DHCP servers are non-functional. APIPA exists in all modern versions of Windows except Windows NT.

When a DHCP server fails, APIPA allocates IP addresses in the private range 169.254.0.1 to 169.254.255.254. Clients verify their address is unique on the network using ARP. When the DHCP server is again able to service requests, clients update their addresses automatically.

In APIPA, all devices use the default network mask 255.255.0.0 and all reside on the same subnet.

There are three main forms of IP communications, each with its own unique properties.

Unicast: The most common concept of an IP address is in unicast addressing, available in both IPv4 and IPv6. It normally refers to a single sender or a single receiver, and can be used for both sending and receiving. Usually, a unicast address is associated with a single device or host, but it is not a one-to-one correspondence. Some individual PCs have several distinct unicast addresses, each for its own distinct purpose. Sending the same data to multiple unicast addresses requires the sender to send all the data many times over, once for each recipient.

Broadcast: In IPv4 it is possible to send data to all possible destinations ("all-hosts broadcast"), which permits the sender to send the data only once, and all receivers receive a copy of it. In the IPv4 protocol, the address 255.255.255.255 is used for local broadcast. In addition, a directed (limited) broadcast can be made by combining the network prefix with a host suffix composed entirely of binary 1s. For example, the destination address used for a directed broadcast to devices on the 192.0.2.0/24 network is 192.0.2.255.

Multicast: A multicast address is associated with a group of interested receivers. In IPv4, addresses 224.0.0.0 through 239.255.255.255 (the former Class D addresses) are designated as multicast addresses. In either case, the sender sends a single datagram from its unicast address to the multicast group address and the intermediary routers take care of making copies and sending them to all receivers that have joined the corresponding multicast group.

Chapter at a Glance

1. IP addresses must be unique in a network.
2. IP addresses only have meaning when read in conjunction with a subnet mask.
3. 32 bits (0 or 1) divided into 4 octets.
4. IP address has two portions – network and host.
5. Each octet has a decimal value range of 0 to 255, except for the first octet, which is 1 to 255.
6. The network portion cannot be all 0's nor all 1's.
7. The first octet cannot be 127 (network), this is reserved for loopback and also to check if protocol stack is correctly configured. Errors can easily be resolved by reloading TCP/IP and rebooting.
8. The host portion cannot be all 0's – this defines the network address.
9. The host portion cannot be all 1's – this defines a broadcast in that particular network.
10. The IP address 255.255.255.255 defines a general broadcast.

Class	1st Octet Range (Decimal)	1st octet Struct. (Binary)	Total No. of NW	Maximum of H/NNumber	Address Struct.	Default Mask
A	1 – 127	0XXXXXXX	(2^7)-2 126	(2^24)-2 16,777,214	N.H.H.H	255.0.0.0
B	128 – 191	10XXXXXX	2^14 16,384	(2^16)-2 65,534	N.N.H.H	255.255.0.0
C	192 – 223	110XXXXX	2^21 2,097,152	(2^8)-2 254	N.N.N.H	255.255.255.0
D	224 – 239	1110XXXX	Reserved for multi casting			
E	240 – 255	1111XXX0	Reserved for experimental and future use			

Note that X= 0 or 1, also N = Network portion and H = Host portion

Chapter 3 – Understanding Subnetting

Understanding Subnetting:

Subnetting allows you to create multiple logical networks that exist within a single Class A, B, or C network. If you do not subnet, you are only able to use one network from your Class A, B, or C network, which is unrealistic. Each data link on a network must have a unique network ID, with every node on that link being a member of the same network. If you break a major network (Class A, B, or C) into smaller subnetworks, it allows you to create a network of interconnecting subnetworks. Each data link on this network would then have a unique network/subnetwork ID.

In order to subnet a network, extend the natural mask using some of the bits from the host ID portion of the address to create a subnetwork ID. For example, given a Class C network of 204.17.5.0 which has a natural mask of 255.255.255.0, you can create subnets in this manner:

```
204.17.5.0
11001100.00010001.00000101.00000000
255.255.255.224
11111111.11111111.11111111.11100000
---------------------------------|sub|-----
```

By extending the mask to be 255.255.255.224, you have taken three bits (indicated by "sub") from the original host portion of the address and used them to make subnets. With these three bits, it is possible to create eight subnets. With the remaining five host ID bits, each subnet can have up to 32 host addresses, 30 of which can actually be assigned to a device since host ids of all zeros or all ones are not allowed (it is very important to remember this). So, with this in mind, these subnets have been created.

```
204.17.5.0   255.255.255.224 host address range 1   to 30
204.17.5.32  255.255.255.224 host address range 33  to 62
204.17.5.64  255.255.255.224 host address range 65  to 94
204.17.5.96  255.255.255.224 host address range 97  to 126
204.17.5.128 255.255.255.224 host address range 129 to 158
204.17.5.160 255.255.255.224 host address range 161 to 190
204.17.5.192 255.255.255.224 host address range 193 to 222
204.17.5.224 255.255.255.224 host address range 225 to 254
```

Note: There are two ways to denote these masks. First, since you are using three bits more than the "natural" Class C mask, you can denote these addresses as having a 3 bit subnet mask. Or, secondly, the mask of 255.255.255.224 can also be denoted as /27 as there are 27 bits that are set in the mask. This

second method is used with CIDR. Using this method, one of these networks can be described with the notation prefix/length.

For example, 204.17.5.32/27 denotes the network 204.17.5.32 with subnet mask of 255.255.255.224.

Take a look at how a Class B network might be subnetted. If you have network 172.16.0.0, then you know that its natural mask is 255.255.0.0 or 172.16.0.0/16. Extending the mask to anything beyond 255.255.0.0 means you are subnetting. You can quickly see that you have the ability to create a lot more subnets than with the Class C network. If you use a mask of 255.255.248.0 (/21), how many subnets and hosts per subnet does this allow for?

You are using five bits from the original host bits for subnets. This allows you to have 32 subnets. After using the five bits for subnetting, you are left with 11 bits for host addresses. This allows each subnet so have 2048 host addresses, 2046 of which could be assigned to device.

Chapter 4 – Easy Subnetting Steps

Steps of Subnetting

1. Find the number of networks required.

2. Find the number of bits to borrow (Use the chart below).

...	10	9	8	7	6	5	4	3	2	1	0
...	1024	512	256	128	64	32	16	8	4	2	1

For Example, if we require 16 networks, number of bits to be borrowed =4.

...	10	9	8	7	6	5	4	3	2	1	0
...	1024	512	256	128	64	32	16	8	4	2	1

3 Find the Increment number on the chart.

In the above example, increment no. = 16

4.Write the New mask (256 – Increment)

5. Write the new network numbers. Use the increment to write the numbers.

Note : First network will be the given network (with new subnet mask), new network will be previous network plus increment and so on.

Write the range of valid hosts and the broadcast address for each network.

Note to the instructor: Explain the subnetting steps by solving examples each from Class A, Class B and Class C

Chapter 5 – Subnetting Exercise

1. You have a Class C address of 192.168.5.0. You would like to break it into 7 Subnets. Write the new Subnet Mask, First, Last and Broadcast addresses for the new Subnetworks.

2. You have a Class B address of 150.5.0.0. You would like to break it into 15 Subnets. Write the new Subnet Mask, First, Last and Broadcast addresses for the First 5 Subnetworks.

3. You have a Class A address of 50.0.0.0. You would like to break it into 50 Subnets. Write the new Subnet Mask, First, Last and Broadcast addresses for the First 5 Subnetworks.

4. If you have sub-netted a network 172.16.0.0 with a mask of /20. Which of the following addresses are broadcast addresses? (Choose all that apply)

a. 172.16.32.255

b. 172.16.47.255

c. 172.16.79.255

d. 172.16.159.255

5. What would your subnet mask be if you want 5 networks with 20 hosts each?

6. You are required to break the 172.15.0.0 network into subnets having a capacity of 450 hosts with the maximum allowed subnets. What would your mask be?

7. Convert 1101 1001 into Decimal and Hex.

8. If your mask is 255.255.255.224, which of the following addresses are valid IP Addresses? (Choose all that apply)

a. 192.165.4.37

b. 195.5.2.63

c. 172.6.5.32

d. 11.5.1.94

9. If your mask on a Class C network is /29, how many subnets and host per

subnet do you have?

10. What is the binary range of Class A, Class B and Class C addresses?

11. If you routers ID is 192.168.1.60/240, what is the range of valid addresses that you can configure for a PC connected to the same Interface?

Chapter 6 – OSI Reference Model

Layering Benefits & Reasons

1. To divide the interrelated aspects of network operation into less complex operations.

2. To define standard interfaces to achieve compatibility and multi-vendor integration.

3. To achieve a modular approach to networking protocols so new applications and services can be deployed without redesigning other layers.

4. To keep changes in one area from affecting other layers.

5. To ease troubleshooting using data packets which will have specific information about each layer.

5. TCP and UDP uses port numbers to multiplex from the Transport layer through

Layer 7	Application	Network Processes to Applications
Layer 6	Presentation	Data Representation
Layer 5	Session	Inter-host Communication
Layer 4	Transport	End-to-end Communications
Layer 3	Network	Address and Best Path
Layer 2	Data Link	Access to Media
Layer 1	Physical	Binary Transmission

Application Layer (Layer 7)

1. The application layer is the OSI layer that is closest to the user.
2. It provides network services to the user's applications.
3. It differs from the other layers in that it does not provide services to any other OSI layer, but rather, only to applications outside the OSI model.
4. Examples of such applications are spreadsheet programs, word processing programs, and bank terminal programs.
5. The application layer establishes the availability of intended communication partners, synchronizes and establishes agreement on procedures for error recovery and control of data integrity.
6. Examples protocols include ftp, tftp, http, https, DNS, SMTP, telnet.

Presentation Layer (Layer 6)

1. Defines data format for transmission.
2. Ensures arriving data from the network can be used by the application and information sent by the application can be transmitted on the network.
3. Performs encryption and decryption.
4. Example representations include ASCII, EBCDIC, JPEG, TIFF, PICT, MPEG, MIDI, and HTML.

Session Layer (Layer 5)

1. The session layer defines how to start, control and end conversations (called sessions) between applications.
2. This includes the control and management of multiple bi-directional messages using dialogue control.
3. It also synchronizes dialogue between two hosts' presentation layers and manages their data exchange.
4. The session layer offers provisions for efficient data transfer.
5. Examples: - SQL, ASP(AppleTalk Session Protocol).

Transport layer (Layer 4):

1. The transport layer regulates information flow to ensure end-to-end connectivity between host applications reliably and accurately.
2. The transport layer segments data from the sending host's system and reassembles the data into a data stream on the receiving host's system.
3. The boundary between the transport layer and the session layer can be thought of as the boundary between application protocols and data-flow protocols. Whereas the application, presentation, and session layers are concerned with application issues, the lower four layers are concerned with data transport issues.
4. Layer 4 protocols include TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

to the Application layer.

6. Connection-oriented protocols establish and terminate sessions, for example, the TCP 3-way handshake.

7. Ports are defined in RFC 1700.

8. The first 1023 ports are reserved, or well-known ports used by the Operating System.

9. The remaining ports (1024 – 65,535) are available for use by client/server-based applications.

10. Example TCP based applications are:
FTP (Port 20, 21), Telnet (Port 23), SMTP (Port 25), SNMP (Port 161), SSH (Port 22), HTTP (Port 80), HTTPS (Port 443), BGP (Port 179), MPLS Unicast (Port 646)

11. Example UDP based applications are:
DNS (Port 53), TFTP (Port 69), DHCP Server (Port 67), DHCP Client (Port 68), RIP (Port 520), ISAKMP (Port 500)

Chapter 7 – TCP Header

TCP Header

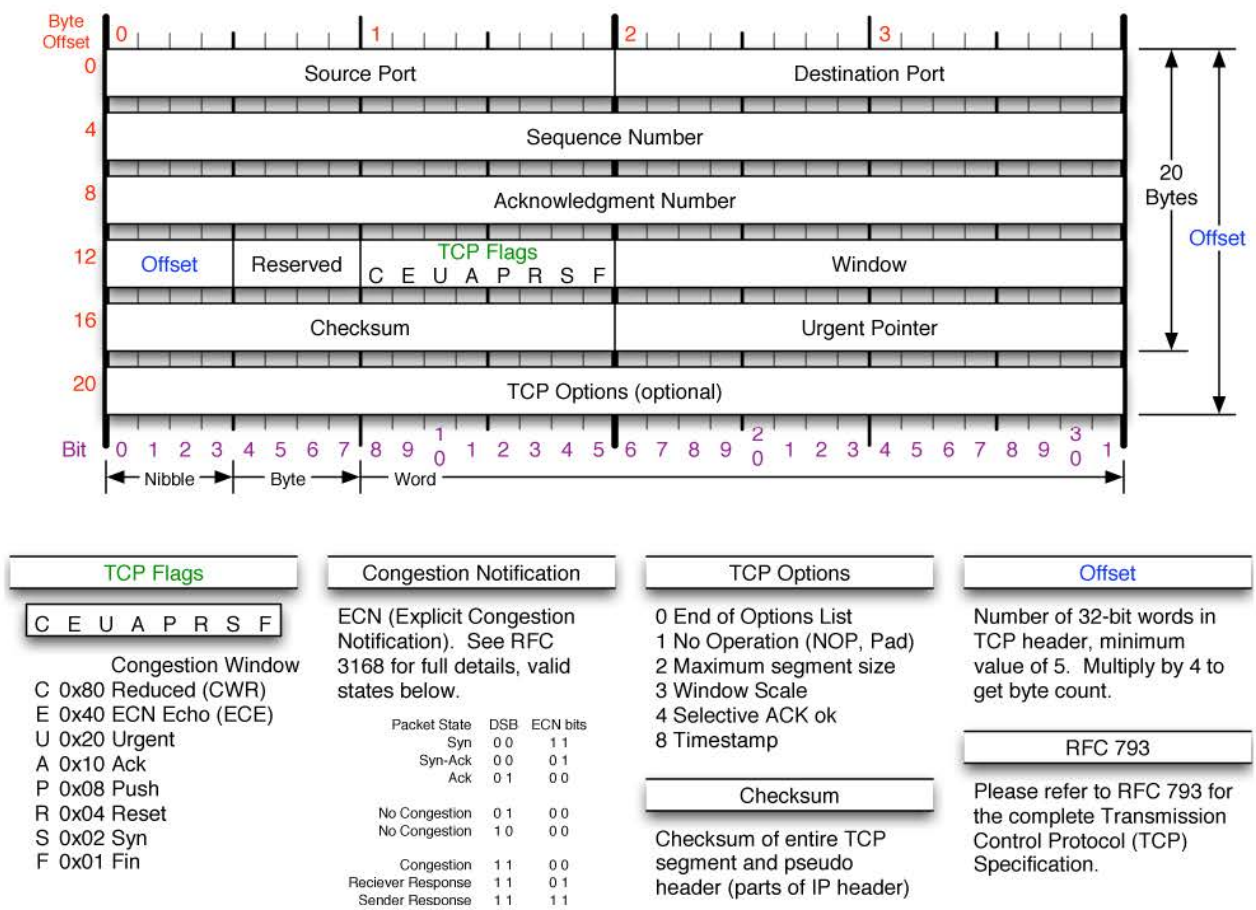


Figure 1: TCP Header

The 16-bit **Source Port** field identifies the application service on the sending host. The 16-bit **Destination Port** field identifies the application service on the remote host.

The 32-bit **Sequence Number** field is used both during connection establishment, and during data transfer. During connection establishment (SYN message), an initial sequence number is randomly chosen.

Subsequently, sequence numbers are used to identify data bytes in a stream.

The 32-bit **Acknowledgment Number** field, as its name suggests, is used to acknowledge a sequence number. During connection setup, this is set to the sending host's initial sequence number + 1. During data transfer, this value is used to acknowledge receipt of a group of data bytes.

The 4-bit **Data Offset field** indicates where data begins in a TCP segment, by identifying the number of 32-bit multiples in the TCP header. A TCP header must end on a 32-bit boundary.

Following the data offset field is the 6-bit **Reserved** (for future use) **field**, which is always set to zeroes.

The 6-bit Control Bits field contains six 1-bit flags, in the following order:

- **URG (Urgent) –** prioritizes specified traffic.
- **ACK (Acknowledgment) –** acknowledges a SYN or receipt of data.
- **PSH (Push) –** forces an immediate send even if window is not full.
- **RST (Reset) –** forcefully terminates an improper connection.
- **SYN (Synchronize) –** initiates a connection.
- **FIN (Finish) –** gracefully terminates a connection when there is further data to send.

The 16-bit **Window field** identifies the number of data octets that the receiver is able to accept.

The 16-bit **Checksum field** is used for error-checking, and is computed using both the TCP segment and select fields from the IP header. The receiving host will discard the segment if it fails the checksum calculation.

The 16-bit **Urgent Pointer field** is used to identify the last byte of prioritized traffic in a segment, when the URG flag is set.

The variable-length **Options field** provides additional optional TCP parameters, outside the scope of this guide.

The variable-length **Padding field** ensures the TCP header ends on a 32-bit boundary, and is always set to zeroes.

Chapter 8 – UDP Header

UDP Header

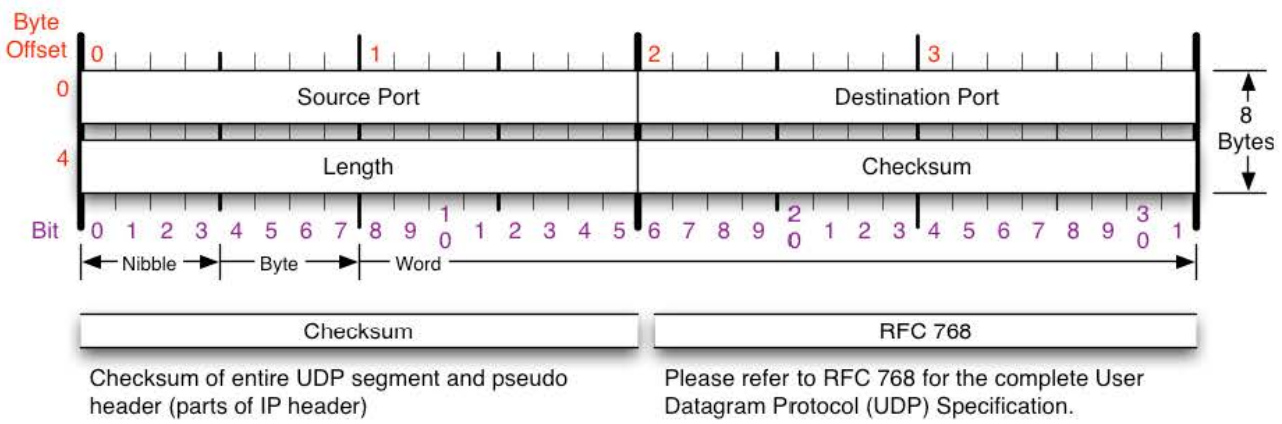


Figure 1: UDP Header

The **User Datagram Protocol (UDP)** is a connectionless transport protocol, and is defined in RFC 768.

UDP, above all, is simple. It provides no three-way handshake, no flowcontrol, no sequencing, and no acknowledgment of data receipt. UDP essentially forwards the segment and takes no further interest.

Thus, UDP is **inherently unreliable**, especially compared to a connectionoriented protocol like TCP. However, UDP **experiences less latency** than TCP, due to the reduced overhead. This makes UDP ideal for applications that require speed over reliability. For example, DNS primarily uses UDP as its transport protocol, though it supports TCP as well.

Like TCP, UDP does provide basic error-checking using a checksum, and uses port numbers to differentiate applications running on the same host.

The following provides a quick comparison of TCP and UDP:

TCP	UDP
Connection-oriented	Connectionless
Guarantees delivery	Does not guarantee delivery
Sends acknowledgments	Does not send acknowledgments
Reliable, but slower than UDP	Unreliable, but faster than TCP
Segments and sequences data	Does not provide sequencing
Resends dropped segments	Does not resend dropped segments
Provides flow control	Does not provide flow control
Performs CRC on data	Also performs CRC on data
Uses port numbers	Also uses port numbers

MODULE 2

ROUTER BASICS

Lab 1 – Router Basics

Router Configuration Sources

Routers can be configured from:

1. Console terminal.
2. Auxiliary port – externally, via modems.
3. Virtual terminals (Telnet) – after installation.

Internal Configuration Components

RAM

Contains dynamic / running configuration

NVRAM

Contains backup of configuration (startup configuration)

Flash

Contains copy of Cisco IOS

ROM

Contains a subset of IOS

Contains bootable IOS image

Interfaces

Network connections which packets enter/exit from routers, e.g. Ethernet, serial, BRI, Token Ring

Console and auxiliary ports

Main command-line interface used for configuration.

Router Startup Sequence – Summary

1. Bootstrap program loaded from ROM
2. Bootstrap runs the POST
3. Bootstrap locates IOS in Flash
4. IOS is expanded and then loaded into RAM
5. Once IOS is loaded into RAM, it looks for startup-config in NVRAM
6. If found, the configuration is loaded into RAM

Router Modes

User EXEC mode (look, but don't change)

Automatically enter this mode when router is turned on.

```
System Configuration Dialog ---
Continue with configuration dialog? [yes/no]:no
Press enter to get started
Router>
```

Lab 2 – Connecting a Dump Terminal / PC to the Router

Task 1: Connect PC to a router

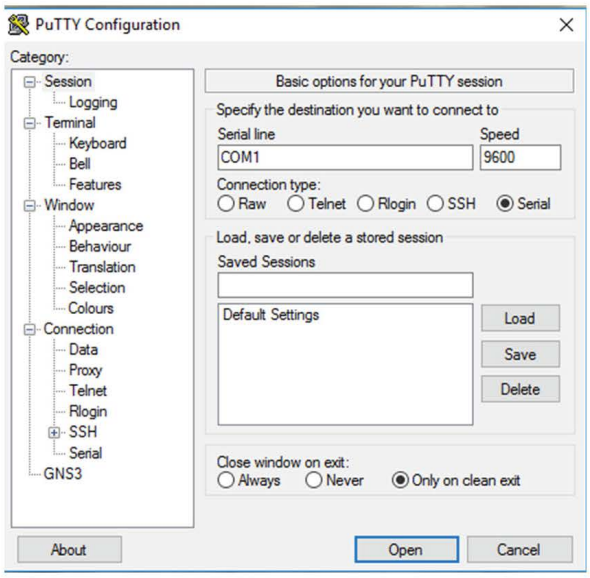
Following are the steps to connect Dump Terminal PC to Router

- Connect the Console Adapter to either Com1 or Com2 on the back of your PC.
- Connect the Rollover cable from the back of the Console Adapter to the Console Port on the Router.
- Make sure the PC is up and running before turning the Router on.

Task 2: Configure hyper terminal

Hyper Terminal Configurations

- Download and Install Putty (visit www.putty.org)
- Open putty
- Select connection type : Serial
- Specify that you are using either Com1 or Com2 (based on what port you connected the console adapter to).
- Make sure speed is set 9600
- Press Open.
- If the router is up, you should see the Router prompt



Now you can perform basic tasks, such as connect to remote devices, perform basic tests.

```
--- System Configuration Dialog ---
Continue with configuration dialog? [yes/no]:no
```

Note: Always do "no"

Router>

Privileged EXEC mode

High-level testing commands

Set operating parameters

Command to enter:

```
Router>enable
Router#
```

Global configuration mode

Commands apply to features that affect the system as a whole

Enter from privileged EXEC mode with command:

```
Router#configure terminal
Router(config)#
```

Interface mode

Configure interface, such as Ethernet, serial

Enter from global configuration mode with command:

```
Router(config)#interface fastethernet0/0
```

Or

```
Router(config)#interface serial0/0
Router(config-if)#
```

Setup mode

Helps new user to create a configuration for the first time, via a series of questions

Prompted at bootup or enter setup at:

```
Router#setup
--- System Configuration Dialog ---
Continue with configuration dialog? [yes/no]:
```

Rommon mode

Provides router with a small subset of IOS and helps router boot if IOS not found in Flash

Rommon 1>

(You will learn more about rommon mode at the end of the course)

Lab 3 – Stepping Through Different Command Modes and Getting Help

Task 1: Stepping Through Different Command Modes

Router> indicates that you are in User Exec Mode.

On the Router >

```
Router>enable
Press Enter
```

Your prompt should look like

```
Router#
```

The Router# indicates that you are in Privileged Exec Mode

```
Router#disable
```

It will take you from Privileged Mode to User Mode. Your prompt should look like

```
Router>
```

You can also type in Router#en to go into Privileged Mode from User Mode. The reason being, there is no other command in User Exec mode that starts with the letters "en"

Similarly, you can type Router#disa instead of Router#disable at the Privilege Mode to go into User Exec Mode.

If you want to exit out completely, type Router#logout

Logout will log you out of the router. You should see the prompt asking you press **Enter** to get started.

```
Press Enter.
```

You can also type

```
Router#exit
```

to logout of the Router from either User or Privilege Mode.

Press Enter to get started again.

Type? It displays all the commands that can be type in the current Mode (User Exec). It will give a short description about each command and stop after each page.

Press the Enter Key.

What happens?

Press the Spacebar Key. What happens? Write it down.

```
Router#?
```

If you want to exit out of help without seeing all the commands press Ctrl-C or Esc.

```
Router>enable
Router#?
```

followed by the space bar key until you return back to the prompt.

Did you see more commands when Typed Router#? in Privileged Mode than in User Mode?

Cisco Help is Context sensitive. It displays help based on where you typed?

If you wanted to find out about all the commands that start with a specific letter, you can type that letter followed by? It will only display commands that start with that letter.

```
Router#s?
```

What does it show you?

To go to the Global Configuration mode, type from the Privileged Mode.

```
Router#config terminal
```

You can also type

```
Router#conf t
```

to have the same effect.

Your prompt should look like

```
Router(config)#
```

This is the prompt for Global Configuration Mode.

```
Router#exit
```

To go down one level you could use exit.

```
Router#disable
```

This should take you to User Exec Mode. Can we get to configuration mode from here?

```
Router>config t
```

What happens?

```
Router>en
```

Your prompt should look like

```
Router#
Router#config t
```

Your Prompt should like

```
Router(config)#
```

Can we logout from here?

```
Router#logout
```

What happens?

Task 2:

How to Configure a Specific Interface?

To go to configure a specific interface, you have to go into that interface. The command that will allow you to go into a specific interface is as follows:

```
Router#interface <Type> <Slot/Port>
Router#interface fastethernet 0/0
```

You could also have typed

```
Router#int f0/0
```

This allows you to configure the Ethernet interface 0/0. The prompt should look like

```
Router(config-if)#
```

If you wanted to configure the Serial interface, type

```
Router#interface Serial 0/0 or Router#int S 0/0
```

Does your prompt change?

The prompt for all your interfaces is the generic

```
Router(config-if)#
```

To go back to Global Configuration,

Type

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

Type

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

to go back into interface configuration mode.

To go back directly back into Privileged Mode, you can either type Ctrl-Z or end.

```
Router(config-if)#end
Router(config)#config t
Router(config)#int f0/0
Router(config-if)#Ctrl-Z
```

In the Privileged Mode,

Lab 4 – Stepping through Context-sensitive Help to set the Time for the Router

Task 1: Configure Clock

What command will allow you to set the Clock?

```
Type Router#clock ? What should you type next?
Type Router#clock set ? What should you type next? (HH:MM:SS)
Type Router#clock set 17:25:00 and Press Enter.
What is response?
Type Router#clock set 17:25:00 1 ? What should you type next? (DD:MM:YYYY)
Type Router#clock set 17:01:00 30 january? What should you type next?
Type Router# clock set 17:25:00 30 january 2017 and Press Enter.
```

Verification:

```
Router> show clock
*17:01:05.415 UTC Mon Jan 30 2017
```

Note: Time Zone is set to UTC

Can we change the time zone?

Lab 5 – Configure Terminal History

Task 1: Checking Terminal history

What happened when we pressed up arrow?

```
Router#show history
```

It shows the last set of commands you have typed.

By default, the router will keep track of the last 10 commands.

Task 2:

Change the terminal history size to 100

```
Router#terminal history size 100
```

Changes the history size to 100

Task 2: Verfiy the history

```
Router#show terminal
```

To verify the change. (Towards the bottom of the output)

Lab 6 – Setting Router Name and Banner

Task 1:

Change the Router hostname to DELHI

```
Router(config)#hostname DELHI
DELHI(config)#
```

Task 2:

```
Router(config)#banner motd #Welcome to KBITS, Delhi#
```

Displayed when router is accessed

Displayed prior to prompting for a password

Task 3:

```
Router(config)#banner login #Welcome to CCNAX Courses at KBITS#
```

Displayed after successful login

Lab 7 – Save Configuration and Display Various Status Commands

Task 1:

Save the configuration

```
Router#copy running-config startup-config
```

Task 2:

To delete startup configuration

```
Router#erase startup-config
```

Then reload

```
Router#reload
```

Task 3:

Verifying Configuration

```
Router(config)#show running-config
```

Task 5:

Display IOS version, image file name and location, router is uptime

```
Router#show version
```

Task 6:

Display backup configuration

```
Router#show startup-config
```

Task 7:

Display current, active configuration

```
Router#show running-config
```

Task 8:

Display flash memory

```
Router#show flash
```

Lab 8 – Setting Router Passwords

Task 1:

Configure Console Password

To set the Line Console Password

```
Router(config)#line con 0
Router(config)#login
Router(config)#password xxxxxx
```

Task 2:

Configure Enable Password

To setup enable password

```
Router(config)#enable password xxxxxxx
```

Task 3: Configure Encrypted Enable Mode Password

To set the enable secret

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

Note: Enable secret overrides the enable password.

Lab 9 – Editing Keys

Task 1:

Test various editing keys on console

Press **CTRL – P** It will show you the Previous Command.

Press **CTRL - P** It will show you the command you typed before the previous command.

Press **CTRL – N** It will show you the Next Command.

Where is the cursor at? Let us say that you want to change something at the beginning of the line. Rather than using the arrow keys to scroll to the beginning of the line, you can accomplish the same by pressing

CTRL – A

Press **CTRL-A**. Takes the cursor should be at the beginning of the line.

Press **CTRL – E.** Takes the cursor to the end of the line.

Lab 10 – Some Show Commands

Task 1:

Type and verify various show commands

All show commands are typed in Privilege Exec Mode (#).

```
Router#show interfaces serial 0/0
Serial0/0 is administratively down, line protocol is down
    Hardware is GT96K Serial
    MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
      reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation HDLC, loopback not set
```

What is the status of the line? What is the Encapsulation type on the Serial interface?

```
Router#show version
Cisco IOS Software, 3700 Software (C3725-ADVENTERPRISEK9-M), Version 12.4(15)T14, RELEASE
SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Tue 17-Aug-10 12:08 by prod_rel_team
```

What does this command display?

What is the name of the file that was used to boot the Router?

How many interfaces does your router have?

```
Router#show ip interface brief
Interface              IP-Address      OK? Method  Status    Protocol
FastEthernet0/0        unassigned      YES unset   administratively   down down
Serial0/0              unassigned      YES unset   administratively   down down
FastEthernet0/1        unassigned      YES unset   administratively   down down
Serial0/1              unassigned      YES unset   administratively   down down
```

Lab 11 – Disabling Domain-Lookup, Synchronizing the Console Line and Session Time-Out

Task 1:

Disabling Domain-lookup

In global configuration mode, Type

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

This command prevents the router from doing a Name lookup if you mistype a command?

Task 2:

Synchronizing the console line

In global configuration mode, Type

```
Router(config)#line console 0
Router(config-line)#logging synchronous
```

This prevents console messages from getting inserted into your command as you are typing.

Task 3:

Prevent the session from getting timed

```
Router(config-line)# no exec-timeout
```

This command prevents the session from getting timed out after 2 minutes of idle time.

Lab 12 – Creating Aliases

Task 1:

Create a shortcut shr for the show run command.

```
Router(config)#alias exec shr show running-config
```

Create a shortcut shs for the show start command.

```
Router(config)#alias exec shs show startup-config
```

Create a shortcut ship for the show ip int brief command.

```
Router(config)#alias exec ship sh ip int brief
```

Create a shortcut shv for the show version command.

```
Router(config)#alias exec shv sh ver
```

Create a shortcut cc for the config t command.

```
Router(config)#alias exec cc config t
```

Task 2:

Try the different aliases you have created by typing them one at a time.

Lab 13-
Setting a Line Console Password

Task 1:

Set the Line Console Password to newyork

```
Router>en
Router#config t
Router(config)#line console 0
Router(config-line)#login
Router(config-line)#password newyork
Router(config-line)#end
```

Task 2: Testing the Line Console Password

Type Logout to exit out of the router's console. You should see a message that says

```
"Press Return to get started".
```

Press Enter

Do you get a prompt for password to get into User Exec mode?

Type newyork (The password that was set).

Are you in User Exec mode?

Type enable to get into Privilege Exec mode.

Did it prompt you for a password?

Lab 14 – Setting a Enable Password

Task 1: Setting the Enable Password

Set the Enable Password to cisco@123 on Router

```
Router>en
Router#config t
Router(config-line)#enable password cisco@123
Router(config-line)#end
```

Task 2: Testing the Enable Password

Type Logout to exit out of the router's console. You should see a message that says

```
"Press Return to get started"
```

Press Enter.

Do you get a prompt for password to get into User Exec mode?

Type newyork (The Console password that was set in previous lab)

Are you in User Exec mode?

Type enable to get into Privilege Exec mode.

Did it prompt you for a password?

Type cisco@123. (The enable password that was set in this lab).

Are you in Privilege Exec Mode?

Lab 15 – Setting a Enable Secret Password

Task 1:

Set the Enable Secret Password to kbits@123 on router

```
Router>en
Router#config t
Router(config-line)#enable secret kbits@123
Router(config-line)#end
```

Task 2:

Type Logout to exit out of the router's console. You should see a message that says

```
"Press Return to get started"
```

Press Enter

Do you get a prompt for password to get into User Exec mode?

Type newyork (The Console password that was set).

Are you in User Exec mode?

Type en to get into Privilege Exec mode.

Did it prompt you for a password?

Type cisco@123. (The enable password that was set).

Did it work?

Type kbits@123

Did it work?

Type show running-config

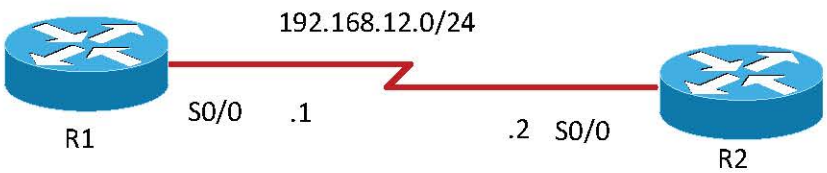
Do you see both passwords?

Which password works?

MODULE 3

BASICS IPV4 LABS

Lab 1 – Basic Serial Connection (HDLC)



Task 1:

Finding the DCE (Clock Source) and DTE

```
R1#show controller serial 0/0
MK5 unit 0, NIM slot 1, NIM type code 7, NIM version 1
idb = 0x6150, driver structure at 0x34A878, regaddr = 0x8100300
IB at 0x6045500: mode=0x0108, local_addr=0, remote_addr=0
N1=1524, N2=1, scaler=100, T1=1000, T3=2000, TP=1
buffer size 1524
DTE V.35 serial cable attached
```

```
R2#show controllers serial 0/0
M1T-E3 pa: show controller:
PAS unit 0, subunit 0, f/w version 2-55, rev ID 0x2800001, version 2
idb = 0x6080D54C, ds = 0x6080F304, ssb=0x6080F4F4
Clock mux=0x30, ucmd_ctrl=0x0, port_status=0x1
line state: up
E3 DTE cable, received clockrate 50071882
```

Look for the word DCE or DTE in the above output.

If it says DCE, you will provide the Clock (Speed for the link). It is normally specified by the Telephone Company based on your contract.

Task 2:

Configure the Serial Interface on R1 with IP Addresses and bringing them up.

Interface	Type	Clock Rate	IP Address	Subnet Mask
S0/0	DCE	128000	192.168.1.1	255.255.255.0

Go into Global configuration mode by typing

```
R1#config t
R1(config)#int serial0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R2(config-if)#no shutdown
```

Task 3:

Configure the Serial Interface on R2 with IP Addresses and bringing them up.

Interface	Type	Clock Rate	IP Address	Subnet Mask
S0/0	DTE	-	192.168.1.2	255.255.255.0

Go into Global configuration mode by typing

```
R1#config t
R1(config)#int serial0/0
R1(config-if)#ip address 192.168.1.2 255.255.255.0
R2(config-if)#no shutdown
```

Verifying the connection

Make sure both routers are configured before proceeding to the following section.

Task 4: Verify your configuration

```
R1#show ip interface brief
```

What is the status of your Serial line?

Type Ping 192.168.1.Y where Y is your partner's IP address.

```
R1#ping 192.168.1.2
R2#ping 192.168.1.1
```

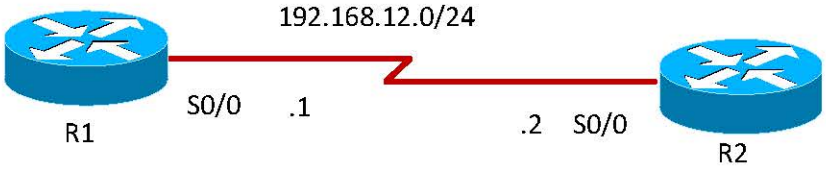
Are you successful?

```
Router#show interface S0/0
```

What is the encapsulation type?

Can you **authenticate** the routers with this type of encapsulation?

Lab 2 – Basic Serial Connection (PPP)



Task 1:

Change the encapsulation to ppp. To change the encapsulation of the interface to PPP, type the following commands on **both** routers:

```
RX>en
RX#config t
RX(config)#int S 0/0
RX(config-if)#encapsulation ppp
RX(config-if)#end
```

Task 2: Verify the encapsulation

```
Router#show interface S0/0
```

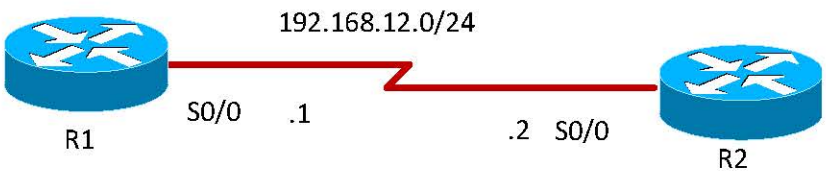
What is the encapsulation type?

Ping your Partner's router.

Are you successful?

What are the advantages of using PPP over the Cisco Proprietary HDLC?

Lab 3 – Configure PPP Authentication PAP



Task 1:

Configure username ROUTER2 and password cisco on R1

```
R1(config)#username ROUTER2 password cisco
```

Task 2:

Change the authentication to ppp and enable pap authentication on R1

```
R1(config)#interface serial0/0
R1(config-if)#encapsulation ppp
R1(config-if)#ppp authentication pap
```

Task 3:

Configure R2 to authenticate the serial0/0 connection using the username and password configured on server R1.

```
R2(config)#interface serial0/0
R2(config-if)#encapsulation ppp
R2(config-if)#ppp pap sent-username ROUTER2 password cisco
```

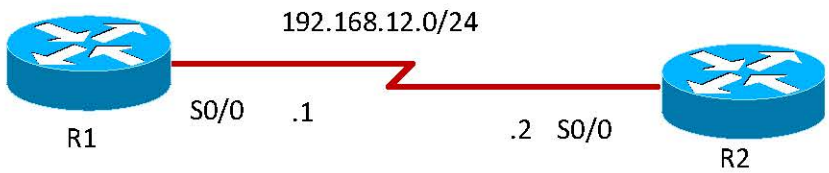
Task 4:

Verify by debugging

```
R1#debug ppp authentication
R1#debug ppp negotiation
```

Check the debug output

Lab 4 – Configure PPP Authentication CHAP



CHAP can be done by two methods

A. LOCAL DATABASE

Task 1:

R1 need to be authenticating server configure the following command in R1. Create username R2 and password kbits@123

```
R1(config)#username R2 password kbits@123
```

Task 2:

Change the authentication to ppp and enable chap authentication on R1

```
R1(config)#interface serial 0/0
R1(config-if)#no shutdown
R1(config-if)#encapsulation ppp
R1(config-if)#ppp authentication CHAP
```

Task 3:

Configure PPP encapsulation on Serial. R2 need to be authenticating Client. Configure username R1 and password cisco@123 on R2. Change encapsulation ppp on serial 0/0.

```
R2(config)#username R1 password kbits@123
R2(config)#interface serial 0/0
R2(config-if)#no shutdown
R2(config-if)#encapsulation ppp
```

Task 4:

To test configuration, enable debug

```
R1# debug PPP authentication
R1(config-if)#shutdown
R1(config-if)#no shutdown
```

B. INTERFACE CONFIGURATION

Task 1:

R1 need to be authenticating server. Configure username R1 and password cisco@123 on R1.

```
R1(config)#username R2 password cisco@123
R1(config)#interface serial 1/0
R1(config-if)#no shutdown
```

Task 2:

Configure PPP encapsulation and enable CHAP on Serial.

```
R1(config-if)#encapsulation ppp
R1(config-if)#ppp authentication chap
```

Task 3:

Configure R2 as Client. Configure CHAP username and password on interface. R2 need to be authenticating Client then configure the following commands in R2.

```
R2(config)#interface serial 1/0
R2(config-if)#no shutdown
R2(config-if)#encapsulation ppp
R2(config-if)#ppp CHAP hostname R1
R2(config-if)#ppp CHAP password cisco@123
```

Task 4:

To test configuration, enable debug on R1

```
R1#debug ppp authentication
R1(config-if)#shutdown
R1(config-if)#no shutdown
```

What is HDLC (High-Level Data Link Control)?

There are a couple of different encapsulation methods for PtP connections. Making sure you choose the correct one for your network can be a daunting task. Hopefully, this article will provide you with some information to make your choice.

1. HDLC is the primary (and default method on Cisco devices) for encapsulation on synchronous serial links. There are two types of HDLC: Standard and Cisco. Standard HDLC only supports single-protocol

environments, where Cisco HDLC can support multiple environments.

2. HDLC works best for connection between two Cisco devices where you control the link. It doesn't offer much in the way of security, but is really easy to use. Great for connections between routers in the office or leased lines through the Net.

What is PPP?

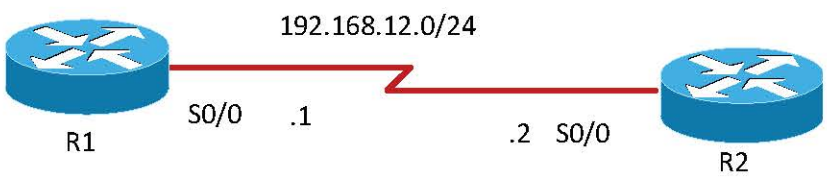
1. PPP uses NCP (Network Control Program) to encapsulate multiple protocols (TCP/IP, Novell, IPX). PPP also uses LCP (Link Control Protocol) to establish and maintain the link. PPP can be configured to provide different features, such as Authentication, Compression, Multilink and Error Detection.

2. PPP is a more robust encapsulation method, providing some security (CHAP is the best way to go), error detection and the ability to load balance over multiple links. It is more difficult to setup and the debug options can be a pain to sort through. I would use this protocol to load balance or across unsecured WAN connections.

MODULE 4

ROUTER MANAGEMENT

Lab 1 – Configuring Telnet Management



Task 1:

Testing the Telnet Password.

Type

```
R1#telnet 192.168.1.2
```

What message do you get?

Task 2:

Set the Telnet Password on R2 to cisco@123

```
R1>en
R1#config t
R1(config)#line vty 0 4
R1(config-line)#login
R1(config-line)#password cisco@123
```

Task 3:

Testing the Telnet Password again

```
R1#telnet 192.168.1.2
Trying 192.168.1.2 ... Open
User Access Verification
Password:
```

Did you get a password prompt?

Do you see your partner's Router prompt?

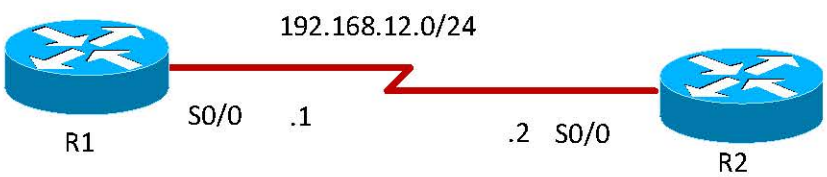
Type

```
R2> en
```

To exit the remote session, type

```
R2>quit
```

Lab 2 – Configuring Telnet with local user name and password



Task 1:

Create username admin and password cisco on R2.

```
R1(config)#username admin password cisco
```

Task 2:

Set the Telnet to use local username/password to authenticate users.

```
R1>en
R1#config t
R1(config)#line vty 0 4
R1(config-line)#login local
```

Task 3:

Testing the Telnet Password again

```
R1#telnet 192.168.1.2
Trying 192.168.1.2 ... Open
User Access Verification
Username: admin
Password:
```

Did you get a password prompt?

Do you see your partner's Router prompt?

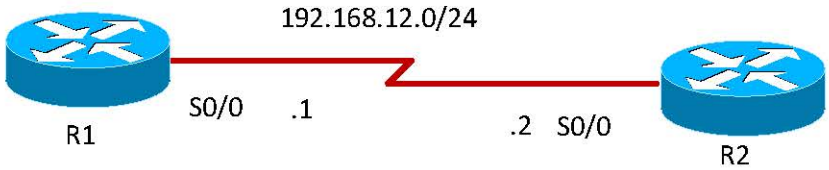
Type

```
R2> en
```

To exit the remote session, type

```
R2>quit
```

Lab 3 – Configuring SSH with Local Username and Password



Task 1:

Create username admin and password cisco on R2.

```
R2(config)#username admin password cisco
```

Task 2:

Configure domain-name kbits.com

```
R2(config)#ip domain-name kbits.com
```

Task 3:

Generate crypto key with modulus of 1024 to enable SSH v2.

```
R2(config)#crypto key generate rsa modulus 1024
The name for the keys will be: R2.kbits.in
% The key modulus size is 1024 bits
 % Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
R2(config)#
*Mar  1 00:27:57.767: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

Task 4:

Set the SSH to use local username/password to authenticate users.

```
R1>en
R1#config t
R1(config)#line vty 0 4
R1(config-line)#login local
```

Task 5:

Test to establish SSH connection to R2 from R1.

```
R1#ssh -l admin 192.168.1.2
```

MODULE 5

BASIC ROUTING

Chapter 1 – Basic Routing

Basic Terms

Routing Tables

Routers build routing tables initially based on their **directly connected** networks.

In addition to directly connected networks, routers can learn about destinations in one of three ways:

Static Routes: Manually added to the routing tables by the administrator.

Default Routes: Manually added to the routing table by the administrator to define a Default Gateway for the router. If the routing table does not have an entry for a destination network, send the packet to the Default Route.

Dynamically: Learned through a Routing Protocol.

Routing tables are used to send data along specific paths to reach a particular destination.

Routers need to exchange routing tables so they can route data to networks that are not directly connected to them.

Routers require a Routing Protocol in order to exchange routing tables with their neighboring routers and advertise networks.

Static Routes

Static Routes are User-defined, manually created routes.

The administrator creates Static Routes in a Cisco Router using the ip route Command

```
Syntax: ip route destination-network subnet-mask Next-Hop-Router-IP-Address {distance}
```

For example,

```
R1>enable
R1#config terminal
R1(config)#ip route 2.0.0.0 255.0.0.0 192.168.1.2
```

Default Routes

Default Routes define a router as the default router for your router. When there is no entry for the destination network in a routing table, the router will forward the packet to its default router. Default routes help in reducing the size of your routing table.

For example,

```
R1>enable
R1#config terminal
R2(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.1
```

Routable and Routing Protocols

A **Routable Protocol** is a network protocol that transports data across a network with a structure, which allows it to be routed to the specified destination network.

A **Routing Protocol** is a method by which routers exchange information about the networks they can reach. Exchange of information allows routing tables to be built and exchanged. The process of updating routers is called convergence.

Routing Protocols determine the best path for the transport of data using some criteria, such as distance or metric. Examples include bandwidth, delay, hops and reliability.

Routing Protocols are divided into two groups:

Interior and Exterior

Interior Routing Protocols Include:

- Routing Information Protocol (RIP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Open Shortest Path First (OSPF)
- Intersystem-Intersystem (IS-IS)

Exterior Routing Protocols include:

- Border Gateway Protocol (BGP)

Two main Types of Interior Routing Protocols are Distance Vector and Link State.

The Routing updates include the entire routing table.

It uses a periodic update.

Routing Update packets are sent as broadcast. Unicast packets can also be specified.

Examples of Distance Vector Routing Protocols are RIP v1, RIP v2, IGRP.

Link State Routing Protocols

The Routing updates include only new changes to the routing table which saves bandwidth.

Handles larger networks and is more scalable than Distance Vector Routing Protocols.

Example OSPF, IS-IS.

Administrative Distance

Rating of the Trustworthiness of a routing information source.

The Number is between 0 and 255

The higher the value, the lower the trust. For example, 255 signify no trust and therefore it is ignored.

Lowest administrative distance is always chosen as the routing protocol to use to transport data.

Default administrative distances for common protocols are as follows:

Connected=0 Static Routes=1 EIGRP=90 OSPF=110 RIP=120

Protocol	AD Value (Default)
Connected	0
Static	1
EIGRP	90
RIP	120
OSPF	110
IS-IS	115
EIGRP External	170
eBGP	20
iBGP	200
Not available	255

Chapter 2 – Distance Vector Routing Protocols

Common Characteristics of Distance Vector Routing Protocols

Neighbors: As far as the routers are concerned the neighboring router is the one that shares a common data link. These routers have at least one interface on the same network.

Periodic Updates: The interval that the routers wait for before they advertise their routing table to neighboring routers.

RIP for IPv4 – 30 Seconds

RIPng for IPv6 – 30 Seconds

Broadcast Update: are used by routers to find other routers when they come online. They send their routing table to broadcast address of **255.255.255.255,** if the neighboring router talks the same routing protocol, it will respond and routers now know of each other.

Route Invalidation Timers: is the time that must pass before a Router considers a route to be invalid. If network 2.0.0.0 is connected to Router A and it goes down, Router A will notify its neighboring router, Router B of that fact. But what if Router A goes down?

This problem is handled by **Route Invalidation Timer** for each entry in the routing table. When Router B first hears about network 2.0.0.0 from Router A, it will set a route invalidation timer for that route. Since Router A was the one that gave him the news it expects Router A to keep updating that information on regular periodic updates, however if Router A fail to do so and misses x number of periodic updates, Router B will set that route in the routing table to unreachable.

Asynchronous Updates (Random Jitters or Time Jitters): Periodic Updates can collide and cause further delays in convergence. A Random Jitter will attempt to overcome this by introducing an offset value to the periodic update time, thus reducing the probability of updates colliding.

Routing Loops and Solutions

Routing Loops can occur if the network's slow convergence on a new configuration causes inconsistent routing entries.

Solutions to Routing Loops

Counting to Infinity: Distance Vector Routing Protocols define a maximum value for hops. The maximum Hop Count is 15 is commonly used.

Spilt Horizon: Spilt Horizon has two flavors, Simple Split Horizon and Spilt Horizon with Poison Reverse.

The logic behind Simple Spilt Horizon is that it is never useful to send information about a route back in the direction from which the information originally came. So if Router A learns about a Route through Router B, it will never send the same route back to Router A. This is known as suppressing routes.

Split Horizon with Poison Reverse does not work based on suppression, and it will include every route in its updates but it will tag them as unreachable. Let's say Router B receives a corrupted update believing that it can reach network 1.0 through Router C, Simple Split Horizon will not be able to avoid the loop, whereas Poison Reverse will definitely fix the problem. Router B will say 1.0 can be reached via Router C, but this time Router C will poison that route eliminating the routing loop.

Triggered Updates: Also known as flash updates. Changes to the network topology are sent instantaneously to neighboring routers.

Hold down Times: If the hop count to a given destination increases, the router sets a hold down timer for that route. By implementing this refinement we have reduced the likelihood of a bad or corrupted information getting into the routing table, but once again understand that nothing is free and in this case the trade off is convergence time.

Chapter 3 – Routing Information Protocols (RIP)

RIP Overview

The Routing Information Protocol (RIP) uses broadcast UDP data packets to exchange routing information. Cisco software sends routing information updates every 30 seconds, which is termed advertising. If a device does not receive an update from another device for 180 seconds or more, the receiving device marksthe routes served by the nonupdating device as unusable. If there is still no update after 240 seconds, the device removes all routing table entries for the nonupdating device. A device that is running RIP can receive a default network via an update from another device that is running RIP, or the device can source the default network using RIP. In both cases, the default network is advertised through RIP to other RIP neighbors. The Cisco implementation of RIP Version 2 (RIPv2) supports plain text and message digest algorithm 5 (MD5) authentication, route summarization, classlessinterdomain routing (CIDR), and variable-length subnet masks (VLSMs).

RIP Routing Updates

The Routing Information Protocol (RIP) sends routing-update messages at regular intervals and when the network topology changes. When a device receives a RIP routing update that includes changes to an entry, the device updates its routing table to reflect the new route. The metric value for the path is increased by 1, and the sender is indicated as the next hop. RIPdevices maintain only the best route (the route with the lowest metric value) to a destination. After updating its routing table, the device immediately begins transmitting RIP routing updates to inform other network devices of the change. These updates are sent independently of the regularly scheduled updates that RIP devices send.

RIP Routing Metric

The Routing Information Protocol (RIP) uses a single routing metric to measure the distance between the source and the destination network. Each hop in a path from the source to the destination is assigned a hop-count value, which is typically 1. When a device receives a routing update that contains a new or changed destination network entry, the device adds 1 to the metric value indicated in the update and enters the network in the routing table. The IP address of the sender is used as the next hop. If an interface network is not specified in the routing table, it will not be advertised in any RIP update

Authentication in RIP

The Cisco implementation of the Routing InformationProtocol (RIP) Version 2 (RIPv2) supports authentication, key management, route summarization, classless interdomain routing (CIDR), and variable-length subnet masks (VLSMs). By default, the software receives RIP Version 1 (RIPv1) and RIPv2 packets, but sends only RIPv1 packets. You can configure the software to receive and send only RIPv1 packets. Alternatively, you can configure the software to receive and send only RIPv2 packets. To override the default behavior, you can configure the RIP version that an interface sends. Similarly, you can also control how packets received from an interface are processed. RIPv1 does not support authentication. If you are sending and receiving RIP v2 packets, you can enable RIP authentication on an interface. The key chain determines the set of keys that can be used on the interface. Authentication, including default authentication, is performed on that interface only if a key chain is configured.

Cisco supports two modes of authentication on an interface on which RIPis enabled:

- Plain-text authentication
- Message digest algorithm 5 (MD5) authentication.

Plain-text authentication is the default authentication in every RIPv2 packet.

Keychains and Keychain Management

Keychain management allows you to create and maintain keychains, which are sequences of keys (sometimes called shared secrets). You can use keychains with features that secure communications with other devices by using key-based authentication. The device allows you to configure multiple keychains.

Some routing protocols that support key-based authentication can use a keychain to implement a hitless key rollover for authentication.

Lifetime of a Key

To maintain stable communications, each device that uses a protocol that is secured by key-based authentication must be able to store and use more than one key for a feature at the same time. Based on the send and accept lifetimes of a key, keychain management provides a secure mechanism to handle key rollover. The device uses the lifetimes of keys to determine which keys in a keychain are active.

Each key in a keychain has two lifetimes, as follows:

Accept lifetime - The time interval within which the device accepts the key during key exchange with another device.

Send lifetime - The time interval within which the device sends the key during key exchange with another device.

You define the send and accept lifetimes of a key using the following parameters:

Start-time - The absolute time that the lifetime begins.

End-time - The end time can be defined in one of the following ways:

- The absolute time that the lifetime ends
- The number of seconds after the start time that the lifetime ends
- Infinite lifetime (no end-time)

During a key send lifetime, the device sends routing update packets with the key. The device does not accept communication from other devices when the key sent is not within the accept lifetime of the key on the device.

We recommend that you configure key lifetimes that overlap within every keychain. This practice avoids failure of neighbor authentication due to the absence of active keys.

Exchange of Routing Information

Routing Information Protocol (RIP v1) is a broadcast protocol (255.255.255.255), and for RIPv2 (224.0.0.9)

RIP Operation

RIP defines two types of messages.

Request Message

Response Message

When a RIP router comes online, it sends a broadcast Request Message on all of its RIP enabled interfaces. All the neighbouring routers which receive the Request message respond back with the Response Message containing their Routing table. The Response Message is also gratuitously sent when the Update timer expires. On receiving the Routing table, the router processes each entry of the routing table as per the following rules

If there are no route entries matching the one received then the route entry is added to the routing table automatically, along with the information about the router from which it received the routing table.

If there are matching entries but the hop count metric is lower than the one already in its routing table, then the routing table is updated with the new route.

If there are matching entries but the hop count metric is higher than the one already in its routing table, then the routing entry is updated with hop count of 16 (infinite hop). The packets are still forwarded to the old route. A Holddown timer is started and all the updates for that from other routers are ignored. If after the Holddown timer expires and still the router is advertising with the same higher hop count then the value is updated into its routing table. Only after the timer expires, the updates from other routers are accepted for that route.

Timers

The routing information protocol uses the following timers as part of its operation:

- Update Timer
- Invalid Timer
- Flush Timer
- Holddown Timer

Update Timer

The update timer controls the interval between two gratuitous Response Messages. By default the value is 30 seconds. The response message is broadcast to all its RIP enabled interface.

Invalid Timer

The invalid timer specifies how long a routing entry can be in the routing table without being updated. This is also called as expiration Timer. By default, the value is 180 seconds. After the timer expires the hop count of the routing entry will be set to 16, marking the destination as unreachable.

Flush Timer

The flush timer controls the time between the route is invalidated or marked as unreachable and removal of entry from the routing table. By default the value is 240 seconds. This is 60 seconds longer than Invalid timer. So for 60 seconds the router will be advertising about this unreachable route to all its neighbours. This timer must be set to a higher value than the invalid timer.

Hold-down Timer

The hold-down timer is started per route entry, when the hop count is changing from lower value to higher value. This allows the route to get stabilized. During this time no update can be done to that routing entry. This is not part of the RFC 1058. This is Cisco's implementation. The default value of this timer is 180 seconds.

Summary

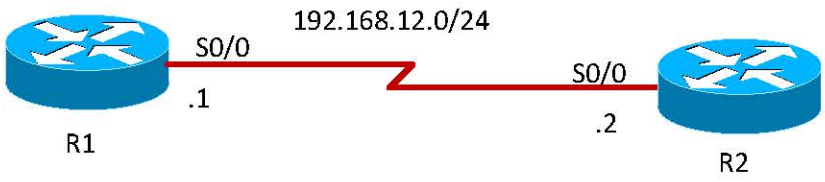
RIP Version 1 Features and Concepts

- Distance Vector
- Operating from UDP port 520
- Metric used by Rip is hop count
- Maximum hop count is 15, 16th hop is unreachable
- Periodic Update = 30sec
- Random Jitter (RIP_JITTER) = 15% (4.5 sec) so the Periodic Update can vary from 25.5 sec to 30 seconds.
- Invalidation timer = 180 sec (6 times the update timer)
- Holddown timer = 180 sec (6 times the update timer)
- Split horizon with Poisoned reverse with triggered update is used for stability of the operation.

RIP Version 2 Features and Concepts

- Route updates include subnet masks
- Supports authentication of Routing Updates
- Multicast address used for Routing Updates
- Automatic summarization

Lab 1 - Basic Static Routes



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
S 0/0	192.168.12.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
S 0/0	192.168.12.2	255.255.255.0

Task 1

Configure the hostname R1 on Router 1 and R2 on Router 2.

Task 2

Configure the Serial Link between R1 & R2 using HDLC as the encapsulation. Set the clock rate at 128 kbps.

Task 3

Configure the loopback interface on R1 & R2.

Task 4

Configure the static routes on R1 and R2 as follows:

- 2.0.0.0/8 on R1
- 1.0.0.0/8 on R2

On R1

```
enable
configure terminal
!
hostname R1
!
no ip domain lookup
!
line con 0
 no exec-timeout
 logging synchronous
!
interface Loopback0
 ip address 1.1.1.1 255.0.0.0
!
interface Serial0/0
 ip address 192.168.12.1 255.255.255.0
 no shutdown
!
ip route 2.0.0.0 255.0.0.0 192.168.12.2
!
```

```
On R2
enable
configure terminal
!
hostname R2
!
no ip domain lookup
!
line con 0
 no exec-timeout
 logging synchronous
!
interface Loopback0
 ip address 2.2.2.2 255.0.0.0
!
interface Serial0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
ip route 1.0.0.0 255.0.0.0 192.168.12.1
!
```

Verfication

On Both Routers check interface status

R1#show ip interface brief					
Interface	IP-Address	OK?	Method	Status	Protocol
Serial0/0	192.168.12.1	YES	manual	up	up
Loopback0	1.1.1.1	YES	manual	up	up

R2#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
Serial0/0	192.168.12.2	YES	manual	up	up
Loopback0	2.2.2.2	YES	manual	up	up

On Both Routers check routing table

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
     1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S       2.0.0.0/24 [1/0] via 192.168.1.2
C       1.0.0.0/8 is directly connected, Loopback0
C       192.168.12.0/24 is directly connected, Serial0/0
```

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
S       1.0.0.0/8 [1/0] via 192.168.1.1
C       2.0.0.0/8 is directly connected, Loopback0
C       192.168.12.0/24 is directly connected, Serial0/0
```

What networks do you see listed?

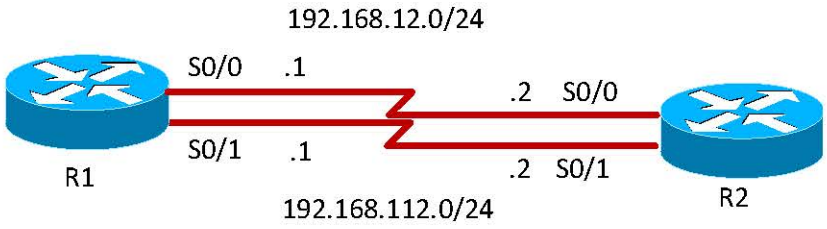
Ping your partner's Loopback Interface address.

```
R1#ping 192.168.12.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/23/52 ms
R1#ping 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/21/72 ms
```

```
R2#ping 192.168.12.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/23/52 ms
R2#ping 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/21/72 ms
```

Are you successful?

Lab 2 –
Load balancing using Static Routes



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
S 0/0	192.168.12.1	255.255.255.0
S0/1	192.168.112.1	

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
S 0/0	192.168.12.2	255.255.255.0
S0/1	192.168.112.2	

Task 1

Configure the hostname R1 on Router 1 and R2 on Router 2.

Task 2

Configure the Serial Link between R1 & R2 using HDLC as the encapsulation. Set the clock rate at 128 kbps.

Task 3

Configure the loopback interface on R1 & R2.

Task 4

Configure the static routes on R1 as follows

• 2.0.0.0/8 on R1 via 192.168.12.2

Configure the static routes on R2 as follows

• 1.0.0.0/8 on R2 via 192.168.12.1

Task 5

Configure another static routes on R1:

• 2.0.0.0/8 on R1 via 192.168.112.2

Configure the static routes on R2 as follows:

• 1.0.0.0/8 on R2 via 192.168.112.1

On R1

```
enable
configure terminal
!
hostname R1
!
no ip domain lookup
!
line con 0
 no exec-timeout
 logging synchronous
!
interface Loopback0
 ip address 1.1.1.1 255.0.0.0
!
interface Serial0/0
 ip address 192.168.12.1 255.255.255.0
 no shutdown
!
interface Serial0/1
 ip address 192.168.112.1 255.255.255.0
 no shutdown
!
ip route 2.0.0.0 255.0.0.0 192.168.12.2
ip route 2.0.0.0 255.0.0.0 192.168.112.2 10
!
```

On R2

```
enable
configure terminal
!
hostname R2
!
no ip domain lookup
!
line con 0
 no exec-timeout
 logging synchronous
!
interface Loopback0
 ip address 2.2.2.2 255.0.0.0
!
interface Serial0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
interface Serial0/1
 ip address 192.168.112.2 255.255.255.0
 no shutdown
!
ip route 1.0.0.0 255.0.0.0 192.168.12.1
ip route 1.0.0.0 255.0.0.0 192.168.112.1
!
```

Task 6:

Verfication

On R1

Type

```
R1#show ip route
S       2.0.0.0/8 [1/0] via 192.168.112.2
                  [1/0] via 192.168.12.2
```

On R2

Type

```
R2#show ip route static
S       1.0.0.0/8 [1/0] via 192.168.112.1
                  [1/0] via 192.168.12.1
```

```
R1#show ip route 2.0.0.0
Routing entry for 2.0.0.0/8
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 192.168.112.2
      Route metric is 0, traffic share count is 1
    192.168.12.2
      Route metric is 0, traffic share count is 1
```

Do you see an Asterisks (*) against one of the routes?

Note: The Asterisks represents the next path the router will take to get the packet to the destination.

On R2

```
R2#show ip route 1.0.0.0
Routing entry for 1.0.0.0/8
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 192.168.112.1
      Route metric is 0, traffic share count is 1
    192.168.12.1
      Route metric is 0, traffic share count is 1
```

Do you see an Asterisks (*) against one of the routes?

Note: The Asterisks represents the next path the router will take to get the packet to the destination.

Test 1

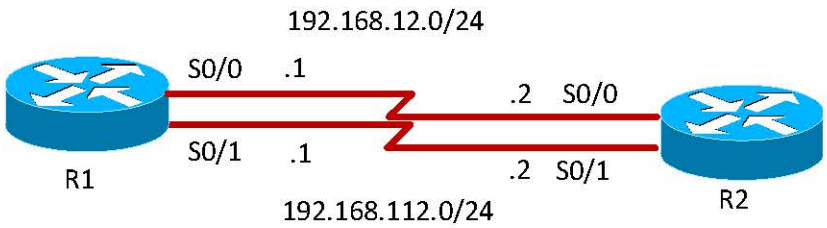
On router R1 bring down S0/0 interface to simulate link failure. Check the route again for 2.0.0.0 on R1. You should see another path on R1 to reach 2.0.0.0.

Test 2

On R1 try to traceroute 2.2.2.2. Are you successful?

```
R1#traceroute 2.2.2.2
Tracing the route to 2.2.2.2
 1 192.168.112.2 36 msec
   192.168.12.2 4 msec
   192.168.112.2 8 msec
```

Lab 3 – Static Routes Floating



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
S 0/0	192.168.12.1	255.255.255.0
S0/1	192.168.112.1	

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
S 0/0	192.168.12.2	255.255.255.0
S0/1	192.168.112.2	

Task 1

Configure the hostname R1 on Router 1 and R2 on Router 2.

Task 2

Configure the Serial Link between R1 & R2 using HDLC as the encapsulation. Set the clock rate at 128 kbps.

Task 3

Configure the loopback interface on R1 & R2.

Task 4

Configure the static routes on R1 as follows

• 2.0.0.0/8 on R1 via 192.168.12.2

Configure the static routes on R2 as follows

• 1.0.0.0/8 on R2 via 192.168.12.1

Task 5

Configure another static routes on R1 with **AD** value of **10** as follows:

• 2.0.0.0/8 on R1 via 192.168.112.2

Configure the static routes on R2 as follows with **AD** value of **10** as follows:

• 1.0.0.0/8 on R2 via 192.168.112.1

On R1

```
enable
configure terminal
!
hostname R1
!
no ip domain lookup
!
line con 0
 no exec-timeout
 logging synchronous
!
interface Loopback0
 ip address 1.1.1.1 255.0.0.0
!
interface Serial0/0
 ip address 192.168.12.1 255.255.255.0
 no shutdown
!
interface Serial0/1
 ip address 192.168.112.1 255.255.255.0
 no shutdown
!
ip route 2.0.0.0 255.0.0.0 192.168.12.2
ip route 2.0.0.0 255.0.0.0 192.168.112.2 10
!
```

On R2

```
enable
configure terminal
!
hostname R2
!
no ip domain lookup
!
line con 0
 no exec-timeout
 logging synchronous
!
interface Loopback0
 ip address 2.2.2.2 255.0.0.0
!
interface Serial0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
interface Serial0/1
 ip address 192.168.112.2 255.255.255.0
 no shutdown
!
ip route 1.0.0.0 255.0.0.0 192.168.12.1
ip route 1.0.0.0 255.0.0.0 192.168.112.1 10
!
```

Task 6:

Verfication

On R1

Type

```
R1#show ip route 2.0.0.0
Routing entry for 2.0.0.0/8
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 192.168.12.2
      Route metric is 0, traffic share count is 1
```

Do you see an Asterisks (*) against one of the routes?

Note: The Asterisks represents the next path the router will take to get the packet to the destination.

On R2

Type

```
R2#show ip route 1.0.0.0
Routing entry for 1.0.0.0/8
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 192.168.12.1
      Route metric is 0, traffic share count is 1
```

Do you see an Asterisks (*) against one of the routes?

Note: The Asterisks represents the next path the router will take to get the packet to the destination.

Test 1

On router R1 bring down S0/0 interface to simulate link failure. Check the route again for 2.0.0.0 on R1. You should see another path on R1 to reach 2.0.0.0.

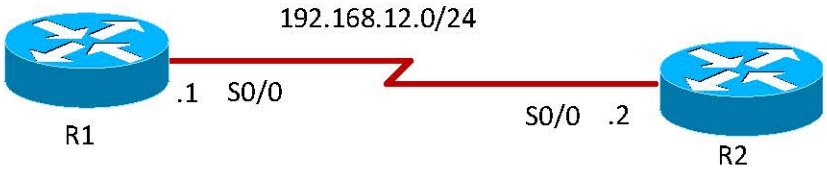
```
R1(config)#interface serial 0/0
R1(config-if)#shutdown
*Mar  1 00:19:31.087: %LINK-5-CHANGED: Interface Serial0/0, changed state to administratively down
*Mar  1 00:19:32.087: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to
down
R1#show ip route 2.0.0.0
Routing entry for 2.0.0.0/8
  Known via "static", distance 10, metric 0
  Routing Descriptor Blocks:
  * 192.168.112.2
      Route metric is 0, traffic share count is 1
```

Test 2

On R1 try to ping 2.2.2.2. Are you successful?

```
R1#ping 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/17/44 ms
```

Lab 4 – Default Routing



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
S 0/0	192.168.12.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
S 0/0	192.168.12.2	255.255.255.0

Task 1

Configure the hostname R1 on Router 1 and R2 on Router 2.

Task 2

Configure the Serial Link between R1 & R2 using HDLC as the encapsulation. Set the clock rate at 128 kbps.

Task 3

Configure the loopback interface on R1 & R2.

Task 4

Configure the static routes on R1 and R2 as follows:

- 2.0.0.0/8 on R1
- 1.0.0.0/8 on R2

Task 5

Configure additional loopbacks on R2 as follows:

- Loopback 1 10.2.1.0/24
- Loopback 2 10.2.2.0/24
- Loopback 3 10.2.3.0/24
- Loopback 4 10.2.4.0/24

(Note: These loopbacks are subnets of 10.0.0.0/8. Recall what you have learned in subnetting classes.)

Task 6

On R1 Create a Default route towards R2

On R1

```
enable
configure terminal
!
interface Loopback0
 ip address 1.1.1.1 255.0.0.0
!
interface Serial0/0
 ip address 192.168.12.1 255.255.255.0
 no shutdown
!
ip route 0.0.0.0 0.0.0.0 192.168.12.1
!
```

On R2

```
enable
configure terminal
!
interface loopback0
 ip address 2.2.2.2 255.0.0.0
!
interface loopback 1
 ip address 10.2.1.1 255.255.255.0
!
interface loopback 2
 ip address 10.2.2.1 255.255.255.0
!
interface loopback 3
 ip address 10.2.3.1 255.255.255.0
!
interface loopback 4
 ip address 10.2.4.1 255.255.255.0
!
interface Serial0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
ip route 1.0.0.0 255.0.0.0 192.168.12.1
!
```

Task 7

Verfication

```
R1#show ip route
Codes: C - connected, S - static, ...
Gateway of last resort is 192.168.12.2 to network 0.0.0.0
C    192.168.12.0/24 is directly connected, Serial0/0
C    1.0.0.0/8 is directly connected, Loopback0
C    192.168.112.0/24 is directly connected, Serial0/1
S*   0.0.0.0/0 [1/0] via 192.168.12.2
R1#show ip route 0.0.0.0
Routing entry for 0.0.0.0/0, supernet
  Known via "static", distance 1, metric 0, candidate default path
  Routing Descriptor Blocks:
  * 192.168.12.2
      Route metric is 0, traffic share count is 1
```

Do you see the default routes on R1?

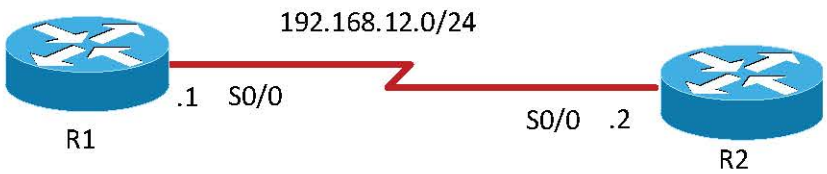
Do you see a route with an Asterisk?

What is the gateway of last resort?

MODULE 6

RIP LABS

Lab 1 – Basic RIPv1 Configuration



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
S 0/0	192.168.12.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
S 0/0	192.168.12.2	255.255.255.0

Task 1

Configure the hostname R1 on Router 1 and R2 on Router 2.

Task 2

Configure the Serial Link between R1 & R2 using HDLC as the encapsulation. Set the clock rate at 128 kbps.

Task 3

Configure the loopback interface on R1 & R2.

Task 4

Configure the RIPv1 on R1 and R2 as follows:

- 1.0.0.0/8 on R1
- 192.168.12.0/24 on R1 and R2
- 1.0.0.0/8 on R2

On R1

```
enable
configure terminal
!
interface Loopback0
 ip address 1.1.1.1 255.0.0.0
!
interface Serial0/0
 ip address 192.168.12.1 255.255.255.0
 no shutdown
!
Router rip
 version 1
 network 192.168.12.0
 network 1.0.0.0
!
```

On R2

```
enable
configure terminal
!
interface loopback0
 ip address 2.2.2.2 255.0.0.0
!
interface Serial0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
router rip
 version 1
 network 192.168.12.0
 network 2.0.0.0
!
```

Task 4

Verfication

On R1 and R2

```
R1#show ip route rip
R    2.0.0.0/8 [120/1] via 192.168.12.2, 00:00:13, Serial0/0
R2#show ip route rip
R    2.0.0.0/8 [120/1] via 192.168.12.1, 00:00:13, Serial0/0
```

What networks do you see listed?

Ping your partner's Loopback Interface address.

Are you successful?

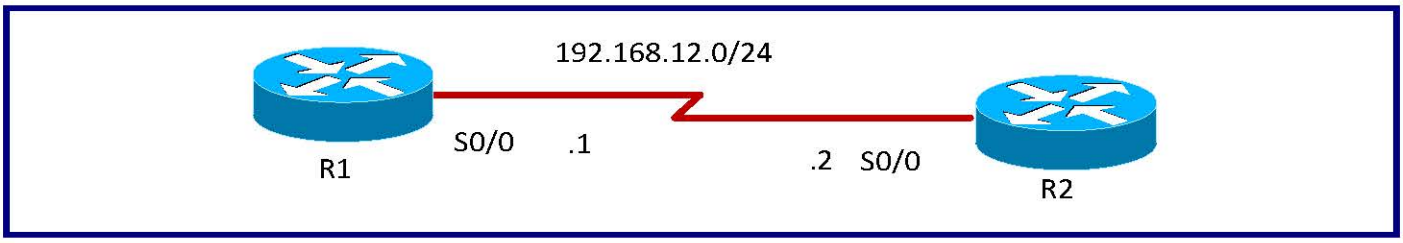
On R1

```
R1#show ip route 2.0.0.0
Routing entry for 2.0.0.0/8
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 192.168.12.2 on Serial0/0, 00:00:22 ago
  Routing Descriptor Blocks:
  * 192.168.12.2, from 192.168.12.2, 00:00:22 ago, via Serial0/0
      Route metric is 1, traffic share count is 1
```

To verify RIP protocol further

```
R1#show ip protocols
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 9 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive version 1
    Interface          Send  Recv  Triggered RIP  Key-chain
    Serial0/0          1     1
    Loopback0          1     1
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    1.0.0.0
    192.168.12.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.12.2         120      00:00:28
  Distance: (default is 120)
```

Lab 2 – RIPv1 Operations



On R1

```
R1#debug ip rip
RIP protocol debugging is on
R1#
*Mar  1 03:06:41.087: RIP: received v1 update from 192.168.12.2 on Serial0/0
*Mar  1 03:06:41.087:      2.0.0.0 in 1 hops
*Mar  1 03:06:47.603: RIP: sending v1 update to 255.255.255.255 via Loopback0 (1.1.1.1)
*Mar  1 03:06:47.603: RIP: build update entries
*Mar  1 03:06:47.603:   network 2.0.0.0 metric 2
*Mar  1 03:06:47.607:   network 192.168.12.0 metric 1
*Mar  1 03:06:53.879: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (192.168.12.1)
*Mar  1 03:06:53.879: RIP: build update entries
*Mar  1 03:06:53.883:   network 1.0.0.0 metric 1
```

Interesting Facts

- Does not include the directly connected network (192.168.12.0) in its update when the update is send via serial 0/0
- Does not include 2.0.0.0 network although it does exist in its routing table (Split Horizon)
- The destination address is a Broadcast (255.255.255.255)
- It does not send periodic updates at constant intervals (Time Jitters)
- Updates are also send via loopback 0.
- Can we stop sendig updates via loopback 0? (Passive-Interface)

On R1

```
R1(config)#interface loopback 0
R1(config-if)#shutdown
*Mar  1 03:09:22.507: %LINK-5-CHANGED: Interface Loopback0, changed state to administratively down
*Mar  1 03:09:22.510: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to down
*Mar  1 03:09:22.515: RIP: sending v1 flash update to 255.255.255.255 via Serial0/0 (192.168.12.1)
*Mar  1 03:09:22.515: RIP: build flash update entries
*Mar  1 03:09:22.515: network 1.0.0.0 metric 16
*Mar  1 03:09:24.779: RIP: received v1 update from 192.168.12.2 on Serial0/0
*Mar  1 03:09:24.779: 1.0.0.0 in 16 hops (inaccessible)
*Mar  1 03:09:27.591: RIP: received v1 update from 192.168.12.2 on Serial0/0
*Mar  1 03:09:27.595: 1.0.0.0 in 16 hops (inaccessible)
*Mar  1 03:09:27.595: 2.0.0.0 in 1 hop
```

Interesting Facts

When a route goes down, the router does not wait for periodic update. It sends a **flash update** with a Poisoned route with a metric of 16.

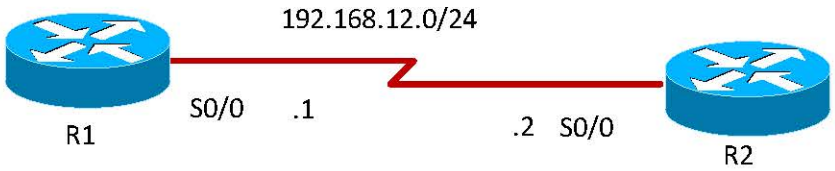
Note: R2 also sends an immediate Triggered Update back, indicating that you can't reach 1.0.0.0 cannot be reached through it.

To stop debugging

Type

```
R1#undebug all
```

Lab 3 – RIP Passive-Interface Configuration



On R1

Task 1

Configure passive-interface on R1 and R2

```
R1(config)#router rip
R1(config-router)#passive-interface Loopback 0
```

On R2

```
R2(config)#router rip
R2(config-router)#passive-interface Loopback 0
```

Task 2

Verfication

```
R1(config)#debug ip rip
*Mar  1 04:17:12.974: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (192.168.12.1)
*Mar  1 04:17:12.974: RIP: build update entries
*Mar  1 04:17:12.974:   network 1.0.0.0 metric 1
```

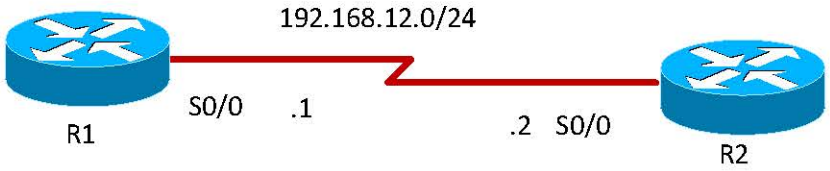
Interesting Facts

- The router stops advertising via the Loopback 0.
- The command is useful for cutting down unnecessary broadcast over an interface that only has hosts on it and no router.

Important note:

Passive-interface in RIP Stops send broadcast and multicast updates, but continues to receive it.

Lab 4 – Basic RIPv2 Configuration



Task 1

Configure RIP version 2 on both routers

```
R1
!
router rip
 version 2
 network 192.168.12.0
 network 1.0.0.0
!
```

```
R2
!
router rip
 version 2
 network 192.168.12.0
 network 2.0.0.0
!
```

Task 2

Verfication

On R1 and R2

```
R1#show ip route rip
R    2.0.0.0/8 [120/1] via 192.168.12.2, 00:00:13, Serial0/0
R2#show ip route rip
R    2.0.0.0/8 [120/1] via 192.168.12.1, 00:00:13, Serial0/0
```

What networks do you see listed?

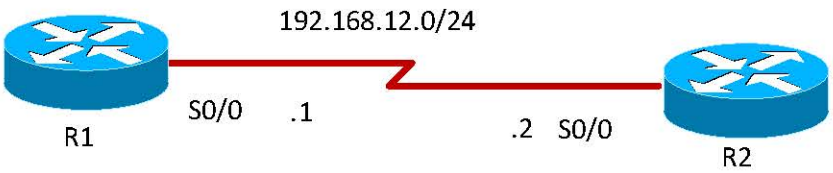
Ping your partner's Loopback Interface address.

Are you successful?

To verfy further

```
R1#show ip protocol
```

Lab 5 – Basic RIPv2 Operation



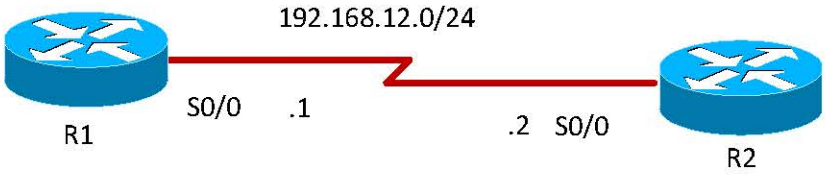
On R1

```
R1#debug ip rip
*Mar  1 00:05:49.743: RIP: sending v2 update to 224.0.0.9 via Serial0/0 (192.168.12.1)
*Mar  1 00:05:49.747: RIP: build update entries
*Mar  1 00:05:49.747:   1.0.0.0/8 via 0.0.0.0, metric 1, tag 0
R1#
*Mar  1 00:06:07.771: RIP: sending v2 update to 224.0.0.9 via Loopback0 (1.1.1.1)
*Mar  1 00:06:07.775: RIP: build update entries
*Mar  1 00:06:07.775:   2.0.0.0/8 via 0.0.0.0, metric 2, tag 0
*Mar  1 00:06:07.779:   192.168.12.0/24 via 0.0.0.0, metric 1, tag 0
*Mar  1 00:06:07.791: RIP: ignored v2 packet from 1.1.1.1 (sourced from one of our addresses)
R1#
*Mar  1 00:06:12.207: RIP: received v2 update from 192.168.12.2 on Serial0/0
*Mar  1 00:06:12.207:      2.0.0.0/8 via 0.0.0.0 in 1 hops
```

Interesting Facts

- Update is a V2 update
- Includes the subnet mask
- The destination address.

Lab 6 – RIPv2 Auto-Summary



Task 1

Configure following loopbacks on R1 and R2 respectively.

Configure additional loopbacks on R1 as follows:

- Loopback 1 10.1.1.0/24
- Loopback 2 10.1.2.0/24
- Loopback 3 10.1.3.0/24
- Loopback 4 10.1.4.0/24

Configure additional loopbacks on R2 as follows:

- Loopback 1 10.2.1.0/24
- Loopback 2 10.2.2.0/24
- Loopback 3 10.2.3.0/24
- Loopback 4 10.2.4.0/24

On R1

```
enable

configure terminal

!

interface loopback 1

 ip address 10.1.1.1 255.255.255.0

!

interface loopback 2

 ip address 10.1.2.1 255.255.255.0

!

interface loopback 3

 ip address 10.1.3.1 255.255.255.0

!

interface loopback 4

 ip address 10.1.4.1 255.255.255.0

!
```

On R2

```
enable
configure terminal
!
interface loopback 1
 ip address 10.2.1.1 255.255.255.0
!
interface loopback 2
 ip address 10.2.2.1 255.255.255.0
!
interface loopback 3
 ip address 10.2.3.1 255.255.255.0
!
interface loopback 4
 ip address 10.2.4.1 255.255.255.0
!
```

Task 2

Advertised the loopback networks on R1 and R2

On R1 and R2

```
router rip
network 10.0.0.0
!
```

Task 3

Check routing table.

Can you see subnets of 10.0.0.0 network in the routing table of R1 and R2?

If no, why? (auto-summary)

```
R2#show ip route rip
R    1.0.0.0/8 [120/1] via 192.168.12.1, 00:00:13, Serial0/0
     10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
R       10.0.0.0/8 [120/1] via 192.168.12.1, 00:00:13, Serial0/0
```

Task 4

Turn off auto-summary on R1 and R2

On R1 and R2

```
router rip
 no auto-summary
!
```

Task 5

Check routing table.

Can you see subnets of 10.0.0.0 network in the routing table of R1 and R2?

```
R1#show ip route rip
R    2.0.0.0/8 [120/1] via 192.168.12.2, 00:00:28, Serial0/0
     10.0.0.0/8 is variably subnetted, 9 subnets, 2 masks
R       10.2.1.0/24 [120/1] via 192.168.12.2, 00:00:28, Serial0/0
R       10.2.2.0/24 [120/1] via 192.168.12.2, 00:00:28, Serial0/0
R       10.2.3.0/24 [120/1] via 192.168.12.2, 00:00:28, Serial0/0
R       10.2.4.0/24 [120/1] via 192.168.12.2, 00:00:28, Serial0/0
```

```
R2#show ip route rip
R    1.0.0.0/8 [120/1] via 192.168.12.1, 00:00:10, Serial0/0
     10.0.0.0/24 is subnetted, 8 subnets
R       10.1.3.0 [120/1] via 192.168.12.1, 00:00:10, Serial0/0
R       10.1.2.0 [120/1] via 192.168.12.1, 00:00:10, Serial0/0
R       10.1.1.0 [120/1] via 192.168.12.1, 00:00:10, Serial0/0
R       10.1.4.0 [120/1] via 192.168.12.1, 00:00:10, Serial0/0
```

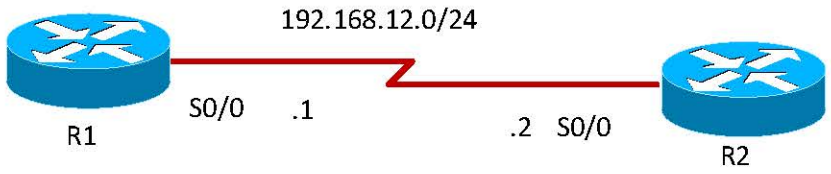
What networks do you see listed?

Ping your partner's loopback interface address sourced from your loopback.

Are you successful?

```
R1#ping 10.2.1.1 source 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/18/48 ms
```

Lab 7 – Configuring RIPv2 Text Authentication



Task 1

Configure Clear Text authentication between R1, R2 Use ccie as the key-string with a key-id of 1.

R1

```
key chain AUTH
 key 1
  key-string ccie
!
interface Serial 0/0
 ip rip authentication key-chain AUTH
```

R2

```
key chain AUTH
 key 1
  key-string ccie
!
interface Serial 0/0
 ip rip authentication key-chain AUTH
```

Task 2

Verify the authentication by debugging on R1 and R2.

```
R1#debug ip rip
*Mar  1 00:55:25.959: RIP: received packet with text authentication ccie
*Mar  1 00:55:25.959: RIP: received v2 update from 192.168.12.2 on Serial0/0
*Mar  1 00:55:25.963:      2.0.0.0/8 via 0.0.0.0 in 1 hops
*Mar  1 00:55:25.963:      10.2.1.0/24 via 0.0.0.0 in 1 hops
*Mar  1 00:55:25.967:      10.2.2.0/24 via 0.0.0.0 in 1 hops
*Mar  1 00:55:25.971:      10.2.3.0/24 via 0.0.0.0 in 1 hops
*Mar  1 00:55:25.975:      10.2.4.0/24 via 0.0.0.0 in 1 hops
```

Can you see password for authentication?

MODULE 7

EIGRP

Chapter 1 – Enhanced IGRP (EIGRP)

Enhanced Interior Gateway Routing Protocol

Enhanced Interior Gateway Routing Protocol (EIGRP) is an interior gateway protocol suited for many different topologies and media. In a well designed network, EIGRP scales well and provides extremely quick convergence times with minimal network traffic.

EIGRP Theory of Operation

Some of the many advantages of EIGRP are:

- Very low usage of network resources during normal operation; only hello packets are transmitted on a stable network.
- When a change occurs, only routing table changes are propagated, not the entire routing table; this reduces the load the routing protocol itself places on the network.
- Rapid convergence times for changes in the network topology (in some situations convergence can be almost instantaneous).
- EIGRP is an enhanced distance vector protocol, relying on the Diffused Update Algorithm (DUAL) to calculate the shortest path to a destination within a network.

Basic Theory

A typical distance vector protocol saves the following information when computing the best path to a destination: the distance (total metric or distance, such as hop count) and the vector (the next hop). For instance, all the routers in the network in Figure 1 are running Routing Information Protocol (RIP). Router Two chooses the path to Network A by examining the hop count through each available path.

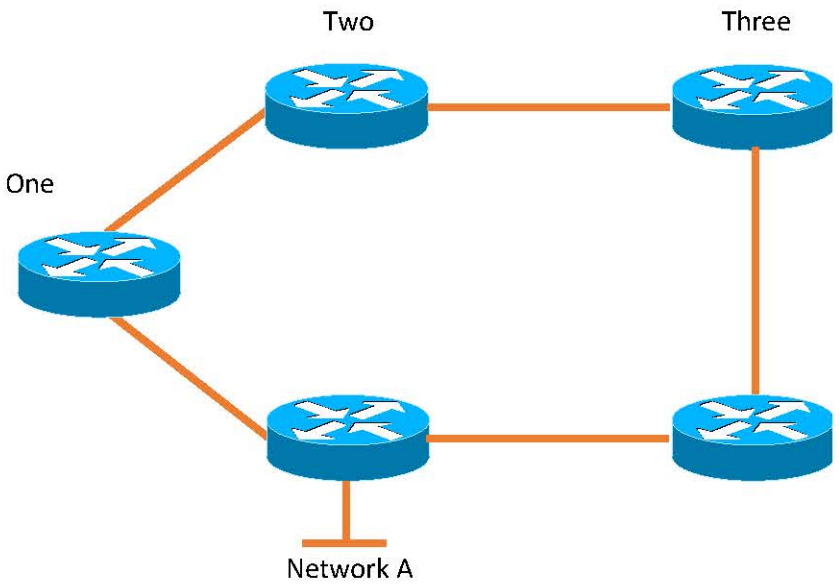


Figure 1

Since the path through Router Three is three hops, and the path through Router One is two hops, Router Two chooses the path through One and discards the information it learned through Three. If the path between Router One and Network A goes down, Router Two loses all connectivity with this destination until it times out the route of its routing table (three update periods, or 90 seconds), and Router Three re advertises the route (which occurs every 30 seconds in RIP). Not including any hold down time, it will take between 90 and 120 seconds for Router Two to switch the path from Router One to Router Three. EIGRP, instead of counting on full periodic updates to re converge, builds a topology table from each of its neighbor's advertisements (rather than discarding the data), and converges by either looking for a likely loop free route in the topology table, or, if it knows of no other route, by querying its neighbors. Router Two saves the information it received from both Routers One and Three. It chooses the path through one as its best path (the successor) and the path through three as a loop free path (a feasible successor). When the path through Router One becomes unavailable, Router Two examines its topology table and, finding a feasible successor, begins using the path through Three immediately.

From this brief explanation, it is apparent that EIGRP must provide:

- A system where it sends only the updates needed at a given time; this is accomplished through neighbor discovery and maintenance.
- A way of determining which paths a router has learned are loop free.
- A process to clear bad routes from the topology tables of all routers on the network.
- A process for querying neighbors to find paths to lost destinations.

Neighbor Discovery and Maintenance

To distribute routing information throughout a network, EIGRP uses non periodic incremental routing updates. That is, EIGRP only sends routing updates about paths that have changed when those paths change. The basic problem with sending only routing updates is that you may not know when a path through a neighboring router is no longer available. You cannot time out routes, expecting to receive a new routing table from your neighbors. EIGRP relies on neighbor relationships to reliably propagate routing table changes throughout the network; two routers become neighbors when they see each other's hello packets on a common network. EIGRP sends hello packets every 5 seconds. The rate at which EIGRP sends hello packets is called the hello interval, and you can adjust it per interface with the **ip hello interval eigrp** command. The hold time is the amount of time that a router will consider a neighbor alive without receiving a hello packet. The hold time is typically three times the hello interval, by default, 15 seconds. You can adjust the hold time with the **ip hold-time eigrp** command.

Note: If you change the hello interval, the hold time is not automatically adjusted to account for this change you must manually adjust the hold time to reflect the configured hello interval

It is possible for two routers to become EIGRP neighbors even though the hello and hold timers do not match. The hold time is included in the hello packets so each neighbor should stay alive even though the

hello interval and hold timers do not match.

Building the Topology Table

Now that these routers are talking to each other, what are they talking about? Their topology tables, of course! EIGRP, unlike RIP and IGRP, does not rely on the routing (or forwarding) table in the router to hold all of the information it needs to operate. Instead, it builds a second table, the topology table, from which it installs routes in the routing table. To see the basic format of the topology table on a router running EIGRP, issue the show ip eigrp topology

EIGRP Metrics

EIGRP uses the minimum bandwidth on the path to a destination network and the total delay to compute routing metrics. Although you can configure other metrics, we do not recommend it, as it can cause routing loops in your network. The bandwidth and delay metrics are determined from values configured on the interfaces of routers in the path to the destination network.

For instance, in Figure 2 below, Router One is computing the best path to Network A.

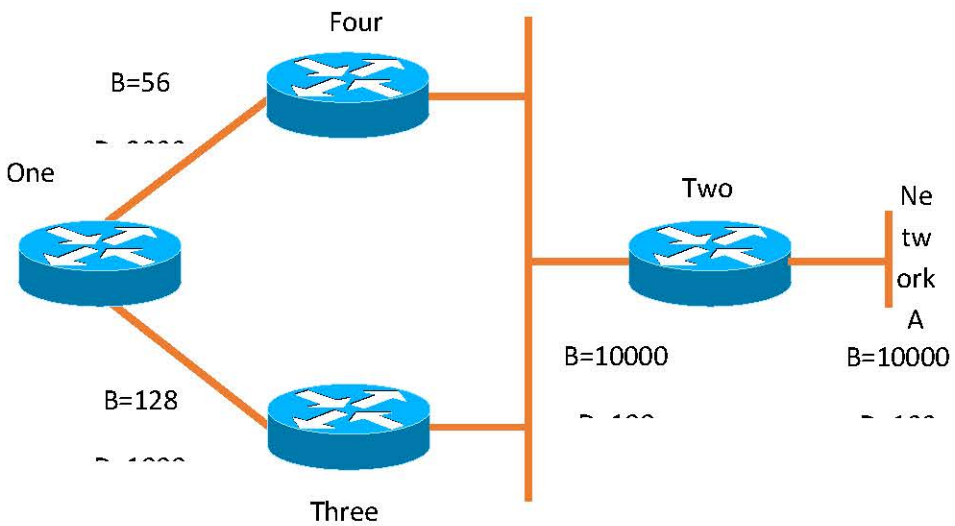


Figure: 2

It starts with the two advertisements for this network: One through Router Four, with a minimum bandwidth of 56 and a total delay of 2200; and the other through Router Three, with a minimum bandwidth of 128 and a delay of 1200. Router One chooses the path with the lowest metric.

Let us compute the metrics. EIGRP calculates the total metric by scaling the bandwidth and delay metrics.

EIGRP uses the following formula to scale the bandwidth:

Bandwidth = {[10^7/bandwidth (Min)] + [delay (sum)/10]} * 256

- Where bandwidth is the least bandwidth of all outgoing interfaces on the route to the destination network represented in kilobits.
- Where delay is the sum of the delays configured on the interfaces, on the route to the destination network, in tens of microseconds so you must divide by 10 before you use it in this formula.

Chapter at a Glance

- Was Cisco proprietary routing protocol.
- Became open standard in February 2013.
- First released in 1994 with IOS version 9.21.
- Advance Distance Vector/Hybrid routing protocol that has the behavior

of distance vector with several Link State features, such as dynamic neighbor discovery.

Rapid Convergence: EIGRP uses DUAL to achieve rapid convergence. It stores a backup route if one is available, so it can quickly re-converge incase a route goes down. If no backup route exists, EIGRP will send a query to its neighbor/s to discover an alternate path. These queries are propagated until an alternate route is found.

Reduced Bandwidth Usage/Incremental Updates: In EIGRP updates are still sent to directly connected neighbors, much like distance vector protocols, but these updates are:

Non-Periodic: The updates are not sent at regular intervals, rather when a metric or a topology change occurs.

Partial: Updates will include the routes that are changed and not every route in the routing table.

Bounded: Updates are sent to affected routers only.

Another issue regarding bandwidth usage is the fact that EIGRP by default will only consume 50% of the bandwidth of the link during convergence. This parameter can be adjusted to a higher or lower value enter the following command in interface sub-config mode:

ip bandwidth-percent eigrp <AS number> <number that represents the %age>

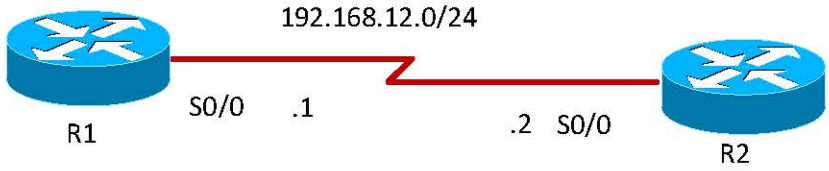
Classless Routing Protocol: This means that advertised routes will include their subnet mask, this feature will eliminate the issue pertaining to discontiguous networks. VLSM and Manual Summarization is also supported on any router within the enterprise.

Security: With IOS version 11.3 or better, EIGRP can authenticate using only MD5, the reason EIGRP does not support clear text is because, EIGRP can only be used within CISCO routers, and all Cisco routers support MD5 authentication. But the routes are not encrypted, so a sniffer can easily see the password/s.

Multiple Network Layer Protocol Support: EIGRP can support IP, IPX, and AppleTalk, whereas the other routing protocols support only one routed protocol. EIGRP will also perform auto-redistribution with NLSP, IPXRIP, RTMP. EIGRP supports incremental SAP and RIP updates, 224 HOPS, and it uses bandwidth + delay which is far better than just Ticks and Hops used by IPXRIP. For RTMP it supports event driven updates, but it must run in a clientless networks (WAN), and also a better metric calculation.

Use of Multicast Instead Of Broadcast: EIGRP uses multicast address of **224.0.0.10** instead of broadcast.

Lab 1 – Basic EIGRP Configuration



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
S 0/0	192.168.12.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
S 0/0	192.168.12.2	255.255.255.0

Task 1

Configure R1 and R2 as per the topology and put the ip addresses as given in the table.

Task 2

Configure EIGRP on R1 and R2 routers in AS 1. Disable Auto-summary.

On R1

```
router eigrp 1
 network 1.0.0.0
 network 192.168.12.0
 no auto-summary
!
```

On R2

```
router eigrp 1
 network 2.0.0.0
 network 192.168.12.0
 no auto-summary
!
```

Task 3

Verfication on R1

```
R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address         Interface      Hold Uptime   SRTT   RTO  Q   Seq
                                   (sec)         (ms)       Cnt Num
0   192.168.12.2    Se0/0          14 00:00:14    52    468  0   3
```

What and whose time do you see?

```
R1#show ip route eigrp
D    2.0.0.0/8 [90/2297856] via 192.168.12.2, 00:08:04, Serial0/0
```

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.4.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 1.0.0.0/8, 1 successors, FD is 128256
        via Connected, Loopback0
P 2.0.0.0/8, 1 successors, FD is 2297856
        via 192.168.12.2 (2297856/128256), Serial0/0
P 192.168.12.0/24, 1 successors, FD is 2169856
        via Connected, Serial0/0
```

```
R1#show ip eigrp topology 2.0.0.0
IP-EIGRP (AS 1): Topology entry for 2.0.0.0/8
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2297856
  Routing Descriptor Blocks:
  192.168.12.2 (Serial0/0), from 192.168.12.2, Send flag is 0x0
      Composite metric is (2297856/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

What do you see?

Are the metrics advertised correct?

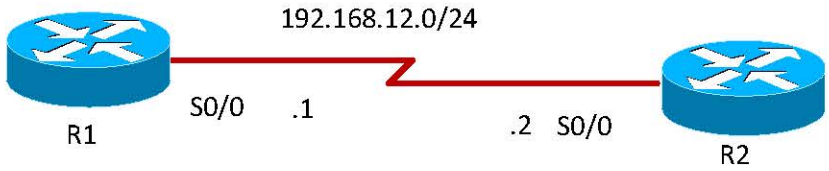
Can you compute the EIGRP metric using the formulae?

Metric = {[10000000/bandwidth (min)] + [delay (sum)/10]}*256

What is the significance of Successor?

What is FD?

Lab 2 – EIGRP Hello-Interval and Hold-Time



Task 1

Configure hello-interval and hold-time 3 and 9 sec respectively on R1 and R2.

On R1

```
interface Serial0/0
 ip hello-interval eigrp 1 3
 ip hold-time eigrp 1 9
 !
```

On R2

```
interface Serial0/0
 ip hello-interval eigrp 1 3
 ip hold-time eigrp 1 9
 !
```

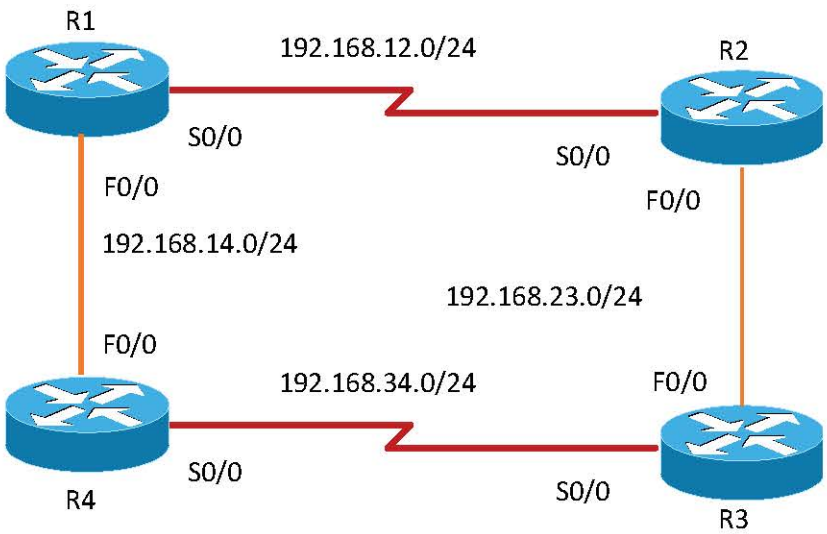
Task 2

Verfication

```
R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address         Interface     Hold Uptime   SRTT   RTO   Q  Seq
                                  (sec)         (ms)       Cnt Num
0   192.168.12.2    Se0/0         6  00:29:23   52   468  0  3
```

What timer and whose timers do you see?

Lab 3 – EIGRP Equal Cost Load Balancing



Interface IP Address Configuration

Task 1

Configure R1, R2, R3 and R4 as give in the topology

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
S 0/0	192.168.12.1	255.255.255.0
F0/0	192.168.14.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
S 0/0	192.168.12.2	255.255.255.0
F0/0	192.168.23.2	255.255.255.0

R3

Interface	IP Address	Subnet Mask
Loopback 0	3.3.3.3	255.0.0.0
S 0/0	192.168.34.3	255.255.255.0
F0/0	192.168.23.3	255.255.255.0

R4

Interface	IP Address	Subnet Mask
Loopback 0	4.4.4.4	255.0.0.0
S 0/0	192.168.34.4	255.255.255.0
F0/0	192.168.14.4	255.255.255.0

Task 2

Configure EIGRP 1 on all router. Disable auto-summary on all routers.

R1

```
!
router eigrp 1
 network 1.0.0.0
 network 192.168.12.0
 network 192.168.14.0
 no auto-summary
!
```

R2

```
!
router eigrp 1
 network 2.0.0.0
 network 192.168.12.0
 network 192.168.23.0
 no auto-summary
!
```

R3

```
!
router eigrp 1
 network 3.0.0.0
 network 192.168.12.0
 no auto-summary
!
```

R2

```
!
router eigrp 1
 network 2.0.0.0
 network 192.168.12.0
 no auto-summary
!
```

Task 3

Verfication

```
R1#show ip route
```

Do you see all the routes?

```
R1#show ip eigrp neighbor
```

Who are your neighbors?

```
R1#show ip eigrp topology
```

Do you see multiple paths?

```
R1#traceroute 4.4.4.4
```

What is the output of traceroute?

MODULE 8

OSPF

Chapter 1 – OSPF

Introduction

The Open Shortest Path First (OSPF) protocol, defined in RFC 2328, is an Interior Gateway Protocol used to distribute routing information within a single Autonomous System.

Background Information

OSPF protocol was developed due to a need in the internet community to introduce a high functionality non proprietary Internal Gateway Protocol (IGP) for the TCP/IP protocol family. The OSPF protocol is based on link state technology, which is a departure from the Bellman Ford vector based algorithms used in traditional Internet routing protocols such as RIP. OSPF has introduced new concepts such as authentication of routing updates, Variable Length Subnet Masks (VLSM), route summarization, and so forth.

OSPF versus RIP

The rapid growth and expansion of today's networks has pushed RIP to its limits. RIP has certain limitations that can cause problems in large networks:

- RIP has a limit of 15 hops. A RIP network that spans more than 15 hops (15 routers) is considered unreachable.

- RIP cannot handle Variable Length Subnet Masks (VLSM). Given the shortage of IP addresses and the flexibility VLSM gives in the efficient assignment of IP addresses, this is considered a major flaw.
- Periodic broadcasts of the full routing table consume a large amount of bandwidth. This is a major problem with large networks especially on slow links and WAN clouds.
- RIP converges slower than OSPF. In large networks convergence gets to be in the order of minutes.RIP routers go through a period of a hold down and garbage collection and slowly time out information that has not been received recently. This is inappropriate in large environments and could cause routing inconsistencies.
- RIP has no concept of network delays and link costs. Routing decisions are based on hop counts. The path with the lowest hop count to the destination is always preferred even if the longer path has a better aggregate link bandwidth and less delays.
- RIP networks are flat networks. There is no concept of areas or boundaries. With the introduction of classless routing and the intelligent use of aggregation and summarization, RIP networks seem to have fallen behind.

Some enhancements were introduced in a new version of RIP called RIP2. RIP2 addresses the issues of VLSM, authentication, and multicast routing updates. RIP2 is not a big improvement over RIP (now called RIP 1) because it still has the limitations of hop counts and slow convergence which are essential in today's large networks.

OSPF, on the other hand, addresses most of the issues previously presented:

- With OSPF, there is no limitation on the hop count.
- The intelligent use of VLSM is very useful in IP address allocation.
- OSPF uses IP multicast to send link state updates. This ensures less processing on routers that are not listening to OSPF packets. Also, updates are only sent in case routing changes occur instead of periodically. This ensures a better use of bandwidth.
- OSPF has better convergence than RIP. This is because routing changes are propagated instantaneously and not periodically.
- OSPF allows for better load balancing.
- OSPF allows for a logical definition of networks where routers can be divided into areas. This limits the explosion of link state updates over the whole network. This also provides a mechanism for aggregating routes and cutting down on the unnecessary propagation of subnet information.

What Do We Mean by Link States?

OSPF is a link state protocol. We could think of a link as being an interface on the router. The state of the link is a description of that interface and of its relationship to its neighboring routers. A description of the interface would include, for example, the IP address of the interface, the mask, the type of network it is connected to, the routers connected to that network and so on. The collection of all these link states would form a link state database.

Shortest Path First Algorithm

- OSPF uses a Dijkstra's algorithm shorted path first algorithm in order to build and calculate the shortest path to all known destinations.
- Upon initialization or due to any change in routing information, a router generates a link state advertisement. This advertisement represents the collection of all link states on that router.
- All routers exchange link states by means of flooding. Each router that receives a link state update should store a copy in its link state database and then propagate the update to other routers.
- After the database of each router is completed, the router calculates a Shortest Path Tree to all destinations. The router uses the Dijkstra's algorithm in order to calculate the shortest path tree. The destinations, the associated cost and the next hop to reach those destinations form the IP routing table.

Areas and Border Routers

As previously mentioned, OSPF uses flooding to exchange link state updates between routers. Any change in routing information is flooded to all routers in the network. Areas are introduced to put a boundary on the explosion of link state updates. Flooding and calculation of the Dijkstra algorithm on a router is limited to changes within an area. All routers within an area have the exact link state database. Routers that belong to multiple areas, and connect these areas to the backbone area are called area border routers (ABR). ABRs must therefore maintain information describing the backbone areas and other attached areas.

An area is interface specific. A router that has all of its interfaces within the same area is called an internal router (IR). A router that has interfaces in multiple areas is called an area border router (ABR). Routers that act as gateways (redistribution) between OSPF and other routing protocols (IGRP, EIGRP, ISIS, RIP, BGP, Static) or other instances of the OSPF routing process are called autonomous system boundary router (ASBR). Any router can be an ABR or an ASBR.

The Backbone and Area 0

OSPF has special restrictions when multiple areas are involved. If more than one area is configured, one of these areas has be to be area 0. This is called the backbone. When designing networks it is good practice to start with area 0 and then expand into other areas later on.

The backbone has to be at the center of all other areas, i.e. all areas have to be physically connected to the backbone. The reasoning behind this is that OSPF expects all areas to inject routing information into the backbone and in turn the backbone will disseminate that information into other areas.

Neighbors

Routers that share a common segment become neighbors on that segment. Neighbors are elected via the Hello protocol. Hello packets are sent periodically out of each interface using IP multicast. Routers become neighbors as soon as they see themselves listed in the neighbor's Hello packet. This way, a two way communication is guaranteed. Two routers will not become neighbors unless they agree on the following:

Area id: Two routers having a common segment; their interfaces have to belong to the same area on that segment. Of course, the interfaces should belong to the same subnet and have a similar mask.

Authentication: OSPF allows for the configuration of a password for a specific area. Routers that want to become neighbors have to exchange the same password on a particular segment.

Hello and Dead Intervals: OSPF exchanges Hello packets on each segment. This is a form of keepalive used by routers in order to acknowledge their existence on a segment and in order to elect a designated router (DR) on multi-access segments. The Hello interval specifies the length of time, in seconds, between the hello packets that a router sends on an OSPF interface. The dead interval is the number of seconds that a router's hello packets have not been seen before its neighbors declare the OSPF router down. OSPF requires these intervals to be exactly the same between two neighbors. If any of these intervals are different, these routers will not become neighbors on a particular segment. The router interface commands used to set these timers are:

```
ip ospf hello interval seconds

ip ospf dead interval seconds.
```

Stub area flag: Two routers have to also agree on the stub area flag in the Hello packets in order to become neighbors. Stub areas will be discussed in a later section. Keep in mind for now that defining stub areas will affect the neighbor election process.

Adjacencies:

Adjacency is the next step after the neighboring process. Adjacent routers are routers that go beyond the simple Hello exchange and proceed into the database exchange process. In order to minimize the amount of information exchange on a particular segment, OSPF elects one router to be a designated router (DR), and one router to be a backup designated router (BDR), on each multi access segment. The BDR is elected as a backup mechanism in case the DR goes down. The idea behind this is that routers have a central point of contact for information exchange. Instead of each router exchanging updates with every other router on the segment, every router exchanges information with the DR and BDR. The DR and BDR relay the information to everybody else.

DR Election

DR and BDR election is done via the Hello protocol. Hello packets are exchanged via IP multicast packets on each segment. The router with the highest OSPF priority on a segment will become the DR for that segment. The same process is repeated for the BDR. In case there is a tie, the router with the highest RID will become a DR. The default for the interface OSPF priority is one.

Enabling OSPF on the Router

Enabling OSPF on the router involves the following two steps in config mode:

1. Enabling an OSPF process using the router ospf <process id> command.
2. Assigning areas to the interfaces using the network **<network ID or IP address> <wildcard mask> <area id>** command.

For example,

```
Router(config)#router ospf 1

Router(config-router)#network 192.168.12.0 0.0.0.255 area 0
```

The OSPF **process-id** is a numeric value local to the router. It does not have to match process ids on other routers. It is possible to run multiple OSPF processes on the same router, but is not recommended as it creates multiple database instances that add extra overhead to the router.

The network command is a way of assigning an interface to a certain area. The mask is used as a shortcut and it helps putting a list of interfaces in the same area with one line configuration line. The mask contains wildcard bits where 0 is a **"match"** and 1 is a **"do not care"** bit, e.g. 0.0.255.255 indicates a match in first two bytes of the network number.

The **area-id** is the area number we want the interface to be in. The area id can be an integer between 0 and 4294967295 or can take a form similar to an IP address A.B.C.D.

Chapter at a Glance

- OSPF Version 1 was specified in RFC 1131 in 1988. This protocol was finalized in 1989.
- OSPF Version 2 (Current version). The most recent specifications are specified in RFC 2328.
- Scales better than Distance Vector Routing protocols. It virtually has no practical Hop Count Limit.
- Provides Load Balancing (Equal Cost).
- Introduces the concept of Area's to ease management and control traffic.
- Provides Authentication.
- Uses Multicast versus Broadcasts.
- Convergence is faster than in Distance Vector Routing protocols. The reason for that is it floods the changes to all neighboring routers simultaneously rather than in a chain.
- Supports Variable Length Subnet Masking (VLSM), FLSM and Supernetting.
- Provides bit-based Route summarization.
- There are no periodic updates. Updates are only sent when there are changes.
- Router only sends changes in updates and not the entire full tables.
- OSPF uses a Cost Value, instead of hop count. Cost is based on the speed of the link. Cost = 108/Bandwidth.
- Classless Routing Protocol.
- It relies on IP to deliver the Packets. Use port 89.
- Area is a logical grouping of OSPF routers.
- Areas divide an OSPF domain into sub-domains.

- Areas allow OSPF to be extremely scalable.
- Areas reduce the Memory, CPU utilization and amount of traffic in a network.
- Most of the traffic can be restricted to within the area.
- Routers within an area will have no detailed knowledge of the topology outside of their area.
- Reduced size of the Database reduces Memory requirements for the routers.
- Area's identified by a 32-bit Area ID. Can be denoted in Decimal format (0) or Dotted format (0.0.0.0)
- OSPF requires one area to be Area 0, known as the backbone area.
- Backbone area or Area 0, connects all the other area to each other.

Three types of Traffic may be defined in relation to areas:

- **Intra-area** traffic consists of packets that are passed between routers within a single area.
- **Inter-area** traffic consists of packets that are passed between routers in different areas.
- **External** traffic consists of packets that are passed between a router within the OSPF domain and a router within another Autonomous system.

Some OSPF terms:

Priority

The Pri field indicates the priority of the neighbor router. The router with the highest priority becomes the designated router (DR). If the priorities are the same, then the router with the highest router ID becomes the DR. By default, priorities are set to 1. A router with a priority of 0 never becomes a DR or a backup designated router (BDR); it is always a DROTHER, meaning a router that is neither the DR nor the BDR.

State

The State field indicates the functional state of the neighbor router. Refer to OSPF Neighbor States for more information about states. FULL means the router is fully adjacent with its neighbor. The neighbor is the DR, so it is Router 1.

Dead Time

The Dead Time field indicates the amount of time remaining that the router waits to receive an OSPF hello packet from the neighbor before declaring the neighbor down. On broadcast and point to point media, the default dead interval is 40 seconds. On non broadcast and point to multipoint links, the default dead interval is 120 seconds.

Address

The Address field indicates the IP address of the interface to which this neighbor is directly connected. In the case of unnumbered links, this field shows the IP address of the interface to which the neighbor is unnumbered. When OSPF packets are transferred to the neighbor, this address will be the destination address.

Interface

The Interface field indicates the interface on which the OSPF neighbor has formed adjacency. In the above example the neighbor can be reached through FastEthernet0/0.

OSPF Neighbor State:

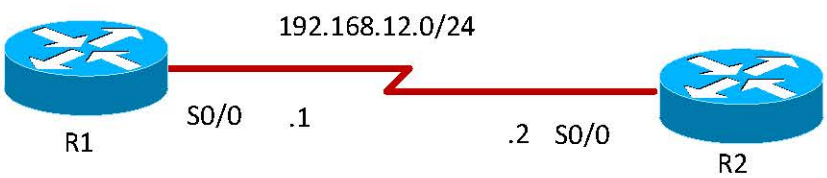
- **Init State:** First Hello is sent
- **2-Way:** Neighbor discovered, but adjacency not built
- **Exstart:** Neighbor's form a Master/Slave Relationship. Based on the Highest IP address. Initial: sequence number established.

- **Exchange:** The router's exchange Database Description packets to tell each other about the routes it knows about. A request list is created.
- **Loading:** Link State Request is sent to each other and based on the LSR's received; Link State Update packets are sent back in both directions.
- **Full:** All neighbors have a consistent Database.

OSPF over ethernet terms:

- **DR:** The neighbor is the DR.
- **BDR:** The neighbor is the BDR.
- **DROTHER:** The neighbor is neither a DR nor BDR.

Lab 1 – OSPF Point-to-Point Configuration



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
S 0/0	192.168.12.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
S 0/0	192.168.12.2	255.255.255.0

Configure the Interface IP addresses based on the above table

On R1

```
!
router ospf 1
network 1.0.0.0 0.255.255.255 area 0
network 192.168.12.0 0.0.0.255 area 0
!
```

On R2

```
!
router ospf 1
network 1.0.0.0 0.255.255.255 area 0
network 192.168.12.0 0.0.0.255 area 0
!
```

Task 1

Test the Configuration

```
On R2
R2#show ip route ospf
    1.0.0.0/32 is subnetted, 1 subnets
O       1.1.1.1 [110/65] via 192.168.12.2, 00:00:02, Serial0/0
```

On R1

```
R1#show ip route ospf
    2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/65] via 192.168.12.1, 00:00:02, Serial0/0
```

What routes do you see?

```
R2#show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           0   FULL/  -        00:00:38    192.168.12.1    Serial0/0
```

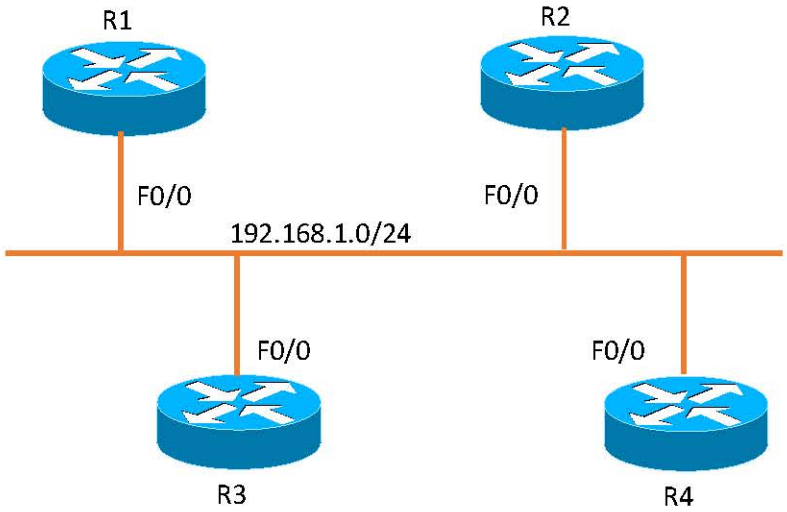
```
R2#show ip ospf interface serial 0/0
Serial0/0 is up, line protocol is up
  Internet Address 192.168.12.2/24, Area 0
  Process ID 1, Router ID 2.2.2.2, Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    . . .
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 1.1.1.1
  Suppress hello for 0 neighbor(s)
```

What type of netwok do you see?

```
R1#show ip ospf database
            OSPF Router with ID (1.1.1.1) (Process ID 1)
                Router Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         658         0x80000004 0x00B0DF 3
2.2.2.2         2.2.2.2         656         0x80000002 0x005237 3
```

What type of LSA can you see?

Lab 2 – OSPF over Ethernet Configuration



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
F 0/0	192.168.1.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
F 0/0	192.168.1.2	255.255.255.0

R3

Interface	IP Address	Subnet Mask
Loopback 0	3.3.3.3	255.0.0.0
F 0/0	192.168.1.3	255.255.255.0

R4

Interface	IP Address	Subnet Mask
Loopback 0	4.4.4.4	255.0.0.0
F 0/0	192.168.1.4	255.255.255.0

Task 1

Configure OSPF on All Ethernet segments in Area 0. Advertise all networks on all routers. Hard Code the Router-id based on the following: -

- R1 – 1.1.1.1
- R2 – 2.2.2.2
- R3 – 3.3.3.3
- R4 – 4.4.4.4

Task 2

Verfication

```
R1#show ip route ospf
     2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/11] via 192.168.1.2, 00:01:16, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/11] via 192.168.1.3, 00:01:16, FastEthernet0/0
     4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/11] via 192.168.1.4, 00:01:16, FastEthernet0/0
```

```
R1#show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   2WAY/DROTHER    00:00:32    192.168.1.2     FastEthernet0/0
3.3.3.3           1   FULL/BDR        00:00:32    192.168.1.3     FastEthernet0/0
4.4.4.4           1   FULL/DR         00:00:37    192.168.1.4     FastEthernet0/0
```

Can you see DR/BDR?

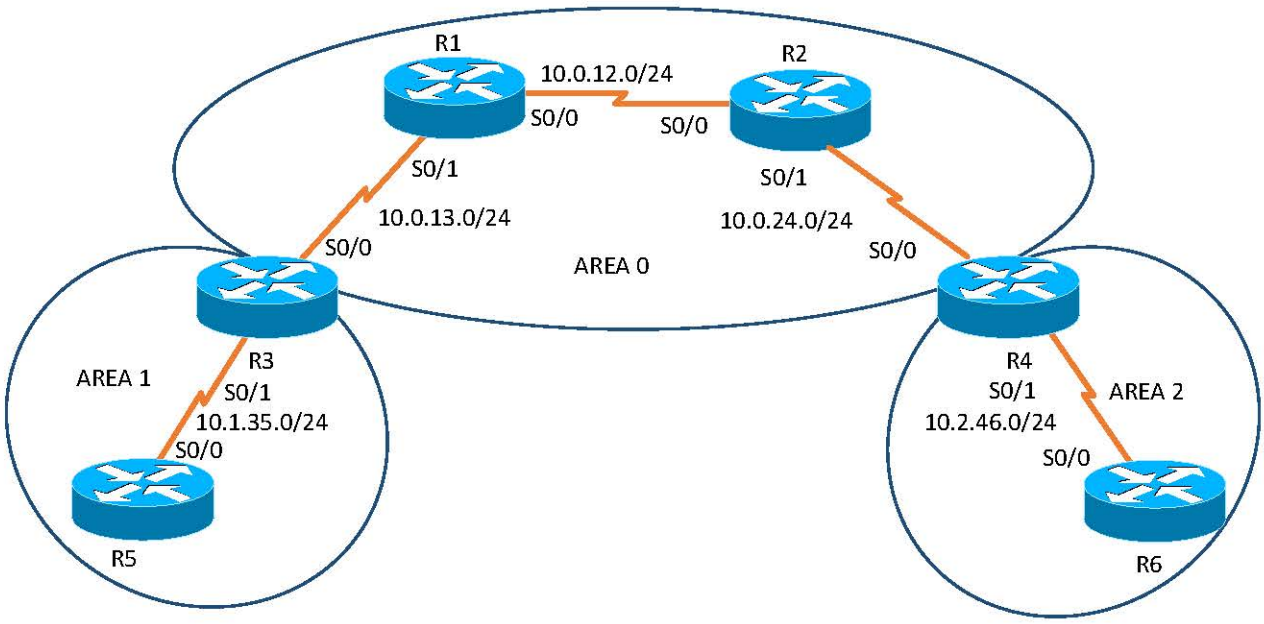
What is criterion for electing DR/BDR?

```
R1#show ip ospf database
            OSPF Router with ID (1.1.1.1) (Process ID 1)
                Router Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         272         0x80000001 0x008FA3 2
2.2.2.2         2.2.2.2         275         0x8000001A 0x004DBF 2
3.3.3.3         3.3.3.3         234         0x80000004 0x0069AC 2
4.4.4.4         4.4.4.4         233         0x80000005 0x0057B0 2
                Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
192.168.1.4     4.4.4.4         233         0x80000001 0x008803
```

What type of LSA do you see?

Can you see LSA 2?

Lab 3 – Configuring a Multi-Area OSPF Network



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	10.0.1.1	255.255.255.0
S 0/0	10.0.12.1	255.255.255.0
S 0/1	10.0.13.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	10.0.2.2	255.255.255.0
S 0/0	10.0.12.2	255.255.255.0
S 0/1	10.0.24.2	255.255.255.0

R3

Interface	IP Address	Subnet Mask
Loopback 0	10.0.3.3	255.255.255.0
Loopback 1	10.1.3.3	255.255.255.0
S 0/0	10.1.13.3	255.255.255.0
S 0/1	10.1.35.3	255.255.255.0

R4

Interface	IP Address	Subnet Mask
Loopback 0	10.0.4.4	255.255.255.0
Loopback 1	10.2.4.4	

S 0/0	10.2.24.4	255.255.255.0
S 0/1	10.2.46.4	255.255.255.0

R5

Interface	IP Address	Subnet Mask
Loopback 0	5.5.5.5	255.0.0.0
S 0/0	10.1.35.5	255.255.255.0

R6

Interface	IP Address	Subnet Mask
Loopback 0	6.6.6.6	255.0.0.0
S 0/0	10.1.46.6	255.255.255.0

Configure the Interface IP addresses based on the above table

Task 1

Configure OSPF on R1 and R2 in Area 0. Advertise all loopback networks on R1 and R2 routers in Area 0. Hard Code the Router-id based on the following: -

- R1 – 1.1.1.1
- R2 – 2.2.2.2

Task 2

Configure OSPF on R3 S0/0 interface network in Area 0 and S0/1 in Area 1. Advertise loopback 0 network on R3 in Area 0 and loopback 1 on R3 routers in Area 1. Hard Code the Router-id based on the following: -

- R3 – 3.3.3.3

Task 3

Configure OSPF on R4 S0/0 interface network in Area 0 and S0/1 in Area 2. Advertise loopback 0 network on R4 in Area 0 and loopback 1 on R4 routers in Area 2. Hard Code the Router-id based on the following: -

- R4 – 4.4.4.4

Task 4

Configure OSPF on R5 S0/0 interface network in Area 1. Advertise loopback 0 network on R5 in Area 1 Hard Code the Router-id based on the following: -

- R5 – 5.5.5.5

Task 5

Configure OSPF on R6 S0/0 interface network in Area 2. Advertise loopback 0 network on R6 in Area 2. Hard Code the Router-id based on the following: -

- R5 – 6.6.6.6

```
On R1
!
enable
configure terminal
!
interface Loopback0
 ip address 10.0.1.1 255.255.255.0
!
interface Serial0/0
 ip address 10.0.12.1 255.255.255.0
 no shutdown
!
interface Serial0/1
 ip address 10.0.13.1 255.255.255.0
 no shutdown
!
router ospf 1
 router-id 1.1.1.1
 network 10.0.12.0 0.0.0.255 area 0
 network 10.0.13.0 0.0.0.255 area 0
 network 10.0.1.0 0.0.0.255 area 0
!
```

```
On R2
!
interface Loopback0
 ip address 10.0.2.2 255.255.255.0
!
interface Serial0/0
 ip address 10.0.12.2 255.255.255.0
 no shutdown
!
interface Serial0/1
 ip address 10.0.24.2 255.255.255.0
 no shutdown
!
router ospf 1
 router-id 2.2.2.2
 network 10.0.12.0 0.0.0.255 area 0
 network 10.0.24.0 0.0.0.255 area 0
 network 10.0.2.0 0.0.0.255 area 0
!
```

```
On R3
!
interface Loopback0
 ip address 10.0.3.3 255.255.255.0
!
interface Loopback1
 ip address 10.1.3.3 255.255.255.0
!
interface Serial0/0
 ip address 10.0.13.3 255.255.255.0
 no shutdown
!
interface Serial0/1
 ip address 10.1.35.3 255.255.255.0
 no shutdown
!
router ospf 1
 router-id 3.3.3.3
 network 10.0.13.0 0.0.0.255 area 0
 network 10.1.35.0 0.0.0.255 area 1
 network 10.0.3.0 0.0.0.255 area 0
 network 10.1.3.0 0.0.0.255 area 1
!
```

```
On R4
!
interface Loopback0
 ip address 10.0.4.4 255.255.255.0
!
interface Loopback1
 ip address 10.2.4.4 255.255.255.0
!
interface Serial0/0
 ip address 10.0.24.4 255.255.255.0
 no shutdown
!
interface Serial0/1
 ip address 10.2.46.4 255.255.255.0
 no shutdown
!
router ospf 1
 router-id 4.4.4.4
 network 10.0.24.0 0.0.0.255 area 0
 network 10.2.46.0 0.0.0.255 area 2
 network 10.0.4.0 0.0.0.255 area 0
 network 10.2.4.0 0.0.0.255 area 2
!
```

```
On R5
!
interface Loopback0
 ip address 10.1.5.5 255.255.255.0
!
interface Serial0/0
 ip address 10.1.35.5 255.255.255.0
 no shutdown
!
router ospf 1
 router-id 5.5.5.5
 network 10.1.35.0 0.0.0.255 area 1
 network 10.1.5.0 0.0.0.255 area 1
!
```

```
On R6
!
interface Loopback0
 ip address 10.2.6.6 255.255.255.0
!
interface Serial0/0
 ip address 10.2.46.6 255.255.255.0
 no shutdown
!
router ospf 1
 router-id 6.6.6.6
 network 10.2.6.0 0.0.0.255 area 2
 network 10.2.46.0 0.0.0.255 area 2
 !
```

Task 6

Verfication

```
R1#show ip route ospf
    10.0.0.0/8 is variably subnetted, 12 subnets, 2 masks
O IA    10.2.6.6/32 [110/193] via 10.0.12.2, 00:21:16, Serial0/0
O IA    10.2.4.4/32 [110/129] via 10.0.12.2, 00:30:39, Serial0/0
O       10.0.4.4/32 [110/129] via 10.0.12.2, 00:30:39, Serial0/0
O       10.0.2.2/32 [110/65] via 10.0.12.2, 00:30:39, Serial0/0
O IA    10.1.5.5/32 [110/129] via 10.0.13.3, 00:22:19, Serial0/1
O IA    10.1.3.3/32 [110/65] via 10.0.13.3, 00:29:43, Serial0/1
O       10.0.24.0/24 [110/128] via 10.0.12.2, 00:30:39, Serial0/0
O IA    10.2.46.0/24 [110/192] via 10.0.12.2, 00:27:06, Serial0/0
O IA    10.1.35.0/24 [110/128] via 10.0.13.3, 00:24:31, Serial0/1
```

What type of route can you see on R1?

```
R1#show ip ospf database
          OSPF Router with ID (1.1.1.1) (Process ID 1)
Router Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         1868        0x8000000A 0x005D0A 5
2.2.2.2         2.2.2.2         124         0x80000008 0x001037 5
3.3.3.3         3.3.3.3         1869        0x80000005 0x00FC38 3
4.4.4.4         4.4.4.4         1714        0x80000006 0x00D436 3
Summary Net Link States (Area 0)
```

```
Link ID         ADV Router      Age         Seq#       Checksum
10.1.3.3        3.3.3.3         157         0x80000002 0x0069B4
10.1.5.5        3.3.3.3         1421        0x80000001 0x00C317
10.1.35.0       3.3.3.3         1553        0x80000001 0x00A022
10.2.4.4        4.4.4.4         55          0x80000002 0x002AEC
10.2.6.6        4.4.4.4         1358        0x80000001 0x00844F
10.2.46.0       4.4.4.4         1709        0x80000001 0x00FCB5
```

Can you see LSA 3?

What is the name of LSA 3?

On R3 or R4 (ABR Routers)

```
R3#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 3.3.3.3
  It is an area border router
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  . . . (Output ommited for brevity)
 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
    5.5.5.5              110      00:27:17
    1.1.1.1              110      00:34:42
    2.2.2.2              110      00:34:42
    4.4.4.4              110      00:26:14
  Distance: (default is 110)
```

What is an ABR?

MODULE 9

BGP

Chapter 1 – BGP

Border Gateway Protocol (BGP)

The Border Gateway Protocol (BGP) is an inter-autonomous system routing protocol. An autonomous system is a network or group of networks under a common administration and with common routing policies. BGP is used to exchange routing information for the Internet and is the protocol used between Internet service providers (ISP). Customer networks, such as universities and corporations, usually employ an Interior Gateway Protocol (IGP) such as RIP or OSPF for the exchange of routing information within their networks. Customers connect to ISPs, and ISPs use BGP to exchange customer and ISP routes. When BGP is used between autonomous systems (AS), the protocol is referred to as External BGP (eBGP). If a service provider is using BGP to exchange routes within an AS, then the protocol is referred to as Interior BGP (iBGP).

Features

- Inter-domain routing protocol also known as EGP.
- Latest version is 4.
- It supports both IPv4 and IPv6 address family.
- Defined in RFC 1771.
- Autonomous system is a set of routers under a single technical administration, using an IGP & common metrics to route packets within the AS, and using an EGP to route packets to other AS.
- Internet Assigned Numbers Authority (IANA) is responsible for allocating AS numbers.
- AS number is a 16 bit number.
- Range between 1 – 65535
- 64512 – 65535 reserved for private use like private IP addresses.

- An autonomous system can be connected to more than one ISP. This type of AS is known as a multihomed AS. This may be done for redundancy or/and to increase performance through load balancing.
- BGP exchanges routes between AS in a loop free manner.

When to use BGP

- An AS allows packets to transit through it to reach other AS (for example, ISP).
- An AS has multiple connections to other AS.
- The flow of traffic entering & leaving your AS must be manipulated.

Size of an internet BGP router

- > 130MB routing table
- > 648698 (as on 11-03-2017) routes
- > 6500 AS numbers

When not to use BGP (instead use static routes)

- A single connection to the Internet or other AS
- Routing policy & route selection not a concern
- Lack of hardware requirements on the routers
- Lack of understanding of route

BGP Terminology

- Advanced D.V. protocol.
- Runs on top of the TCP port 179 hence provides reliability.
- Incremental, flash updates.
- Periodic keepalives to verify connectivity.
- Rich metric (called path vectors or attributes).
- Extremely Scalable.
- BGP routers contain two tables.
- IGP routing table.
- BGP routing table.
- Information can be exchanged.
- Peers/Neighbors – any two routers that have formed a TCP connection in order to exchange BGP routing information.

Internal BGP

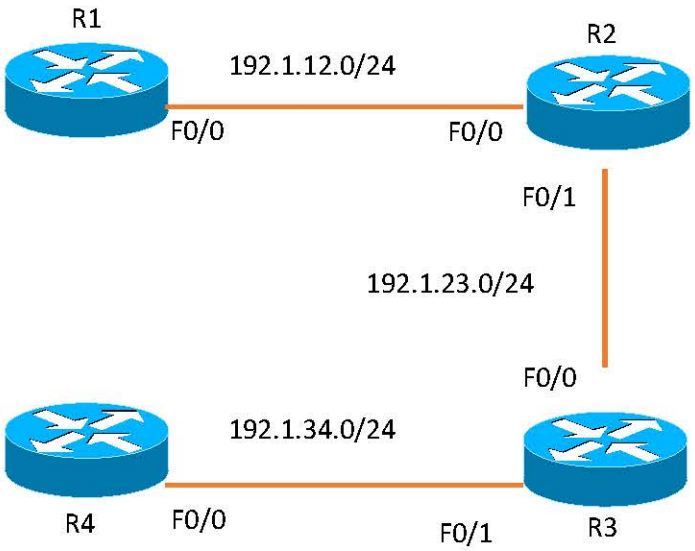
- When BGP neighbors belong to the same AS.
- Neighbors do not have to be directly connected but need to be able to reach each other.

External BGP

- BGP neighbors belong to different AS and should be able to reach each other.
- Neighbors should be directly connected.
- Used to connect different Autonomous Systems to each other.

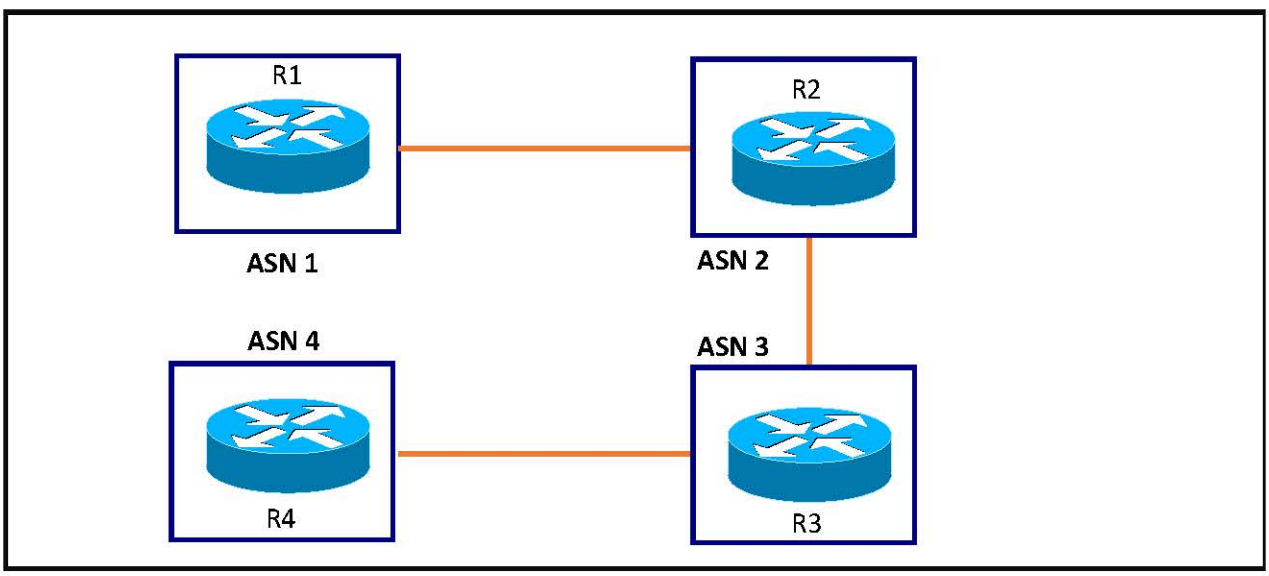
Note: In CCNA, we are looking into the configuration of eBGP only. No iBGP and BGP path selection will be discussed, as it is beyond the scope of CCNA.

Lab 1 – Configuring eBGP



Physical Layout

BGP Logical Layout



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
Loopback 1	11.1.0.1	255.255.0.0
F 0/0	192.1.12.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
Loopback 1	12.1.0.1	255.255.0.0
F 0/0	192.1.12.2	255.255.255.0

F 0/1	192.1.23.2	255.255.255.0

R3

Interface	IP Address	Subnet Mask
Loopback 0	3.3.3.3	255.0.0.0
Loopback 1	13.1.0.1	255.255.0.0
F 0/0	192.1.23.3	255.255.255.0
F 0/1	192.1.34.3	255.255.255.0

R4

Interface	IP Address	Subnet Mask
Loopback 0	4.4.4.4	255.0.0.0
Loopback 1	14.1.0.1	255.255.0.0
F 0/0	192.1.34.4	255.255.255.0

Task 1

Configure a BGP neighbor relationship between R1 and R2. R1 should be in AS 1 and R2 should be in AS 2. Advertise the loopback networks in BGP. Hard-code the Router ID for the BGP routers as 11.11.11.11 for R1 and 22.22.22.22 for R2.

```
R1
!
router bgp 1
 bgp router-id 11.11.11.11
 network 1.0.0.0
 network 11.1.0.0 mask 255.255.0.0
 neighbor 192.1.12.2 remote-as 2
!
R2
!
router bgp 2
 bgp router-id 22.22.22.22
 network 2.0.0.0
 network 12.1.0.0 mask 255.255.0.0
 neighbor 192.1.12.1 remote-as 1
!
```

Task 2

Configure a BGP neighbor relationship between R2 and R3. R2 should already be configured in AS 2 and R3 should be in AS 3. Advertise the loopback network of R3 in BGP. Hard-code the Router ID for R3 as 33.33.33.33

```
R2
!
router bgp 2
neighbor 192.1.23.3 remote-as 3
!
R3
!
router bgp 3
 bgp router-id 33.33.33.33
 network 3.0.0.0
 network 13.1.0.0 mask 255.255.0.0
 neighbor 192.1.23.2 remote-as 2
!
```

Task 3

Configure a BGP neighbor relationship between R3 and R4. R3 should already be configured in AS 3 and R4 should be in AS 4. Advertise the loopback network of R4 in BGP. Hard-code the Router ID for R4 as 44.44.44.44. Establish the neighbor relationship based on Loopback 0 addresses. You are allowed to create a static route on each router to accomplish this task.

```
R3
ip route 4.0.0.0 255.0.0.0 192.1.34.4
!
router bgp 3
neighbor 4.4.4.4 remote-as 4
neighbor 4.4.4.4 update-source loo0
neighbor 4.4.4.4 ebgp-multihop
!
R4
ip route 3.0.0.0 255.0.0.0 192.1.34.3
!
router bgp 4
 bgp router-id 44.44.44.44
 network 4.0.0.0
 network 14.1.0.0 mask 255.255.0.0
 neighbor 3.3.3.3 remote-as 3
 neighbor 3.3.3.3 update-source loop 0
 neighbor 3.3.3.3 ebgp-mulithop
!
```

Task 3

Verify by following command

```
On R1, R2, R3 and R4
R1#show ip bgp summary
R1#show ip bgp
R1#show ip route
```

MODULE 10

IPv6

Chapter 1 – IPv6 Address

Internet Protocol (IP) version 6

Internet Protocol (IP) version 6 is a new IP protocol designed to replace IP version 4, which is deployed today and used throughout the world.

The current IP version, IPv4, has proven to be robust, easily implemented, interoperable, and has stood the test of scaling an internetwork to a global utility the size of the Internet today. However, the initial design of IPv4 did not anticipate the rapid growth of the Internet and the exhaustion of the IPv4 address

The lifetime of IPv4 has been extended with techniques such as private address space with Network Address Translation (NAT). Although these techniques seem to increase the address space and satisfy the traditional client-server setup, they fail to meet the requirements of IP address growth.

The need to reach always-on environments (such as residential Internet through broadband, cable modem, or DSL) precludes IP-address conversion, pooling, and temporary allocation techniques. Also, the plug-and-play capabilities required by consumer Internet appliances further increase the address requirements.

The designers and users of the early Internet could not have anticipated the recent rapid growth of the Internet and the impending exhaustion of the IPv4 address space. The IPv6 address protocol meets the current requirements of the new applications and the never ending growth of the Internet.

The IPv6 address space makes more addresses available but it must be approached with careful planning. Successful deployment of IPv6 can be achieved with existing IPv4 infrastructures. With proper planning and design, the transition between IP version 4 and 6 is possible today as well.

The Internet Engineering Task Force (IETF) designed the IPv6 addressing scheme to provide interoperability with existing IPv4 network architecture and to allow the coexistence of IPv6 networks with existing IPv4 networks

IPv6 Address Format

IPv6 uses 16-byte hexadecimal number fields separated by colons (:) to represent the 128-bit addressing

format that makes the address representation less cumbersome and error-prone. Here is an example of a valid IPv6 address: 2001:db8:130F:0000:0000:09C0:876A:130B.

Additionally, to shorten the IPv6 address and make the address easier to represent, IPv6 uses the following conventions:

- Leading zeros in the address field are optional and can be compressed. For example: The following hexadecimal numbers can be represented as shown in a compressed format:

Example 1: 0000 = 0 (compressed form)

Example 2: 2001:db8:130F:0000:0000:09C0:876A:130B

=2001:db8:130F:0:0:9C0:876A:130B (compressed form)

- A pair of colons (::) represents successive fields of 0. However, the pair of colons is allowed just once in a valid IPv6 address.

Example 1: 2001:db8:130F:0:0:9C0:876A:130B

=2001:db8:130F::9C0:876A:130B (compressed form)

Example 2: FF01:0:0:0:0:0:1 = FF01::1 (compressed form)

An address parser can easily identify the number of missing zeros in an IPv6 address by separating the two parts of the address and filling in the 0s until the 128-bit address is complete. However, if two ::s are placed in the same address, then there is no way to identify the size of each block of zeros. The use of the :: makes many IPv6 addresses very small.

Network Prefix

In IPv6 there are references to prefixes which, in IPv4 terms, loosely equate to subnets. The IPv6 prefix is made up of the left-most bits and acts as the network identifier. The IPv6 prefix is represented using the IPv6-prefix or prefix-length format just like an IPv4 address is represented in the classless interdomain routing (CIDR) notation.

The /prefix-length variable is a decimal value that indicates the number of high-order contiguous bits of the address that form the prefix, which is the network portion of the address. For example: **2001:db8:8086:6502::/64** is an acceptable IPv6 prefix. If the address ends in a double colon, the trailing double colon can be omitted. So the same address can be written as **2001:db8:8086:6502/64.** In either case, the prefix length is written as a decimal number 64 and represents the left-most bits of the IPv6 address

IPv6 Address Types

Prefix	Designation and Explanation	IPv4 Equivalent
::/128	**Unspecified** This address may only be used as a source address by an initialising host before it has learned its own address.	**0.0.0.0**
::/0	**Default Route** This address is used to configure default gateway or default route	**0.0.0.0/0**
::1/128	**Loopback** This address is used when a host talks to itself over IPv6. This often happens when one program sends data to another.	**127.0.0.0**
fc00::/7	**Unique Local Addresses (ULAs)** These addresses are reserved for local use in home and enterprise environments and are not public address space.	**Private, or RFC 1918 address space:** **10.0.0.0/8** **172.16.0.0/12** **192.168.0.0/16**

fe80::/10	**Link-Local Addresses** These addresses are used on a single link or a non-routed common access network, such as an Ethernet LAN. They do not need to be unique outside of that link.	**NA**
2000::/3	**Multicast** These addresses are used to identify multicast groups. They should only be used as destination addresses, never as source addresses.	**224.0.0.0/4**

IPv6 Neighbor Discovery

The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and track neighboring devices.

The IPv6 static cache entry for neighbor discovery feature allows static entries to be made in the IPv6 neighbor cache. Static routing requires an administrator to manually enter IPv6 addresses, subnet masks, gateways, and corresponding Media Access Control **(MAC)** addresses for each interface of each device into a table. Static routing enables more control but requires more work to maintain the table. The table must be updated each time routes are added or changed.

- IPv6 Neighbor Solicitation Message
- IPv6 Router Advertisement Message
- IPv6 Neighbor Redirect Message

IPv6 Neighbor Solicitation Message

A value of **135** in the Type field of the ICMP packet header identifies a neighbor solicitation message. Neighbor solicitation messages are sent on the local link when a node wants to determine the link-layer address of another node on the same local link (see the figure below). When a node wants to determine the link-layer address of another node, the source address in a neighbor solicitation message is the IPv6 address of the node sending the neighbor solicitation message. The destination address in the neighbor solicitation message is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address in the neighbor advertisement message is the IPv6 address of the node (more specifically, the IPv6 address of the node interface) sending the neighbor advertisement message. The destination address in the neighbor advertisement message is the IPv6 address of the node that sent the neighbor solicitation message. The data portion of the neighbor advertisement message includes the link-layer address of the node sending the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate.

Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. When a node wants to verify the reachability of a neighbor, the destination address in a neighbor solicitation message is the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is such a change, the destination address for the neighbor advertisement is the all-nodes multicast address.

IPv6 Router Advertisement Message

Router advertisement (RA) messages, which have a value of 134 in the Type field of the ICMP packet header, are periodically sent out each configured interface of an IPv6 device. For stateless autoconfiguration to work properly, the advertised prefix length in RA messages must always be 64 bits.

RA messages typically include the following information:

- One or more onlink IPv6 prefixes that nodes on the local link can use to automatically configure their IPv6 addresses
- Lifetime information for each prefix included in the advertisement
- Sets of flags that indicate the type of autoconfiguration (stateless or stateful) that can be completed
- Default device information (whether the device sending the advertisement should be used as a default device and, if so, the amount of time, in seconds, the device should be used as a default device)
- Additional information for hosts, such as the hop limit and maximum transmission unit (MTU) a host should use in packets that it originates

RAs are also sent in response to device solicitation messages. Device solicitation messages, which have a value of **133** in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message. Given that device solicitation messages are usually sent by hosts at system startup (the host does not have a configured unicast address), the source address in device solicitation messages is usually the unspecified IPv6 address **(0:0:0:0:0:0:0:0)**. If the host has a configured unicast address, the unicast address of the interface sending the device solicitation message is used as the source address in the message. The destination address in device solicitation messages is the all-devices multicast address with a scope of the link. When an RA is sent in response to a device solicitation, the destination address in the RA message is the unicast address of the source of the device solicitation message.

IPv6 Neighbor Redirect Message

A value of **137** in the type field of the ICMP packet header identifies an IPv6 neighbor redirect message. Devices send neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination.

After forwarding a packet, a device should send a redirect message to the source of the packet under the following circumstances:

- The destination address of the packet is not a multicast address.
- The packet was not addressed to the device.
- The packet is about to be sent out the interface on which it was received.
- The device determines that a better first-hop node for the packet resides on the same link as the source of the packet.
- The source address of the packet is a global IPv6 address of a neighbor on the same link, or a link-local address.

SLAAC Defined

Like IPv4, there are a number of different ways that a host can be addressed in IPv6; the two most common in IPv4 are static addressing and dynamic address configuration via the Dynamic Host Configuration Protocol (DHCP). Often the reason that engineers use DHCP is that it not only provides a method of dynamically assigning addresses, it also provides a way to assign the host devices other service information like DNS servers, domain names, and a number of different custom information.

To perform address configuration on IPv6 there are a couple of familiar methods and a few additional methods, including: static addressing, static addressing with DHCPv6 (stateless), dynamic addressing via DHCPv6 (Stateful), SLAAC alone, or SLAAC with DHCPv6 (Stateless). IPv6 static addressing works exactly the same as IPv4 static addressing so there is no mystery there. IPv6 does, however, provide two different ways of implementing DHCP, either stateful (e.g., when an IPv4 DHCP server tracks the addresses that are given out) and stateless. Stateless DHCP does not track what information is given out to clients and does not give out IPv6 addresses; instead, it provides the extra information that most people relate with **typical** DHCP assignment, e.g., DNS server information. Stateless DHCP is then matched up with another mechanism (such as Static addressing or SLAAC) for IPv6 address assignment.

SLAAC provides the ability to address a host based on a network prefix that is advertised from a local

network router via Router Advertisements (RA). RA messages are sent by default by most IPV6 routers; these messages are sent out periodically by the router and include information including:

- One or more IPv6 prefixes (Link-local scope)
- Prefix lifetime information
- Flag information
- Default device information (Default router to use and its lifetime)

SLAAC is implemented on the IPv6 client by listening for these local RA's and then taking the prefix that is advertised to form a unique address that can be used on the network. For this to work, the prefix that is advertised must advertise a prefix length of 64 bits (i.e., /64); SLAAC will then dynamically form a host identifier that is 64 bits long and will be suffixed to the end of the advertised prefix to form an IPv6 address. Originally, the host identifier was formed using the **EUI-64** rules (the same that are used to form link local addresses) and many devices still use this method. However, some Microsoft operating systems by default do not use this original method. Instead, they take advantage of some additional privacy extensions that were defined in RFC4941.

If the hosts (H1-H4) were using the EUI-64 method of host identification, the IPv6 addresses created using SLAAC would be:

- H1 – 2000:1234:5678::12FF:FE34:5678
- H2 – 2000:1234:5678::EBFF:FEA4:C1AE
- H3 – 2000:1234:5678::BAFF:FE24:C4AE
- H4 – 2000:1234:5678::84FF:FE67:AEFC

The EUI-64 process will be outlined for H1 as follows:

The prefix **2000:1234:5678::/64** will be learned from **R1's** RA messages and will be the initial prefix.

The client identifier would then be created from the MAC address that is assigned to H1, in this case 0200:1234:5678. The first step of **EUI-64** conversion is to split the MAC address in half and place **FF:FE** in the middle, which results in **0200:12FF:FE34:5678.** Then the **seventh bit** will be **flipped**, in this case the first **8 bits is 00000010 (0x02)**. Next, the seventh bit is flipped and the **bit becomes 0, resulting in 00000000 (0x00)**; this gives a final host identifier result of **0000:12FF:FE34:5678**. When the prefix and the host identifier are brought together, it results in an IPv6 address that is used for H1 of **2000:1234:5678:0000:0000:12FF :FE34:5678**, which can be shortened to **2000:1234:5678::12FF:FE34:5678**.

IPv6 Transition Technologies

Dual Stack

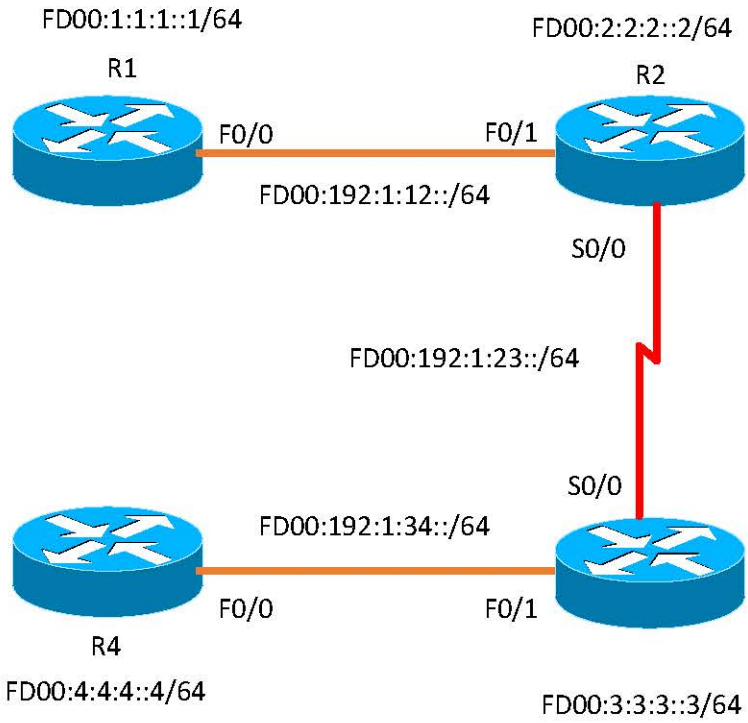
Dualstack is the basic strategy to use for large agencies that are adopting

IPv6. It involves configuring devices to be able to run IPv4 and IPv6 simultaneously. IPv4 communication uses the IPv4 protocol stack, and IPv6 communication uses the IPv6 protocol stack

Tunneling

Tunnels encapsulate IPv6 traffic within IPv4 packets, and are primarily used for communication between IPv6 (or dual-stack) sites or for connection to remote IPv6 networks or hosts over an IPv4 backbone.

Lab 1 – Configuring IPv6 ND



Task 1

Enable IPv6 routing on R1, R2, R3 and R4. Assign IPv6 addresses to the F 0/0 interface of the routers as follows:

- R1 – FD00:192:1:12::1 /64 , FE80::1 link-local
- R2 – FD00:192:1:12::2 /64, FE80::2 link-local
- R3 – FD00:192:1:34::3 /64, FE80::3 link-local
- R4 – FD00:192:1:34::4 /64, FE80::4 link-local

Task 2

Configure the Serial Link between R2 and R3 using the folloing IPV6 addresses:

- R2 – FD00:192:1:23::2/64, FE80::2 link-local
- R3 – FD00:192:1:23::3/64, FE80::3 link-local

Task 3

Configure the Loopback0 interface on all routers as follows:

- R1 – Loopback0 – FD00:1:1:1::1/64
- R2 – Loopback0 – FD00:2:2:2::2/64
- R3 – Loopback0 – FD00:3:3:3::3/64
- R4 – Loopback0 – FD00:4:4:4::4/64

On R1

```
conf t
!
ipv6 unicast-routing
!
interface F 0/0
 ipv6 address FD00:192:1:12::1/64
 ipv6 address FE80::1 link
 no shut
!
interface Loopback 0
 ipv6 address FD00:1:1:1::1/64
!
```

On R2

```
conf t
!
ipv6 unicast-routing
!
interface F 0/0
 ipv6 address FD00:192:1:12::2/64
 ipv6 address FE80::2 link
 No shut
!
interface S0/0
 ipv6 address FD00:192:1:23::2/64
 ipv6 address FE80::2 link
 no shut
!
interface Loopback 0
 ipv6 address FD00:2:2:2::2/64
!
```

On R3

```
conf t
!
ipv6 unicast-routing
!
interface F 0/0
 ipv6 address FD00:192:1:34::3/64
 ipv6 address FE80::3 link
 No shut
!
interface S0/0
 ipv6 address FD00:192:1:23::3/64
 ipv6 address FE80::3 link
 no shut
!
interface Loopback 0
 ipv6 address FD00:3:3:3::3/64
!
```

On R4

```
conf t
!
ipv6 unicast-routing
!
interface F 0/0
 ipv6 address FD00:192:1:34::4/64
 ipv6 address FE80::4 link
 no shut
!
interface Loopback 0
 ipv6 address FD00:4:4:4::4/64
!
```

Task 3

Verify the ND using the debug commands

```
R1#debug ipv6 nd
R1#ping FD00:192:1:12::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FD00:192:1:12::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/44/88 ms
R1#
ICMPv6-ND: DELETE -> INCMP: FD00:192:1:12::2
ICMPv6-ND: Sending NS for FD00:192:1:12::2 on FastEthernet0/0
ICMPv6-ND: Received NA for FD00:192:1:12::2 on FastEthernet0/0 from FD00:192:1:12::2
ICMPv6-ND: Neighbour FD00:192:1:12::2 on FastEthernet0/0 : LLA c201.14c8.0000
ICMPv6-ND: INCMP -> REACH: FD00:192:1:12::2
ICMPv6-ND: Received NS for FD00:192:1:12::1 on FastEthernet0/0 from FE80::2
ICMPv6-ND: DELETE -> INCMP: FE80::2
ICMPv6-ND: Neighbour FE80::2 on FastEthernet0/0 : LLA c201.14c8.0000
ICMPv6-ND: INCMP -> STALE: FE80::2
ICMPv6-ND: Sending NA for FD00:192:1:12::1 on FastEthernet0/0
ICMPv6-ND: STALE -> DELAY: FE80::2
ICMPv6-ND: DELAY -> PROBE: FE80::2
ICMPv6-ND: Sending NS for FE80::2 on FastEthernet0/0
ICMPv6-ND: Received NA for FE80::2 on FastEthernet0/0 from FE80::2
ICMPv6-ND: PROBE -> REACH: FE80::2
```

```
R1#show ipv6 neighbors
IPv6 Address                Age  Link-layer Addr State   Interface
FE80::2                      1   c201.14c8.0000  REACH   Fa0/0
FD00:192:1:12::2             1   c201.14c8.0000  REACH   Fa0/0
```

Lab 2 – Configuring IPv6 SLAAC

(Build on lab 1)

Task 1

Romove the static IPv6 address from R4 and configure the R4 to receive the IPv6 from R3.

Task 2

Verfiy by debug ipv6 nd

```
R1#debug ipv6 nd
. . .
ICMPv6-ND: Sending RS on FastEthernet0/0
ICMPv6-ND: Received RA from FE80::3 on FastEthernet0/0
ICMPv6-ND: Prefix Information change for FD00:192:1:34::/64, 0x0 -> 0xE0
ICMPv6-ND: Adding prefix FD00:192:1:34::/64 to FastEthernet0/0
ICMPv6-ND: Sending NS for FD00:192:1:34:C003:14FF:FEC8:0 on FastEthernet0/0
ICMPv6-ND: Autoconfiguring FD00:192:1:34:C003:14FF:FEC8:0 on FastEthernet0/0
. . .
```

Task 3

Verfiy ip address

```
R4#show ipv6 interface brief
FastEthernet0/0          [up/up]
   FE80::C003:14FF:FEC8:0
   FD00:192:1:34:C003:14FF:FEC8:0
```

Lab 3 – Configuring IPv6 with Static and Default Routes

(Build on Lab 1)

Task 1

Configure static route on R1 router to route all loopbacks on R2, R3 and R4.

Task 2

Configure static route on R2 router to route all loopbacks on R1, R3 and R4.

Task 3

Configure static route on R3 router to route all loopbacks on R1, R2 and R4.

Task 4

Configure static default route on R4 router to route all loopbacks on R1, R2and R3. Also configure static route for indirectly connected link on respective routers.

```
R1
!
ipv6 route FD00:2:2:2::/64 FD00:192:1:12::2
ipv6 route FD00:3:3:3::/64 FD00:192:1:12::2
ipv6 route FD00:4:4:4::/64 FD00:192:1:12::2
ipv6 route FD00:192:1:23::/64 FD00:192:1:12::2
ipv6 route FD00:192:1:34::/64 FD00:192:1:12::2
!
R2
!
ipv6 route FD00:1:1:1::/64 FD00:192:1:12::1
ipv6 route FD00:3:3:3::/64 FD00:192:1:23::3
ipv6 route FD00:4:4:4::/64 FD00:192:1:23::3
ipv6 route FD00:192:1:34::/64 FD00:192:1:23::3
!
R3
ipv6 route FD00:4:4:4::/64 FD00:192:1:34::4
ipv6 route FD00:2:2:2::/64 FD00:192:1:23::2
ipv6 route FD00:1:1:1::/64 FD00:192:1:23::2
ipv6 route FD00:192:1:12::/64 FD00:192:1:23::2
R4
!
ipv6 route ::/0  FD00:192:1:34::3
!
```

Task 5

Verify by checking routes on R1

```
R1#show ipv6 route static
S   FD00:2:2:2::/64 [1/0]
     via FD00:192:1:12::2
S   FD00:3:3:3::/64 [1/0]
     via FD00:192:1:12::2
S   FD00:4:4:4::/64 [1/0]
     via FD00:192:1:12::2
S   FD00:192:1:23::/64 [1/0]
     via FD00:192:1:12::2
S   FD00:192:1:34::/64 [1/0]
     via FD00:192:1:12::2
```

Task 6

Verify by checking routes on R4

```
R4#show ipv6 route static
S   ::/0 [1/0]
     via FD00:192:1:34::3
```

Task 7

Verify by ping test from R4's loopback 0 to R1's loopback 0 for end to end conntectivity.

```
R4#ping ipv6 FD00:1:1:1::1 source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FD00:1:1:1::1, timeout is 2 seconds:
Packet sent with a source address of FD00:4:4:4::4
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/56/80 ms
```

Lab 4 – Configuring IPv6 with RIPng

(Build on Lab 1)

Task 1

Configure RIPng on all routers to route all loopbacks. Enable RIPng under the following interfaces:

- R1 – F 0/0, Loopback 0
- R2 – F 0/0, Loopback 0, S 0/0
- R3 – F 0/0, Loopback 0, S 0/0
- R4 – F 0/0, Loopback 0

R1

```
ipv6 unicast-routing
interface Loopback 0
 ipv6 rip 100 enable
!
interface F 0/0
 ipv6 rip 100 enable
!
```

R2

```
ipv6 unicast-routing
interface Loopback 0
 ipv6 rip 100 enable
!
interface S 0/0
 ipv6 rip 100 enable
!
interface F 0/0
 ipv6 rip 100 enable
```

R3

```
ipv6 unicast-routing
interface Loopback 0
 ipv6 rip 100 enable
!
interface F 0/0
 ipv6 rip 100 enable
!
interface S0/0
 ipv6 rip 100 enable
!
```

R4

```
ipv6 unicast-routing
interface Loopback 0
 ipv6 rip 100 enable
!
interface F 0/0
 ipv6 rip 100 enable
!
```

Task 2

Verify routing table

```
R1#show ipv6 route rip
IPv6 Routing Table - 10 entries
R   FD00:2:2:2::/64 [120/2]
     via FE80::2, FastEthernet0/0
R   FD00:3:3:3::/64 [120/3]
     via FE80::2, FastEthernet0/0
R   FD00:4:4:4::/64 [120/4]
     via FE80::2, FastEthernet0/0
R   FD00:192:1:23::/64 [120/2]
     via FE80::2, FastEthernet0/0
R   FD00:192:1:34::/64 [120/3]
     via FE80::2, FastEthernet0/0
```

Task 3

Verify by ping test from R4's loopback 0 to R1's loopback 0 for end to end conntectivity.

```
R4#ping ipv6 FD00:1:1:1::1 source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FD00:1:1:1::1, timeout is 2 seconds:
Packet sent with a source address of FD00:4:4:4::4
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/56/80 ms
```

Lab 5 – Configuring IPv6 with EIGRP

(Build on Lab 1)

Task 1

Disable RIP NG on all routers on all interfaces.

R1 interface loopback 0 no ipv6 rip 100 enable ! interface f 0/0 no ipv6 rip 100 enable !	**R2** interface loopback 0 no ipv6 rip 100 enable ! interface f 0/0 no ipv6 rip 100 enable ! interface s0/0 no ipv6 rip 100 enable !
R3 interface loopback 0 no ipv6 rip 100 enable ! interface f 0/0 no ipv6 rip 100 enable ! interface s0/0 no ipv6 rip 100 enable !	**R4** interface loopback 0 no ipv6 rip 100 enable ! interface f 0/0 no ipv6 rip 100 enable !

Task 2

Configure EIGRPv6 in AS 100 on all routers to route all loopbacks. Configure the EIGRP Router-id's as follows:

- R1 – 1.1.1.1
- R2 – 2.2.2.2
- R3 – 3.3.3.3
- R4 – 4.4.4.4

R1

```
interface loopback 0
 ipv6 eigrp 100
!
interface f 0/0
 ipv6 eigrp 100
!
ipv6 router eigrp 100
 router-id 1.1.1.1
 no shut
!
```

R2

```
interface loopback 0
 ipv6 eigrp 100
!
interface f 0/0
 ipv6 eigrp 100
!
interface s 0/0
 ipv6 eigrp 100
!
ipv6 router eigrp 100
 router-id 2.2.2.2
 no shut
!
```

R3

```
interface loopback 0
 ipv6 eigrp 100

interface f 0/0
 ipv6 eigrp 100
!
interface s 0/0
 ipv6 eigrp 100
!
ipv6 router eigrp 100
 router-id 3.3.3.3
 no shutdown
!
```

R4

```
interface loopback 0
 ipv6 eigrp 100
!
interface f 0/0
 ipv6 eigrp 100
!
ipv6 router eigrp 100
 router-id 4.4.4.4
 no shutdown
!
```

Task 3

Verify ipv6 route

```
R1#show ipv6 route eigrp
D   FD00:2:2:2::/64 [90/409600]
    via FE80::2, FastEthernet0/0
D   FD00:3:3:3::/64 [90/2323456]
    via FE80::2, FastEthernet0/0
D   FD00:4:4:4::/64 [90/2349056]
    via FE80::2, FastEthernet0/0
D   FD00:192:1:23::/64 [90/2195456]
    via FE80::2, FastEthernet0/0
D   FD00:192:1:34::/64 [90/2221056]
    via FE80::2, FastEthernet0/0
```

Task 4

Verify by ping test from R4's loopback 0 to R1's loopback 0 for end to end conntectivity.

```
R4#ping ipv6 FD00:1:1:1::1 source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FD00:1:1:1::1, timeout is 2 seconds:
Packet sent with a source address of FD00:4:4:4::4
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/56/80 ms
```

Lab 6 – Configuring IPv6 with OSPFv3

(Build on lab 1)

Task 1

Disable EIGRP on all routers on all interfaces. Disable the protocol on the router as well.

R1 ! interface loopback 0 no ipv6 eigrp 100 ! interface f 0/0 no ipv6 eigrp 100 ! no ipv6 router eigrp 100 !	**R2** ! interface loopback 0 no ipv6 eigrp 100 ! interface f 0/0 no ipv6 eigrp 100 ! interface s 0/0 ! no ipv6 router eigrp 100 !
R3 ! interface loopback 0 no ipv6 eigrp 100 ! interface f 0/0 no ipv6 eigrp 100 ! interface s 0/0 ! no ipv6 router eigrp 100 !	**R4** ! interface loopback 0 no ipv6 eigrp 100 ! interface f 0/0 no ipv6 eigrp 100 ! no ipv6 router eigrp 100 !

Task 2

Configure the routers in OSPFv3 area 0 and advertise their directly connected interfaces in this area. Configure the OSPF Router-id's as follows:

- R1 – 1.1.1.1
- R2 – 2.2.2.2
- R3 – 3.3.3.3
- R4 – 4.4.4.4

R1 ! ipv6 router ospf 1 router-id 1.1.1.1 ! interface loopback0 ipv6 ospf 1 area 0 ! interface f 0/0 ipv6 ospf 1 area 0 !	**R2** ! ipv6 router ospf 1 router-id 2.2.2.2 ! interface loopback0 ipv6 ospf 1 area 0 ! interface f 0/0 ipv6 ospf 1 area 0 ! interface s 0/0 ipv6 ospf 1 area 0 !

R3 ! ipv6 router ospf 1 router-id 3.3.3.3 ! interface loopback0 ipv6 ospf 1 area 0 ! interface f 0/0 ipv6 ospf 1 area 0 ! interface s 0/0 ipv6 ospf 1 area 0 !	**R4** ! ipv6 router ospf 1 router-id 4.4.4.4 ! interface loopback0 ipv6 ospf 1 area 0 ! interface f 0/0 ipv6 ospf 1 area 0 !

Task 3

Ensure that the loopback interfaces are advertised with their correct mask.

R1 ! interface loopback0 ipv6 ospf network point-to-point !	R2 ! interface loopback0 ipv6 ospf network point-to-point !
R3 ! Interface Loopback0 ipv6 ospf network point-to-point !	R4 ! interface loopback0 ipv6 ospf network point-to-point !

Task 4

Check IPv6 routes

```
R1#show ipv6 route ospf
O   FD00:2:2:2::2/128 [110/10]
     via FE80::2, FastEthernet0/0
O   FD00:3:3:3::3/128 [110/74]
     via FE80::2, FastEthernet0/0
O   FD00:4:4:4::4/128 [110/84]
     via FE80::2, FastEthernet0/0
O   FD00:192:1:23::/64 [110/74]
     via FE80::2, FastEthernet0/0
O   FD00:192:1:34::/64 [110/84]
     via FE80::2, FastEthernet0/0
```

MODULE 11

IPv4 And IPv6 Acess List For Traffic Filtering

Chapter 1 – IPv4 Access-list and IPv6 Access-list

Overview of IPv4 Access List

Used to define the type of traffic that should be allowed or restricted from crossing a router (entering or exiting a router interface).

Set of rules that help control flow of packets into or out of a router.

Statements that specify how the router will handle the traffic flow through specified interfaces.

Uses of Access Lists

Filter packet flow in/out of router interfaces.

Restrict/reduce contents of routing updates, e.g. from RIP, IGRP.

Identify packets that will initiate dial-on-demand connections (interesting packets).

Types of access lists

Standard Access Lists: Check source address of packets and permit or deny the packets based on network, subnet or host address.

Extended Access Lists: Check both source and destination addresses for filtering. Packets can be filtered based on protocols within a suite (e.g. TCP/IP) and port numbers. Extended Access Lists add more granularity than Standard Access Lists.

Access Lists Operation and Application

Operate in sequential, logical order, following a top-down order of tests.

If no conditions, or tests, are met, a final implicit deny will drop that particular packet.

Routers stop processing once the first instance of a condition is met in the written access list.

Only one access list per protocol per interface is permitted.

Access lists can be inbound or outbound, with reference to a router interface.

Location and sequential order can affect performance of router.

Written in global configuration mode (by the **access-list** command) and grouped, or linked in interface mode for the appropriate router interface (by the **access-group** command).

Verifying Access Lists

Router#show interface or **show [ip|ipx] interface**

Use to see if an interface is grouped to an access list

Returns IP addresses and all configuration parameters.

Router#show access-lists

Shows details of all access lists configured.

Router#show [ip|ipx] access-list

Shows access lists for a specified protocol.

Access Lists Types and Numbers

Protocol	Type	Access List Number Range
IPv4	Standard	1-99
	Extended	100-199

Wildcard Mask Bits

0 indicates that the corresponding bit should be checked.

1 indicates that the corresponding bit should be ignored.

Examples

00000111 indicates that only the last three bits in the corresponding octet should be ignored

0.0.0.0 indicates any IP address – check all bits in all four octets

255.255.255.255 indicates that all bits should be ignored – use the **any** statement.

Match any IP address 0.0.0.0 255.255.255.255 – any address, ignore all bits.

Match a specific IP address w.x.y.z 0.0.0.0 – check all bits so they match – use the **host** command, as follows: host **w.x.y.z**

Configuring Standard IP Access Lists

Creating the accessing list

```
Router(config)#access-list [1-99][permit|deny] source_address  wildcard_mask
Applying it to an interface
Router(config-if)#ip access-group [1-99][in|out]
```

Note the last statement in the access-group statement. In or out specifies incoming or outgoing traffic. By default, all access lists are applied to outgoing traffic, i.e. if the in or out statement is omitted, out will be applied.

Examples

Permitting only a specific network

To allow only traffic from 172.16.0.0 to pass through the router

```
!
access-list 1 permit 172.16.0.0   0.0.255.255
(Implicit deny all – not necessary to write)
interface f0/0
 ip access-group 1 out
!
```

Denying a specific host

To deny only the host 172.16.4.10 and permit everyone else to communicate through the router.

```
!
access-list 1 deny host 172.16.4.10
access-list 1 permit ip any
interface F 0/0
 ip access-group 1 in
!
```

Configuring Extended IP Access Lists

Overview

Extended IP Access Lists filter based on source and destination addresses, specific protocols and even ports defined by TCP or UDP.

Extended IP Access Lists offer more granularity than Standard Access Lists and can be used in a wider range of situations in providing access security to a network through a router.

Configuration

Creating the access list

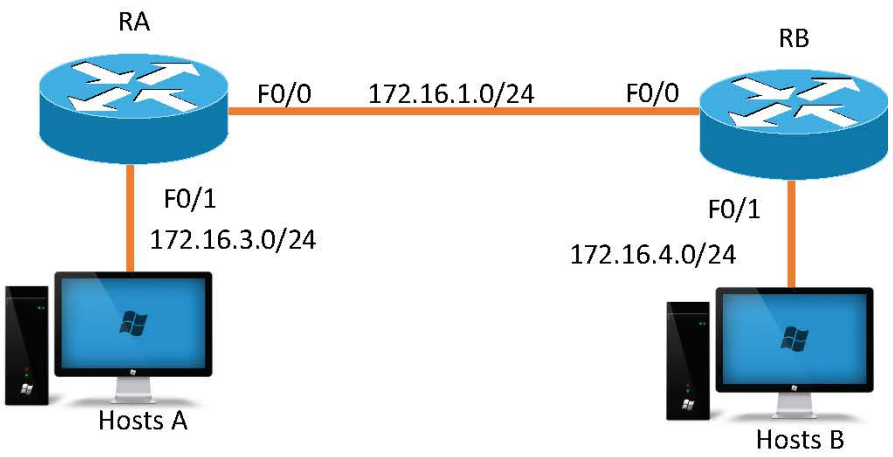
```
Router(config)#access-list [100-199] [permit|deny] [ip|tcp|icmp] [source_address] [source_mask] [destination_address]  [destination_mask] [eq|neq|lt|gt] [port_number]
```

Applying it to an interface

```
Router(config-if)#ip access-group [100-199] [in|out]
```

Examples

1. Blocking only FTP traffic from one network



The aim here is to block all FTP traffic from 172.16.3.0 entering 172.16.4.0 by creating an extended access list at R1

```
!
access-list 101 deny tcp 172.16.3.0 0.0.0.255 172.16.4.0 0.0.0.255 eq 20
access-list 101 deny tcp 172.16.3.0 0.0.0.255 172.16.4.0 0.0.0.255 eq 21
access-list 101 permit ip any any
int F 0/0
ip access-group 101 out
!
```

Note the third line in the access list – it permits all other IP-based traffic from anywhere going anywhere.

3.Denying all web-based (www) traffic entering a network

The aim here is to block all networks from accessing the www service on the 172.16.4.0 network.

```
!
Router(config)#access-list 101 deny tcp any any eq 80
Router(config)#access-list 101 permit ip any any
Router(config)#int F 0/0
Router(config-if)#ip access-group 101 out
!
```

4. Denying a host from executing a ping statement to a network

The aim here is to stop the host 172.16.3.5 from pinging other hosts on the 172.16.4.0 network.

```
!
access-list 101 deny icmp host 172.16.3.5 any echo
access-list 101 permit ip any any
!
int F 0/0
ip access-group 101 out
!
```

Verifying Access Lists

Show Access-lists displays the definition of all access lists that are created on the router.

Show IP access-lists displays the definition of IP access lists on this router.

Show IP interface displays the interface that is using a given access-list.

IPv6 access-list

IPv6 only has named extended access-lists.

IPv6 access-lists have **three invisible statements** at the bottom:

- permit icmp any any nd-na
- permit icmp any any nd-ns
- deny ipv6 any any

The two permit statements are required for neighbor discovery which is an important protocol in IPv6, it's the replacement for ARP.

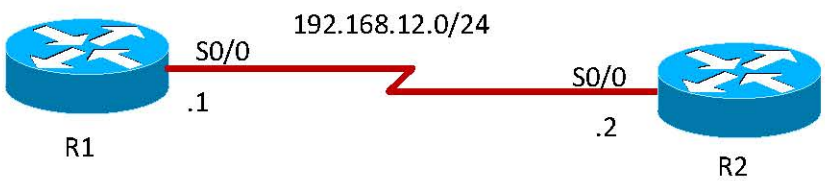
Note:

When you use a deny ipv6 any any at the bottom of your access-list, make sure you also add the two permit statements for neighbor discovery just before the final statement or this traffic will be dropped.

Example

```
!
Router(config)# ipv6 access-list FILTER
Router(config-ipv6-acl)# deny tcp any any eq 80
Router(config-ipv6-acl)# permit any any
!
Router(config)# interface f0/0
Router(config-if)# ipv6 traffic-filter FILTER out
!
```

Lab 1 –Denying a Host Using Standard Access Lists



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
S 0/0	192.168.12.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
S 0/0	192.168.12.2	255.255.255.0

Task 1

Configure the hostname R1 on Router 1 and R2 on Router 2.

Task 2

Configure the Serial Link between R1 & R2 using HDLC as the encapsulation. Set the clock rate at 128 kbps.

Task 3

Configure the loopback interface on R1 & R2.

Task 4

Configure the RIPv1 on R1 and R2 as follows:

- 1.0.0.0/8 on R1
- 192.168.12.0/24 on R1 and R2
- 2.0.0.0/8 on R2

On R1

```
enable
configure terminal
!
interface Loopback0
 ip address 1.1.1.1 255.0.0.0
!
interface Serial0/0
 ip address 192.168.12.1 255.255.255.0
 no shutdown
!
router rip
 version 2
 network 192.168.12.0
 network 1.0.0.0
!
```

On R2

```
enable
configure terminal
!
interface Loopback0
 ip address 2.2.2.2 255.0.0.0
!
interface Serial0/0
 ip address 192.168.12.2 255.255.255.0
 no shutdown
!
router rip
 version 2
 network 192.168.12.0
 network 2.0.0.0
!
```

Task 5

Create a Standard Access-list on R1 that blocks host 2.2.2.2 on R2 from accessing R1.

```
!
access-list 1 deny 2.2.2.2 0.0.0.0
access-list 1 permit any
!
```

Task 6

Applying the access-list to the Serial Interface

```
!
interface S 0/0
ip access-group 1 in
!
```

Task 7

On R2 Test the Standard Access list

```
R2#ping 1.1.1.1 source 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 2.2.2.2
UUUUU
Success rate is 0 percent (0/5)
```

Task 7

On R2 verify ACL

```
R1#show access-lists
Standard IP access list 1
    10 deny   2.2.2.2 (15 matches)
    20 permit any (7 matches)
```

Task 8

Deleting the Access-List

```
!
no access-list 1
interface S 0/0
no ip access-group 1 in
!
```

Lab 2 – Denying a Network Using Standard Access Lists

(Builds on Lab 1)

Task 1

Create a Standard Access-list on R2 that blocks Network 1.0.0.0 on R1 from accessing R2.

```
access-list 1 deny 1.0.0.0 0.255.255.255
access-list 1 permit any
```

Task 2

Applying the access-list to the Serial Interface

```
interface f 0/0
ip access-group 1 in
```

Task 3

On R2 test the Standard Access list

```
R1#ping 2.2.2.2 source 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
UUUUU
Success rate is 0 percent (0/5)
Are you successful?
```

Why or why not?

Task 4

Delete the access-list on R1

```
!
no access-list 10
interface S 0/0
no ip access-group 10 in
!
```

Task 5

On R2 verify ACL

```
R2#show access-lists 1
Standard IP access list 1
    10 deny   1.0.0.0, wildcard bits 0.255.255.255 (15 matches)
    20 permit any (34 matches)
```

Can you see the match?

Lab 3 – Denying an Entire Network from using Telnet

(Builds on Lab 2)

Task 1

Create an extended access on R1 list that blocks anyone from accessing the router via telnet.

```
!
access-list 100 deny tcp any any eq 23
access-list 100 permit ip any any
!
```

Task 2

Apply the access-list to the Serial Interface

```
!
interface Serial 0/0
ip access-group 100 in
!
```

Task 3

Test the ACL on R2

```
R2#telnet 1.1.1.1
Trying 1.1.1.1 ...
% Destination unreachable; gateway or host down
```

Are you successful?

Task 4

Deleting the Access-List

```
!
no access-list 100
interface S 0/0
no ip access-group 100 in
!
```

Task 5

On R2 verify ACL

```
R1#show access-lists
Extended IP access list 100
    10 deny tcp any any eq telnet (3 matches)
    20 permit ip any any (1 match)
```

Can you see the match?

Lab 4 – Denying a Host from Pinging R2's Loopback 0

(Builds on Lab 3)

Task 1

Create an extended access list on R2 that blocks Host from Pinging R2's Loopback.

```
access-list 100 deny icmp any host 2.2.2.2 echo
access-list 100 permit ip any any
```

Task 2

Applying the access-list to the Serial Interface

```
int S 0/0
ip access-group 101 in
```

Task 3

Test the Extended Access list from R1

```
R1#ping 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5)
```

Are you successful?

Task 4

Test the extended ACL from R2

```
R2#ping 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/43/72 ms
```

Are you successful?

Task 5

On R2 verify ACL

```
R2#show access-lists
Extended IP access list 100
    10 deny icmp any host 2.2.2.2 echo (15 matches)
    20 permit ip any any (1 match)
```

Lab 5 – Named Access List

(Builds on Lab 4)

Task 1

Create a Named Standard Access-list on R1 that blocks Network 2.0.0.0 from coming into R1.

```
ip access-list standard DENY-2
deny 2.0.0.0 0.255.255.255
permit any
```

Task 2

Apply the access-list to the Serial Interface

```
interface S 0/0
ip access-group DENY-2 in
```

Task 3

Create a Named Extended Access-list on R2 that blocks Network 1.0.0.0 from accessing the 2.0.0.0 Network

```
ip access-list extended DENY-1-2
deny ip 1.0.0.0 0.255.255.255 2.0.0.0 0.255.255.255
permit ip any any
```

Task 4

Applying the access-list to the Serial Interface

```
interface f 0/0
ip access-group DENY-1-2 in
```

Task 5

Test the Named Access list on R1 and R2

```
R1#ping 2.2.2.2 source 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
UUUUU
Success rate is 0 percent (0/5)
```

```
R2#ping 1.1.1.1 source 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 2.2.2.2
UUUUU
Success rate is 0 percent (0/5
```

Are you successful?

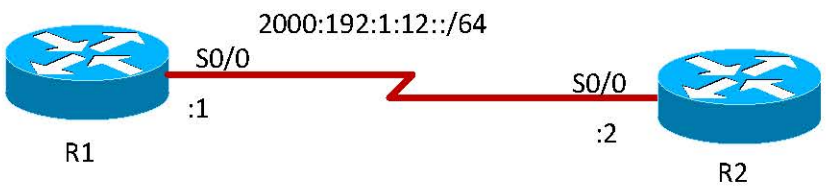
Task 6

Verify ACL on R1 and R2

```
R1#show access-lists
Standard IP access list DENY-2
    10 deny   2.0.0.0, wildcard bits 0.255.255.255 (15 matches)
    20 permit any (1 match)
```

```
R2#show access-lists
Extended IP access list DENY-1-2
    10 deny ip 1.0.0.0 0.255.255.255 2.0.0.0 0.255.255.255 (15 matches)
    20 permit ip any any (9 matches)
```

Lab 5 – IPv6 Access List



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	2000:1:1:1::1	/64
S 0/0	2000:192:1:12::1	/64
Link Local	FE80::1	/64

R2

Interface	IP Address	Subnet Mask
Loopback 0	2000:2:2:2::2	/64
S 0/0	2000:192:1:12::2	/64
Link Local	FE80::2	/64

Task 1

Configure the hostname R1 on Router 1 and R2 on Router 2.

Task 2

Configure the Serial Link between R1 & R2 using IPv6 addresses given above

Task 3

Configure the loopback interface on R1 & R2.

Task 4

Configure the RIPng on R1 and R2 as follows:

- Loopback 0 on R1
- Serial 0/0 on R1 and R2
- Loopback 0 on R2

```
On R1
hostname R1
!
ipv6 unicast-routing
!
interface Loopback0
 ipv6 address 2000:1:1:1::1/64
 ipv6 rip 1 enable
!
interface Serial0/0
 ipv6 address FE80::1 link-local
 ipv6 address 2000:192:1:12::1/64
 ipv6 rip 1 enable
 no shutdown
!
ipv6 router rip 1
!
```

```
On R2
hostname R2
!
ipv6 unicast-routing
!
interface Loopback0
 ipv6 address 2000:2:2:2::2/64
 ipv6 rip 1 enable
!
interface Serial0/0
 ipv6 address FE80::2 link-local
 ipv6 address 2000:192:1:12::2/64
 ipv6 rip 1 enable
 no shutdown
!
ipv6 router rip 1
!
```

Task 3

Create IPv6 Access-list on R1 that blocks Network 2000:2:2:2::2/64 from telnetting the 2000:1:1:1::1/64 network.

```
On R1
!
ipv6 access-list DENY-TELNET
 deny tcp 2000:2:2:2::/64 2000:1:1:1::/64 eq telnet
 permit ipv6 any any
!
```

Task 4

Apply IPv6 ACL on Serial 0/0

```
interface Serial0/0
 ipv6 traffic-filter DENY-TELNET in
!
```

Task 6

Test by telneting from R2 as follows

```
R2#telnet 2000:1:1:1::1 /source-interface loopback 0
Trying 2000:1:1:1::1 ...
% Destination unreachable; gateway or host down
```

Task 7

Check access-list on R1

```
R1#show ipv6 access-list
IPv6 access list DENY-TELNET
deny tcp 2000:2:2:2::/64 2000:1:1:1::/64 eq telnet (1 match) sequence 10
permit ipv6 any any (33 matches) sequence 20
!
```

MODULE 12

WAN TECHNOLOGIES

Chapter 1 – WAN Technologies

WAN Concepts and Terminology

Wide-area networks (WANs) connect networks, users, and services across broad geographic areas. Companies use WANs to connect company sites for information exchange.

Three WAN Connection Types

WAN services are generally leased from service providers on a subscription basis. There are three main types of WAN connections (services):

- **Leased-line** - Provides a preestablished connection through the service provider's network (WAN) to a remote network. Leased lines provide a reserved connection for the client but are costly. Leased-line connections are typically synchronous serial connections with speeds up to 45 Mbps (E3).
- **Circuit-switched** - Provides a dedicated circuit path between sender and receiver for the duration of the "call." Circuit switching is used for basic telephone service or Integrated Services Digital Network (ISDN). Circuit-switched connections are best for clients that require only sporadic WAN usage.
- **Packet-switched** - Devices transportpackets using virtual circuits (VCs) that provide end-to-end connectivity. Programmed switching devices provide physical connections. Packet headers are used to identify the destination. Packet switching offers leased-line-type services over shared lines, but at a much lower cost. Packet-switched networks typically use serial connections with speeds ranging from 56 Kbps to E3.

WAN Cabling

The router end of the cable connects to the DB-60 port on a serial WAN interface card (using a DB-60 connector). The connector on the other end of the serial cable is specified according to the standard used. The ports on either end of a WAN connection are specified as DTE (data terminal equipment) or data

communications equipment (DCE). DCE converts user data into the service provider's preferred format. The port configured as DTE requires external clocking from the CSU/DSU or another DCE device

Physical Parameters for WAN Connections

A WAN service provider assigns your organization the parameters required for making the WAN link connection.

- **Customer premises equipment (CPE)** is located on the subscriber's premises. It includes both equipment owned by the subscriber and devices leased by the service provider.
- **Demarcation, or demarc,** marks the point where CPE ends and the local loop begins. Usually, it is located in the telecommunications closet.
- **Local loop, or "last-mile,"** is the cabling from the **demarc** into the WAN service provider's central office (CO).
- The **central office** is a switching facility that provides a point of presence for WAN service. The central office is the entry point to the WAN cloud and the exit point from the WAN for called devices.
- A switching point for calls.
- The toll network is a collection of trunks inside the WAN cloud.

Layer 2 Encapsulation Protocols

- **High-level data link control (HDLC)** is the default encapsulation type on point-to-point dedicated links and circuit switched connections. HDLC should be used for communication between Cisco devices.
- **Point-to-Point Protocol (PPP)** provides connections between devices over several types of physical interfaces, such as asynchronous serial, HSSI, ISDN, and synchronous. PPP works with several network layer protocols, including IP and IPX. PPP uses PAP and CHAP for basic security.
- **X.25/Link Access Procedure, Balanced (LAPB)** defines connections between DTE and DCE for remote terminal access. LAPB is a data link layer protocol specified by X.25. Frame Relay is the industry-standard switched data link layer protocol. Frame Relay (based on X.25) can handle multiple virtual circuits.
- **Asynchronous Transfer Mode (ATM)** is the international standard for cell relay using fixed-length (53-byte) cells for multiple service types. Fixed-length cells allow hardware processing, which greatly reduces transit delays. ATM takes advantage of high-speed transmission media, such as E3, T3, and SONET.

Chapter 2 – Frame Relay

Overview

1. Frame Relay defines the interconnection process between the Customer Premises Equipment (CPE) device, such as a router, acting as a DTE and the service provider's local access switching equipment, acting as a DCE.
2. Frame Relay is a Layer 2 packet-switched WAN protocol.
3. Frame Relay can be configured in a point-to-point or multipoint environment, through the use of sub-interfaces.
4. Frames are encapsulated in one of two formats:

    i) **Cisco** - default, proprietary

    ii) **IETF** - use to connect to routers from different vendors (e.g. Lucent)

Frame Relay Terminology

Local Access Rate (AR)

Clock speed of connection to the Frame Relay cloud

Also known as local access loop, local loop.

Data Link Connection Identifier (DLCI)

1. Number that identifies the logical circuit between the CPE and FR switch (Layer 2).
2. DLCIs between each pair of routers are used to create a PVC.
3. DLCIs only have local significance.
4. DLCI numbers:
   - 0-15 reserved for signaling
   - 16-991 available for use
   - 992-1007 eserved for layer 2 management
   - 1008-1023 in-channel signaling
5. Inverse ARP maps DLCI number (Layer 2) to IP address (Layer 3).

Local Management Interface (LMI)

1. Protocol used for communication between Frame Relay switch and CPE
2. Signaling standard.
3. LMI is responsible for managing the connection and maintaining status between the 2 devices.
4. Keepalive packets verify that data is flowing between the 2 devices
5. LMI provides congestion notification.
6. 3 LMI standards:
   - Cisco (default)
   - ANSI
   - ITU Q.933a
7. LMI sent every 10 seconds by default
8. For IOS 10.3 and later, LMI type is auto-sensed
   (Frame Relay switch will send this to the router)
9. To set the LMI type,

```
Router(config)#lmi-type [cisco|ansi|itu]
```

Committed Information Rate (CIR)

Minimum guaranteed bandwidth for data transfer, within the Frame Relay cloud.

Oversubscription

When the sum of CIRs on all virtual circuits coming into a device exceed the access line speed. Once oversubscription occurs, packets are dropped.

Committed Burst (Bc)

Maximum number of bits the Frame Relay network agrees to transfer.

Excess Burst

Maximum number of uncommitted bits that the Frame Relay switch will attempt to transfer beyond

the CIR.

Dependent on the service provider.

Forward Explicit Congestion Notification (FECN)

When the Frame Relay switch recognizes congestion, it will set the DE bit to 1 in the Frame Relay packet bound for the destination. The destination router may drop the packet upon arrival.

Backward Explicit Congestion Notification (BECN)

When the router detects congestion, it sets the BECN bit to 1 and sends a packet to the source router, so the source router can reduce its rate of transmission of packets

Discard Eligibility (DE)

When the router detects congestion, this bit is turned to 1 for oversubscribed traffic. Packets with a DE bit equal to 1 will be discarded first by receiving routers.

Subinterfaces

Provide a method of separating one physical network connection into multiple logical connections, i.e. one local loop can support many PVCs.

A single physical interface (s 0/0) can simulate multiple logical interfaces (s 0/0.1, s 0/0.2, and so on), called subinterfaces.

Subinterfaces can be configured to support 2 connection types:

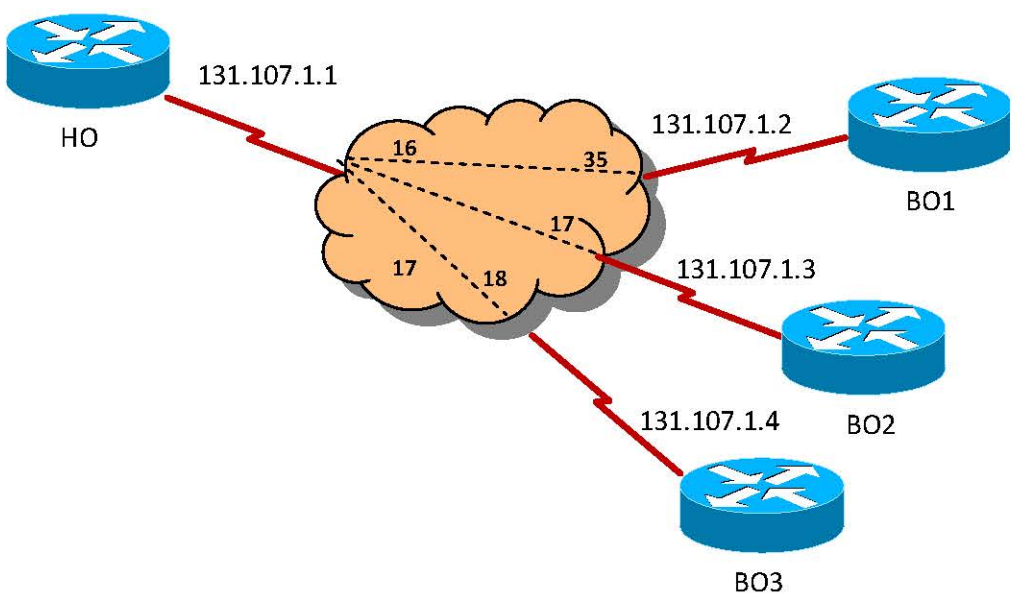
Point-to-point

1. Does not forward broadcasts or routing updates.
2. PVC connection is established from one subinterface to another Interfaces are on the same subnet, each subinterface has its own local and unique DLCI number.

Multipoint

1. Forwards broadcasts and routing updates.
2. A single interface establishes multiple PVCs to multiple interfaces or subinterfaces on remote routers.
3. All participating subinterfaces are on the same subnet, with unique local DLCIs.
4. Total number of subinterfaces = 1, ..., 4294967293.
5. Subinterfaces can be added at any time, even during normal operation.

Frame Relay – An Example



Once the DLCIs have been assigned by the service provider, mappings are created as follows (by Inverse ARP):

Router HO	
Local DLCI	**Destination Address**
16	131.107.1.2
17	131.107.1.3
18	131.107.1.4

Router BO1	
Local DLCI	**Destination Address**
35	131.107.1.1

Router B02	
Local DLCI	**Destination Address**
16	131.107.1.1

Router B03	
Local DLCI	**Destination Address**
18	131.107.1.1

So, if router B1 wishes to forward a packet to router HO, it sends the packet through its local DLCI 35, as all packets sent on that DLCI will get to 131.107.1.1 (Router HO).

Now BO1, BO2 and BO3 can each ping HO and vice versa, as there is a mapping of the path to get to the destination, butB1, B2 and B3 cannot ping each other.

For B1 to ping B2, for example, there must be a mapping from B1 to B2 via HO. This is done by the following command done at each respective router...

At router BO1

```
frame-relay map ip 131.107.1.3   35
```

At router BO2

```
frame-relay map ip 131.107.1.2   16
```

Similar mappings would need to be made for interconnecting B1 to B3 and B2 to B3.

Note: Mappings must be two-way for two-way communication, for example, ping.

This set up of routers is called a hub and spoke topology.

Verifying Frame Relay Configuration – Useful Commands

```
show frame-relay pvc
```

Shows DLCIs used and their status

Shows LMI type

Shows number of FECN and BECN bits received

```
show ip route
```

Shows routing table

```
show frame-relay map
```

Shows IP address to DLCI mapping

Shows if link to remote site is up or down

```
show frame-relay lmi
```

Shows lmi traffic status

Shows if link to Frame Relay switch from CPE is up or down

Chapter 3 – Metro Ethernet

Overview

A **metropolitan-area Ethernet, Ethernet MAN, or metro Ethernet network is a metropolitan area network** (MAN) that is based on Ethernet standards. It is commonly used to connect subscribers to a larger service network or the Internet. Businesses can also use metropolitan-area Ethernet to connect their own offices to each other.

An Ethernet interface is much cheaper than a synchronous digital hierarchy (SONET/SDH) or plesiochronous digital hierarchy (PDH) interface of the same bandwidth. Another distinct advantage of an Ethernet-based access network is that it can be easily connected to the customer network, due to the prevalent use of Ethernet in corporate and, more recently, residential networks. A typical service provider's network is a collection of switches and routers connected through optical fiber. The topology could be a ring, hub-and-spoke (star), or full or partial mesh. The network will also have a hierarchy: core, distribution (aggregation), and access. The core in most cases is an existing IP/MPLS backbone but may migrate to newer forms of Ethernet transport in the form of 10Gbit/s, 40Gbit/s, or 100Gbit/s speeds or even possibly 400Gbit/s to Terabit Ethernet network in the future.

Ethernet on the MAN can be used as pure Ethernet, Ethernet over SDH, Ethernet over MPLS, or Ethernet over DWDM. Pure Ethernet-based deployments are cheaper but less reliable and scalable and thus are usually limited to small scale or experimental deployments. SDH-based deployments are useful when there is an existing SDH infrastructure already in place, its main shortcoming being the loss of flexibility in bandwidth management due to the rigid hierarchy imposed by the SDH network. MPLS-based deployments are costly but highly reliable and scalable and are typically used by large service providers.

Metropolitan area network (MAN) topology

Familiar network domains are likely to exist regardless of the transport technology chosen to implement Metropolitan area networks: Access, aggregation/distribution, and core.

- Access devices normally exist at a customer's premises, unit, or wireless base station. This is the network that connects customer equipment, and may include ONT and/or Residentialoi gateway, or office router.
- Aggregation occurs on a distribution network such as an ODN segment. Often Passive Optical Network, microwave or Digital Subscriber Line technologies are employed, but some of them using point-to-point Ethernet over "home-run" direct fibre. This part of the network includes nodes such as Multi Tenanted Unit switches, Optical line terminals in an outside plant or central office cabinet, Ethernet in the First Mile equipment, or provider bridges.
- A MAN may include the transport technologies MPLS, PBB-TE and T-MPLS, each with its own resiliency and management solutions.
- A core network often uses IP-MPLS to connect different MANs together.

Much of the functionality of Ethernet MANs such as virtual private lines or virtual private networks is implemented by the use of **Ethernet VLAN tags** that allow differentiation of each part of the network. Logical differentiation of the physical network helps to identify the rights that the traffic has and to ease the management of hosts' access rights with respect to other users and networks.

Chapter 3 – Broadband PPPoE

Overview

The Point-to-Point Protocol over Ethernet (PPPoE) is a network protocol for encapsulating PPP frames inside Ethernet frames. It appeared in 1999, in the context of the boom of DSL as the solution for tunneling packets over the DSL connection to the ISP's IP network, and from there to the rest of the Internet. A 2005 networking book noted that "Most DSL providers use PPPoE, which provides authentication, encryption, and compression." Typical use of PPPoE involves leveraging the PPP facilities for authenticating the user with a username and password, predominately via the PAP protocol and less often via CHAP.

On the customer-premises equipment, PPPoE may be implemented either in a unified residential gateway device that handles both DSL modem and IP routing functions or in the case of a simple DSL modem (without routing support), PPPoE may be handled behind it on a separate Ethernet-only router or even directly on a user's computer. (Support for PPPoE is present in most operating systems, ranging from Windows XP, Linux to Mac OS X.

PPPoE was developed by UUNET, Redback Networks (now Ericsson) and RouterWare (now Wind River Systems) and is available as an informational RFC 2516.

In the world of DSL, PPPoE was commonly understood to be running on top of ATM (or DSL) as the underlying transport, although no such limitation exists in the PPPoE protocol itself. Other usage scenarios are sometimes distinguished by tacking as a suffix another underlying transport. For example, PPPoEoE, when the transport is Ethernet itself, as in the case of Metro Ethernet networks. (In this notation, the original use of PPPoE would be labeled PPPoEoA, although it should not be confused with PPPoA, which is a different encapsulation protocol.)

PPPoE has been described in some books as a "layer 2.5" protocol, in some rudimentary sense similar to MPLS because it can be used to distinguish different IP flows sharing an Ethernet infrastructure, although the lack of PPPoE switches making routing decision based on PPPoE headers limits applicability in that respect.

PPPoE stages

The PPPoE has two distinct stages:

PPPoE discovery

Since traditional PPP connections are established between two end points over a serial link or over an ATM virtual circuit that has already been established during dial-up, all PPP frames sent on the wire are sure to reach the other end. But Ethernet networks are multi-access where each node in the network can access every other node. An Ethernet frame contains the hardware address of the destination node (MAC address). This helps the frame reach the intended destination.

Hence before exchanging PPP control packets to establish the connection over Ethernet, the MAC address of the two end points should be known to each other so that they can be encoded in these control packets. The PPPoE Discovery stage does exactly this. In addition it also helps establish a Session ID that can be used for further exchange of packets.

PPP session

Once the MAC address of the peer is known and a session has been established, the Session stage will start.

Chapter 4 – MPLS

Overview

Multiprotocol Label Switching (MPLS), originating in IPv4, was initially proposed to improve forwarding speed. Its core technology can be extended to multiple network protocols, such as IPv6, Internet Packet Exchange (IPX), and Connectionless Network Protocol (CLNP). That is what the term multiprotocol means.

MPLS integrates both Layer 2 fast switching and Layer 3 routing and forwarding, satisfying the networking requirements of various new applications

MPLS Overview
Basic Concepts of MPLS

1. FEC

As a forwarding technology based on classification, MPLS groups packets to be forwarded in the same manner into a class called the forwarding equivalence class (FEC). That is, packets of the same FEC are handled in the same way.

The classification of FECs is very flexible. It can be based on any combination of source address, destination address, source port, destination port, protocol type and VPN. For example, in the traditional IP forwarding using longest match, all packets to the same destination belongs to the same FEC.

2. Label

A label is a short fixed length identifier for identifying a FEC. A FEC may correspond to multiple labels in scenarios where, for example, load sharing is required, while a label can only represent a single FEC.

A label is carried in the header of a packet. It does not contain any topology information and is local significant.

A label is four octets, or 32 bits, in length. **Figure 1** illustrates its format.

Figure 1, MPLS Label

A label consists of four fields:

Label: Label value of 20 bits. Used as the pointer for forwarding.

Exp: For QoS, three bits in length.

S: Flag for indicating whether the label is at the bottom of the label stack, one bit in length. 1 indicates that the label is at the bottom of the label stack. This field is very useful when there are multiple levels of MPLS labels.

TTL: Time to live (TTL) for the label. Eight bits in length. This field has the same meaning as that for an IP packet.

3. LSR

Label switching router (LSR) is a fundamental component on an MPLS network. All LSRs support MPLS.

4. LSP

Label switched path (LSP) means the path along which a FEC travels through an MPLS network. Along an LSP, two neighboring LSRs are called upstream LSR and downstream LSR respectively. In Figure 2, R2 is the downstream LSR of R1, while R1 is the upstream LSR of R2.

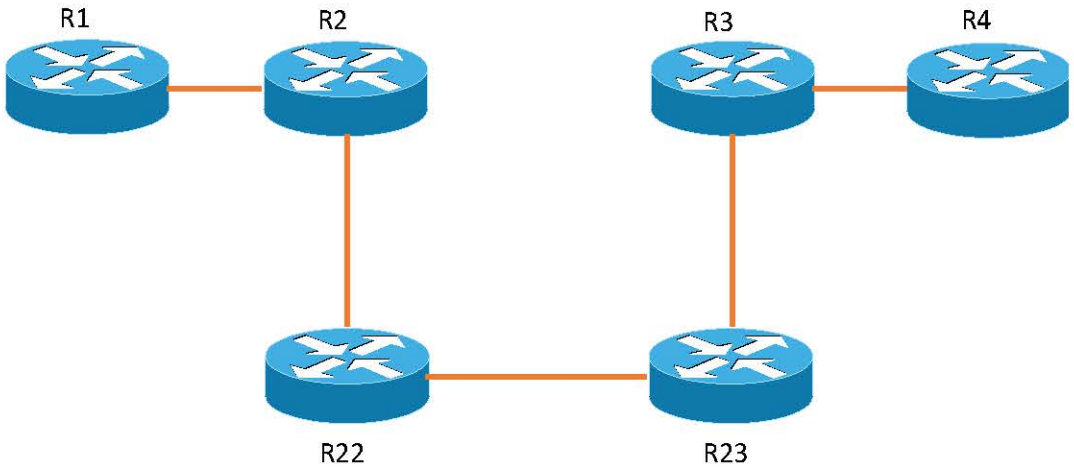


Figure 2, Diagram for an LSP

An LSP is a unidirectional path from the ingress of the MPLS network to the egress. It functions like a virtual circuit in ATM or frame relay. Each node of an LSP is an LSR.

5. LDP

Label Distribution Protocol (LDP) means the protocol used by MPLS for control. An LDP has the same functions as a signaling protocol on a traditional network. It classifies FECs, distributes labels, and establishes and maintains LSPs.

MPLS supports multiple label distribution protocols of either of the following two types:

Those dedicated for label distribution, such as LDP and Constraint-based Routing using LDP (CR-LDP). The existing protocols that are extended to support label distribution, such as Border Gateway Protocol (BGP) and Resource Reservation Protocol (RSVP).

In addition, you can configure static LSPs

Structure of the MPLS network

As shown in Figure 3, the element of an MPLS network is LSR. LSRs in the same routing or administrative domain form an MPLS domain.

In an MPLS domain, LSRs residing at the domain border to connect with other networks are label edge routers (LERs), while those within the MPLS domain are core LSRs. All core LSRs, which can be routers running MPLS or ATM-LSRs upgraded from

ATM switches, use MPLS to communicate, while LERs interact with devices outside the domain that use traditional IP technologies.

Each packet entering an MPLS network is labeled on the ingress LER and then forwarded along an LSP to the egress LER. All the intermediate LSRs are called transit LSRs.

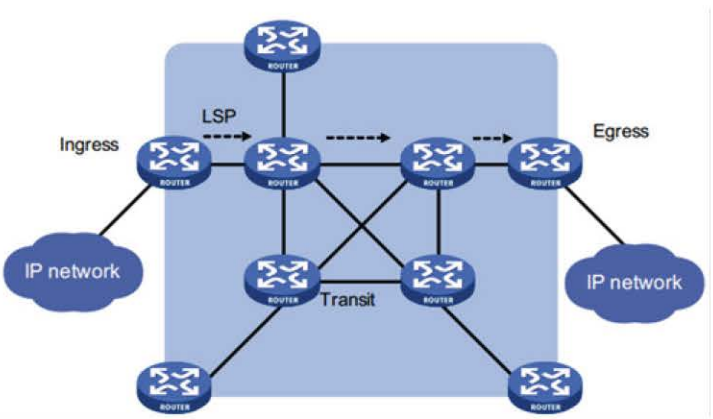
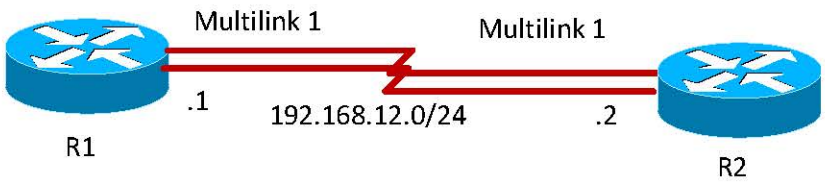


Figure 3, MPLS Network Topology

How MPLS Operates?

- First, the LDP protocol and the traditional routing protocol (such as OSPF and ISIS) work together on each LSR to establish the routing table and the label information base (LIB) for intended FECs.
- Upon receiving a packet, the ingress LER completes the Layer 3 functions, determines the FEC to which the packet belongs, labels the packet, and forwards the labeled packet to the next hop along the LSP.
- After receiving a packet, each transit LSR looks up its label forwarding table for the next hop according to the label of the packet and forwards the packet to the next hop. None of the transit LSRs performs Layer 3 processing.
- When the egress LER receives the packet, it removes the label from the packet and performs IP forwarding.
- Obviously, MPLS is not a service or application, but actually a tunneling technology and a routing and switching technology platform combining label switching with Layer 3 routing. This platform supports multiple upper layer protocols and services, as well as secures transmission of information to a certain degree.

Lab 1 – MLPPP on WAN interfaces using local authentication



Task 1

Change encapsulation to PPP on both R1 and R2.

Task 2

Configure R1's S0/0 and S0/1 in multilink group 1.

Task 3

Configure R2's S0/0 and S0/1 in multilink group 1.

Task 4

Interface configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	10.1.1.1	255.255.255.0
Multilink 1	192.1.12.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	10.2.2.2	255.255.255.0
Multilink 1	192.1.12.2	255.255.255.0

```
R1
interface Serial0/0
 encapsulation ppp
 ppp multilink
 ppp multilink group 1
 no shutdown
!
interface Serial0/1
 encapsulation ppp
 ppp multilink
 ppp multilink group 1
 no shutdown
!
interface Multilink1
 ip address 192.1.12.1 255.255.255.0
 ppp multilink
 ppp multilink group 1
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.0
!
```

```
R2
interface Serial0/0
 encapsulation ppp
 ppp multilink
 ppp multilink group 1
 no shutdown
!
interface Serial0/1
 encapsulation ppp
 ppp multilink
 ppp multilink group 1
 no shutdown
!
interface Multilink1
 ip address 192.1.12.2 255.255.255.0
 ppp multilink
 ppp multilink group 1
!
interface Loopback0
 ip address 10.2.2.2 255.255.255.0
!
```

Task 5

Configure RIP v2 as the routing protocol on R1 and R2. Make sure auto summary is disabled.

```
R1
!
router rip
 version 2
 network 10.0.0.0
 network 192.1.12.0
 no auto-summary
!
```

```
R2
router rip
 version 2
 network 10.0.0.0
 network 192.1.12.0
 no auto-summary
!
```

Task 6

Configure PPP Chap authentication on R1 and R2 using local username and password. Configure username R2 and password cisco on R1, and username R1 and password cisco on R2.

On R1

```
!
username R2 password cisco
interface Multilink 1
 ppp authentication chap
!
```

On R2

```
!
username R2 password cisco
interface Multilink 1
 ppp authentication chap
!
```

Task 7

Check RIP routes

On R1

```
R1#show ip route rip
     10.0.0.0/24 is subnetted, 2 subnets
R       10.2.2.0 [120/1] via 192.1.12.2, 00:00:09, Multilink1
```

Task 7

Use various show commands to check the output.

```
R1
R1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
Serial0/0              unassigned      YES NVRAM  up                    up
Serial0/1              unassigned      YES NVRAM  up                    up
Multilink1             192.1.12.1      YES manual up                    up
Loopback0              10.1.1.1        YES manual up                    up
```

```
R1#show interfaces multilink 1
Multilink1 is up, line protocol is up
  Hardware is multilink group interface
  Internet address is 192.1.12.1/24
  MTU 1500 bytes, BW 3088 Kbit/sec, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open, multilink Open
. . . . .
```

```
R1#show ppp multilink
Multilink1
  Bundle name: R2
  Remote Username: R2
  Remote Endpoint Discriminator: [1] R2
  Local Username: R1
  Local Endpoint Discriminator: [1] R1
  Bundle up for 00:08:02, total bandwidth 3088, load 1/255
  Receive buffer limit 24000 bytes, frag timeout 1000 ms
    0/0 fragments/bytes in reassembly list
    0 lost fragments, 8 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x29 received sequence, 0x2A sent sequence
  Member links: 2 active, 0 inactive (max not set, min not set)
    Se0/0, since 00:08:03
    Se0/1, since 00:08:03
No inactive multilink interfaces
```

Task 8

Verify connectivity by pinging R2's loopback from R1's loopback.

On R1

```
R1#ping 10.2.2.2 source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/16/32 ms
```

Are you successful?

Lab 2 – PPPoE client-side and server-side interfaces using local

R1 as PPPoE Client

Task 1

Configure R1 as PPPoE Client. Enables a PPPoE session on the FastEthernet interface.

Task 2

Configures a PPPoE client and specifies dial-on-demand routing (DDR) functionality.

Task 3

Defines a dialer rotary group and set the maximum transmission unit (MTU) size 1492.

Task 4

Specify that the IP address for a particular interface is obtained via PPP/IP Control Protocol (IPCP) address negotiation.

Task 5

Set PPP as the encapsulation type.

Task 6

Specify the dialing pool that the dialer interface uses to connect to a specific destination subnetwork.

Task 7

Enable remote Password Authentication Protocol (PAP) support for an interface and uses the username and password parameters in the PAP authentication packet to the peer.

```
R1
enable
configure terminal
interface FastEthernet 0/0
 no ip address
 pppoe enable
 pppoe-client dial-pool-number 1
!
interface dialer 1
 mtu 1492
 ip address negotiated
 encapsulation ppp
 dialer pool 1
 ppp pap sent-username user1 password cisco1
!
```

R2 as PPPoE Server

Task 1

Creates a PPPoE profile by creating BBA group 1.

Task 2

Creates a virtual template for a PPPoE profile with an identifying number 1 to be used for cloning virtual access interfaces.

Task 3

Creates a loopback 1 interface and assign it an IP address 192.168.1.2/24.

Task 4

Enable PPPoE session on the FastEthernet0/0 interface by assigning it BBA1.

Task 5

Create a virtual template interface 1 that can be configured and applied dynamically to create virtual access interfaces. Sets the MTU size 1492.

Task 6

Enable IP processing on an interface without explicitly assigning an IP address to the interface (ip unnumbered).

Task 7

Specify an IP address from a specific IP address pool "pool1" to be returned to a remote peer connecting to this interface.

Task 8

Enable PPP PAP authentication protocol.

Task 9

Configure local username **"user1"** and password **"cisco1"** for

Task 10

Configure a local pool **"pool1"** of IP addresses to be used when a remote peer connects to a point-to-point interface.

```
enable
configure terminal
!
bba-group pppoe bba1
 virtual-template 1
!
interface Loopback1
 ip address 192.2.0.2 255.255.255.0
!
interface FastEthernet0/0
 no ip address
 pppoe enable group bba1
!
interface Virtual-Template1
 description pppoe bba1
 mtu 1492
 ip unnumbered Loopback1
 peer default ip address pool pool1
 ppp authentication pap
!
ip local pool pool1 192.168.1.10 192.168.1.100
!
```

Task 11

Use the following verification commands to verify results.

On R1

```
R1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        unassigned      YES unset  up                    up
Virtual-Access1        unassigned      YES unset  up                    up
Virtual-Access2        unassigned      YES unset  up                    up
Dialer1                192.168.1.10    YES IPCP   up                    up
```

```
R1#show ip route connected
     192.2.0.0/32 is subnetted, 2 subnets
C       192.2.0.2 is directly connected, Dialer1
C       192.2.0.10 is directly connected, Dialer1
```

```
R1#show pppoe session
     1 client session
Uniq ID  PPPoE  RemMAC         Port            VT  VA        State
         SID  LocMAC                          VA-st
   N/A      1  c201.12c0.0000  Fa0/0           Di1 Vi2       UP
             c200.12c0.0000                   UP
```

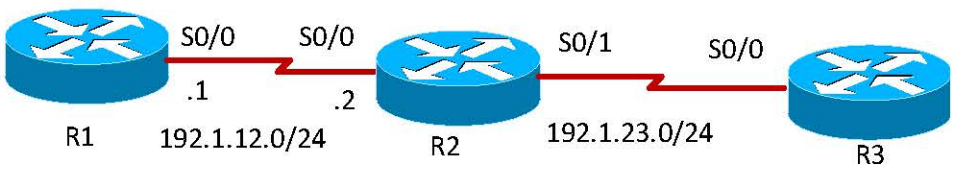
On R2

```
R2#show ip interface brief
Interface          IP-Address      OK? Method Status       Protocol
FastEthernet0/0    unassigned      YES manual up           up
Virtual-Access1    unassigned      YES unset  up           up
Virtual-Access1.1  192.168.1.2     YES TFTP   up           up
Virtual-Template1  192.168.1.2     YES TFTP   down         down
Virtual-Access2    unassigned      YES unset  down         down
Loopback1          192.168.1.2     YES manual up           up
```

```
R2#show ip route connected
    192.2.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.2.0.0/24 is directly connected, Loopback1
C       192.2.0.10/32 is directly connected, Virtual-Access1.1
```

```
R2#show pppoe session
    1 session  in LOCALLY_TERMINATED (PTA) State
    1 session  total
Uniq ID  PPPoE  RemMAC        Port          VT  VA     State
         SID  LocMAC                        VA-st
    1     1  c200.12c0.0000  Fa0/0          1  Vi1.1   PTA
             c201.12c0.0000                    UP
```

Lab 3 – Point-to-Point GRE



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	10.1.1.1	255.255.255.0
S 0/0	192.1.12.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
S 0/0	192.1.12.2	255.255.255.0
S 0/1	192.1.23.2	255.255.255.0

R3

Interface	IP Address	Subnet Mask
Loopback 0	10.3.3.3	255.255.255.0
S 0/0	192.1.23.3	255.255.255.0

Task 1

Configure a Default Route on R1 & R3 towards R2, the ISP Router.

R1

```
!
ip route
0.0.0.0
0.0.0.0
192.1.12.2
!
```

R3

```
!
ip route
0.0.0.0
0.0.0.0
192.1.23.2
!
```

Task 2

Configure a Point-to-Point GRE tunnel between R1 and R3. Use 192.168.13.0/24 as the Tunnel Network IP.

R1

```
!
interface tunnel 1
 ip add 192.168.13.1 255.255.255.0
 tunnel source 192.1.12.1
 tunnel destination 192.1.23.3
!
```

R3

```
!
interface tunnel 1
 ip add 192.168.13.3 255.255.255.0
 tunnel source 192.1.23.3
 tunnel destination 192.1.12.1
!
```

Task 3

Configure EIGRP in AS 13 to route the internal networks (Loopbacks) on the GRE Tunnel between R1 and

R3.

```
R1
!
router eigrp 13
 no auto-summary
 network 192.168.13.0
 network 10.0.0.0
!
```

```
R3
!
router eigrp 13
 no auto-summary
 network 192.168.13.0
 network 10.0.0.0
!
```

Task 4

Use the following verification commands to check the configuration

```
R1#show ip interface brief
Interface              IP-Address      OK? Method Status            Protocol
Serial0/0              192.1.12.1      YES manual up                up
Loopback0              10.1.1.1        YES manual up                up
Tunnel1                192.168.13.1    YES manual up                up
```

```
R1#show ip route eigrp
     10.0.0.0/24 is subnetted, 2 subnets
D       10.3.3.0 [90/297372416] via 192.168.13.3, 00:03:09, Tunnel1
```

```
R1#ping 10.3.3.3 source 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/24/48 ms
```

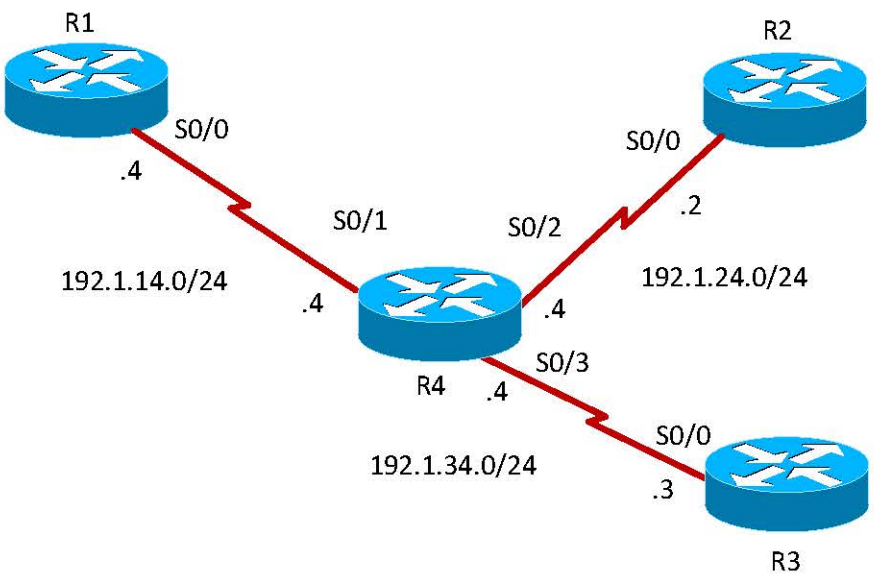
On R3

```
R3#show ip interface brief
Interface              IP-Address      OK? Method Status            Protocol
Serial0/0              192.1.23.3      YES manual up                up
Loopback0              10.1.1.1        YES manual up                up
Tunnel1                192.168.13.3    YES manual up                up
```

```
R3#show ip route eigrp
     10.0.0.0/24 is subnetted, 2 subnets
D       10.1.1.0 [90/297372416] via 192.168.13.1, 00:03:09, Tunnel1
```

```
R3#ping 10.1.1.1 source 10.3.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.3.3.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/24/48 ms
```

Lab 4 – Multipoint GRE



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	10.1.1.1	255.255.255.0
S 0/0	192.1.15.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	10.2.2.2	255.255.255.0
S 0/0	192.1.25.2	255.255.255.0

R3

Interface	IP Address	Subnet Mask
Loopback 0	10.3.3.3	255.255.255.0
S 0/0	192.1.35.3	255.255.255.0

R4

Interface	IP Address	Subnet Mask
S 0/0	192.1.14.4	255.255.255.0
S 0/1	192.1.24.4	255.255.255.0
S 0/2	192.1.34.4	255.255.255.0

Task 1

Configure a Default Route on R1, R2, R3 & R4 towards R5, the ISP Router.

R1
ip route 0.0.0.0 0.0.0.0 192.1.14.5
R2
ip route 0.0.0.0 0.0.0.0 192.1.24.5
R3
ip route 0.0.0.0 0.0.0.0 192.1.34.5

Task 2

Configure a MultiPoint GRE tunnel between R1, R2, R3 Use 192.168.1.0/24 as the Tunnel Network IP. Use the following parameters for your MGRE Tunnel:

- **NHRP Parameters**
- NHRP ID – 1234
- NHRP Hub – R1

```
R1
interface tunnel 1
 ip address 192.168.1.1 255.255.255.0
 ip nhrp network-id 1234
 tunnel source s 0/0
 tunnel mode gre multipoint
!
```

```
R2
interface tunnel 1
 ip address 192.168.1.2 255.255.255.0
 ip nhrp network-id 1234
 ip nhrp nhs 192.168.1.1
 ip nhrp map 192.168.1.1 192.1.14.1
 tunnel source s 0/0
 tunnel mode gre multipoint
!
```

```
R3
interface tunnel 1
 ip address 192.168.1.3 255.255.255.0
 ip nhrp network-id 1234
 ip nhrp nhs 192.168.1.1
 ip nhrp map 192.168.1.1 192.1.14.1
 tunnel source s 0/0
 tunnel mode gre multipoint
!
```

Task 3

Configure EIGRP in AS 1234 to route the internal networks (Loopbacks) on the GRE Tunnel on all the MGRE Routers. Configure the Multicast mapping on all the Routers. Disable Split horizon on R1 to allow it propagate routes from the Spoke routers to the other spoke routers.

Note: You might need to bounce the Tunnel interface to make the Routing work. Bring up the Hub router before the Spoke Routers.

```
R1
!
interface tunnel 1
 ip nhrp map multicast dynamic
 no ip split-horizon eigrp 1234
!
router eigrp 1234
 no auto-summary
 network 192.168.1.0
 network 10.0.0.0
!
```

```
R2
!
interface tunnel 1
 ip nhrp map multicast 192.1.14.1
!
router eigrp 1234
 no auto-summary
 network 192.168.1.0
 network 10.0.0.0
!
```

```
R3
!
interface tunnel 1
 ip nhrp map multicast 192.1.14.1
!
router eigrp 1234
 no auto-summary
 network 192.168.1.0
 network 10.0.0.0
!
```

Task 4

As EIGRP changes the Next-hop address in its routing update, all traffic will be routed thru the hub. This might create a bottleneck in the network. Configure EIGRP such that it does not change the next-hop attribute when it sends the routing update for the Spoke network to the other spoke networks. This will allow the spoke to communicate directly to each other over the Internet.

```
R1
!
interface tunnel 1
 no ip next-hop-self eigrp 1234
!
router eigrp 1234
 no auto-summary
 network 192.168.1.0
 network 10.0.0.0
!
```

Task 5

Verify your configuration by show commands

```
R1#show ip nhrp
192.168.1.2/32 via 192.168.1.2, Tunnel1 created 00:01:49, expire 01:58:10
  Type: dynamic, Flags: unique registered
  NBMA address: 192.1.24.2
192.168.1.3/32 via 192.168.1.3, Tunnel1 created 00:01:44, expire 01:58:15
  Type: dynamic, Flags: unique registered
  NBMA address: 192.1.34.3
```

```
R2#show ip nhrp
192.168.1.1/32 via 192.168.1.1, Tunnel1 created 00:07:47, never expire
  Type: static, Flags: used
  NBMA address: 192.1.14.1
```

```
R3#show ip nhrp
192.168.1.1/32 via 192.168.1.1, Tunnel1 created 00:08:39, never expire
  Type: static, Flags: used
  NBMA address: 192.1.14.1
```

```
R1#show ip route eigrp
     10.0.0.0/24 is subnetted, 3 subnets
D       10.3.3.0 [90/297372416] via 192.168.1.3, 00:08:07, Tunnel1
D       10.2.2.0 [90/297372416] via 192.168.1.2, 00:08:29, Tunnel1
```

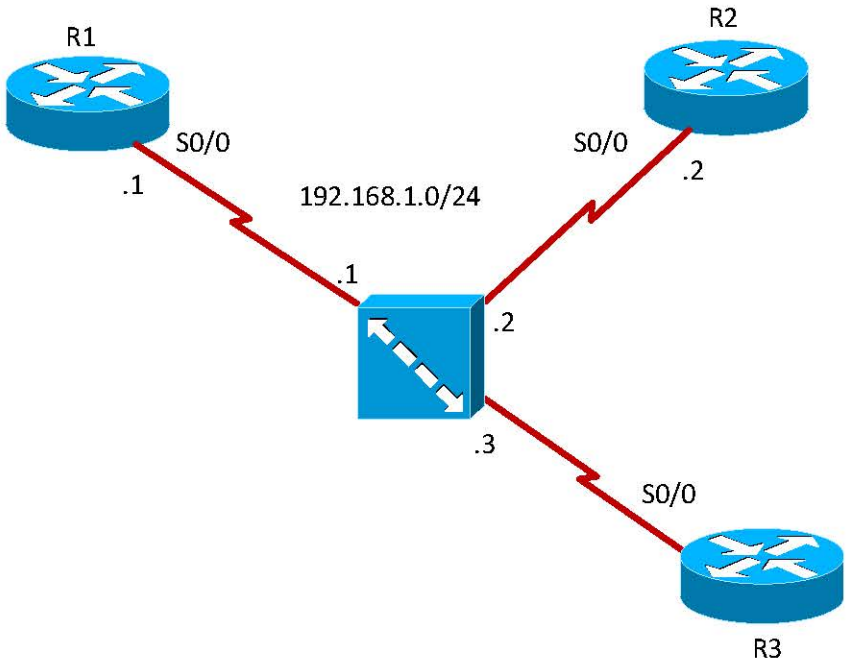
```
R2#show ip route eigrp
     10.0.0.0/24 is subnetted, 3 subnets
D       10.3.3.0 [90/310172416] via 192.168.1.1, 00:08:30, Tunnel1
D       10.1.1.0 [90/297372416] via 192.168.1.1, 00:09:08, Tunnel1
```

```
R3#show ip route eigrp
     10.0.0.0/24 is subnetted, 3 subnets
D       10.2.2.0 [90/310172416] via 192.168.1.1, 00:09:52, Tunnel1
D       10.1.1.0 [90/297372416] via 192.168.1.1, 00:10:06, Tunnel1
```

```
R2#ping 10.3.3.3 source 10.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 10.2.2.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/72/108 ms
```

```
R2#traceroute 10.3.3.3 source 10.2.2.2
Type escape sequence to abort.
Tracing the route to 10.3.3.3
  1 192.168.1.1 72 msec 4 msec 84 msec
  2 192.168.1.3 80 msec 72 msec 76 msec
```

Lab 5 – Frame Relay Hub and Spoke using Inverse ARP



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	10.1.1.1	255.255.255.0
S 0/0	192.168.1.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	10.2.2.2	255.255.255.0
S 0/0	192.168.1.2	255.255.255.0

R3

Interface	IP Address	Subnet Mask
Loopback 0	10.3.3.3	255.255.255.0
S 0/0	192.168.1.3	255.255.255.0

Task 1

Configure a frame relay switch on GNS3 to give out following DLCI

Source	Port	DLCI
	1	102
	1	103
Destination	Port	DLCI
	2	201
	3	301

Task 2

Configure encapsulation frame relay on interface serial 0/0 on R1, R2 and R3.

```
R1 (HUB)
!
configure terminal
interface serial 0/0
 ip address 192.168.1.1 255.255.255.0
 encapsulation frame-relay
 no shutdown
!
inter loopback 0
 ip add 10.1.1.1 255.255.255.0
!
```

```
R2
!
configure terminal
interface serial 0/0
 ip address 192.168.1.2 255.255.255.0
 encapsulation frame-relay
 no shutdown
!
inter loopback 0
 ip add 10.2.2.2 255.255.255.0
!
```

```
R3
!
configure terminal
interface serial 0/0
 ip address 192.168.1.3 255.255.255.0
 encapsulation frame-relay
no shutdown
!
inter loopback 0
 ip add 10.3.3.3 255.255.255.0
!
```

Task 3

Check frame relay map on all routers.

```
R1#show frame-relay map
Serial0/0 (up): ip 192.168.1.2 dlci 102(0x66,0x1860), dynamic,
        broadcast,, status defined, active
Serial0/0 (up): ip 192.168.1.3 dlci 103(0x67,0x1870), dynamic,
        broadcast,, status defined, active
R2#show frame-relay map
Serial0/0 (up): ip 192.168.1.1 dlci 201(0xC9,0x3090), dynamic,
        broadcast,, status defined, active
R3#show frame-relay map
Serial0/0 (up): ip 192.168.1.1 dlci 301(0x12D,0x48D0), dynamic,
        broadcast,, status defined, active
```

How many mappings do you have at the Hub?

How many mappings do you have at the two spokes?

Task 4

Ping from one spoke to the other.

```
R3#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Are you successful?

Task 5

On the Spokes put in the Frame-relay map statements.

```
R2
!
interface serial 0/0
 frame-relay map ip 192.168.1.3 201
!
R3
!
interface serial 0/0
 frame-relay map ip 192.168.1.2 301
!
```

Task 6

Configure RIP v2 on all interface of R1, R2 and R3.

```
R1
!
router rip
version 2
 network 10.0.0.0
 network 192.168.1.0
 no auto-summary
!
```

```
R2
!
router rip
 version 2
 network 10.0.0.0
 network 192.168.1.0
 no auto-summary
!
```

```
R3
!
router rip
 version 2
 network 10.0.0.0
 network 192.168.1.0
 no auto-sutmmary
!
```

Task 7

Check routing table on routers.

Can you see all the route on Spokes?

```
R1#show ip route rip
    10.0.0.0/24 is subnetted, 3 subnets
R       10.3.3.0 [120/1] via 192.168.1.3, 00:00:04, Serial0/0
R       10.2.2.0 [120/1] via 192.168.1.2, 00:00:19, Serial0/0
```

```
R2#show ip route rip
    10.0.0.0/24 is subnetted, 3 subnets
R       10.3.3.0 [120/2] via 192.168.1.3, 00:00:05, Serial0/0
R       10.1.1.0 [120/1] via 192.168.1.1, 00:00:05, Serial0/0
R3#show ip route rip
    10.0.0.0/24 is subnetted, 3 subnets
R       10.2.2.0 [120/2] via 192.168.1.2, 00:00:24, Serial0/0
R       10.1.1.0 [120/1] via 192.168.1.1, 00:00:24, Serial0/0
```

What happened to Split-horizon of RIP?

```
R1#show ip interface serial 0/0
Serial0/0 is up, line protocol is up
  Internet address is 192.168.1.1/24
  Broadcast address is 255.255.255.255
  . . .
  Split horizon is disabled
  . . .
```

It is disabled by default for RIP for frame relay interfaces.

Task 8

Ping for R2's loopback 0 to R3's loopback 0

```
R2#ping 10.3.3.3 source 10.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 10.2.2.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/24/52 ms
```

Lab 6 – Frame Relay Full Mesh using Inverse ARP

(Build on Lab 5)

Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	10.1.1.1	255.255.255.0
S 0/0	192.168.1.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	10.2.2.2	255.255.255.0
S 0/0	192.168.1.2	255.255.255.0

R3

Interface	IP Address	Subnet Mask
Loopback 0	10.3.3.3	255.255.255.0
S 0/0	192.168.1.3	255.255.255.0

Task 1

Configure a frame relay switch on GNS3 to give out following DLCI

Source	Port	DLCI
	1	102
	1	103
Destination	**Port**	**DLCI**
	2	201
	3	301

Source	Port	DLCI
	2	203
	3	302

Note: Reverse DLCI on Port 3 will be automatically maped.

Task 2

Configure encapsulation frame relay on interface serial 0/0 on R1, R2 and R3.

```
On R1 (HUB)
!
configure terminal
interface serial 0/0
 ip address 192.168.1.1 255.255.255.0
 encapsulation frame-relay
 no shutdown
!
inter loopback 0
 ip add 10.1.1.1 255.255.255.0
!
```

```
On R2
!
configure terminal
interface serial 0/0
 ip address 192.168.1.2 255.255.255.0
 encapsulation frame-relay
 no shutdown
!
inter loopback 0
 ip add 10.2.2.2 255.255.255.0
!
```

```
On R3
!
configure terminal
interface serial 0/0
 ip address 192.168.1.3 255.255.255.0
 encapsulation frame-relay
no shutdown
!
inter loopback 0
 ip add 10.3.3.3 255.255.255.0
!
```

Task 3

Check frame relay map on all routers.

```
R1#show frame-relay map
Serial0/0 (up): ip 192.168.1.2 dlci 102(0x66,0x1860), dynamic,
            broadcast,, status defined, active
Serial0/0 (up): ip 192.168.1.3 dlci 103(0x67,0x1870), dynamic,
            broadcast,, status defined, active
R2#show frame-relay map
Serial0/0 (up): ip 192.168.1.1 dlci 201(0xC9,0x3090), dynamic,
            broadcast,, status defined, active
Serial0/0 (up): ip 192.168.1.3 dlci 203(0xCB,0x30B0), dynamic,
            broadcast,, status defined, active
R3#show frame-relay map
Serial0/0 (up): ip 192.168.1.1 dlci 301(0x12D,0x48D0), dynamic,
            broadcast,, status defined, active
Serial0/0 (up): ip 192.168.1.2 dlci 302(0x12E,0x48E0), dynamic,
            broadcast,, status defined, active
```

How many mappings do you have at the Hub?

How many mappings do you have at the two spokes?

Task 4

Ping from one spoke to the other.

```
R3#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 0 percent (5/5)
```

Are you successful?

Task 6

Configure RIP v2 on all interface of R1, R2 and R3.

```
On R1
!
router rip
version 2
 network 10.0.0.0
 network 192.168.1.0
 no auto-summary
!
```

```
On R2
!
router rip
 version 2
 network 10.0.0.0
 network 192.168.1.0
 no auto-summary
!
```

```
On R3
!
router rip
 version 2
 network 10.0.0.0
 network 192.168.1.0
 no auto-summary
!
```

Task 7

Check routing table on routers.

Can you see all the route on Spokes?

```
R1#show ip route rip
    10.0.0.0/24 is subnetted, 3 subnets
R       10.3.3.0 [120/1] via 192.168.1.3, 00:00:04, Serial0/0
R       10.2.2.0 [120/1] via 192.168.1.2, 00:00:19, Serial0/0
```

```
R2#show ip route rip
    10.0.0.0/24 is subnetted, 3 subnets
R       10.3.3.0 [120/2] via 192.168.1.3, 00:00:05, Serial0/0
R       10.1.1.0 [120/1] via 192.168.1.1, 00:00:05, Serial0/0
```

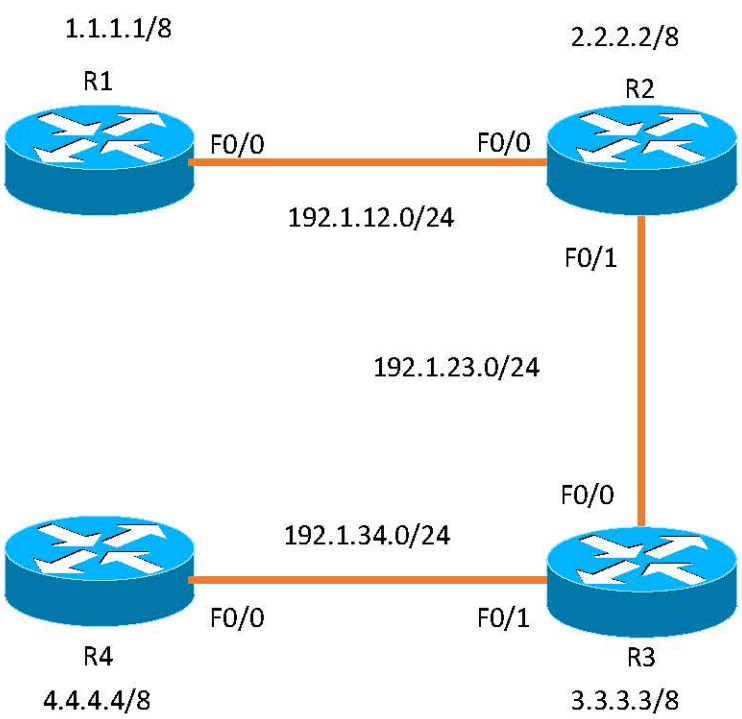
```
R3#show ip route rip
    10.0.0.0/24 is subnetted, 3 subnets
R       10.2.2.0 [120/2] via 192.168.1.2, 00:00:24, Serial0/0
R       10.1.1.0 [120/1] via 192.168.1.1, 00:00:24, Serial0/0
```

Task 8

Ping for R2's loopback 0 to R3's loopback 0

```
R2#ping 10.3.3.3 source 10.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 10.2.2.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/24/52 ms
```

Lab 6 – Configuring MPLS Unicast Routing



Interface configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	1.1.1.1	255.0.0.0
F 0/0	192.1.12.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
Loopback 0	2.2.2.2	255.0.0.0
F 0/0	192.1.12.2	255.255.255.0
F 0/1	192.1.23.2	255.255.255.0

R3

Interface	IP Address	Subnet Mask
Loopback 0	3.3.3.3	255.0.0.0
F 0/0	192.1.23.3	255.255.255.0
F 0/1	192.1.34.3	255.255.255.0

R4

Interface	IP Address	Subnet Mask
Loopback 0	4.4.4.4	255.0.0.0
F 0/0	192.1.34.4	255.255.255.0

Task 1

Configure OSPF between all the SP routers (R1, R2, R3, and R4). OSPF process should use Loopback0 on each router as the router -id. Advertise all links in OSPF except Loopback 1's on R1 and R4. Loopback 0's should appear with a /8 mask in the routing table.

```
R1
!
interface loopback 0
ip ospf network point-to-point
!
router ospf 1
router-id 1.1.1.1
network 1.1.1.1 0.0.0.0 area 0
network 192.1.12.1 0.0.0.0 area 0
!
```

```
R2
!
interface loopback 0
ip ospf network point-to-point
!
router ospf 1
router-id 2.2.2.2
network 2.2.2.2 0.0.0.0 area 0
network 192.1.12.2 0.0.0.0 area 0
network 192.1.23.2 0.0.0.0 area 0
!
```

```
R3
!
interface loopback 0
ip ospf network point-to-point
!
router ospf 1
router-id 3.3.3.3
network 3.3.3.3 0.0.0.0 area 0
network 192.1.23.3 0.0.0.0 area 0
network 192.1.34.3 0.0.0.0 area 0
!
```

```
R4
!
interface loopback 0
ip ospf network point-to-point
!
router ospf 1
router-id 4.4.4.4
network 4.4.4.4 0.0.0.0 area 0
network 192.1.34.4 0.0.0.0 area 0
!
```

Task 2

Configure MPLS on all the physical links in the SP Network. Use LDP to distribute labels. The LDP neighbour relationships should be formed based on the most reliable interface. The Labels should be assigned from the range X00 –X99, where X is the router number.

R1
```
!
mpls ldp router-id loopback0
mpls label protocol ldp
mpls label range 100 199
!
interface f 0/0
mpls ip
!
```

R2
```
!
mpls ldp router-id loopback0
mpls label protocol ldp
mpls label range 200 299
!
interface f 0/0
mpls ip
!
interface f 0/1
mpls ip
!
```

R3
```
!
mpls ldp router-id loopback0
mpls label protocol ldp
!
mpls label range 300 399
!
interface f 0/0
mpls ip
!
interface f 0/1
mpls ip
```

R4
```
!
mpls ldp router-id loopback0
mpls label protocol ldp
!
mpls label range 400 499
!
interface f 0/0
mpls ip
!
```

Task 3

Check Lable Information Base (LIB).

```
R1#show mpls ldp bindings
  tib entry: 1.0.0.0/8, rev 4
        local binding:  tag: imp-null
        remote binding: tsr: 2.2.2.2:0, tag: 200
  tib entry: 2.0.0.0/8, rev 6
        local binding:  tag: 100
        remote binding: tsr: 2.2.2.2:0, tag: imp-null
  tib entry: 3.0.0.0/8, rev 8
        local binding:  tag: 101
        remote binding: tsr: 2.2.2.2:0, tag: 201
  tib entry: 4.0.0.0/8, rev 10
        local binding:  tag: 102
        remote binding: tsr: 2.2.2.2:0, tag: 202
  tib entry: 192.1.12.0/24, rev 2
        local binding:  tag: imp-null
        remote binding: tsr: 2.2.2.2:0, tag: imp-null
  tib entry: 192.1.23.0/24, rev 12
        local binding:  tag: 103
        remote binding: tsr: 2.2.2.2:0, tag: imp-null
  tib entry: 192.1.34.0/24, rev 14
        local binding:  tag: 104
        remote binding: tsr: 2.2.2.2:0, tag: 203
```

Task 4

Check Lable Forwarding Information Base (LFIB).

```
R1#show mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
100    Pop tag     2.2.2.0/24       0          Fa0/0       192.1.12.2
101    201         3.3.3.0/24       0          Fa0/0       192.1.12.2
102    202         4.4.4.0/24       0          Fa0/0       192.1.12.2
103    Pop tag     192.1.23.0/24    0          Fa0/0       192.1.12.2
104    203         192.1.34.0/24    0          Fa0/0       192.1.12.2
```

```
R2#show mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
200    Pop tag     1.1.1.0/24       0          Fa0/0       192.1.12.1
201    Pop tag     3.3.3.0/24       0          Fa0/1       192.1.23.3
202    303         4.4.4.0/24       0          Fa0/1       192.1.23.3
203    Pop tag     192.1.34.0/24    0          Fa0/1       192.1.23.3
```

```
R3#show mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
300    Pop tag     192.1.12.0/24    0          Fa0/0       192.1.23.2
301    200         1.1.1.0/24       0          Fa0/0       192.1.23.2
302    Pop tag     2.2.2.0/24       0          Fa0/0       192.1.23.2
303    Pop tag     4.4.4.0/24       0          Fa0/1       192.1.34.4
```

```
R4#show mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
400    300         192.1.12.0/24    0          Fa0/0       192.1.34.3
401    301         1.1.1.0/24       0          Fa0/0       192.1.34.3
402    302         2.2.2.0/24       0          Fa0/0       192.1.34.3
403    Pop tag     3.3.3.0/24       0          Fa0/0       192.1.34.3
404    Pop tag     192.1.23.0/24    0          Fa0/0       192.1.34.3
```

Task 5

Trace route from R1's loopback 0 to R4's loopback 0. Can you see the label path?

```
R1#traceroute 4.4.4.4 source 1.1.1.1
Type escape sequence to abort.
Tracing the route to 4.4.4.4
 1 192.1.12.2 [MPLS: Label 202 Exp 0] 120 msec 128 msec 92 msec
 2 192.1.23.3 [MPLS: Label 303 Exp 0] 124 msec 128 msec 108 msec
 3 192.1.34.4 156 msec 128 msec 104 msec
```

Task 6

Check MPLS LDP Neighbor. What TCP port no. LDP uses?

```
R1#show mpls ldp neighbor
   Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0
      TCP connection: 2.2.2.2.23081 - 1.1.1.1.646
      State: Oper; Msgs sent/rcvd: 16/16; Downstream
      Up time: 00:05:26
      LDP discovery sources:
        FastEthernet0/0, Src IP addr: 192.1.12.2
      Addresses bound to peer LDP Ident:
        192.1.12.2      192.1.23.2      2.2.2.2
```

MODULE 13

INTERNET VPNS

Chapter 1 – IPSec VPNs

Virtual Private Networks (VPNs)

A Virtual Private Network (VPN) provides a secure tunnel across a public (and thus, insecure) network. This provides a mechanism for organizations to connect users and offices together, without the high costs of dedicated leased lines. VPNs are most often used across the Internet, the world's largest public network, providing users with access to email, documents, printers, and systems as if they were actually at their central office. VPNs are generally used for two purposes:

Client VPNs - connect home or "roaming" users to an office.

Site-to-Site VPNs - connect remote offices to a main office.

What is IPSEC?

IPSEC, short for IP Security, is a suite of protocols, standards, and algorithms to secure traffic over an untrusted network, such as the Internet. IPSEC is supported on both Cisco IOS devices and PIX Firewalls. IPSEC provides three core services:

Confidentiality – prevents the theft of data, using encryption.

Integrity – ensures that data is not tampered or altered, using a hashing algorithm.

Authentication – confirms the identity of the host sending data, using pre-shared keys or a Certificate Authority (CA).

Anti-replay – prevents duplication of encrypted packets, by assigning a unique sequencing number. The IPSEC standard is outlined in RFC 2401

Confidentiality and Encryption

Data sent in clear-text across the Internet can easily be intercepted and stolen. Because of this, sensitive data should be encrypted when sent across an untrusted network or domain.

Keys are generated values used to both encrypt and decrypt data. The longer the key, the more secure

that key is. The length of a key is measured in bits. Two "types" of keys exist: Symmetric keys can be used to both encrypt and decrypt data. More specifically, the same key is used to both encrypt a packet (at the sending device) and then decrypt that packet (at the receiving device). Symmetric key encryption is efficient, but does not scale well in large environments.

Symmetric keys are not openly shared during data transmit, and must instead be installed on each machine prior to the transfer of data. This can be accomplished using a variety of (inefficient and insecure) methods: email, sneaker-net, and even snail-mail. Each device on a network would require every other device's symmetric key, and thus the lack of scalability.

Asymmetric keys require a separate key for encryption (the public key) and decryption (the private key). Public keys are openly exchanged between devices to encrypt data during transfer. Private keys are never exchanged.

Consider the above diagram. Assume we are using a public/private key infrastructure:

- Both Router A and Router B have their own unique **private** key.
- Both Router A and Router B exchange unique **public** keys.
- When Router B encrypts data destined for Router A, it uses Router A's **public** key. (and vice versa)
- Router A decrypts the data using its private key.

Only the private keys can decrypt the data. Thus, even if the data and the **public** key were intercepted, confidentiality is ensured.

Diffie-Hellman (D-H) Public Key Exchange is the most common standard used to create and exchange keys across insecure mediums. D-H is not used to encrypt data, but rather to generate the keys that are used to encrypt and decrypt data.

A variety of popular standards and protocols utilize D-H key exchange, including SSL (Secure Socket Layer), SSH (Secure Shell), and IPSEC.

The generated public keys encrypt data payload using one of several available encryption algorithms:

- **DES (Data Encryption Standard) – 56-bit key**
- **3DES (Triple Data Encryption Standard) – 168-bit key**
- **AES (Advanced Encryption Standard) - 128, 192, or 256-bit key**
- **Blowfish – up to a 448-bit key**

Additionally, the strength of a key is determined by the D-H group used to generate that key. There are several D-H groups:

- **Group 1 – 768 bits**
- **Group 2 – 1024 bits**
- **Group 5 – 2048 bits**

Data Integrity and Hashing

Data sent across the Internet can not only be stolen, but can also be maliciously altered. To combat this, a hashing algorithm computes and appends a specific hash value as each packet is sent. Once the data is received, it is run through the hashing algorithm again. If the hash value is different, the packet was altered in transit. Hashed Message Authentication Code (HMAC) is used to perform this hashing function. HMAC utilizes a secret key when computing the hash value, thus preventing an attacker from altering the packet and then recomputing the correct hash. Two HMAC algorithms are commonly used:

- **HMAC-MD5 (Message-Digest 5) – 128-bit hashed key**
- **HMAC-SHA1 (Secure Hash Algorithm) – 160-bit hashed key**

Authentication

Another concern when sending data across the Internet is the source or origin of that data. It is possible to masquerade or spoof one's identity or address. For an IPSEC VPN tunnel to be established, both sides

of the tunnel must be authenticated. To accomplish this, either **pre-shared** keys or **RSA digital signatures** are used. When using pre-shared keys, a secret string of text is used on each device to authenticate each other. This string must be pre-agreed upon and identical on each device. This string is then hashed into a digital signature.

When using **RSA Digital signatures**, a **Certificate Authority (CA)** is used to apply a verified digital signature. One of the above options must be correctly configured before the VPN tunnel will become active.

The IPSEC Protocols

IPSEC uses one of two protocol headers for securing data:

- **Authentication Header (AH)**
- **Encapsulation Security Payload (ESP)**

Authentication Header (AH), or IP protocol 51, provides no confidentiality of data. It does not encrypt any data at all. However, AH provides both authentication and integrity services. Because AH does not perform encryption, it is a quicker standard than ESP.

AH uses a hash algorithm to compute a hash value on both the payload and header of a packet, ensuring integrity of the packet. However, this causes a very specific problem. AH will not work through a NATed device.

NAT changes the IP header of a packet during translation, but the hash value is not changed. Thus, the receiving device will believe the packet has been altered in transit, and reject the packet.

Encapsulation Security Payload (ESP), or IP protocol 50, performs confidentiality, authentication, and integrity services. Thus, ESP does perform encryption, and is inherently more secure than AH.

ESP introduces both an additional header and trailer to a packet. ESP also uses a hash algorithm for data integrity. However, the hash does not include the IP header of the packet, and thus ESP will (usually) work through a NATed device.

ESP and AH can be used separately, or used in conjunction with each other.

Transport vs. Tunnel Modes

Each IPSEC protocol (AH or ESP) can operate in one of two modes:

- **Transport mode** – Original IP headers are left intact. Used when securing communication from one device to another single device.
- **Tunnel mode** – the entire original packet is hashed and/or encrypted, including both the payload and any original headers. A temporary IP header is applied to the packet during transit. Used to tunnel traffic from one site to another.

IKE and IPSEC Security Associations

IPSEC VPN peers establish a **Security Association (SA)**, a "connection" or "policy" between the two endpoints of the VPN tunnel. An SA is a **one-way** virtual tunnel between the VPN peers. Thus, for full communication to occur, two SA's must be established, one for each direction. Before the SA can be established, several parameters must be negotiated between VPN peers, and keys must be both created and exchanged. The Internet Key Exchange (IKE) protocol controls this negotiation process, on UDP port 500. IKE Policy Sets are created to negotiate several parameters, including:

- The **encryption algorithm** (such as DES, 3DES, or AES)
- The **hashing algorithm** (such as MD5 or SHA-1)
- The **authentication method** (such as shared keys or RSA signatures)
- The **Diffie-Hellman** (D-H) group for creating and sharing keys
- The **SA Lifetime**, measured in seconds or in kilobytes sent

IKE policies are often referred to as **Internet Security Association and Key Management Protocol (ISAKMP)** policies. Multiple IKE policies can be created on a VPN peer. During the negotiation process, VPN peers share their list of configured IKE policies. The SA will only be established if there is an exact matching policy between the peers. There are two phases to this negotiation process:

IKE Phase 1 establishes the initial tunnel (referred to as the IKE or ISAKMP SA). Peers are authenticated, encryption and hashing algorithms are negotiated, and keys are exchanged based on the IKE Policy Sets. Two modes can be used for Phase 1 negotiation:

- **Main Mode – slower, but more secure**
- **Aggressive Mode – faster, but less secure**

IKE Phase 2 establishes the IPSEC tunnel **(IPSEC SA),** which details the AH or ESP parameters for securing data. These parameters are contained in an **IPSEC Transform Set.**

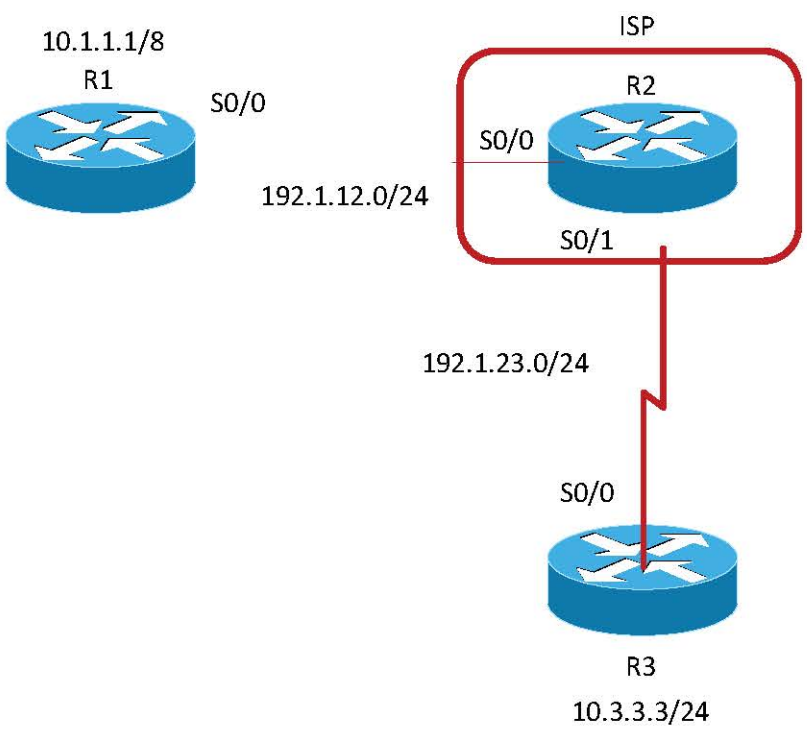
IKE Phase 1 negotiates parameters for the tunnel (key exchange) itself, while IKE Phase 2 negotiates parameters for the data traversing that tunnel.

The Five Steps of IPSEC

The operation of IPSEC can be described in five steps:

1. Any traffic that should be secured and sent across the tunnel is identified as **interesting traffic**, usually using an access-list.
2. **IKE (Internet Key Exchange)** Phase 1 is initiated. Peers are authenticated, keys are exchanged, and IKE Policy Sets are negotiated. If successful, the IKE SA is established.
3. **IKE (Internet Key Exchange)** Phase 2 is initiated. IPSEC Transform Sets are negotiated, and if successful, the IPSEC SA is established.
4. Data is actually transferred, using the agreed upon security policy.
5. The session is torn down once the SA Lifetime expires.

Lab 1 – LAN-To-LAN IPSec Tunnel Using Crypto Maps



Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
Loopback 0	10.1.1.1	255.255.255.0
S 0/0	192.1.12.1	255.255.255.0

R2

Interface	IP Address	Subnet Mask
S 0/0	192.1.12.2	255.255.255.0
S 0/1	192.1.23.2	255.255.255.0

R3

Interface	IP Address	Subnet Mask
Loopback 0	10.3.3.3	255.255.255.0
S 0/0	192.1.23.3	255.255.255.0

Task 1

Configure R1, R2 and R3 as given in the topology. Also configure a Default Route on R1 & R3 towards R2, the ISP Router.

On R1 ip route 0.0.0.0 0.0.0.0 192.1.12.2
On R3 ip route 0.0.0.0 0.0.0.0 192.1.23.2

Task 2

Configure an IPSec Tunnel to encrypt traffic from 10.1.1.0/24 on R1 (Loopback 0) to the 10.3.3.0/24 on R3 (Loopback 0) using the following parameters for IPSec:

- ISAKMP Parameters
  - Authentication : Pre-shared
  - Encryption : 3DES
  - Group : 2
  - Hash : MD5
  - Pre-Shared Key : **cisco**
- IPSec Parameters
  - Encryption : ESP-3DES
  - Authentication : ESP-SHA-HMAC

```
On R1
!
crypto isakmp policy 10
 authentication pre-share
 hash md5
 Group 2
 encryption 3des
!
crypto isakmp key cisco address 192.1.23.3
!
crypto ipsec transform-set t-set esp-3des esp-sha-hmac
!
access-list 101 permit ip 10.1.1.0 0.0.0.255 10.3.3.0 0.0.0.255
!
crypto map I-MAP 10 ipsec-isakmp
 set peer 192.1.23.3
 set transform-set t-set
 match address 150
!
interface S 0/0
 crypto map I-MAP
!
```

```
On R3
!
crypto isakmp policy 10
 authentication pre-share
 hash md5
 group 2
 encryption 3des
!
crypto isakmp key cisco address 192.1.12.1
!
crypto ipsec transform-set t-set esp-3des esp-sha-hmac
!
access-list 150 permit ip 10.3.3.0 0.0.0.255 10.1.1.0 0.0.0.255
!
crypto map I-MAP 10 ipsec-isakmp
 set peer 192.1.12.1
 set transform-set t-set
 match address 150
!
interface S 0/0
 crypto map I-MAP
!
```

Task 3

Verify ping test and show commands

```
On R1
R1#ping 10.3.3.3 source 10.1.1.1
!!!!!
. . .
R1#show crypto isakmp sa
(Ask your trainer to explain output)
R1#show crypto ipsec sa
(Ask your trainer to explain output)
```

Lab 6 – Configuring DMVPN

(Build on mGRE Lab)

Task 1

Configure IPSec to encrypt the traffic passing thru the MGRE tunnel. Make sure the packet does not duplicate the IP addresses in the Header. Use the following parameters for the IPSec Tunnel:

- ISAKMP Parameters
  - Authentication : Pre-shared
  - Encryption : 3DES
  - Group : 2
  - Hash : MD5
- Pre-Shared Key : **cisco**
  - IPSec Parameters
  - Encryption : ESP-3DES
  - Authentication : ESP-SHA-HMAC

```
On R1
!
Crypto isakmp policy 10
 Authentication pre-share
 Hash md5
 Group 2
 Encryption 3des
!
Crypto isakmp key cisco address 0.0.0.0
!
crypto ipsec transform-set t-set esp-3des esp-sha-hmac
 mode transport
!
crypto ipsec profile IPSEC
 set transform-set t-set
!
Interface Tunnel 1
 Tunnel protection ipsec profile IPSEC
!
```

```
On R2
!
crypto isakmp policy 10
 authentication pre-share
 hash md5
 group 2
 encryption 3des
!
crypto isakmp key cisco address 0.0.0.0
!
crypto ipsec transform-set t-set esp-3des esp-sha-hmac
 mode transport
!
crypto ipsec profile IPSEC
 set transform-set t-set
!
interface tunnel 1
 tunnel protection ipsec profile IPSEC
!
```

```
On R3
!
crypto isakmp policy 10
 authentication pre-share
 hash md5
 group 2
 encryption 3des
!
crypto isakmp key cisco address 0.0.0.0
!
crypto ipsec transform-set t-set esp-3des esp-sha-hmac
 mode transport
!
crypto ipsec profile IPSEC
 set transform-set t-set
!
interface tunnel 1
 tunnel protection ipsec profile IPSEC
!
```

Task 2

Verify ping test and show commands

```
On R1

R1#ping 10.3.3.3 source 10.1.1.1

.!!!!

. . .

R1#show crypto isakmp sa

(Ask your trainer to explain output)

R1#show crypto ipsec sa

(Ask your trainer to explain output)
```

MODULE 14

BASIC QOS CONCEPTS

Chapter 1 – QoS Concepts and Congestion Management

Overview

QoS refers to the tools that networking devices use to apply some different treatment to packets in the network as they pass through the device. For example, the **WAN** edge router would queue packets waiting for the WAN interface to be available. The router also use a queue scheduling algorithm to determine which packets should be sent next, using some other order than the arrival order - giving some packets better service, and some worse service.

Characteristics of network traffic:

Bandwidth

Bandwidth refers to the speed of a link, in bits per second (bps). The networking device's QoS tools determine what packet is sent over the link next, so the networking device is in control of which packet gets access to the bandwidth next, and how much of that bandwidth (capacity) each type of traffic gets over time.

Delay

Delay can refer to **one-way delay** - the time required for the message to be sent from the source host to the destination host or **round-trip delay** - the delay from the source to the destination host and then back again.

Jitter

Jitter refers to the variation in one-way delay between consecutive packets sent by the same packet.

Loss

Loss refers to the amount of lost messages, usually as a percentage of packets sent.

QoS Techniques:

Classification and Marking

Classification and marking refers to a type of QoS tool that classifies packets based on their header contents, and then marks the packets by changing some bits in specific header fields. Marking each packet as a member of a network class so that packet's class would be easily recogonised through out the network.

Classification Tools:

ACLs:

QoS tools support the ability to simply refer to an IP ACL, with this kind of logic:

For any packet matched by the ACL with a permit action, consider that packet a match for QoS, so do a particular QoS action.

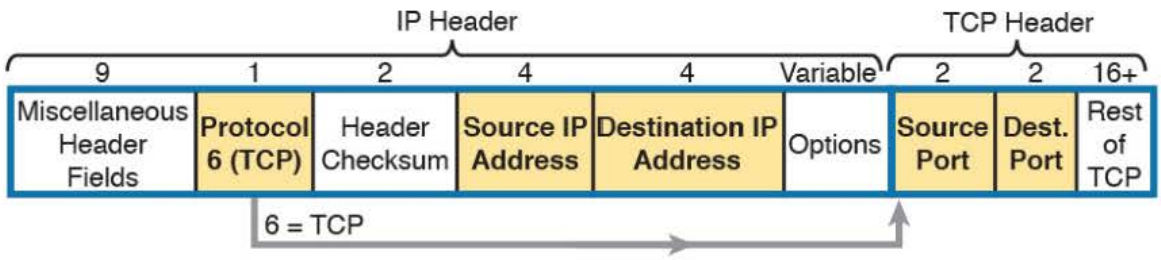


Figure 1 - Classification with Five Fields Used by Extended ACLs

Network-Based Application Recognition (NBAR):

NBAR provides classification abilities beyond that of access-lists, including:

- Ability to classify services that use dynamic port numbers. This is accomplished using the stateful inspection of traffic flows.
- Ability to classify services based on sub-protocol information. For example, NBAR can classify HTTP traffic based on payload, such as the host, URL, or MIME type

NBAR recognizes applications using NBAR **Packet Description Language Modules (PDLMs)**, which are stored in flash on IOS devices. Updated PDLMs are provided by Cisco so that IOS devices can recognize newer application types.

Marking

Layer-2 Marking

Layer-2 marking can be accomplished for a variety of frame types:

- Ethernet – using the 802.1p Class of Service (CoS) field.
- Frame Relay – using the Discard Eligible (DE) bit.
- MPLS - using the EXP field.

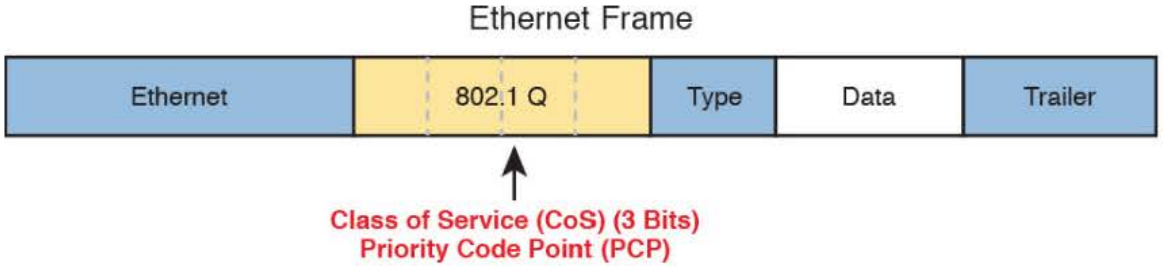


Figure 2 - Class of Service Field in 802.1Q/p Header

Ethernet:

Marking Ethernet frames is accomplished using the 3-bit 802.1p **Class of Service (CoS)** field. The CoS field is part of the 4-byte 802.1Q field in an Ethernet header, and thus is only available when 802.1Q VLAN frame tagging is employed. The CoS field provides 8 priority values:

Type	Decimal	Application
Routine	0	Best effort forwarding
Priority	1	Medium priority forwarding
Immediate	2	High priority forwarding
Flash	3	VoIP call signaling forwarding
Flash-Override	4	Video conferencing forwarding
Critical	5	VoIP forwarding
Internet	6	Inter-network control (Reserved)
Network Control	7	Network control (Reserved)

Frame Relay:

Frame Relay frames provide a less robust marking mechanism, compared to the Ethernet CoS field. Frame Relay frames reserve a 1-bit field, to prioritize which traffic should be dropped during periods of congestion. Frame Relay identifies this bit as the **Discard Eligible (DE) field**, A value of 0 indicates a lower likelihood to get dropped, while a value of 1 indicates a higher likelihood to get dropped.

MPLS:

MPLS employs a 3-bit EXP (Experimental) field within the 4-byte MPLS header. **The EXP** field provides similar QoS functionality to the **Ethernet CoS field.**

Layer-3 Marking:

Layer-3 marking is accomplished using the 8-bit **Type of Service (ToS) field**, part of the IP header. A mark in this field will remain unchanged as it travels from hop-to-hop, unless a Layer-3 device is explicitly configured to overwrite this field.

There are two marking methods that use the ToS field:

IP Precedence - uses the first three bits of the ToS field. The **IP Precedence** provides similar QoS functionality to the **Ethernet CoS field.**

Differentiated Service Code Point (DSCP) – uses the first six bits of the ToS field. When using DSCP, the ToS field is often referred to as the Differentiated Services (DS) field. These values determine the per-hop behavior (PHB) received by each classification of traffic.

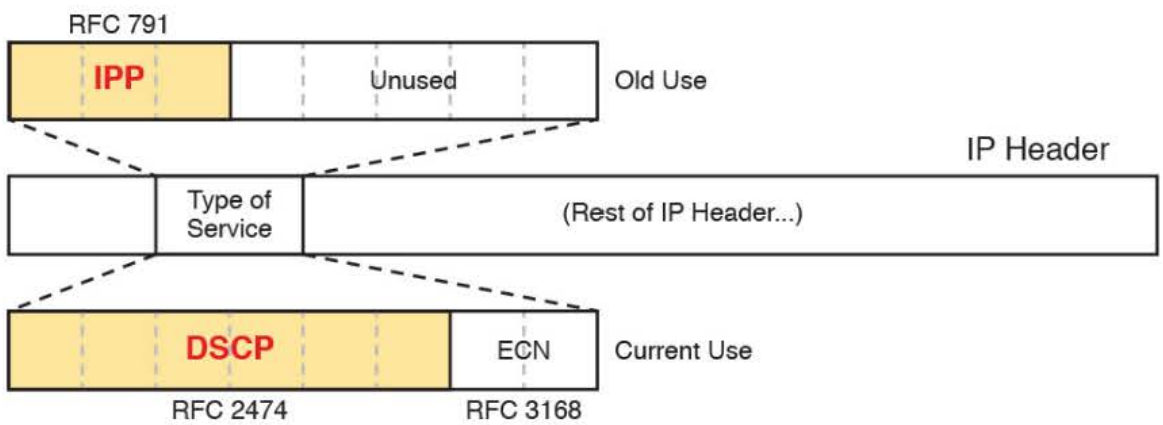


Figure 3 - IP Precedence and Differentiated Services Code Point Fields

Differentiated Service Code Point (DSCP)

DSCP utilizes the **first six bits** of the ToS header to identify the priority of a packet. The **first three bits** identify the **Class Selector** of the packet, and is backwards compatible with IP Precedence. The following **three bits** identify the **Drop Precedence** of the packet.

DSCP identifies **six Class Selectors** for traffic (numbered 0 - 5).

Class 0 is default, and indicates **best-effort forwarding.** Packets with a higher Class value should be provided with a better level of service.

Class 5 is the highest DSCP value, and should be reserved for the most sensitive traffic. Within each Class Selector, traffic is also assigned a **Drop Precedence**. Packets with a higher Drop Precedence are **more likely** to be dropped during congestion than packets with a lower Drop Precedence. Remember that this is applied only within the same Class Selector.

The Class Name provides a simple way of identifying the DSCP value. **AF** is short for **Assured Forwarding,** and is the type of service applied to Classes 1 – 4. If a packet is marked AF23, then the Class Selector is 2 (the 2 in 23) and its Drop Precedence is High (the 3 in 23).

Packets marked as **Class 0** (Default) or **Class 5 (Expedited Forwarding or EF)** do not have a **Drop Precedence.**

Device Trust or Trust boundary

The trust boundary refers to the point in the path of a packet flowing through the network at which the networking devices can trust the current QoS markings.

Congestion Management (Queuing)

The term congestion management refers to the QoS toolset for managing the queues that hold packets while they wait their turn to exit an interface.

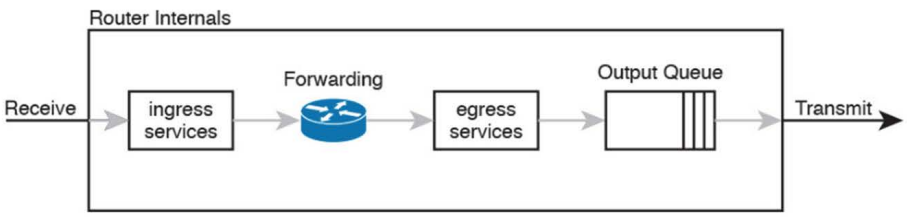
A **queue** is used to store traffic until it can be processed or serialized. Both switch and router interfaces have **ingress** (inbound) queues and **egress** (outbound) queues.

Router queues are susceptible to congestion. Congestion occurs when the rate of ingress traffic is greater than can be successfully processed and serialized on an egress interface. Common causes for congestion include:

- The speed of an ingress interface is higher than the egress interface.
- The combined traffic of multiple ingress interfaces exceeds the capacity of a single egress interface.
- The switch/router CPU is insufficient to handle the size of the forwarding table.

By default, if an interface's queue buffer fills to capacity, new packets will be dropped. This condition is referred to as **tail drop**, and operates on a first come, first-served basis. If a standard queue fills to capacity, any new packets are indiscriminately dropped, regardless of the packet's classification or marking.

QoS provides switches and routers with a mechanism to **queue and service higher priority traffic** before lower priority traffic. QoS also provides a mechanism to **drop lower priority traffic before higher priority traffic**, during periods of congestion. This is known as **Weighted Random Early Detection (WRED).**



The figure shows output queuing, in which the device holds messages until the output interface is available. The queuing system may use a single output queue, with a **first-in, first-out (FIFO)** scheduler.

The classifier can use to previously marked values based on **COS, EXP, IP Precedence or DSCP** to do a more extensive match. The queuing system needs a scheduler **(algorithm)** to decide which packet to take next **(Prioritization)** when the interface becomes available.

Prioritization:

Prioritization refers to the concept of giving priority to one queue over another in some way.

Round Robin Scheduling (Prioritization)

One scheduling algorithm used by Cisco routers and switches uses round robin logic. In its most basic form, round robin cycles through the queues in order, taking turns with each queue. In each cycle, the scheduler either takes one message or takes a number of bytes from each queue by taking enough messages to total that number of bytes. Take some messages from queue 1, move on and take some from queue 2, then take some from queue 3, and so on, starting back at queue 1 after finishing a complete pass through the queues.

Round robin scheduling also includes the concept of weighting (weighted round robin). Basically, the scheduler takes a different number of packets (or bytes) from each queue, giving more preference to one queue over another.

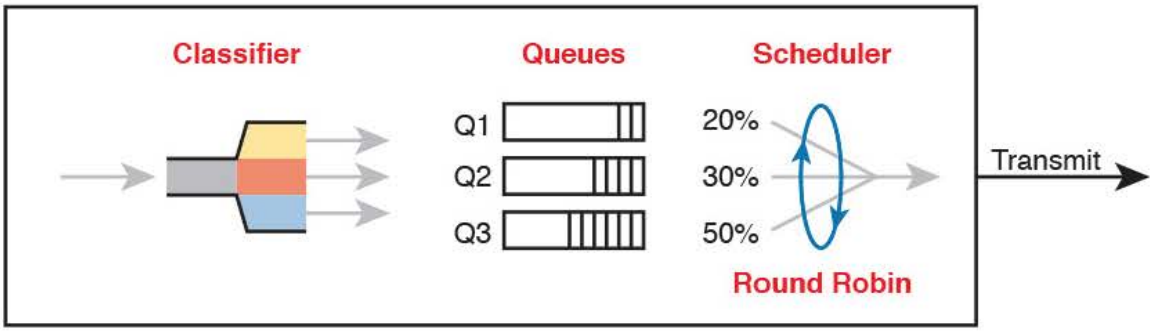


Figure 5 - CBWFQ Round Robin Scheduling

Low Latency Queuing (LLQ)

Low-Latency Queuing (LLQ) is an improved version of CBWFQ that includes one or more strict-priority queues, to alleviate latency issues for real-time applications. Strict-priority queues are always serviced before standard class-based queues.

Shaping and Policing

There are two methods for managing traffic that exceeds a specified rate:

- Traffic shaping
- Traffic policing

These methods are often necessary on the edge separating a customer's network from a provider's network. Providers often force the customer to adhere to a specific policy of service (or committed rate). This policy is referred to as the **Service Level Agreement (SLA)** between the customer and provider. Shaping and policing mechanisms differ in how each handles violations of the SLA.

Shaping is usually implemented on the customer side, and will buffer traffic that exceeds the provider's committed rate. Thus, shaping can slow the traffic rate and siphon out traffic in compliance with the provider's SLA. Buffering traffic will often create delay and jitter, which can negatively impact sensitive traffic types. Shaping also requires sufficient memory to queue buffered traffic. Shaping provides no mechanism to re-mark traffic that exceeds the committed rate.

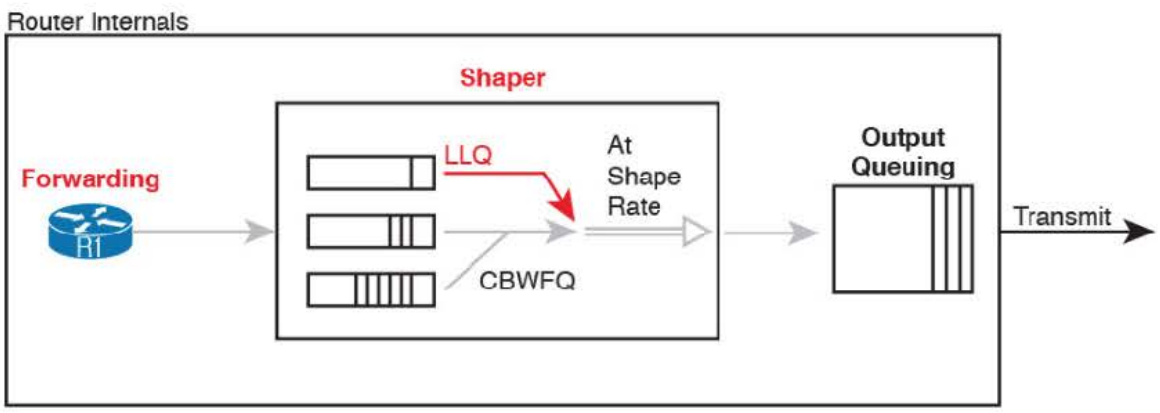


Figure 6 - Shaping Queues: Scheduling with LLQ and CBWFQ

Policing is usually implemented on the provider side, and will either drop or re-mark traffic that exceeds the provider's committed rate. TCP traffic that is dropped will be forced to resend, which may result in TCP global synchronization or starvation issues. Policing can be implemented for both inbound and outbound traffic on an interface. Shaping can only occur on outbound traffic on an interface.

Policing is configured using several parameters. Policing configurations apply the following parameters.

- **Rate** - The effective policing rate in terms of bits per second (bps). Each Catalyst switch supports different rates and different rate increments.
- **Burst** - The number of packets that switches allow in the bucket before determining that the packet is out of profile. Various Catalyst switches support various burst ranges with various increments.
- **Conforming action** - Depending on the Catalyst switch model, optional supported conforming actions include drop, transmit, and mark.

• **Exceed action** - Depending on the Catalyst switch model, optional supported exceed actions for out-of-profile packets are drop, transmit, and mark.

• **Violate action** - Applies to Catalyst switches that support two-rate policers, where there is a second bucket in the leaky token bucket algorithm. The violate action adds a third measurement for out-of-profile traffic. Applicable violate actions are drop, transmit, and mark. RFC 2698 discusses three-color marking, the basis for the addition of violate action on Cisco Catalyst switches.

MODULE 15

INFRASTRUCTURE SERVICE

Chapter 1 – Basic Infrastructure Services

Name Resolution

Name resolution systems provide the translation between alphanumeric names and numerical addresses, alleviating the need for users and administrators to memorize long strings of numbers. There are two common methods for implementing name resolution:

- A static file on each host on the network, containing all the name-toaddress translations (examples include the HOSTS and LMHOSTS files).
- A centralized server that all hosts on the network connect to for name resolution.

The two most common name resolution systems are Domain Name System (DNS) and Windows Internet Name Service (WINS). WINS was used in Microsoft networks to translate IP addresses to NetBIOS names, and is mostly deprecated. DNS is heavily utilized on the Internet and on systems such as Active Directory.

Domain Name System (DNS)

Domain Name System (DNS) translates between domain names and IP addresses, and is supported by nearly every operating system. All Internetbased name resolution utilizes DNS. DNS is organized as a hierarchy.

Consider the following translation:

- www.google.com = 209.85.225.104

The above domain name represents a Fully Qualified Domain Name (FQDN):

- .com represents a top level domain.
- .google represents a secondary level domain
- www represents a host computer in the .google.com domain.

Other top level domains include .**org, .net,** and .**gov.** Top level domains can also include country codes, such as .ca, .nl, and .de.

Resolving Hostnames on Cisco IOS Devices

There are two methods of name resolution on Cisco IOS devices:

A static host table on each device (similar to a HOSTS file).

A centralized DNS server(s) configured on each device.

To manually build a local host table on an IOS device:

```
Router(config)# ip host Router1 172.16.1.1
Router(config)# ip host Router2 172.17.1.2
```

To view the local host table:

```
Router# show hosts
```

To point an IOS device to a centralized DNS server:

```
Router(config)# ip name-server 10.0.1.2
```

To disable DNS lookups on an IOS device:

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

To configure the local domain on an IOS device:

```
Router(config)# ip domain-name CISCO.COM
```

DHCP (Dynamic Host Control Protocol)

Dynamic Host Control Protocol (DHCP) provides administrators with a mechanism to dynamically allocate IP addresses, rather than manually setting the address on each device.

DHCP servers lease out IP addresses to DHCP clients, for a specific period of time. There are four steps to this DHCP process:

- When a DHCP client first boots up, it broadcasts a **DHCPDiscover** message, searching for a DHCP server.
- If a DHCP server exists on the local segment, it will respond with a **DHCPOffer,** containing the "offered" IP address, subnet mask, etc.
- Once the client receives the offer, it will respond with a **DHCPRequest,** indicating that it will accept the offered protocol information.
- Finally, the server responds with a **DHCPACK**, acknowledging the clients acceptance of offered protocol information.

By default, DHCP leases an address for 8 days. Once 50% of the lease expires, the client will try to renew the lease with the same DHCP server. If successful, the client receives a new 8 day lease. If the renewal is not successful, the client will continue "attempting" to renew, until 87.5% of the lease has expired. Once this threshold has been reached, the client will attempt to find another DHCP server to bind to. In addition to IP address and subnet mask information, DHCP can provide the following protocol parameters:

- Default Gateway
- Domain Name and DNS servers
- Time Servers
- WINS servers

These are just a few examples of the many DHCP "options" that exist.

Redundancy and Load Balancing

High availability is critical in most environments. Even a brief outage due to hardware failure may be considered unacceptable.

Cisco supports three protocols to provide transparent Layer-3 redundancy:

- Hot Standby Router Protocol (HSRP)
- Virtual Router Redundancy Protocol (VRRP)
- Gateway Load Balancing Protocol (GLBP)

Hot Standby Router Protocol (HSRP)

Cisco developed the proprietary **Hot Standby Router Protocol (HSRP)** to allow multiple routers or multilayer switches to masquerade as a single gateway. This is accomplished by assigning a **virtual IP and MAC address** to all routers participating in an **HSRP group.**

Routers within the same HSRP group must be assigned the same group number, which can range from 0 to 255. However, most Cisco platforms only support 16 configured HSRP groups. HSRP routers are elected to specific roles:

- **Active Router** – router currently serving as the gateway.
- **Standby Router** – backup router to the Active Router.
- **Listening Router** – all other routers participating in HSRP.

Only **one active** and **one standby** router are allowed per HSRP group. Thus, HSRP provides Layer-3 redundancy, but no inherent load balancing. Hello packets are used to elect HSRP roles and to ensure all routers are functional. If the current active router fails, the standby router will immediately take over as active, and a new standby is elected. By default, **hello packets** are sent every **3 seconds.**

Priority

The role of an HSRP router is dictated by its priority. The priority can range from 0 – 255, with a default of **100.** A higher priority is preferred.

Thus, the router with the **highest priority** is elected the **active router**. The router with the **second highest priority** becomes the **standby router**. If all priorities are equal, whichever router has the **highest IP Address** on its HSRP interface is elected the active router.

HSRP States

A router interface participating in HSRP must progress through several states before settling into a role:

- Disabled
- Initial
- Learn
- Listen

- Speak
- Standby
- Active

A **disabled state** indicates that the interface is either not configured for HSRP, or is administratively shutdown.

An interface begins in an **initial state** when first configured with HSRP, or taken out of an administratively shutdown state.

An interface enters a **learn state** if it does not know the HSRP virtual IP address.

Normally the virtual IP is manually configured on the interface – otherwise, it will be learned from the current Active router via hello packets.

An interface in a **listen state** knows the virtual IP address, but was not elected as either the Active or Standby Router.

Interfaces in a **speak state** are currently participating in the election of an active or standby router. Elections are performed using hello packets, which are sent out every 3 seconds by default.

A **standby state** indicates that the interface is acting as a backup to the active router. The standby router continuously exchanges hello packets with the active router, and will take over if the active router fails.

An interface in an **active state** is the live gateway, and will forward traffic sent to the virtual IP address. Hosts will use the virtual IP address as their default gateway. The active router will respond to ARP requests for the virtual IP with the virtual MAC address.

Note that hello packets are only exchanged in three HSRP states:

- Speak
- Standby
- Active

Interfaces in a listen state will only listen for hello packets. If an active or standby router fails, a listen interface will transition to a speak state to participate in a new election.

NAT (Network Address Translation)

The rapid growth of the Internet resulted in a shortage of available IPv4 addresses. In response, a specific subset of the IPv4 address space was designated as private, to temporarily alleviate this problem.

A **public address** can be routed on the Internet. Thus, devices that must be Internet-accessible must be configured with (or reachable by) public addresses. Allocation of public addresses is governed by the Internet Assigned Numbers Authority (IANA).

A **private address** is intended for internal use within a home or organization, and can be freely used by anyone. However, private addresses can never be routed on the Internet. In fact, Internet routers are configured to immediately drop traffic with private addresses. Three private address ranges were defined in RFC 1918, one for each IPv4 class:

- Class A - 10.x.x.x /8
- Class B - 172.16.x.x /12
- Class C - 192.168.x.x /24

It is possible to translate between private and public addresses, using **Network Address Translation (NAT).** NAT allows a host configured with a private address to be stamped with a public address, thus allowing that host to communicate across the Internet. It is also possible to translate multiple privately-addressed hosts to a single public address, which conserves the public address space.

NAT provides an additional benefit – hiding the specific addresses and addressing structure of the internal (or private) network.

Note: NAT is not restricted to private-to-public address translation, though that is the most common application. NAT can also perform public-to-public address translation, as well as private-to-private

address translation.

NAT is only a temporarily solution to the address shortage problem. IPv4 will eventually be replaced with IPv6, which supports a vast address space.

Both Cisco IOS devices and PIX/ASA firewalls support NAT.

Types of NAT

NAT can be implemented using one of three methods:

Static NAT – performs a static one-to-one translation between two addresses, or between a port on one address to a port on another address. Static NAT is most often used to assign a public address to a device behind a NAT-enabled firewall/router.

Dynamic NAT – utilizes a pool of global addresses to dynamically translate the outbound traffic of clients behind a NAT-enabled device.

NAT Overload or Port Address Translation (PAT) – translates the outbound traffic of clients to unique port numbers off of a single global address. PAT is necessary when the number of internal clients exceeds the available global addresses.

NAT Terminology Specific terms are used to identify the various NAT addresses:

- **Inside Local** – the specific IP address assigned to an inside host behind a NAT-enabled device (usually a private address).
- **Inside Global** – the address that identifies an inside host to the outside world (usually a public address). Essentially, this is the dynamically or statically-assigned public address assigned to a private host.
- **Outside Global** – the address assigned to an outside host (usually a public address).
- **Outside Local** – the address that identifies an outside host to the inside network. Often, this is the same address as the Outside Global. However, it is occasionally necessary to translate an outside (usually public) address to an inside (usually private) address.

For simplicity sake, it is generally acceptable to associate **global** addresses with **public** addresses, and **local** addresses with **private** addresses. However, remember that public-to-public and private-to-private translation is still possible. **Inside** hosts are within the local network, while **outside** hosts are external to the local network.

Network Time Protocol (NTP)

Time and date information can be configured locally on both Cisco routers and switches. However, in environments with a large amount of equipment, this can become unmanageable. As a result, the "time" will be inconsistent throughout the network. This can lead to various inaccuracies, such as the timestamps in syslog messages.

Network Time Protocol (NTP) provides a mechanism to synchronize time throughout the network. An NTP device will form an **association** with NTP devices closer to the time source. NTP devices use a special measurement, called a stratum, to determine how far they are away from the time source.

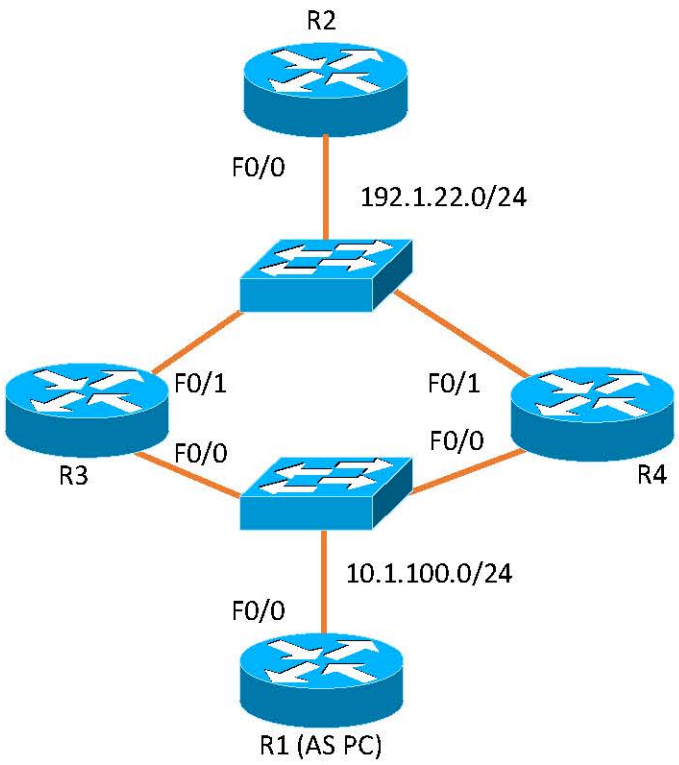
For example, a device with a **stratum** of 1 is directly connected to the time source. A device with a stratum of 2 is one device (or "hop") away from the time source.

NTP can be configured one of two ways:

- **Client/Server** – The NTP client is configured to always get its time information from the NTP server. The server will never get its time from the client.
- **Peer-to-peer** – Peered NTP devices can get their time from each other, depending on who is closest to the time source (i.e., lowest stratum).

NTP associations can be secured using encrypted authentication.

Lab 1 – Basic Configuring HSRP



Interface IP Address Configuration

R1

Interface IP	Address	Subnet Mask
F 0/0	10.1.100.1	255.255.255.0

R2

Interface IP	Address	Subnet Mask
F 0/0	192.1.22.2	255.255.255.0
Loopback 0	8.2.2.2	255.0.0.0

R3

Interface IP	Address	Subnet Mask
F 0/0	10.1.100.3	255.255.255.0
F 0/1	192.1.22.3	255.255.255.0

R4

Interface IP	Address	Subnet Mask
F 0/0	10.1.100.4	255.255.255.0
F 0/1	192.1.22.4	255.255.255.0

Task 1

Configure EIGRP in AS 100 on R2, R3 and R4. Advertise the Loopback 0 and the physical links in EIGRP 100.

R2 ! router eigrp 100 no auto-summary network 192.1.22.0 network 8.0.0.0 !	R3 ! router eigrp 100 no auto-summary network 10.0.0.0 network 192.1.22.0 !
R4 ! router eigrp 100 no auto-summary network 10.0.0.0 network 192.1.22.0 !	

Task 2

Configure HSRP between R3 and R4. Use 192.1.11.34 as the Virtual HSRP address. R3 should be the preferred Router. Have R1 point to the virtual HSRP address as the Default Gateway.

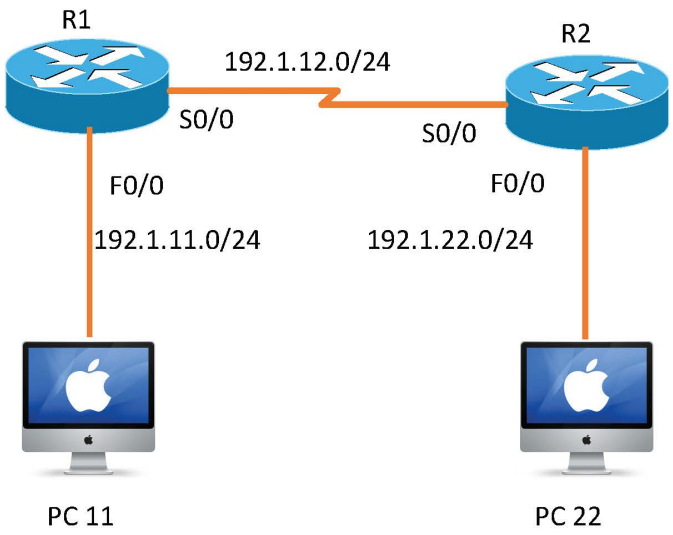
R3 ! interface f 0/1 standby 1 ip 10.1.100.34 standby 1 priority 105 standby 1 preempt !	R3 ! interface f 0/1 standby 1 ip 10.1.100.34 !
R1 ! ip route 0.0.0.0 0.0.0.0 10.1.100.34 !	

Task 3

Verify using various verification commands

- Type **Show standby** on R3 and R4.
- Which router is the Active HSRP Router?
- Which router is the Standby HSRP Router?
- Why?

Lab 2 – Configuring a IOS DHCP Server and IP Helper Address



Interface IP Address Configuration

R1

Interface IP	Address	Subnet Mask
S 0/0	192.1.12.1	255.255.255.0
F0/0	192.1.11.1	255.255.255.0

R2

Interface IP	Address	Subnet Mask
S 0/0	192.1.12.2	255.255.255.0
F0/0	192.1.22.2	255.255.255.0

Task 1

Configure R1 as a DHCP Server with the following parameters:

- IP ADDRESS : 192.1.11.0/24
- NS ADDRESS : 192.1.11.1
- DEFAULT GATEWAY : 192.1.11.1
- Excluded addresses : 192.1.11.1 - 192.1.11.50

```
R1
ip dhcp excluded-address 192.1.11.1 192.1.11.50
!
ip dhcp pool POOL11
   network 192.1.11.0 255.255.255.0
   dns-server 192.1.11.1
   default-router 192.1.11.1
!
```

Task 2

Aslo configure R1 as a DHCP Server with the following parameters:

- IP ADDRESS : 192.1.22.0/24
- DNS ADDRESS : 192.1.22.1
- DEFAULT GATEWAY : 192.1.22.1
- Excluded addresses : 192.1.22.1 - 192.1.22.50

```
R1
ip dhcp excluded-address 192.1.22.1 192.1.22.50
!
ip dhcp pool POOL22
  network 192.1.22.0 255.255.255.0
  dns-server 192.1.22.1
  default-router 192.1.22.1
!
```

Task 2

R2 Broadcast Forwarding configuration (DHCP Relay Agent)

```
R2
!
interface f0/0
 ip helper-address 192.1.12.1
!
```

Task 3

Configure PC 11 (R3) and PC 22 (R4) interface F 0/0 to acquire an ip address using DHCP.

```
R3 and R4
!
interface f 0/0
 ip address dhcp
 no shutdown
!
```

Lab 3 – Configuring NTP

(Build on Lab 2)

Task 1

Configure R1 as a NTP Master with a stratum of 2. Set it with a Time Zone of GMT +4.

```
R3
Clock timezone DST 4
!
!Set the clock to the current date and time using the clock set command
!
Ntp master 2
```

Task 2

Configure R2 to receive its clock from R1. Set it with a Time Zone of GMT +5.5. Configure R2 such that it automatically adjusts the clock based on the time zone

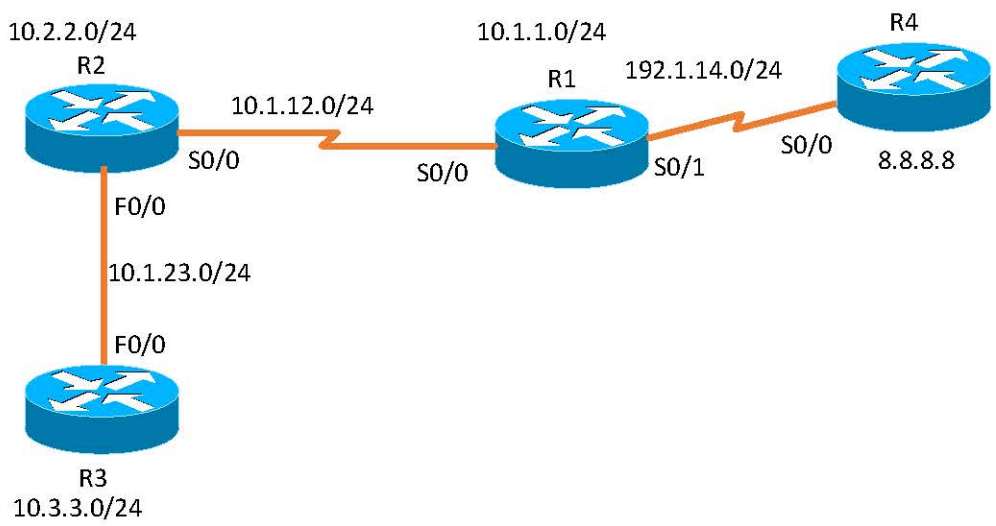
```
R2
Clock timezone DST 5 30
!
Ntp server 192.1.22.3
```

Task 3

Configure R3 and R4 to receive its clock from R2. Set it with a Time Zone of GMT +8. Do not use the NTP Server command to receive the clock. Do not configure any commands under the interface to accomplish this task.

```
On R3 and R4
Clock timezone DST 8
!
Ntp peer 192.1.33.2
!
```

Lab 4 – Configuring NAT



Interface IP Address Configuration

R1

Interface IP	Address	Subnet Mask
S 0/0	10.1.12.1	255.255.255.0
S0/1	192.1.14.1	255.255.255.0
Loopback 0	10.1.1.1	255.255.255.0

R2

Interface IP	Address	Subnet Mask
S 0/0	10.1.12.2	255.255.255.0
F0/0	10.1.23.2	255.255.255.0
Loopback 0	10.2.2.2	255.255.255.0

R3

Interface IP	Address	Subnet Mask
F0/0	10.1.23.3	255.255.255.0
Loopback 0	10.3.3.3	255.255.255.0

R4

Interface IP	Address	Subnet Mask
S 0/0	192.1.14.4	255.255.255.0
Loopback 0	8.8.8.8	255.255.255.0
Loopback 1	4.2.2.2	255.255.255.0

Task 1

Configure EIGRP 100 on all private networks on R1, R2 and R3. Configure a static default route on R1, R2 and R3 towards their respective gateways.

```
On R1, R2 and R3
!
router eigrp 100
 no auto-summary
 network 10.0.0.0
!
ON R1
ip route 0.0.0.0 0.0.0.0 192.1.14.4
!
On R2
ip route 0.0.0.0 0.0.0.0 10.1.12.1
!
On R3
ip route 0.0.0.0 0.0.0.0 10.1.23.2
!
```

Task 2

Translate the 10.1.1.0/24 Network behind R1 into a range of Class C addresses assigned to R1 by the ISP. Use the range 192.1.14.10 – 192.1.14.40 for the pool.

R3 ! interface f 0/1 standby 1 ip 10.1.100.34 standby 1 priority 105 standby 1 preempt !	R3 ! interface f 0/1 standby 1 ip 10.1.100.34 !
R1 ! ip route 0.0.0.0 0.0.0.0 10.1.100.34 !	

Task 3

R2 should use 192.1.14.251 for its Web Server so that people on the outside can access it. The internal web server is at 10.2.2.2.

```
On R1
!
ip nat inside source static 10.2.2.2 192.1.14.251
!
```

Task 3

Translate the 10.3.3.0 Network behind R3 using the 192.1.14.133 address (PAT). The entire should be able to go out simultaneously using this address.

```
On R1
!
access-list 3 permit 10.3.3.0 0.0.0.255
ip nat pool DP 192.1.14.133 192.1.14.133
ip nat inside source list 3 pool DP overload
!
```

Task 4

There is a telnet server at 10.3.3.3 and a web server at 10.3.3.33. Translate these servers to 192.1.14.100 and 192.1.14.101 on the outside. Use Static PAT to accomplish this task. Configute 10.3.3.33/24 as secondary ip on R3 loopback 0.

```
On R1
!
ip nat inside source static tcp 10.3.3.3 80 192.1.14.100 23
ip nat inside source static tcp 10.3.3.33 80 192.1.14.100 80
!
interface loopback 0
 ip address 10.3.3.33 255.255.255.0 secondary
!
```

Task 5

Verify using various test and show commands.

```
On R1
R1#show ip nat translations
(Ask you trainer to explain the output)
```

MODULE 16

LAN SWITCHING TECHNOLOGIES

Chapter 1 – LAN Switching Concepts

Switching Architectures

Network Traffic Models

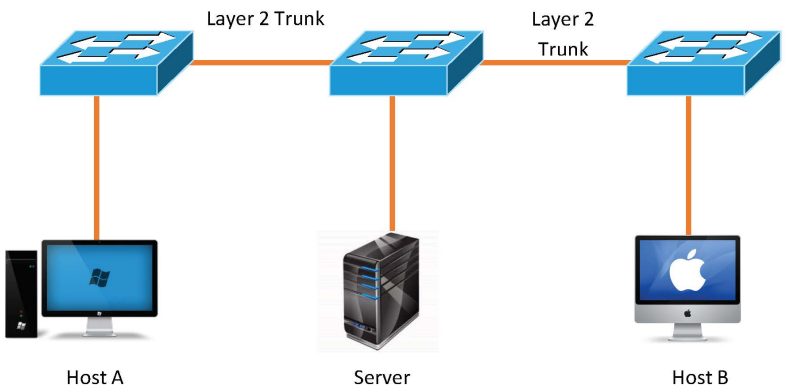
Traffic flow is an important consideration when designing scalable, efficient networks. Fundamentally, this involves understanding two things:

- Where do resources reside?
- Where do the users reside that access those resources?

Legacy networks adhered to the 80/20 design, which dictated that:

- 80 percent of traffic should remain on the local network.
- 20 percent of traffic should be routed to a remote network.

To accommodate this design practice, resources were placed as close as possible to the users that required them. This allowed the majority of traffic to be switched, instead of routed, which reduced latency in legacy networks.
The 80/20 design allowed VLANs to be trunked across the entire campus network, a concept known as end-to-end VLANs:



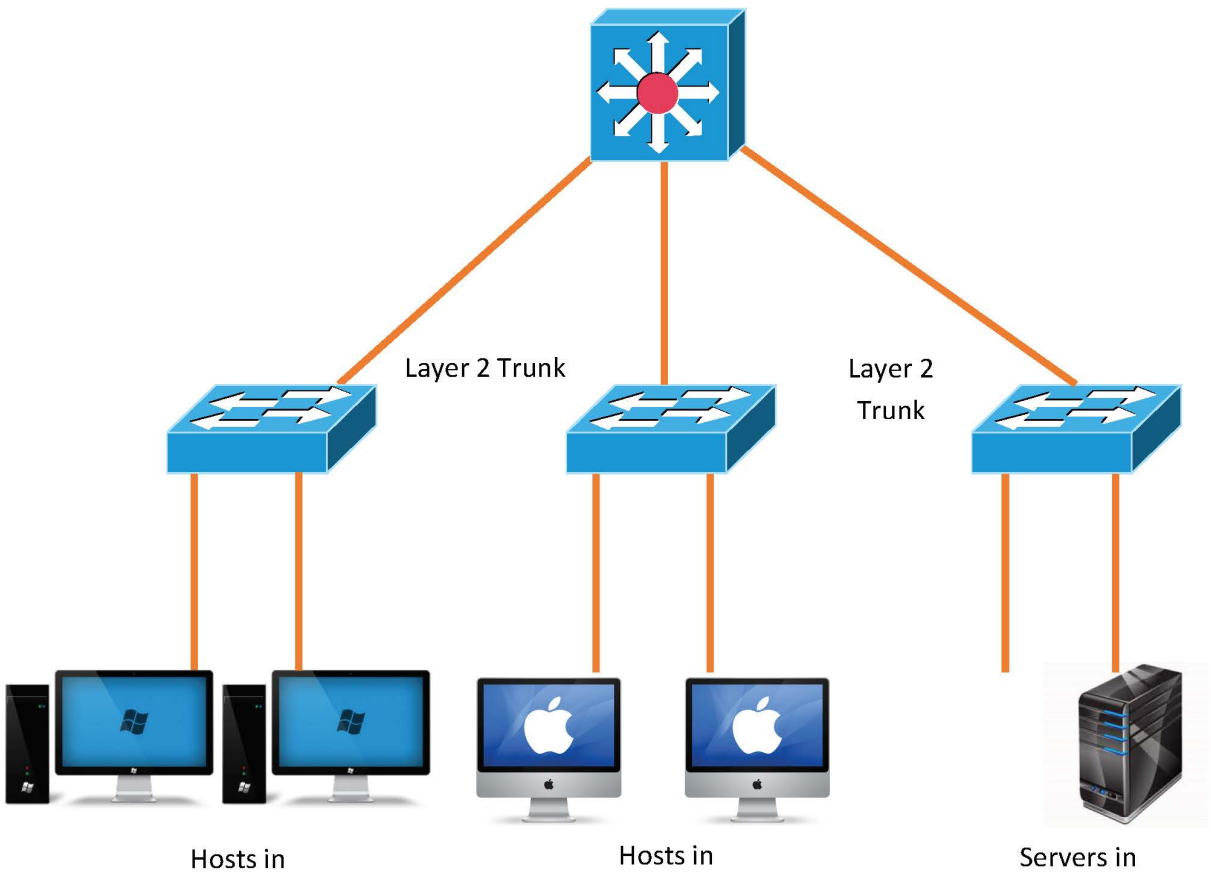
End-to-end VLANs allow a host to exist anywhere on the campus network, while maintaining Layer-2 connectivity to its resources. However, this flat design poses numerous challenges for scalability and performance:

- STP domains are very large, which may result in instability or convergence issues.
- Broadcasts proliferate throughout the entire campus network.
- Maintaining end-to-end VLANs adds administrative overhead.
- Troubleshooting issues can be difficult.

As network technology improved, centralization of resources became the dominant trend. Modern networks adhere to the 20/80 design:

- 20 percent of traffic should remain on the local network.
- 80 percent of traffic should be routed to a remote network.

Instead of placing workgroup resources in every local network, most organizations centralize resources into a datacenter environment. Layer-3 switching allows users to access these resources with minimal latency. The 20/80 design encourages a local VLAN approach. VLANs should stay localized to a single switch or switch block:



This design provides several benefits:

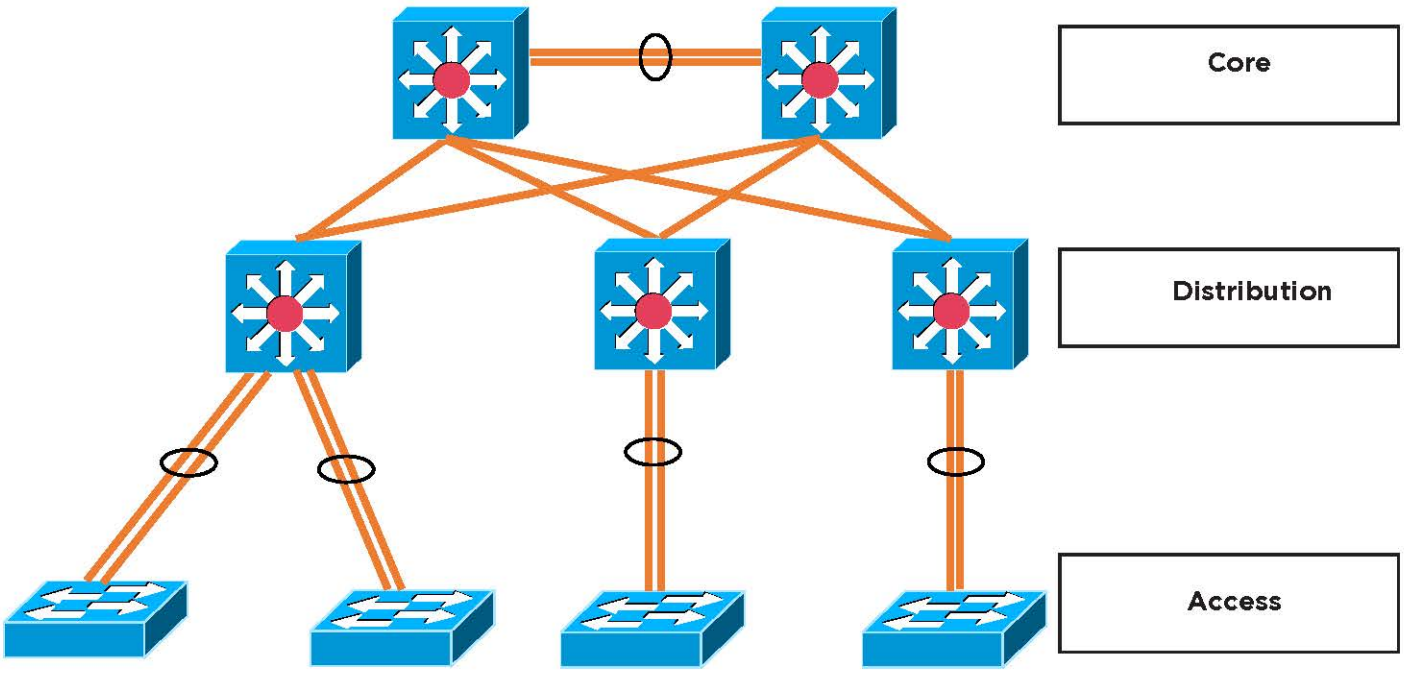
- STP domains are limited, reducing the risk of convergence issues.
- Broadcast traffic is isolated within smaller broadcast domains.
- Simpler, hierarchical design improves scalability and performance.
- Troubleshooting issues is typically easier.

There are nearly no drawbacks to this design, outside of a legacy application requiring Layer-2 connectivity between users and resources. In that scenario, it's time to invest in a better application.

The Cisco Hierarchical Network Model To aid in designing scalable networks, Cisco developed a hierarchical

network model, which consists of three layers:

- Access layer
- Distribution layer
- Core layer



The **access layer** is where users and hosts connect into the network. Switches at the access layer typically have the following characteristics:

- High port density
- Low cost per port
- Scalable, redundant uplinks to higher layers
- Host-level functions such as VLANs, traffic filtering, and QoS

In an 80/20 design, resources are placed as close as possible to the users that require them. Thus, most traffic will never need to leave the access layer. In a 20/80 design, traffic must be forwarded through higher layers to reach centralized resources.

The **distribution layer** is responsible for aggregating access layer switches, and connecting the access layer to the core layer. Switches at the distribution layer typically have the following characteristics:

- Layer-3 or multilayer forwarding
- Traffic filtering and QoS
- Scalable, redundant links to the core and access layers

Historically, the distribution layer was the Layer-3 boundary in a hierarchical network design:

The connection between access and distribution layers was Layer-2.

The distribution switches are configured with VLAN SVIs.

Hosts in the access layer use the SVIs as their default gateway. This remains a common design today.

However, **pushing Layer-3 to the access-layer** has become increasingly prevalent. VLAN SVIs are configured on the access layer switch, which hosts will use as their default gateway.

A routed connection is then used between access and distribution layers, further minimizing STP convergence issues and limiting broadcast traffic.

The core layer is responsible for connecting all distribution layer switches. The core is often referred to as the network backbone, as it forwards traffic from to every end of the network. Switches at the core layer typically have the following characteristics:

- High-throughput Layer-3 or multilayer forwarding
- Absence of traffic filtering, to limit latency
- Scalable, redundant links to the distribution layer and other core switches Advanced QoS functions

Proper **core layer** design is focused on speed and efficiency. In a 20/80 design, most traffic will traverse the core layer. Thus, core switches are often the highest-capacity switches in the campus environment.

Smaller campus environments may not require a clearly defined core layer separated from the distribution layer. Often, the functions of the core and distribution layers are combined into a single layer. This is referred to as a collapsed core design.

Virtual LANs (VLANs) and VTP

Collision vs. Broadcast Domains

A **collision domain i**s simply defined as any physical segment where a **collision** can occur. Hubs can only operate at half-duplex, and thus all ports on a hub belong to the same collision domain.

Layer-2 switches can operate at full duplex. Each individual port on a switch belongs to its own collision domain. Thus, Layer-2 switches create **more collision domains**, which results in **fewer collisions.**

Like hubs though, Layer-2 switches belong to only one broadcast domain. A Layer-2 switch will forward both broadcasts and multicasts out every port but the originating port.

Only Layer-3 devices separate **broadcast domains**. Because of this, Layer-2 switches are poorly suited for large, scalable networks. The Layer-2 header provides no mechanism to differentiate one network from another, only one host from another.

Virtual LANs (VLANs)

By default, a switch will forward both broadcasts and multicasts out every port but the originating port. However, a switch can be logically segmented into separate broadcast domains, using Virtual LANs (or VLANs). Each VLAN represents a unique broadcast domain:

- Traffic between devices within the same VLAN is switched.
- Traffic between devices in different VLANs requires a Layer-3 device to communicate.

Broadcasts from one VLAN will not be forwarded to another VLAN. The logical separation provided by VLANs is **not a Layer-3 function.** VLAN tags are inserted into the **Layer-2 header.**

Thus, a switch that supports VLANs is not necessarily a Layer-3 switch. However, a purely Layer-2 switch cannot route between VLANs.

Remember, though VLANs provide separation for Layer-3 broadcast domains, they are still a Layer-2 function. A VLAN often has a direct relationship with an IP subnet, though this is not a requirement.

Advantages of VLANs VLANs provide the several benefits:

- **Broadcast Control** – eliminates unnecessary broadcast traffic, improving network performance and scalability.
- **Security** – logically separates users and departments, allowing administrators to implement access-lists to control traffic between VLANs.
- **Flexibility** – removes the physical boundaries of a network, allowing a user or device to exist anywhere.

VLANs are very common in LAN and campus networks. For example, user networks are often separated from server networks using VLANs. VLANs can span across WANs as well, though there are only limited scenarios where this is necessary or recommended.

VLAN Port Types A VLAN-enabled switch supports two types of ports:

- **Access ports**
- **Trunk ports**

An access port is a member of only a single VLAN. Access ports are most often used to connect host devices, such as computers and printers. By default on Cisco switches, all switch ports are access ports.

Any host connected to an access port immediately becomes a member of the VLAN configured on that port. This is transparent to the host - it is unaware that it belongs to a VLAN.

It is possible for a VLAN to span more than one switch. There are two methods of connecting a VLAN across multiple switches:

- Create uplink access ports between the switches, one for each VLAN.
- Create a trunk connection between the switches.

A trunk port is not a member of a single VLAN. The traffic from any or all VLANs can traverse trunk links to reach other switches.

VLAN Frame-Tagging

When VLANs span multiple switches, a mechanism is required to identify which VLAN a frame belongs to. This is accomplished through frame tagging, which places a VLAN ID in each frame.

Tagging only occurs when a frame is sent out a trunk port. Traffic sent out access ports is never tagged.

Frame Tagging Protocols

Cisco switches support two frame tagging protocols:

- Inter-Switch Link (ISL)
- IEEE 802.1Q

The tagging protocol can be manually specified on a trunk port, or dynamically negotiated using Cisco's proprietary Dynamic Trunking Protocol (DTP).

VLAN Trunking Protocol (VTP)

Maintaining a consistent VLAN database can be difficult in a large switching environment.

Cisco's proprietary VLAN Trunking Protocol (VTP) simplifies this management - updates to the VLAN database are propagated to all switches using **VTP advertisements.**

VTP requires that all participating switches join a **VTP domain**. Switches must belong to the same domain to share VLAN information, and a switch can only belong to a single domain.

VTP Versions

There are three versions of VTP. VTP version 1 supports the standard 1 – 1005 VLAN range. VTP version 1 is also default on Catalyst switches. VTP version 2 introduces some additional features:

- Token Ring support
- VLAN consistency checks
- Domain-independent transparent pass through

VTPv1 and v2 are **not compatible.** The VTP version is dictated by the VTP server, discussed in detail shortly. If the VTP server is configured for VTPv2, all other switches in the VTP domain will change to v2 as well. Until recently, VTP Version 3 was supported on only limited Cisco switch platforms. VTPv3 was built to be flexible, and can forward both VLAN and other database information, such as Multiple Spanning Tree (MST) protocol.

Other enhancements provided by VTPv3 include:

- Support for the extended 1006-4094 VLAN range.
- Support for private VLANs.
- Improved VTP authentication.

- Protection from accidental database overwrites, by using VTP primary and secondary servers.
- Ability to enable VTP on a per-port basis.

VTP Modes

A switch using VTP must operate in one of three modes:

- Server
- Client
- Transparent

VTP servers are responsible for creating, deleting, or modifying entries in the VLAN database. Each VTP domain must have at least one VTP server, and this is the default mode for Cisco switches.

Servers advertise the VLAN database to all other switches in the VTP domain, including other servers. VTP servers can only advertise the standard 1-1005 VLAN range, and advertisements are only sent out trunk ports.

VTP clients cannot modify the VLAN database, and rely on advertisements from other switches to update VLAN information. A client will also forward VTP advertisements out every trunk port.

Remember: switches must be in the same VTP Domain to share and accept updates to the VLAN database. Only servers can change the VLAN database.

A VTP transparent switch maintains its own local VLAN database, and does not directly participate in the VTP domain.

A transparent switch will never accept VLAN database information from another switch, even a server. Also, a transparent switch will never advertise its local VLAN database to another switch.

Transparent switches will **pass through** advertisements from other switches in the VTP domain.

The VTP version dictates how the pass through is handled:

- **VTP version 1** – the transparent switch will only pass through advertisements from the same VTP domain.
- **VTP version 2** – the transparent switch will pass through advertisements from any VTP domain.

VTP Advertisements - Revision Number

Recall that updates to the VLAN database are propagated using VTP advertisements. VTP advertisements are always sent out trunk ports, on VLAN 1.

VTP advertisements are marked with a 32-bit configuration revision number, to identify the most current VLAN database revision. Any change to the VLAN database increments the configuration revision number by 1. Thus, a higher number represents a newer database revision. A switch will only accept an advertisement if the revision number is higher than the current VLAN database. Advertisements with a lower revision number are ignored.

Important note: While only VTP servers can change the VLAN database, VTP clients can advertise updates, to other clients and even to a server! As long as the revision number is higher, the switch will accept the update.

This can result in a newly-introduced switch advertising a blank or incorrect VLAN database to all other switches in the domain. Switch ports would then lose their VLAN memberships, resulting in a significant network outage.

This can be avoided when implementing a new switch into the VTP domain. Best practice is to configure a new switch as a VTP client, and reset its revision number to zero before deploying into a production network.

There are two methods of resetting the revision number to zero on a switch:

- Change the VTP domain name, and then change it back to the original name.

- Change the VTP mode to transparent, and then change it back to either server or client. Transparent switches always a revision number of 0.

VTP has fallen out of favor, due to the risk of an unintentional overwrite of the VLAN database. Until very recently, Cisco did not support VTP on the Nexus platform of switches.

VTPv3 directly addresses this risk through the use of VTP **primary** and s**econdary** servers. Only the primary server is allowed to update the VLAN database on other switches. Only one primary server is allowed per domain.

VTP Advertisements – Message Types

Three message types exist for VTP advertisements:

- Summary Advertisement
- Subset Advertisement
- Advertisement Request

Both VTP servers and clients will send out a summary advertisement every 300 seconds. Summary advertisements contain the following information about the VTP domain:

- VTP version
- Domain name
- Configuration revision number
- Time stamp
- MD5 digest

Summary advertisements are also sent when a change occurs to the VLAN database. The summary is then followed with a subset advertisement, which actually contains the full, updated VLAN database.

A subset advertisement will contain the following information:

- VTP version
- Domain name
- Configuration revision number
- VLAN IDs for each VLAN in the database
- VLAN-specific information, such as the VLAN name and MTU

Important note: Switches will only accept summary and subset advertisements if the domain name and MD5 digest match. Otherwise, the advertisements are ignored.

If a switch receives a summary advertisement with a revision number higher than its own, it will send out an **advertisement request**. VTP servers will then respond with an updated summary and subset advertisement so that the switch can synchronize to the most current VLAN database.

A switch that is reset or newly joined to the VTP domain will also send out an advertisement request.

VTP pruning eliminates unnecessary broadcast or multicast traffic throughout the switching infrastructure.

EtherChannel

An **EtherChannel** consists of individual Fast Ethernet or Gigabit Ethernet links bundled into a single logical link.

If a link within an EtherChannel fails, traffic previously carried over that failed link changes to the remaining links within the EtherChannel. A trap is sent for a failure, identifying the switch, the EtherChannel, and the failed link.

Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other link of the EtherChannel.

NOTE: All interfaces in each Etherchannel must be the same speed and duplex, same trunking encapsulation

or the same access vlan ID, also the STP cost for each port must be the same and none of the Etherchannel ports can be involved in SPAN, RSPAN configuration or neither 802.1X.

Understanding Port-Channel Interfaces

You create an EtherChannel for Layer 2 interfaces differently from **Layer 3** interfaces. Both configurations involve logical interfaces. With Layer 3 interfaces, you manually create the logical interface by **using the interface port-channel** global configuration command. With **Layer 2** interfaces, the logical interface is dynamically created. With both **Layer 3** and **2 i**nterfaces, you manually assign an interface to the EtherChannel by using the channel-group interface configuration command. This command binds the physical and logical ports together

An **Etherchannel cannot** be configured in both the **PAgP** and **LACP** modes.

Chapter 2 – Switch Port Security

Switch Port Security

Port Security adds an additional layer of security to the switching network.

The MAC address of a host generally does not change. If a specific host will always remain connected to a specific switch port, then the switch can filter all other MAC addresses on that port using Port Security.

Port Security supports both **statically** mapping MAC addresses, and **dynamically** learning addresses from traffic sent on the port

Port Security refers to dynamically learned MAC addresses as **sticky** addresses.

A **violation** occurs if an unauthorized MAC address attempts to forward traffic through a port. There are three violation actions a switch can perform:

- **Shutdown** – If a violation occurs, the interface is placed in an errdisable state. The interface will stop forwarding all traffic, including non-violating traffic, until it is removed from an errdisable state. This is the default action for Port Security.
- **Restrict** – If a violation occurs, the interface will remain online. Legitimate traffic will be forwarded, and unauthorized traffic will be dropped. Violations are logged, either via a syslog message or SNMP trap.
- **Protect** – If a violation occurs, the interface will remain online. Legitimate traffic will be forwarded and unauthorized traffic will be dropped, but no logging will occur.

BPDU Gaurd

Port Fast-enabled ports do not receive **BPDUs.** Receiving a BPDU on a Port Fast-enabled port signals an invalid configuration, such as the connection of an unauthorized device, and the BPDU guard feature puts the port in the error-disabled state.

The BPDU guard feature provides a secure response to invalid configurations because you must manually put the port back in service. Use the BPDU guard feature in a service-provider network to prevent an access port from participating in the spanning tree.

Use the **spanning-tree portfast default** global configuration command to globally enable the Port Fast feature on all nontrunking ports. Configure Port Fast only on ports that connect to end stations; otherwise, an accidental topology loop could cause a data packet loop and disrupt switch and network operation. A Port Fast-enabled port moves directly to the spanning-tree forwarding state when linkup occurs without waiting for the standard forward-delay time.

You can also configure bpduguard under an interface using the command **"spanning-tree bpduguard".**

BPDU Filtering

The **BPDU filtering** feature can be globally enabled on the switch or can be enabled per interface, but the feature operates with some differences.

At the **global level**, you can enable BPDU filtering on Port Fast-enabled interfaces by using the **spanning-tree portfast bpdufilter default** global configuration command. This command prevents interfaces that are in a Port Fast-operational state from sending or receiving BPDUs.

The interfaces still send a few BPDUs at link-up before the switch begins to filter outbound BPDUs. You should globally enable BPDU filtering on a switch so that hosts connected to these interfaces do not receive BPDUs. If a BPDU is received on a Port Fast-enabled interface, the interface loses its Port Fast-operational status, and BPDU filtering is disabled.

At the **interface level**, you can enable BPDU filtering on any interface by using the **spanning-tree bpdufilter enable** interface configuration command without also enabling the Port Fast feature. This command prevents the interface from sending or receiving BPDUs.

802.1x Port Authentication

802.1x Port Authentication forces a host device to authenticate with the switch, before the switch will forward traffic on behalf of that host. This is accomplished using the Extensible Authentication Protocol over LANs (EAPOL). 802.1x only supports RADIUS servers to provide authentication.

Both the switch and the host must support 802.1x to use port authentication:

If the host supports 802.1x, but the switch does not – the host will not utilize 802.1x and will communicate normally with the switch.

If the switch supports 802.1x, but the host does not – the interface will stay in an unauthorized state, and will not forward traffic. A switch interface configured for 802.1x authentication stays in an unauthorized state until a client successfully authenticates.

The only traffic permitted through an interface in an unauthorized state is as follows:

- EAPOL, for client authentication
- Spanning Tree Protocol (STP)
- Cisco Discovery Protocol (CDP)

An interface can be configured in one of three 802.1x states:

- **force-authorized** – The interface will always authorize any client, essentially disabling authentication. This is the default state.
- **force-unauthorized** – The interface will never authorize any client, essentially preventing traffic from being forwarded.
- **auto** – The interface will actively attempt to authenticate the client.

DHCP Snooping

Dynamic Host Control Protocol (DHCP) provides administrators with a mechanism to dynamically assign IP addresses, rather than manually configuring the address on each host.

DHCP servers lease out IP addresses to DHCP clients, for a specific period of time. There are four steps to this DHCP process:

- When a DHCP client first boots up, it broadcasts a DHCPDiscover message, searching for a DHCP server.
- If a DHCP server exists on the local segment, it will respond with a DHCPOffer, containing the offered IP address, subnet mask, etc.
- Once the client receives the offer, it will respond with a DHCPRequest, indicating that it will accept the offered protocol information.
- Finally, the server responds with a DHCPACK, acknowledging the clients acceptance of offered protocol information.

Malicious attackers can place a rogue DHCP server on the trusted network, intercepting DHCP packets while masquerading as a legitimate DHCP server. This is a form of spoofing attack, which intends to gain unauthorized access or steal information by sourcing packets from a trusted source. This is also referred to as a **man-in-the-middle attack.**

DHCP attacks of this sort can be mitigated by using **DHCP Snooping.** Only specified interfaces will accept DHCPOffer packets – **unauthorized** interfaces will discard these packets, and then place the interface in an errdisable state.

Dynamic ARP Inspection

Another common man-in-the-middle attack is ARP spoofing, sometimes referred to as ARP poisoning. A malicious host can masquerade as another host, by intercepting ARP requests and responding with its own MAC address.

Dynamic ARP Inspection (DAI) mitigates the risk of ARP Spoofing, by inspecting all ARP traffic on untrusted ports. DAI will confirm that a legitimate MAC-to-IP translation has occurred, by comparing it against a trusted database.

This MAC-to-IP database can be statically configured, or DAI can utilize the DHCP Snooping table, assuming that DHCP Snooping has been enabled.

If an ARP response does not match the MAC-to-IP entry for a particular IP address, then DAI drops the ARP response and generates a log message.

Chapter 3 – Cisco Discovery Protocol (CDP)

Overview

- Provides details about directly connected Cisco devices, such as address, protocol used
- CDP starts automatically by default for IOS 10.3 and later
- CDP operates at Layer 2, so it is not necessary for the neighboring device to be in the same domain, or share a common network address for communication
- Advertisements about neighbors are multicast to the address 0100.0ccc.cccc
- Routes are learned through hello type updates

CDP Parameters

CDP Timer

- How often updates are sent
- Default = 60 seconds
- To change default time
- Router(config)#cdp timer new_update_time

CDP Holdtime

- The time the CDP packet sent should be kept by the receiving router before being discarded
- Default = 180 seconds
- To change default time
- Router(config)#cdp holdtime new_holdtime

Disabling and enabling CDP

To disable CDP

- Router(config)#no cdp enable

To disable CDP on an interface

- Router(config-if)#no cdp enable

To enable CDP

- Router(config)#cdp run

Showing CDP neighbors

For each connected Cisco device, the following information can be displayed

- Device ID router hostname/domain name
- Local port type and # e.g. Ethernet 0/0
- Holdtime
- Device capability e.g. router, switch
- Hardware platform e.g. 2600, 1900
- IOS version
- Neighbour's remote port type and number
- For a brief summary

Router**#show cdp neighbors**

- For detailed information

Router**#show cdp neighbors detail**

- To look at a single device

Router**#show cdp entry** router_name

- To display information about your local router

Router**#show cdp interface**

Link Layer Discovery Protocol (LLDP)

The Link Layer Discovery Protocol (LLDP) is a vendor-neutral link layer protocol in the Internet Protocol Suite used by network devices for advertising their identity, capabilities, and neighbours on an IEEE 802 local area network, principally wired Ethernet.[1]The protocol is formally referred to by the IEEE as Station and Media Access Control Connectivity Discovery specified in IEEE 802.1AB and IEEE 802.3-2012 section 6 clause 79.

LLDP performs functions similar to several proprietary protocols, such as Cisco Discovery Protocol, Foundry Discovery Protocol, Nortel Discovery Protocol and Link Layer Topology Discovery.

Information gathered with LLDP is stored in the device as a management information database (MIB) and can be queried with the Simple Network Management Protocol (SNMP) as specified in RFC 2922. The topology of an LLDP-enabled network can be discovered by crawling the hosts and querying this database. Information that may be retrieved include:

- System name and description
- Port name and description
- VLAN name
- IP management address
- System capabilities (switching, routing, etc.)
- MAC/PHY information
- MDI power
- Link aggregation

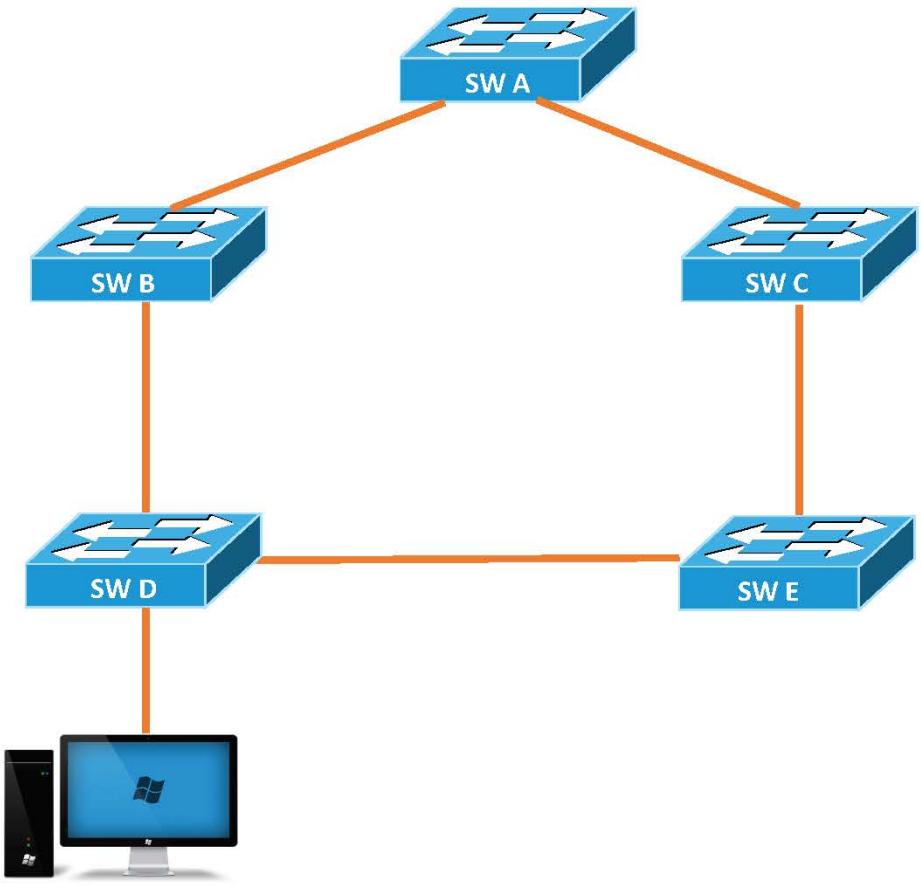
Chapter 4 – Spanning Tree Protocol (STP)

Switching Loops

A Layer-2 switch belongs to only one **broadcast domain,** and will forward both broadcasts and multicasts out every port but the originating port.

When a **switching loop** is introduced into the network, a destructive **broadcast storm** will develop within seconds. A storm occurs when broadcasts are endlessly forwarded through the loop. Eventually, the storm will choke off all other network traffic.

Consider the following example:



Hosts A

If HostA sends out a broadcast, SwitchD will forward the broadcast out all ports in the same VLAN, including the trunk ports connecting to SwitchB and SwitchE. In turn, those two switches will forward that broadcast out all ports, including the trunks to the neighboring SwitchA and SwitchC.

The broadcast will loop around the switches **infinitely.** In fact, there will be two separate broadcast storms cycling in opposite directions through the switching loop. Only powering off the switches or physically removing the loop will stop the storm.

Spanning Tree Protocol (STP) Spanning Tree Protocol (STP) was developed to prevent the broadcast storms caused by switching loops. STP was originally defined in **IEEE 802.1D**.

Switches running STP will build a map or topology of the entire switching network. STP will identify if there are any loops, and then disable or block as many ports as necessary to eliminate all loops in the topology.

A blocked port can be reactivated if another port goes down. This allows STP to maintain redundancy and fault-tolerance.

However, because ports are blocked to eliminate loops, STP does not support load balancing unless an EtherChannel is used. EtherChannel is covered in great detail in another guide.

STP switches exchange Bridge Protocol Data Units (BPDU's) to build the topology database. BPDU's are forwarded out all ports every two seconds, to a dedicated MAC multicast address of **0180.c200.0000.**

Building the STP topology is a multistep convergence process:

- A Root Bridge is elected
- Root ports are identified
- Designated ports are identified
- Ports are placed in a blocking state as required, to eliminate loops

The **Root Bridge** serves as the central reference point for the STP topology. STP was originally developed when Layer-2 bridges were still prevalent, and thus the term Root Bridge is still used for nostalgic reasons. It is also acceptable to use the term **Root Switch**, though this is less common.

Once the full topology is determined, and loops are eliminated, the switches are considered converged. STP is enabled by default on all Cisco switches, for all VLANs.

Electing an STP Root Bridge

The first step in the STP convergence process is electing a Root Bridge, which is the central reference point for the STP topology. As a best practice, the Root Bridge should be the most centralized switch in the STP topology. A Root Bridge is elected based on its Bridge ID, comprised of two components in the original 802.1D standard:

- **16-bit Bridge priority**
- **48-bit MAC address**

The default priority is **32,768**, and the lowest priority wins. If there is a tie in priority, the **lowest MAC address** is used as the **tie-breaker.**

STP Port States

As STP converges the switching topology, a switch port will progress through a series of states:

- **Blocking**
- **Listening**
- **Learning**
- **Forwarding**

Initially, a switch port will start in a blocking state:

- A blocking port will not forward frames or learn MAC addresses.
- A blocking port will still listen for BPDUs from other switches, to learn about changes to the switching topology.

A port will then transition from a blocking to a listening state:

- The switch must believe that the port will not be shut down to eliminate a loop. In other words, the port may become a root or designated port.
- A listening port will not forward frames or learn MAC addresses.
- A listening port will send and listen for BPDUs, to participate in the election of the Root Bridge, root ports, and designated ports.
- If a listening port is not elected as a root or a designated Port, it will transition back to a blocking state.

If a listening port is elected as a root or designated port, it will transition to a learning state:

- A port must wait a brief period of time, referred to as the forward delay, before transitioning

from a listening to learning state.

- A learning port will continue to send and listen for BPDUs.
- A learning port will begin to add MAC addresses to the CAM table.
- However, a learning port cannot forward frames quite yet.

Finally, a learning port will transition to a **forwarding** state:

- A port must wait another forward delay before transitioning from learning to forwarding.

vA forwarding port is fully functional – it will send and listen for BPDUs, learn MAC addresses, and forward frames.

- Root and designated ports will eventually transition to a forwarding state.

Technically, there is a **fifth port state – disabled.** A port in a disabled state has been **administratively shutdown.** A disabled port does not forward frames or participate in STP convergence.

STP Timers

Switches running STP exchange BPDUs to build and converge the topology database. There are three **timers** that are crucial to the STP process:

- Hello timer
- Forward delay timer
- Max age timer

The **hello timer** determines how often switches send BPDUs. By default, BPDUs are sent every **2 seconds.**

The forward delay timer determines how long a port must spend in both a learning and listening state:

- Introducing this delay period ensures that STP will have enough time to detect and eliminate loops.
- By default, the forward delay is 15 seconds.
- Because a port must transition through two forward delays, the total delay time is 30 seconds.

The **max age timer** indicates how long a switch will retain BPDU information from a neighbor switch, before discarding it:

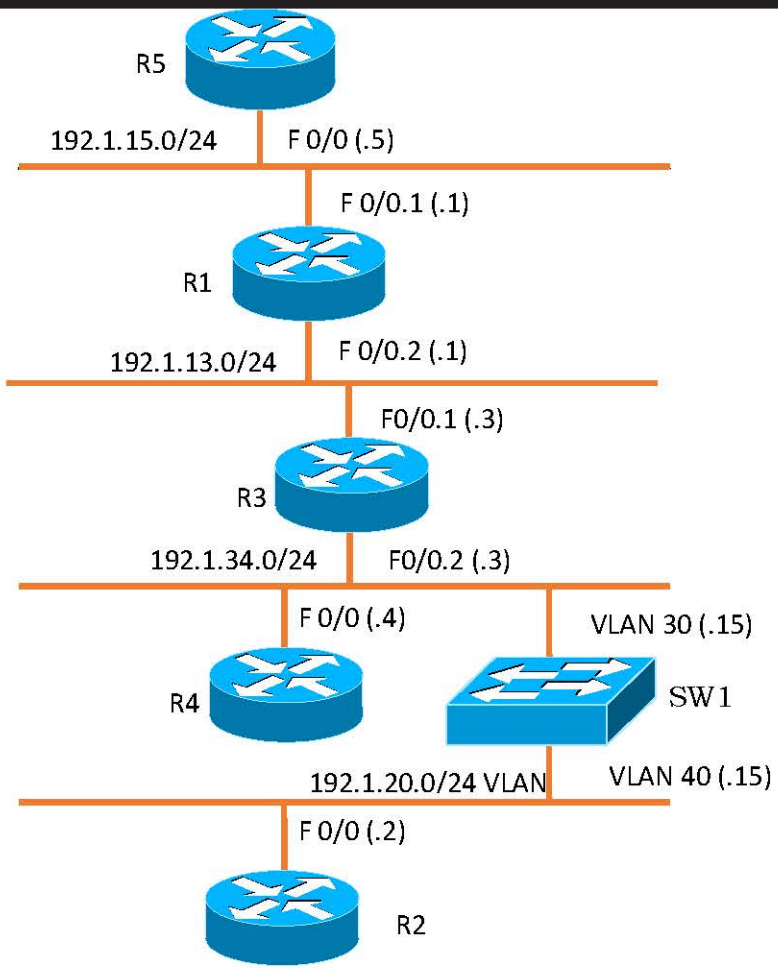
- Remember that BPDUs are sent every two seconds.
- If a switch fails to receive a BPDU from a neighboring switch for the max age period, it will assume there was a change in the switching topology.
- STP will then purge that neighbor's BPDU information.
- By default, the max age timer is 20 seconds.

Timer values can be adjusted. However, this is rarely necessary, and can negatively impact STP performance and reliability.

Timers must be changed on the Root Bridge. The Root Bridge will propagate the new timer values to all switches using BPDUs. Non-root switches will ignore their locally configured timer values.

Note: Trainers are advised to demonstrate the STP on live switches and clearly demonstrate the STP port states and port role.

Lab 1- VTP, Trunking, VLANs and Inter-VLAN Routing



Task 1

Configure the Switches with Hostnames of SW1 and SW2 respectively.

SW1	SW2
Hostname SW1	Hostname SW2

Task 2

Configure both switches to be in a VTP Domain CISCO. SW1 should be configured as a Server and SW2 as a Client.

SW1	SW2
VTP domain CISCO VTP mode server	VTP domain CISCO VTP mode client

Task 3

The CISCO VTP Domain should be password protected using NETMET as the Password.

SW1	SW2
VTP password KBITS	VTP password KBITS

Task 4

Configure Trunking between SW1 and SW2 on all ports that connect the switches to each other. Use an Industry standard encapsulation mechanism.

SW1	sw2
interface range f0/xx – fast0/yy switchport trunk encap dot1q switchport mode trunk !	interface range f0/xx – fast0/yy switchport trunk encap dot1q switchport mode trunk !

Task 5

Create VLAN's based on the Diagram. Assign the appropriate ports to the appropriate VLAN's based on the Logical Diagram. Use an industry standard encapsulation wherever required.

```
SW1
VLAN 10
VLAN 20
VLAN 30
VLAN 40
!
interface f0/1 , f0/3
switchport trunk encap dot1q
 switchport mode trunk
!
interface f0/2
switchport access vlan 40
 switchport mode access
!
interface f0/4
 switchport access vlan 30
 switchport mode access
!
interface f0/5
 switchport access vlan 10
 switchport mode access
```

Task 6

Configure the Routers and SW1 with the Appropriate IP addresses based on the Logical Diagram.

```
R1
!
interface f 0/0
 no shut
!
interface f 0/0.1
 encapsulation dot1q 10
 ip address 192.1.15.1 255.255.255.0
!
interface f0/0.2
 encapsulation dot1q 20
 ip address 192.1.13.1 255.255.255.0
```

```
R2

interface f 0/0
 ip address 192.1.20.2 255.255.255.0
 no shut
!
```

R3 ! interface f 0/0 no shut ! interface f 0/0.1 encapsulation dot1q 20 ip address 192.1.13.3 255.255.255.0 ! interface f0/0.2 encapsulation dot1q 30 ip address 192.1.34.3 255.255.255.0 !	R4 ! interface f 0/0 ip address 192.1.34.4 255.255.255.0 no shut !
R5 ! interface f 0/0 ip address 192.1.15.5 255.255.255.0 no shut !	SW1 ! ip routing ! interface vlan 30 ip address 192.1.34.15 255.255.255.0 ! interface vlan 40 ip address 192.1.20.15 255.255.255.0 !

Task 7

Configure a Loopback 0 interface on each Rotuer with an IP Address of X.X.X.X/8 (where X is the Router # - R1=1, R2=2 ....). Loopback 0 on SW1 as 15.15.15.15/8. Run RIP v2 on all the routers and SW1 such that all networks are reachable from all devices.

R1 ! interface loopback 0 ip address 1.1.1.1 255.0.0.0 ! router rip version 2 no auto-summary network 192.1.15.0 network 192.1.13.0 network 1.0.0.0 !	R2 ! interface loopback 0 ip address 2.2.2.2 255.0.0.0 ! router rip version 2 no auto-summary network 192.1.20.0 network 2.0.0.0 !
R3 ! interface loopback 0 ip address 3.3.3.3 255.0.0.0 ! router rip version 2 no auto-summary network 192.1.13.0 network 192.1.34.0 network 3.0.0.0 !	R4 ! interface loopback 0 ip address 4.4.4.4 255.0.0.0 ! router rip version 2 no auto-summary network 192.1.34.0 network 4.0.0.0 !

R5 ! interface loopback 0 ip address 5.5.5.5 255.0.0.0 ! router rip version 2 no auto-summary network 192.1.15.0 network 5.0.0.0 !	SW1 ! interface loopback 0 ip address 15.15.15.15 255.0.0.0 ! router rip version 2 no auto-summary network 192.1.34.0 network 192.1.20.0 network 15.0.0.0 !

Task 7

Verify by following commands

```
R1#show ip route

R1#ping 2.2.2.2 source 1.1.1.1
```

Lab 2 – Configuring Etherchannels

Task 1

Configure the Trunk Ports connecting SW1 and SW2 to be part of an Etherchannel. The Etherchannel should use an Industry standard protocol.

SW1 ! inteface f0/xx , f0/yy channel-group 1 mode active !	SW2 ! inteface f0/xx , f0/yy channel-group 1 mode active !

Task 2

Configure the Load Balancing mechanism method to be done based on a combination of the Source and Destination IP.

```
SW1
!
port-channel load-balance src-dst-ip
```

```
SW2
!
port-channel load-balance src-dst-ip
```

Task 3

```
Verify the Etherchannel status.
SW1
show etherchannel 1 port-channel

Port-channels in the group:
---------------------------
Port-channel: Po1 (Primary Aggregator)
---------------------------
Age of the Port-channel = 00d:00h:01m:09s
Logical slot/port = 1/0 Number of ports = 0
HotStandBy port = null
Port state = Port-channel Ag-Not-Inuse
Protocol = LACP
```

Lab 3 – Configuring SPAN/RSPAN

Task 1

There is a protocol analyzer connected to SW2 port F0/18. You received a request to monitor and analyze all packets for port F0/16 on SW1, configure the switches to accommodate this request.

```
SW1
!
vlan 90
 remote-span
!
monitor session 1 source interface f0/16
monitor session 1 destination remote vlan 90
!
```

```
SW2
!
monitor session 1 source vlan 90
monitor session 1 destination interface f 0/18
!
```

Lab 4 - Configuring Port Fast

Task 1

Configure the port range from F0/1 – 6 on SW1 in a way that, the link will come up as soon as someone plug in a network cable into some of these ports bypassing STP learning/listening states.

```
SW1
!
interface range f0/1 - 6
spanning-tree portfast
!
```

Output of command:

```
SW1
!
show spanning-tree interface f0/1 portfast
VLAN10 enabled
!
```

Lab 5 - Configuring Port Security

Task 1

Configure VLAN 50 on SW1. Configure Ports F 0/3 and F0/4 on SW2 in VLAN 50. Configure SW2 such that only R3 F 0/1 and R4 F 0/1 can connect to ports F 0/3 and F0/4 on SW2 respectively. If another port tries to connect to these ports, the ports should be error disabled.

```
SW1
VLAN 50
```

```
SW2
!
interface f 0/3
 switchport mode access
 switchport access vlan 50
 switchport port-security
 switchport port-security mac xxxx.xxxx.xxxx
!
interface f 0/4
 switchport mode access
 switchport access vlan 50
 switchport port-security
 switchport port-security mac xxxx.xxxx.xxxx
!
```

Task 2

Configure F 0/5 – F 0/8 in VLAN 50 on SW2. Enable Port Security for these ports such that only 1 MAC address can be connected to them. You would like to learn the MAC address dynamically.

```
SW2
!
interface range f 0/5 – f 0/8
 switchport mode access
 switchport access vlan 50
 switchport port-security
 switchport port-security mac-address sticky
!
```

Task 3

Configure F 0/15 also in VLAN 50 on SW2. Enable Port security for these ports such that 5 MAC addresses can be connected to this port. The first 2 MAC addresses that are allowed to connect are 0001.1010.AB12 and 0001.1010.AB13. The remaining 3 can be learned dynamically.

```
SW2
interface f 0/15
 switchport mode access
 switchport access vlan 50
 switchport port-security
 switchport port-security max 5
 switcport port-security mac-address 0001.1010.ab12
 switcport port-security mac-address 0001.1010.ab13
 switcport port-security mac-address sticky
!
```

Task 4

Verify by testing different mac-addresses.

Task 5

Use following show command to verify the port-security

```
SW2#show switchport port-security
. . .
(Ask your trainer to explain output)
SW2#show switchport port-security interface f 0/15
. . .
(Ask your trainer to explain output)
```

Lab 6 - Configuring BPDU Guard

Task 1

The IT departament just found out that someone in the lobby area just plugged in a switch into port F0/6 on SW1. Configure a command globally on SW1 that if someone connects a hub or a switch to any of the access ports, the port will be disabled. Also make sure that after 4 minutes the disabled port comes up automatically

```
SW1
!
spanning-tree portfast bpduguard
!
errdisable recovery cause bpduguard
errdisable recovery interval 240
!
```

Output of command:

```
SW1
show errdisable recovery
ErrDisable Reason Timer Status
----------------- --------------
udld Disabled
bpduguard Enabled
rootguard Disabled
pagp-flap Disabled
dtp-flap Disabled
link-flap Disabled
Timer interval: 240 seconds
Interfaces that will be enabled at the next timeout:
show spanning-tree summary
Root bridge for: VLAN1, VLAN10, VLAN13, VLAN16, VLAN19, VLAN20, VLAN30
PortFast BPDU Guard is enabled
UplinkFast is disabled
BackboneFast is disabled
...
```

Lab 7 – Configuring BPDU Filter

Task 1

Configure SW2 port F0/15 such that this port won't send or receive any BDPU packets.

```
SW2
!
interface f0/15
 spanning-tree bpdufilter enable
!
```

Task 2

Configure SW1 such that any port configured with portfast should be limited

from sending or receiving BPDU. Don't use any interface level command to accomplish this.

```
SW1
!
spanning-tree portfast bpdufilter default
!
```

Output of command

```
SW1
show spanning-tree summary
Switch is in pvst mode
Root bridge for: none
EtherChannel misconfig guard is enabled
Extended system ID is enabled
Portfast Default is disabled
PortFast BPDU Guard Default is disabled
Portfast BPDU Filter Default is enabled
Loopguard Default is disabled
UplinkFast is disabled
BackboneFast is disabled
Configured Pathcost method used is short
```

MODULE 17

INFRASTRUCTURE MANAGEMENT

Chapter 1 – Router Logging

Router Logging

SNMP Simple Network Management Protocol (SNMP) allows for centralized administration of all network resources. SNMP relies on the use of traps, or criteria for logging specific informational or critical events. Cisco developed **Remote Monitoring (RMON)** to utilize SNMP on routers and switches. RMON alarms identify a specific occurrence on a device. RMON events can then be configured to perform an action once the alarm is tripped.

Configuring RMON Alarms and Events

To configure an RMON alarm:

```
Router(config)#rmon event 1 log trap SNMPCOM description ERROR owner KBITS
```

To configure an RMON event once the ALARM is triggered:

```
Router(config)#snmp-server community SNMPCOM ro
Router(config)#snmp-server host 192.1.12.100 SNMPCOM
Router(config)#snmp-server enable traps
Router(config)#snmp-server trap-source f0/0
Router(config)#snmp-server packetsize 1450
Router(config)#snmp-server location Delhi, India
Router(config)#snmp-server contact Khawar Butt
Router(config)#snmp-server enable traps ospf errors
Router(config)#snmp-server enable traps frame-relay
```

To force RMON to capture every packet coming inbound on an interface:

```
Router(config)# interface f0/0
Router(config-if)# rmon promiscuous
```

Configuring a SYSLOG Server

To direct all logging and debugging information to a centralized syslog server:

```
Router(config)# logging 192.1.12.101
Router(config)# logging buffered
```

Syslog servers used UDP port 514. When using a logging server, it is important to configure the router to include time and date stamps with the logs:

```
Router(config)# service timestamps log datetime localtime msec show-timezone Router(config)#
service timestamps debug datetime localtime msec show-timezone
```

To identify what messages to actually log:

```
Router(config)# logging trap 5
```

The default value is 7, which logs informational and everything else.

Other logging commands include:

```
Router(config)# logging facility LOCAL6
Router(config)# logging source-interface f0/0
```

Troubleshooting **RMON, SNMP,** and Logging Important show commands would include:

```
Router# show snmp
Router# show rmon events
Router# show logging
```

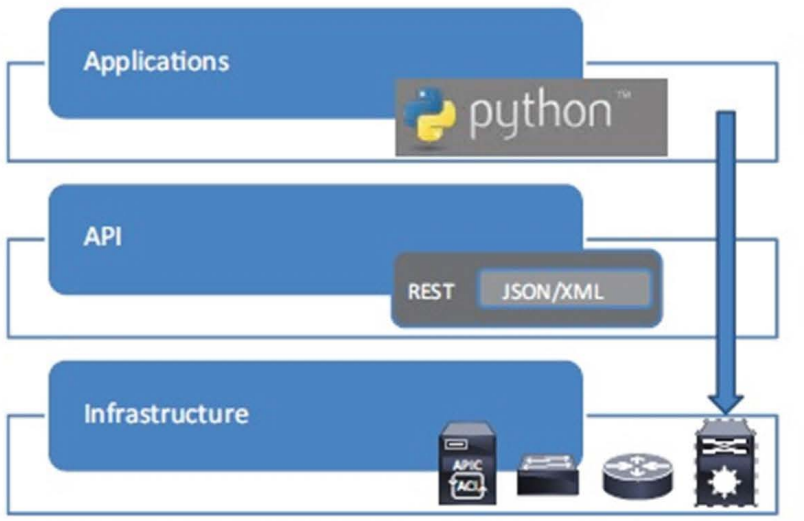
Chapter 2 – SDN (Network Programmability in Enterprise)

Network Programmability Basics

Network programmability is a set of tools to deploy, manage, and troubleshoot a network device. A programmability-enabled network is driven by intelligent software that can deal with a single node or a group of nodes or even or address the network a single unified element. The tool chain uses application programming interfaces or APIs, which serve as the interface to the device or controller. The tool chain also utilizes software that uses the API to gather data or intelligently build configurations.

The term network programmability can have different meanings, depending on perspective. To a network engineer, programmability means interacting with a device or group of devices (driving configurations, troubleshooting, etc.) with a software that sits (logically) above the device. To a developer, network programmability means abstracting the network such that it appears as a single device that can be manipulated with specialized software or within existing software. Both perspectives are correct and drive toward the same goals of using the network to enhance and secure application delivery.

The software component of network programmability can encompass different purposes and either run on the device (on-box) or remotely (off-box). Software built to interact with the network and can address how and/or why the interaction is required. In the case of driving configurations, "How" software addresses the specific device changes when a human determines that configuration is required. "Why" software adds intelligence to automatically react to network or external events, for example, a WAN outage or sudden influx of traffic.



In some cases, the software will be purposely built to interact with the network—for example, day zero deployments or a component of a larger applications, such as Microsoft SharePoint. Figure above describes the relationship between software, the API, and the network.

Network Programmability Benefits

Some of the benefits of network programmability include:

- Time and money cost savings
- Customization
- Reduction of human error
- Innovation

Simplified Networking

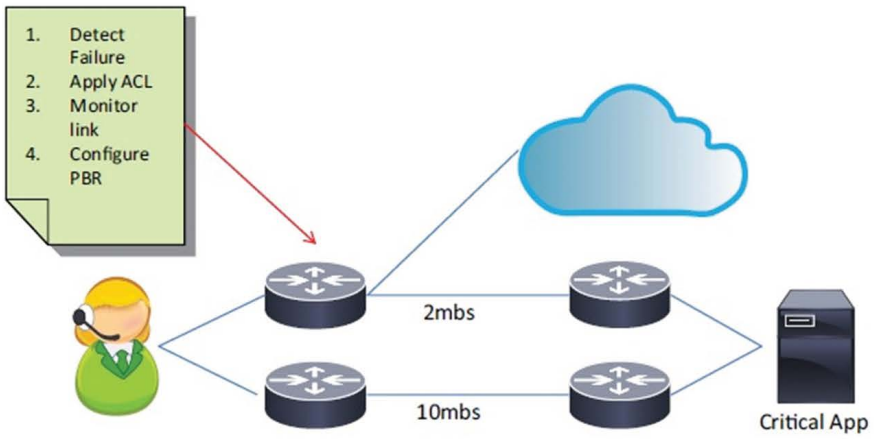
The network is a distributed system, and every new feature required adds configuration complexity and more operational risk. Complexity can lead to increased cost and increased outages. Today a high percentage of the "network down" events are due to misconfiguration (human error). Managing the network programmatically simplifies network management by reducing system variance with automated configurations and streamlined troubleshooting.

For example, **Quality of Service (QoS)** is a critical business feature that, due to complexity, is not commonly configured and is frequently configured incorrectly. A simple change, such as adding a new application, requires a human to access every network device and make a configuration change. Network programmability can simplify QoS deployment and configuration by using a simple application to quickly deliver consistent and accurate configuration changes. Simplified networking reduces man-hours spent operating the network to time spent innovating with the network.

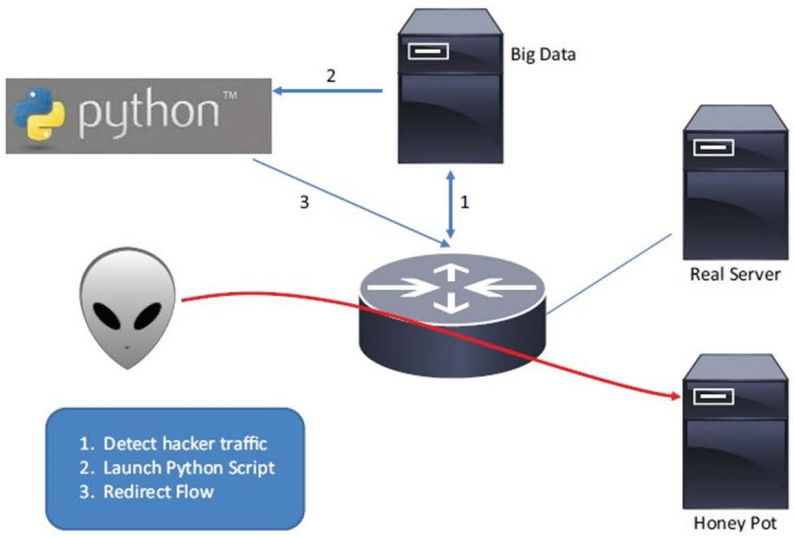
Network Innovation with Programmability

Network programmability can transition the network from simple transport to a sourceof innovation. Today the network is a distributed system of single-purpose appliances that, in most cases, has excess resources. Network programmability unlocks these resources to solve business problems and enhance the application experience. The network is the most logical place for innovation because it has holistic application visibility with distributed policy enforcement. This allows the network to consistently check and adjust the user's application experience. Examples of network innovation include abnormal traffic detection, for example, a deep packet inspection (DPI) tool driving QoS policy, custom integrations to critical mission applications, and automated response to link failure.

Many organizations use redundant WAN links; however, due to cost considerations, the redundant link may only provide enough bandwidth for critical application traffic. Distributed network protocols, for example, OSPF or BGP, are excellent at re-routing traffic in the event of failure, but they generally cannot differentiate between highpriority and low-priority traffic. Network programmability enables automatic and intelligent configuration changes to be also based on business priority. During a failure, a reactive script could change ACLs to block nonpriority traffic from the backup link and even configure policy-based routing to send low-priority traffic over an Internet link, as shown in Figure below:



Network hardware provides a wealth of statistical information that can aid in detecting abnormal traffic or security issues. Data analytic tools can analyze n telemetry from network hardware detect patterns like abnormal flows, for example increased data rate or connections from new countries. If an issue is detected, network programmability can issue an updated QoS configuration to limit data flows or route traffic to a honey pot, as shown in Figure.



Traditionally, Cisco and other network vendors are the innovators in network software and network protocols. If a network administrator needs new features, they are faced with a long process of selling an idea and hoping the vendor will fund its development. Network programmability enables anyone to innovate by creating or extending software without his or her vendor. Network innovation helps organizations differentiate their products and solve problems using the network. Open NX-OS on Cisco Nexus 9000 enables use of the switch as a platform to run user-provided software/applications.

SDN Controllers (or SDN Controllers Platforms)

SDN Controllers (aka SDN Controller Platforms) in a software-defined network (SDN) are the "brains" of the network. It is the application that acts as strategic control point in the SDN network, manage flow control to the switches/routers 'below' **(via southbound APIs)** and the applications and business logic 'above' **(via northbound APIs)** to deploy intelligent networks. Recently, as organizations deploy more SDN networks, the Controllers have been tasked with federating between SDN Controller domains, using common application interfaces, such as OpenFlow and open virtual switch database (OVSDB).

An SDN Controller platform typically contains a collection of "pluggable" modules that can perform different network tasks. Some of the basic tasks including inventorying what devices are within the network and the capabilities of each, gathering network statistics, etc. Extensions can be inserted that enhance the functionality and support more advanced capabilities, such as running algorithms to perform analytics and orchestrating new rules throughout the network.

Two of the most well-known protocols used by SDN Controllers to communicate with the switches/routers is OpenFlow and OVSDB. Others protocols that could be used by an SDN Controller is YANG or NetConf. Other SDN Controller protocols are being developed, while more established networking protocols are

finding ways to run in an SDN environment. For example, the Internet Engineering Task Force (IETF) working group – the Interface to the Routing System (i2rs) – is developing an SDN standard that enables an SDN Controller to leverage proven, traditional protocols, such as OSPF, MPLS, BGP, and IS-IS.

The type of protocols supported can influence the overall architecture of the network – for example, while OpenFlow attempts to completely centralize packet-forwarding decisions, i2rs splits the decision making by leveraging traditional routing protocols to execute distributed routing and allowing applications to modify routing decisions.

SDN Southbound APIs

In a software-defined network (SDN) architecture, southbound application program interfaces (APIs) (or SDN southbound APIs) are used to communicate between the SDN Controller and the switches and routers of the network. They can be open or proprietary.

How Do SDN Southbound APIs Work?

Southbound APIs facilitate efficient control over the network and enable the SDN Controller to dynamically make changes according to real-time demands and needs. OpenFlow, which was developed by the Open Networking Foundation (ONF), is the first and probably most well-known southbound interface. It is an industry standard that defines the way the SDN Controller should interact with the forwarding plane to make adjustments to the network, so it can better adapt to changing business requirements. With OpenFlow, entries can be added and removed to the internal flow-table of switches and potentially routers to make the network more responsive to real-time traffic demands. Besides OpenFlow, Cisco OpFlex (the company's response to OpenFlow) is also a well-known southbound API.

There are a number of switch and router vendors that have announced their support of OpenFlow, including Cisco, Juniper, Big Switch Networks, Brocade, Arista, Extreme Networks, IBM, Dell, NoviFlow, HP, NEC, among others.

While OpenFlow is the most well-known of the SDN protocols for southbound APIs, it is not the only one available or in development. The Network Configuration Protocol (NetConf) uses an Extensible Markup Language (XML) to communicate with the switches and routers to install and make configuration changes; Lisp, also promoted by ONF, is available to support flow mapping. In addition, there are more established networking protocols finding ways to run in an SDN environment, such as OSPF, MPLS, BGP, and IS-IS.

SDN Northbound APIs

In a software-defined network (SDN) architecture, the northbound application program interfaces (APIs) are used to communicate between the SDN Controller and the services and applications running over the network. The northbound APIs can be used to facilitate innovation and enable efficient orchestration and automation of the network to align with the needs of different applications via SDN network programmability.

How Do Northbound APIs Work?

Northbound APIs are arguably the most critical APIs in the SDN environment, since the value of SDN is tied to the innovative applications it can potentially support and enable. Because they are so critical, northbound APIs must support a wide variety of applications, so one size will likely not fit all. This is possibly why SDN northbound APIs are currently the most nebulous component in a SDN environment — a variety of possible interfaces exist in different places up the stack to control different types of applications via an SDN Controller.

It is likely that quite a few different northbound APIs will exist before consolidation occurs – not unlike the early days of the mobile operating system (OS) wars. Examples of the types of network applications that could be optimized via the north bound interface include load balancers, firewalls or other software-defined security (SDSec) services, or orchestration applications across cloud resources.

SDN Northbound APIs are also used to integrate the SDN Controller with automation stacks, such as Puppet, Chef, SaltStack, Ansible and CFEngine, as well as orchestration platforms, such as OpenStack, VMware's vCloudDirector or the open source CloudStack. The goal is to abstract the inner-workings of the network, so that application developers can 'hook' into the network and make changes to accommodate the needs of the application without having to understand exactly what that means for the network.

Recently, the Open Networking Foundation (ONF) turned its focus to the SDN northbound API. They have established a Northbound Working Group that will write code, develop prototypes and look at whether or not to create standards for the interface to drive clarity around what it is and what it can do.

Cisco OpenFlow

Cisco OpenFlow is Cisco's implementation of OpenFlow. OpenFlow is considered the first software-defined networking (SDN) standard, as an open communications protocol in SDNs that enables the SDN Controller to interact with the forwarding plane (switches, routers, etc.) and adapt the network to be responsive to real-time traffic and business requirements.

Cisco has announced support for OpenFlow in the following Cisco products:

- **ISR, ASR, Nexus, and Catalyst Product Lines** – several switching/routing products that work within OpenFlow SDN environments.
- **Cisco Open Network Environment (ONE) Software Controller** – designed to control the network across both Cisco and non-Cisco, OpenFlow-enabled switches to streamline operations, limit costs, and provide a more agile infrastructure.
- **Cisco One Platform Kit (onePK)** – a package of proprietary APIs created by Cisco to enable organizations to create applications to meet their needs.

Cisco has also introduced an alternative protocol to OpenFlow, called **OpFlex.**

OpenFlow Controller

An OpenFlow Controller is a type of SDN Controller that uses the OpenFlow Protocol. An SDN Controller is the strategic point in software-defined network (SDN). An OpenFlow Controller uses the OpenFlow protocol to connect and configure the network devices (routers, switches, etc.) to determine the best path for application traffic. There are also other SDN protocols that a Controller can use such as OpFlex, Yang, and NetConf, to name a few.

SDN Controllers can simplify network management, handling all communications between applications and devices to effectively manage and modify network flows to meet changing needs. When the network control plane is implemented in software, rather than firmware, administrators can manage network traffic more dynamically and at a more granular level. An SDN Controller relays information to the switches/routers (via southbound APIs) and the applications and business logic (via northbound APIs).

In particular, OpenFlow Controllers create a central control point to oversee a variety of OpenFlow-enabled network components. The OpenFlow protocol is designed to increase flexibility by eliminating proprietary protocols from hardware vendors.

Chapter 3 – Router Maintenance Commands

Command	Description
Copy startup-config tftp	Backs up the Startup-config file to a TFTP server
Copy tftp startup-config	Restoring the Startup-config file from a TFTP server
Show Flash	Displays the contents of Flash including the IOS Operating System File.
Copy flash tftp	Backs up the IOS File to a TFTP Server
Copy tftp flash	Upgrades or restores the IOS From a TFTP Server

Tftpdnld	A Rommon mode command used to recover the IOS when it is lost. Requires the setting of the following parameters. (Case-sensitive) IP_ADDRESS=XX.XX.XX.XX IP_SUBNET_MASK=XXX.XXX.XXX.XXX DEFAULT_GATEWAY=XX.XX.XX.XX TFTP_SERVER=XX.XX.XX.XX TFTP_FILENAME=IOS Filename

Lab 1 - Backing up Startup-config to a TFTP Server

Interface IP Address Configuration

R1

Interface	IP Address	Subnet Mask
S 0/0	192.168.1.1	255.255.255.0

PC

IP Address	Subnet Mask	Gateway
192.168.1.254	255.255.255.0	192.168.1.1

On R1

```
enable
config t
hostname R1
!
line console 0
 logging sync
!
no ip domain-lookup
!
interface f 0/0
 ip address 192.168.1.1 255.255.255.0
 no shut
end
!
write
!
```

PC

IP Address : 10.0.0.2
Subnet Mask : 255.0.0.0
Default Gateway : 10.0.0.1

Task 1

Backing up Startup-config to Tftp-Server

- Double-click the **Cisco TFTP Serve**r Icon on your Desktop.
- What is the IP address of the TFTP Server?
- What is the default directory for the TFTP Server?
- Switch to Hyper terminal.
- In Privilege Exec, Ping the IP address of the TFTP Server **(Ping 192.168.1.254)**
- Are you successful?
- Type **copy startup-config tftp**
- Specify the IP address of the TFTP server as the Remote Server.
- Specify **Startup-config** as the destination filename for the file.

Task 2

Verifying the creation of the file

- Open Windows Explorer and browse to the Default TFTP Server folder.
- Do you see the Startup-config file?
- Open it with notepad.

Lab 2 - Restoring the Startup-config from a TFTP Server

(Build on Lab 1)

Task 1

- In Privilege Exec, type e**rases startup-config** to delete the startup-config file.

Task 2

R**estoring Startup-config from the Tftp-Server**

- In Privilege Exec, Ping the IP address of the TFTP Server (**Ping 192.168.1.254)**
- Type **copy tftp startup-config** and follow the prompts to restore the file.

Task 3

Verifying the restoration of the Startup-config file

- In Privilege Exec, type **show start** and check the configuration.

Lab 3 – Backup IOS Using Cisco TFTP Server

(Build on Lab 1)

Task 1

Finding the name of the IOS File

- In Privilege Exec, type Show Flash.
- What is the name of the IOS File?

Task 2

Backing up your IOS to a TFTP Server

- Double-click the Cisco TFTP Server icon on the Desktop, if not already open.
- In Privilege Exec, type Copy flash tftp and follow the prompts using the filename of your IOS.
- Switch to the Cisco TFTP Server program and notice the file being copied.
- Once the copying is done, verify the creation of the file in the default folder for the Cisco TFTP Server

Lab 4 – Upgrading the IOS from a TFTP Server

Task 1

Upgrading IOS from the Tftp-Server

- In Privilege Exec, Ping the IP address of the TFTP Server **(Ping 192.168.1.254)**
- Type **copy tftp flash** and follow the prompts to restore the file.
- Why does it ask you to erase flash before proceeding?

Lab 5 – Recovering IOS from a TFTP Server

Task 1

Simulating a lost or corrupted IOS

- In Privilege Exec, type **erase flash** to delete the flash and simulate a corruption of the IOS.
- Type reload to restart the router.

Task 2

Setting up the TFTP parameters in rommon Mode

- When the router reloads, what mode does it go into and why?

- As the router did not load the **startup-config,** it does not have any IP configuration to connect to the TFTP Server.
- To set IP configuration parameters, use the following commands: (The commands are case-sensitive)
    - **IP ADDRESS=192.168.1.1**
    - **IP_SUBNET_MASK=255.255.255.0**
    - **DEFAULT_GATEWAY=192.168.1.1**
    - **TFTP_SERVER=192.168.1.254**
    - TFTP_FILE=(IOS Filename)

Task 3

Recovering the IOS

- Type **tftpdnld**
- Verify the parameters and type **Y** to start the download.
- Oncae the download is done, reload the router.
- Can you get in?
- Is your old configuration file still valid?
- Why was the configuration file still intact?

MODULE 18

WIRELESS NETWORKS

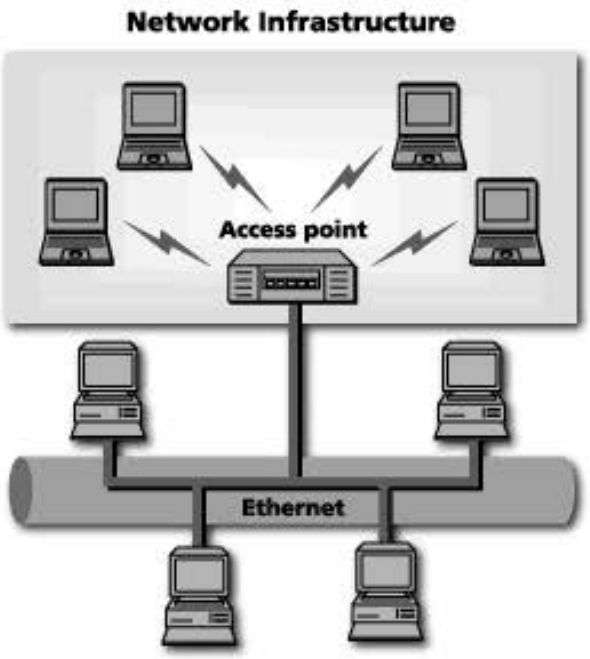
Chapter 1 – introduction wireless networks

Topics to be covered

- Wireless Technology overview
- The IEEE 802.11 WLAN Standards
- Secure Wireless LANs
- Migrating to Wireless LANs (Cutting the cord)
- WLAN, Access point, Wireless Router

Why Wireless ?

- A wireless LAN or WLAN is a wireless local area network that uses radio waves as its carrier.
- The last link with the users is wireless, to give a network connection to all users in a building or campus.
- The backbone network usually uses cables



Common Topologies

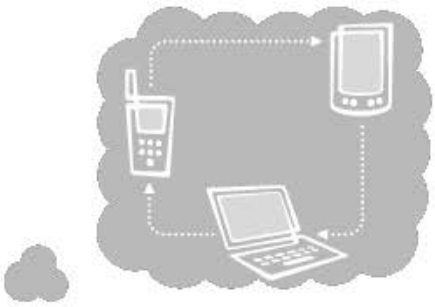
The wireless LAN connects to a wired LAN

- There is a need of an access point that bridges wireless LAN traffic into the wired LAN.
- The access point (AP) can also act as a repeater for wireless nodes, effectively doubling the maximum possible distance between nodes.

Common Topologies

Complete Wireless Networks

- The physical size of the network is determined by the maximum reliable propagation range of the radio signals.
- Referred to as ad hoc networks
- Are self-organizing networks without any centralized control
- Suited for temporary situations such as meetings and conferences.



How do wireless LANs work?

Wireless LANs operate in almost the same way as wired LANs, using the same networking protocols and supporting the most of the same applications.

How are WLANs Different?

- They use specialized physical and data link protocols
- They integrate into existing networks through access points which provide a bridging function
- They let you stay connected as you roam from one coverage area to another
- They have unique security considerations
- They have specific interoperability requirements
- They require different hardware
- They offer performance that differs from wired LANs.

Physical and Data Link Layers

Physical Layer:

The wireless NIC takes frames of data from the link layer, scrambles the data in a predetermined way, then uses the modified data stream to modulate a radio carrier signal.

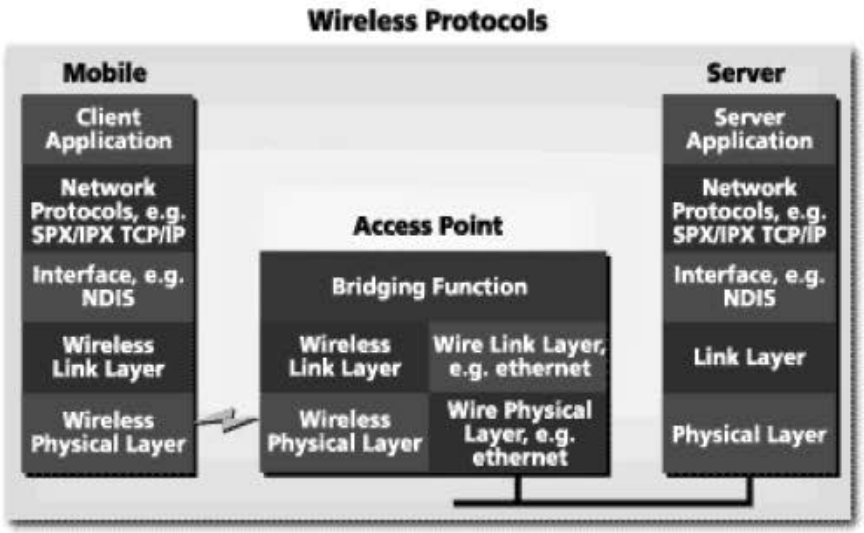
Data Link Layer:

Uses Carriers-Sense-Multiple-Access with Collision Avoidance (CSMA/CA).

Integration With Existing Networks

- Wireless Access Points (APs) - a small device that bridges wireless traffic to your network.
- Most access points bridge wireless LANs into Ethernet networks, but Token-Ring options are available as well.

Integration With Existing Networks



WLAN

A wireless LAN (WLAN) is a wireless computer network that links two or more devices using wireless communication to form a local area network (LAN) within a limited area such as a home, school, computer laboratory, campus, office building etc. This gives users the ability to move around within the area and still be connected to the network. Through a gateway, a WLAN can also provide a connection to the wider Internet.

Access Point

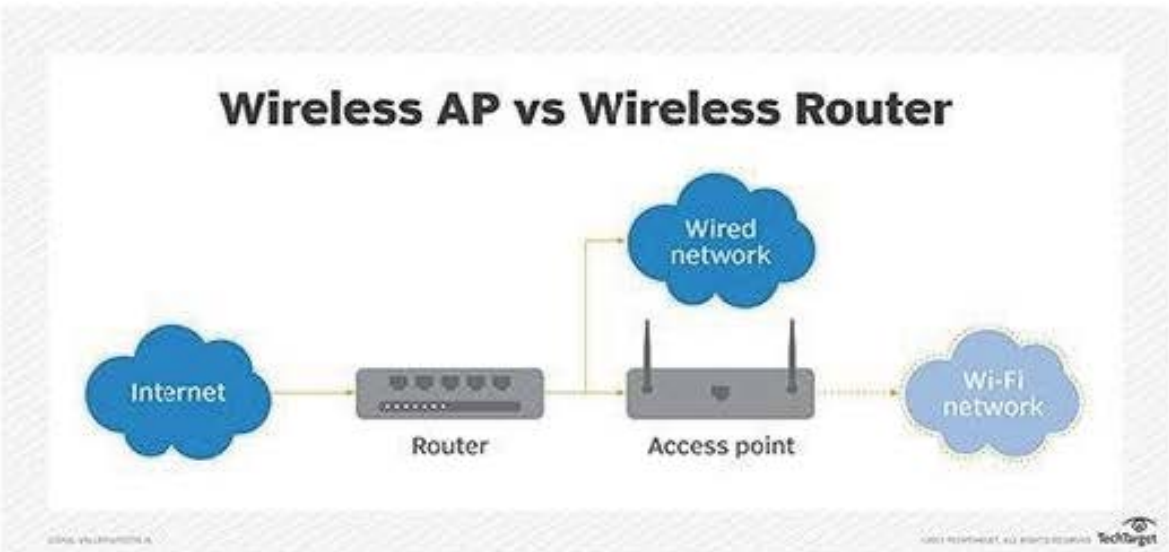
In computer networking, a wireless access point, or more generally just access point, is a networking hardware device that allows other Wi-Fi devices to connect to a wired network. The AP usually connects to a router as a standalone device, but it can also be an integral component of the router itself.

Wireless Router

A wireless router is a device that performs the functions of a router and also includes the functions of a wireless access point. It is used to provide access to the Internet or a private computer network. Depending on the manufacturer and model, it can function in a wired local area network, in a wireless-only LAN, or in a mixed wired and wireless network.

Difference Wireless router and Access point

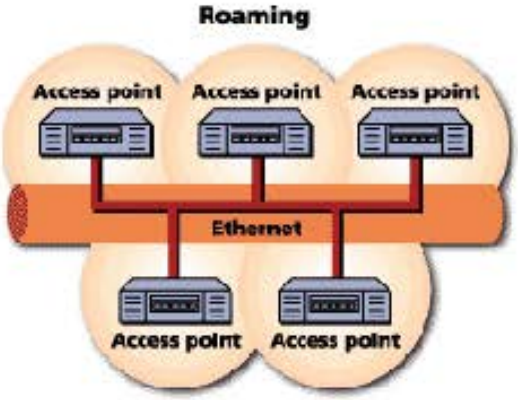
The router acts as a hub that sets up a local area network and manages all of the devices and communication in it. An access point, on the other hand, is a sub-device within the local area network that provides another location for devices to connect from and enables more devices to be on the network.



Chapter 2 – Hardware and IEEE wireless standards

Roaming

- Users maintain a continuous connection as they roam from one physical area to another
- Mobile nodes automatically register with the new access point.
- Methods: DHCP, Mobile IP
- IEEE 802.11 standard does not Address roaming, you may need To purchase equipment from one vendor if your users need to roam from one access point to another.



Security

- In theory, spread spectrum radio signals are inherently difficult to decipher without knowing the exact hopping sequences or direct sequence codes used
- The IEEE 802.11 standard specifies optional security called "Wired Equivalent Privacy" whose goal is that a wireless LAN offer privacy equivalent to that offered by a wired LAN. The standard also specifies optional authentication measures.

Interoperability

- Before the IEEE 802.11 interoperability was based on cooperation between vendors.
- IEEE 802.11 only standardizes the physical and medium access control layers.
- Vendors must still work with each other to ensure their IEEE 802.11 implementations interoperate
- Wireless Ethernet Compatibility Alliance (WECA) introduces the Wi-Fi Certification to ensure cross-vendor interoperability of 802.11b solutions

Hardware

- PC Card, either with integral antenna or with external antenna/RF module.
- ISA Card with external antenna connected by cable.
- Handheld terminals
- Access points

Hardware

CISCO Aironet 350 series

Wireless Handheld Terminal

Semi Parabolic Antenna

BreezeCOM AP

Performance

- 802.11a offers speeds with a theoretically maximum rate of 54Mbps in the 5 GHz band
- 802.11b offers speeds with a theoretically maximum rate of 11Mbps at in the 2.4 GHz spectrum band
- 802.11g is a new standard for data rates of up to a theoretical maximum of 54 Mbps at 2.4 GHz.

What is 802.11?

- A family of wireless LAN (WLAN) specifications developed by a working group at the Institute of Electrical and Electronic Engineers (IEEE)
- Defines standard for WLANs using the following four technologies
  - Frequency Hopping Spread Spectrum (FHSS)
  - Direct Sequence Spread Spectrum (DSSS)
  - Infrared (IR)
  - Orthogonal Frequency Division Multiplexing (OFDM)
- Versions: 802.11a, 802.11b, 802.11g, 802.11e, 802.11f, 802.11i

802.11 - Transmission

- Most wireless LAN products operate in unlicensed radio bands
  - 2.4 GHz is most popular
  - Available in most parts of the world
  - No need for user licensing
- Most wireless LANs use spread-spectrum radio
  - Resistant to interference, secure
  - Two popular methods
    - Frequency Hopping (FH)
    - Direct Sequence (DS)

Frequency Hopping Vs. Direct Sequence

- FH systems use a radio carrier that "hops" from frequency to frequency in a pattern known to both transmitter and receiver
  - Easy to implement
  - Resistance to noise
  - Limited throughput (2-3 Mbps @ 2.4 GHz)
- DS systems use a carrier that remains fixed to a specific frequency band. The data signal is spread onto a much larger range of frequencies (at a much lower power level) using a specific encoding scheme.
  - Much higher throughput than FH (11 Mbps)
  - Better range
  - Less resistant to noise (made up for by redundancy – it transmits at least 10 fully redundant copies of the original signal at the same time)

802.11a

- Employs Orthogonal Frequency Division Multiplexing (OFDM)
  - Offers higher bandwidth than that of 802.11b, DSSS (Direct Sequence Spread Spectrum)
  - 802.11a MAC (Media Access Control) is same as 802.11b
- Operates in the 5 GHz range

802.11a Advantages

- Ultra-high spectrum efficiency
  - 5 GHz band is 300 MHz (vs. 83.5 MHz @ 2.4 GHz)
  - More data can travel over a smaller amount of bandwidth
- High speed
  - Up to 54 Mbps
- Less interference
  - Fewer products using the frequency
    - 2.4 GHz band shared by cordless phones, microwave ovens, Bluetooth, and WLANs

802.11a Disadvantages

- Standards and Interoperability
  - Standard not accepted worldwide
  - No interoperability certification available for 802.11a products
  - Not compatible or interoperable with 802.11b
- Legal issues
  - License-free spectrum in 5 GHz band not available worldwide
- Market
  - Beyond LAN-LAN bridging, there is limited interest for 5 GHz adoption
- Cost
  - 2.4 GHz will still has >40% cost advantage
- Range
  - At equivalent power, 5 GHz range will be ~50% of 2.4 GHz
- Power consumption
  - Higher data rates and increased signal require more power
    - OFDM is less power-efficient then DSSS

802.11a Applications

- Building-to-building connections
- Video, audio conferencing/streaming video, and audio
- Large file transfers, such as engineering CAD drawings
- Faster Web access and browsing
- High worker density or high throughput scenarios
  - Numerous PCs running graphics-intensive applications

802.11a Vs. 802.11b

802.11a vs. 802.11b	802.11a	802.11b
Raw data rates	Up to 54 Mbps (54, 48, 36, 24,18, 12 and 6 Mbps)	Up to 11 Mbps (11, 5.5, 2, and 1 Mbps)
Range	50 Meters	100 Meters
Bandwidth	UNII and ISM (5 GHz range)	ISM (2.4000—2.4835 GHz range)
Modulation	OFDM technology	DSSS technology

802.11g

- 802.11g is a high-speed extension to 802.11b
  - Compatible with 802.11b
  - High speed up to 54 Mbps
  - 2.4 GHz (vs. 802.11a, 5 GHz)
  - Using ODFM for backward compatibility
  - Adaptive Rate Shifting

802.11g Advantages

- Provides higher speeds and higher capacity requirements for applications
  - Wireless Public Access
- Compatible with existing 802.11b standard
- Leverages Worldwide spectrum availability in 2.4 GHz
- Likely to be less costly than 5 GHz alternatives
- Provides easy migration for current users of 802.11b WLANs
  - Delivers backward support for existing 802.11b products
- Provides path to even higher speeds in the future

802.11e Introduces Quality of Service

- Also know as P802.11 TGe
- Purpose: To enhance the 802.11 Medium Access Control (MAC) to improve and manage Quality of Service (QoS)
- Cannot be supported in current chip design
- Requires new radio chips
  - Can do basic QoS in MAC layer

802.11f – Inter Access Point Protocol

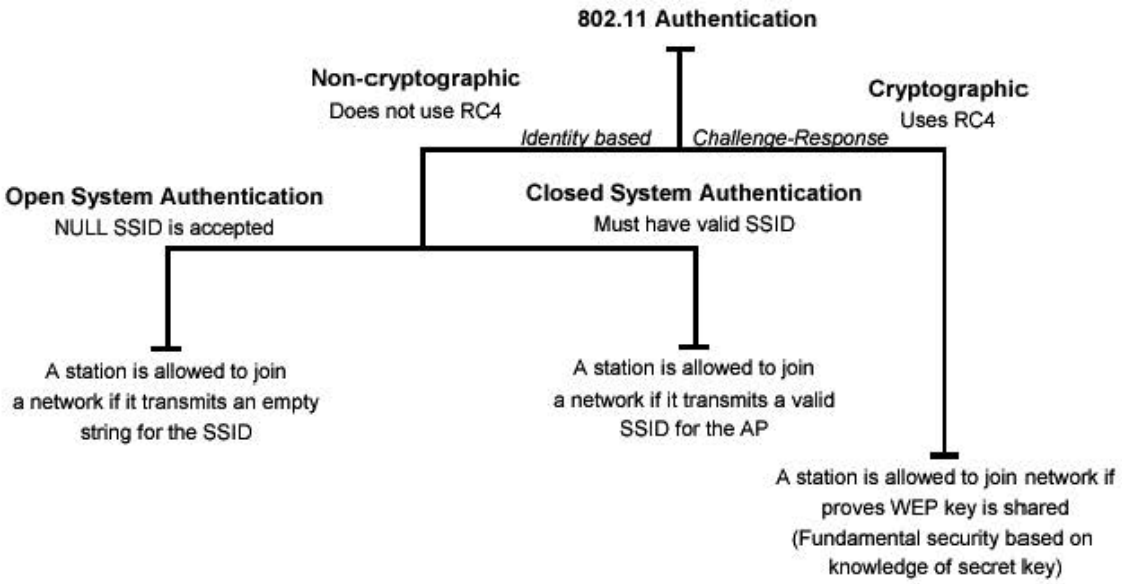
- Also know as P802.11 TGf
- Purpose: To develop a set of requirements for Inter-Access Point Protocol (IAPP), including operational and management aspects

802.11b Security Features

- Wired Equivalent Privacy (WEP) – A protocol to protect link-level data during wireless transmission between clients and access points.
- Services:
  - Authentication: provides access control to the network by denying access to client stations that fail to authenticate properly.
  - Confidentiality: intends to prevent information compromise from casual eavesdropping
  - Integrity: prevents messages from being modified while in transit between the wireless client and the access point.

Chapter 3-Authentication and Wireless topology

Authentication techniques



Privacy

- Cryptographic techniques
- WEP Uses RC4 symmetric key, stream cipher algorithm to generate a pseudo random data sequence. The stream is XORed with the data to be transmitted
- Key sizes: 40bits to 128bits
- Unfortunately, recent attacks have shown that the WEP approach for privacy is vulnerable to certain attack regardless of key size

Data Integrity

- Data integrity is ensured by a simple encrypted version of CRC (Cyclic Redundant Check)
- Also vulnerable to some attacks

Security Problems

- Security features in Wireless products are frequently not enabled.
- Use of static WEP keys (keys are in use for a very long time). WEP does not provide key management.
- Cryptographic keys are short.
- No user authentication occurs – only devices are authenticated. A stolen device can access the network.
- Identity based systems are vulnerable.
- Packet integrity is poor.

Other WLAN Security Mechanisms

- 3Com Dynamic Security Link
- CISCO LEAP - Lightweight Extensible Authentication Protocol

- IEEE 802.1x – Port-Based Network Access Control
- RADIUS Authentication Support
- EAP-MD5
- EAP-TLS
- EAP-TTLS
- PEAP - Protected EAP
- TKIP - Temporal Key Integrity Protocol
- IEEE 802.11i

WLAN Migration – Cutting The Cord

- Essential Questions
- Choosing the Right Technology
- Data Rates
- Access Point Placement and Power
- Antenna Selection and Placement
- Connecting to the Wired LAN
- The Site Survey

Essential Questions

- Why is the organization considering wireless? Allows to clearly define requirements of the WLAN -> development plan
- How many users require mobility?
- What are the applications that will run over the WLAN? Helps to determine bandwidth requirements, a criteria to choose between available technologies. Wireless is a shared medium, not switched!!!

Choose the right technology

- Usually IEEE 802.11b or 802.11a
- 802.11b offers interoperability (WECA Wi-Fi Certification Program)
- 802.11a offers higher data rates (up to 54 mbps) -> higher throughput per user. Limited interoperability.

Data rates

- Data rates affect range
- 802.11b 1 to 11 Mbps in 4 increments
- 802.11a 6 to 54 Mbps in 7 increments
- The minimum data rate must be determined at design time
- Selecting only the highest data rate will require a greater number of APs to cover a specific area
- Compromise between data rates and overall system cost

Access Point Placement and Power

- Typically – mounted at ceiling height.
- Between 15 and 25 feet (4.5m to 8m)
- The greater the height, the greater the difficulty to get power to the unit. Solution: consider devices that can be powered using CAT5 Ethernet cable (CISCO Aironet 1200 Series).
- Access points have internal or external antennas

Antenna Selection and Placement

- Permanently attached.
- Remote antennas connected using an antenna cable.
- Coax cable used for RF has a high signal loss, should not be mounted more than a 1 or 2 meters away from the device.
- Placement: consider building construction, ceiling height, obstacles, and aesthetics. Different materials (cement, steel) have different radio propagation characteristics.

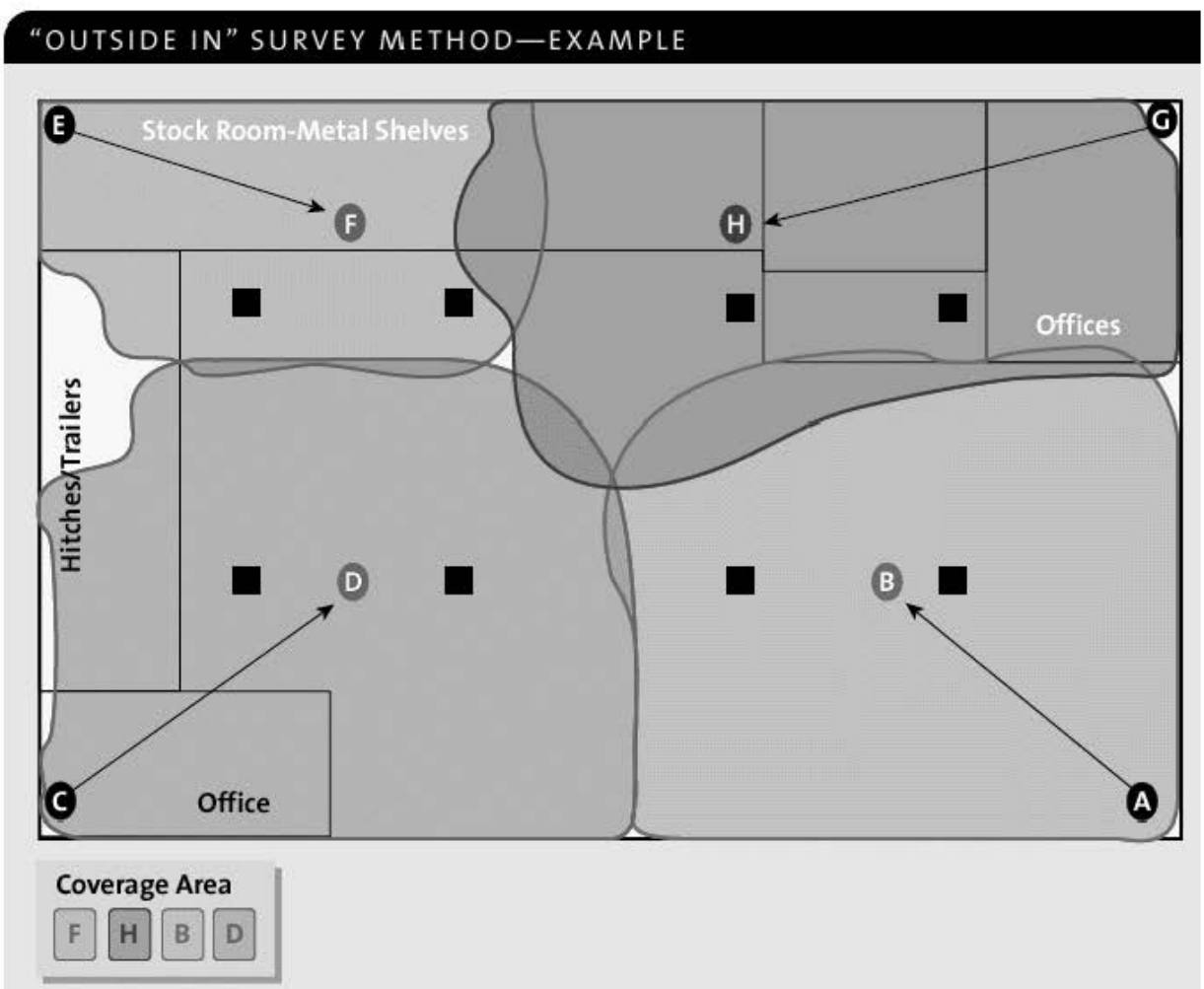
Connecting to the Wired LAN

- Consider user mobility
- If users move between subnets, there are challenges to consider.
- OSes like Windows XP and 2000, Linux support DHCP to obtain the new IP address for the subnet. Certain applications such as VPN will fail.
- Solution: access points in a roaming area are on the same segment.

The Site Survey

- Helps define the coverage areas, data rates, the precise placement of access point.
- Gather information: diagramming the coverage area and measuring the signal strength, SNR (signal to noise ratio), RF interference levels

Site Survey



Chapter 4
Terminology and Explanations

SSID

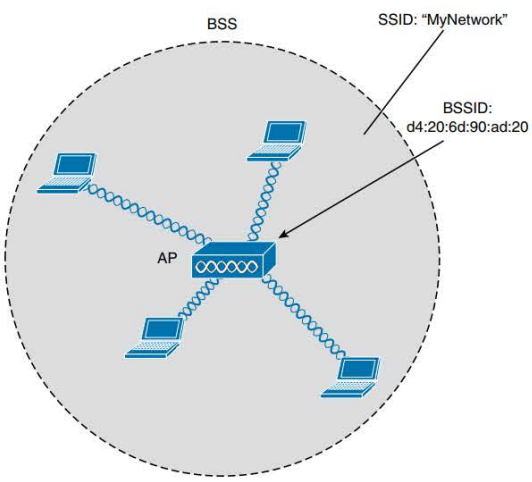
- SSID is short for service set identifier. In layman's terms, an SSID is the name for a Wi-Fi network.
- People typically encounter an SSID most often when they are using a mobile device to connect to a wireless network. For example, if you take your laptop to a coffee shop and attempt to connect to the local Wi-Fi network, your screen will display a list of SSIDs — this is the names of all the networks that are within range of your mobile device. You will select the name of the local network you want to connect to and then enter the password (if necessary) to connect

BSSID

A service set consists of a group of wireless network devices which operates with the same parameters of networking.

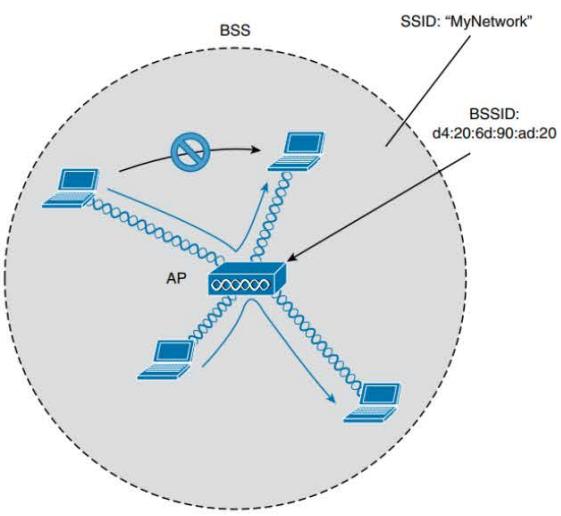
Basic service set identifiers (BSSID) is used to describe sections of a wireless local area network or WLAN.

Basic Service Set



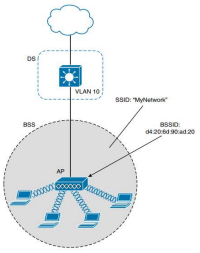
The solution is to make every wireless service area a closed group of mobile devices thatforms around a fixed device; before a device can participate, it must advertise its capabilities and then be granted permission to join. The 802.11 standard calls this a basic service set(BSS). At the heart of every BSS is a wireless access point (AP)

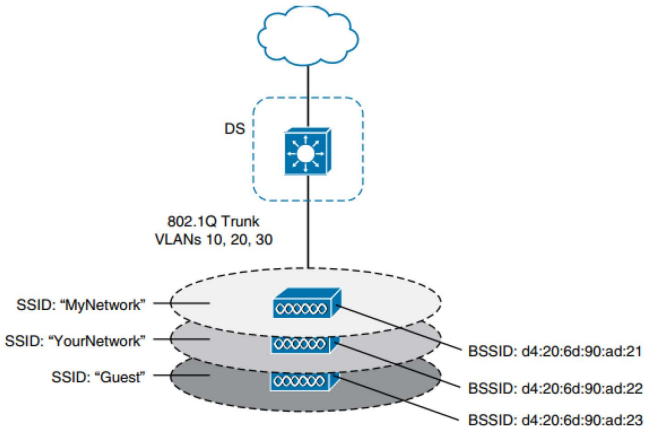
BSS traffic flow



Distribution System

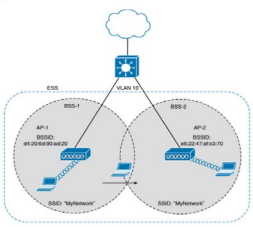
Wireless clients will need to communicate with other devices that are not members of the BSS. Fortunately, an AP can also uplink into an Ethernet network because it has both wireless and wired capabilities. The 802.11 standard refers to the upstream wired Ethernet as the distribution system (DS) for the wireless BSS





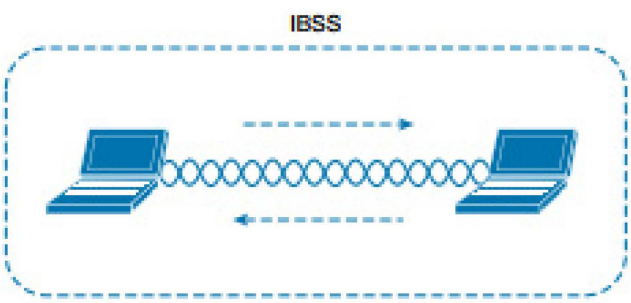
Extended Service Set

When APs are placed at different geographic locations, they can all be interconnected bya switched infrastructure. The 802.11 standard calls this an extended service set (ESS)



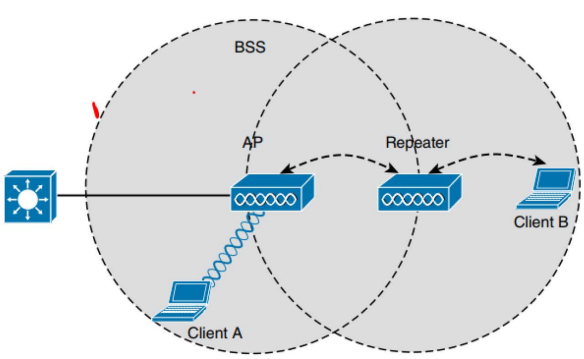
Independent Service Set

The 802.11 standard allows two or more wireless clients to communicate directly with eachother, with no other means of network connectivity. This is known as an ad hoc wireless network, or an independent basic service set (IBSS),



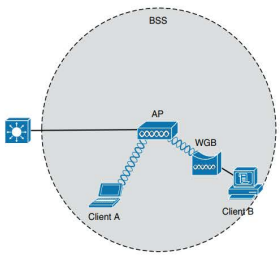
Repeater

Normally, each AP in a wireless network has a wired connection back to the DS or switchedinfrastructure. To extend wireless coverage beyond a normal AP's cell footprint, additionalAPs and their wired connections can be added. In some scenarios, it is not possible to runa wired connection to a new AP because the cable distance is too great to support Ethernetcommunication

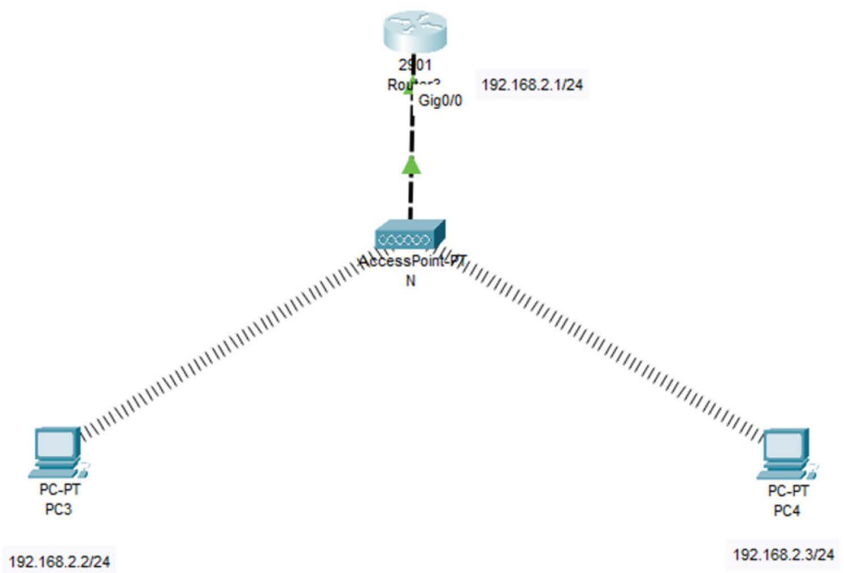


Workgroup Bridge

Suppose you have a device that supports a wired Ethernet link but is not capable of having awireless connection. For example, some mobile medical devices might be designed with onlya wired connection. While it is possible to plug the device into an Ethernet connection whenneeded, a wireless connection would be much more practical. You can use a workgroupbridge (WGB) to connect the device's wired network adapter to a wireless network.



Lab 1 – Communication between two Pcs through A.P. (Access Point) by using DHCP.



Interface IP Address configuration

R1

Interface	IP Address	Subnet Mask
Gig 0/0	192.168.2.1	255.255.255.0

PC1

IP Address	Subnet Mask	Gateway
192.168.2.2	255.255.255.0	192.168.2.1

PC2

IP Address	Subnet Mask	Gateway
192.168.2.3	255.255.255.0	192.168.2.1

Task1

Configure the R1, PC's and AP as per the topology

Task2

Configure the DHCP on router 1 so that pc will get the ip address through DHCP.

On R1

interface GigabitEthernet0/0

ip address 192.168.2.1 255.255.255.0

no shut

!

ip dhcp pool NH2

network 192.168.2.0 255.255.255.0

default-router 192.168.2.1

exit

Task3

Verification on R1

Show ip DHCP Pool NH2 :

Utilization mark (high/low) : 100 / 0

Subnet size (first/next) : 0 / 0

Total addresses : 254

Leased addresses : 2

Excluded addresses : 0

Pending event : none

1 subnet is currently in the pool

Current index IP address range Leased/Excluded/Total

192.168.2.1 192.168.2.1 - 192.168.2.254 2 / 0 / 254

Task4

Lets check the connectivity from pc1 to pc2

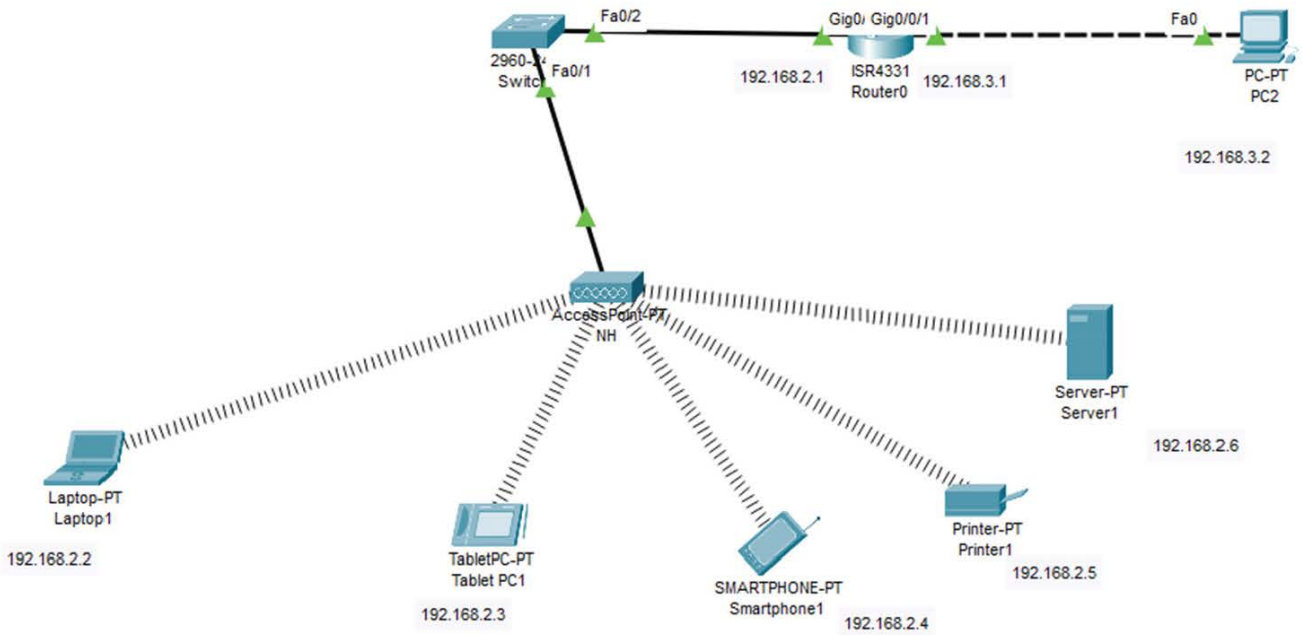
```
Packet Tracer PC Command Line 1.0
C:\>
ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=47ms TTL=128
Reply from 192.168.2.3: bytes=32 time=26ms TTL=128
Reply from 192.168.2.3: bytes=32 time=8ms TTL=128
Reply from 192.168.2.3: bytes=32 time=26ms TTL=128

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 47ms, Average = 26ms
```

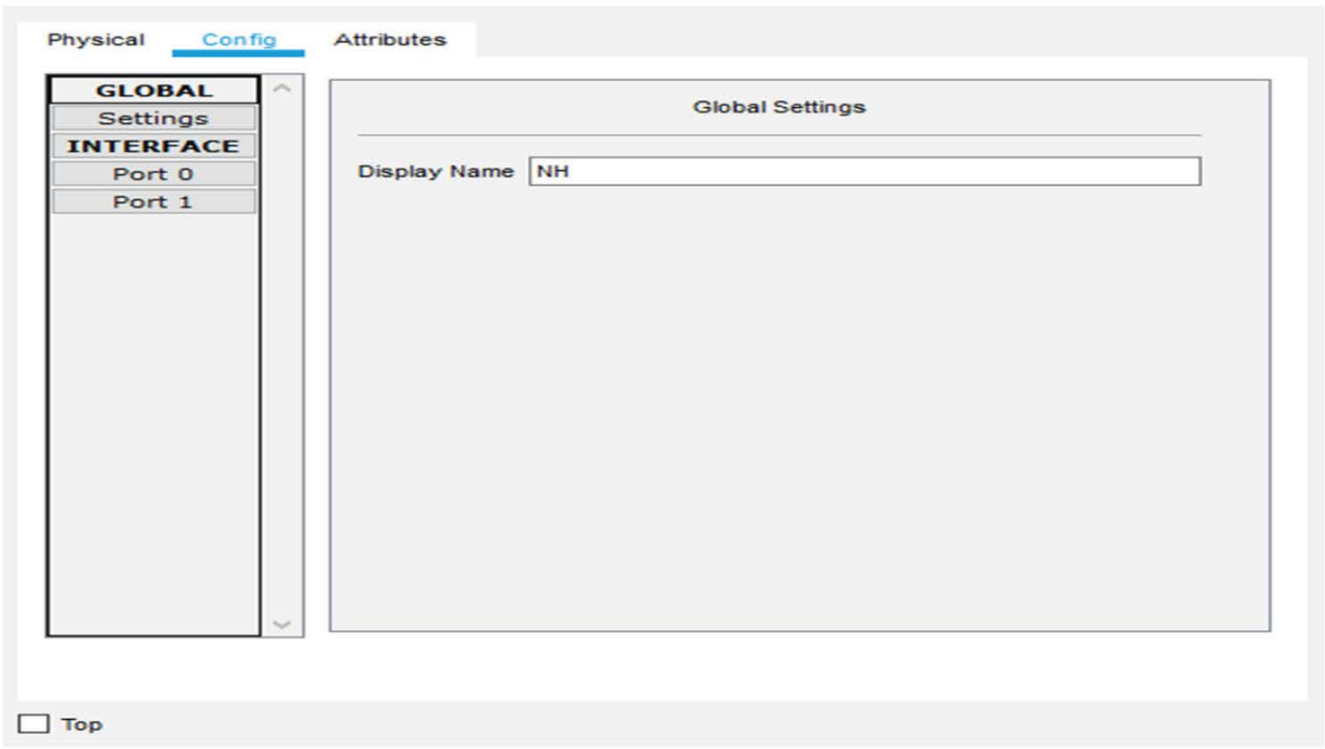
Lab 2:- Communication between different networking devices through A.P. (Access Point) by using DHCP.



Access Point Configuration:

Config<< Global Settings

Change Display Name,Whatever you want to assign

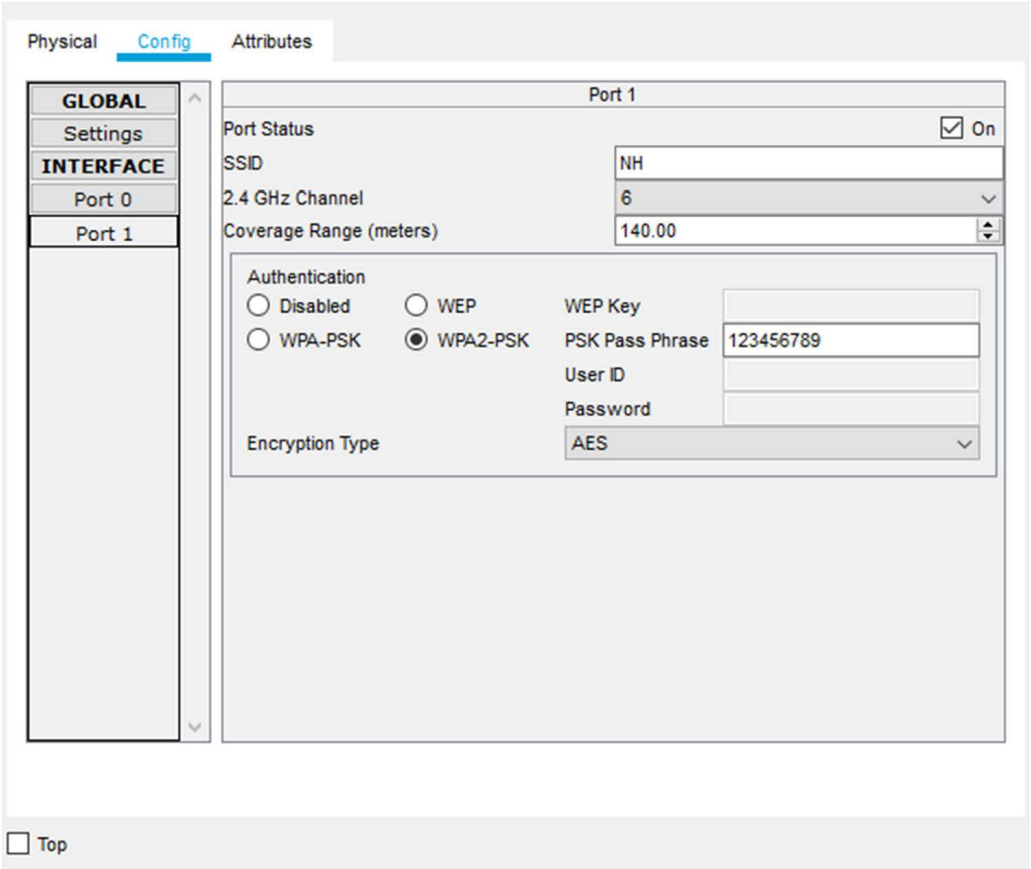


Step 2:

Select Port1

Set SSID Name

select WPAK2-PSK PSK Pass Phrase and set password



Interface IP Address configuration

R1

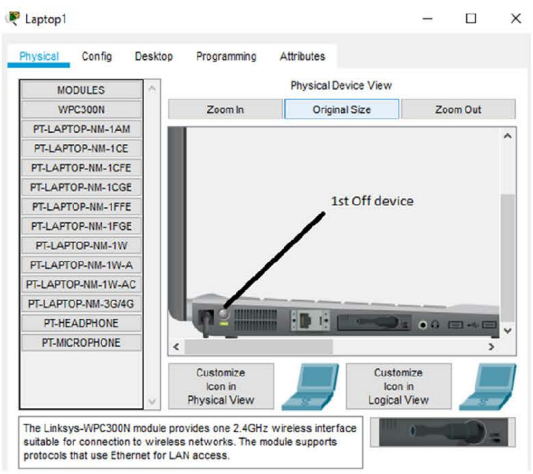
Interface	IP Address	Subnet Mask
Gig 0/0/0	192.168.2.1	255.255.255.0
Gig 0/0/0	192.168.3.1	255.255.255.0

PC2

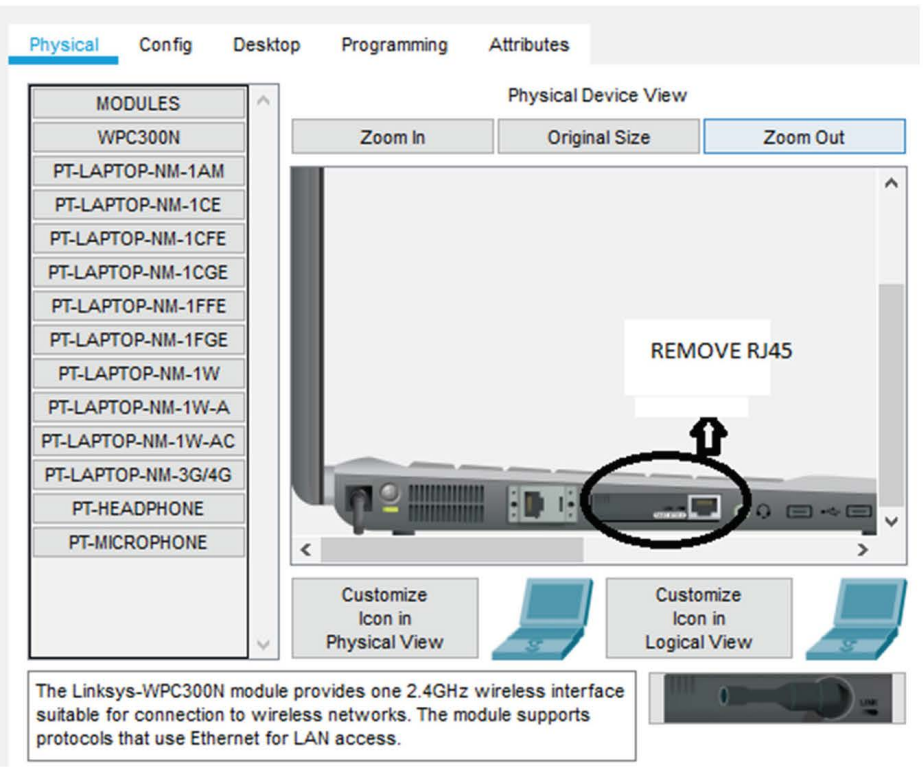
IP Address	Subnet Mask	Gateway
192.168.3.2	255.255.255.0	192.168.2.1

Physically off devices

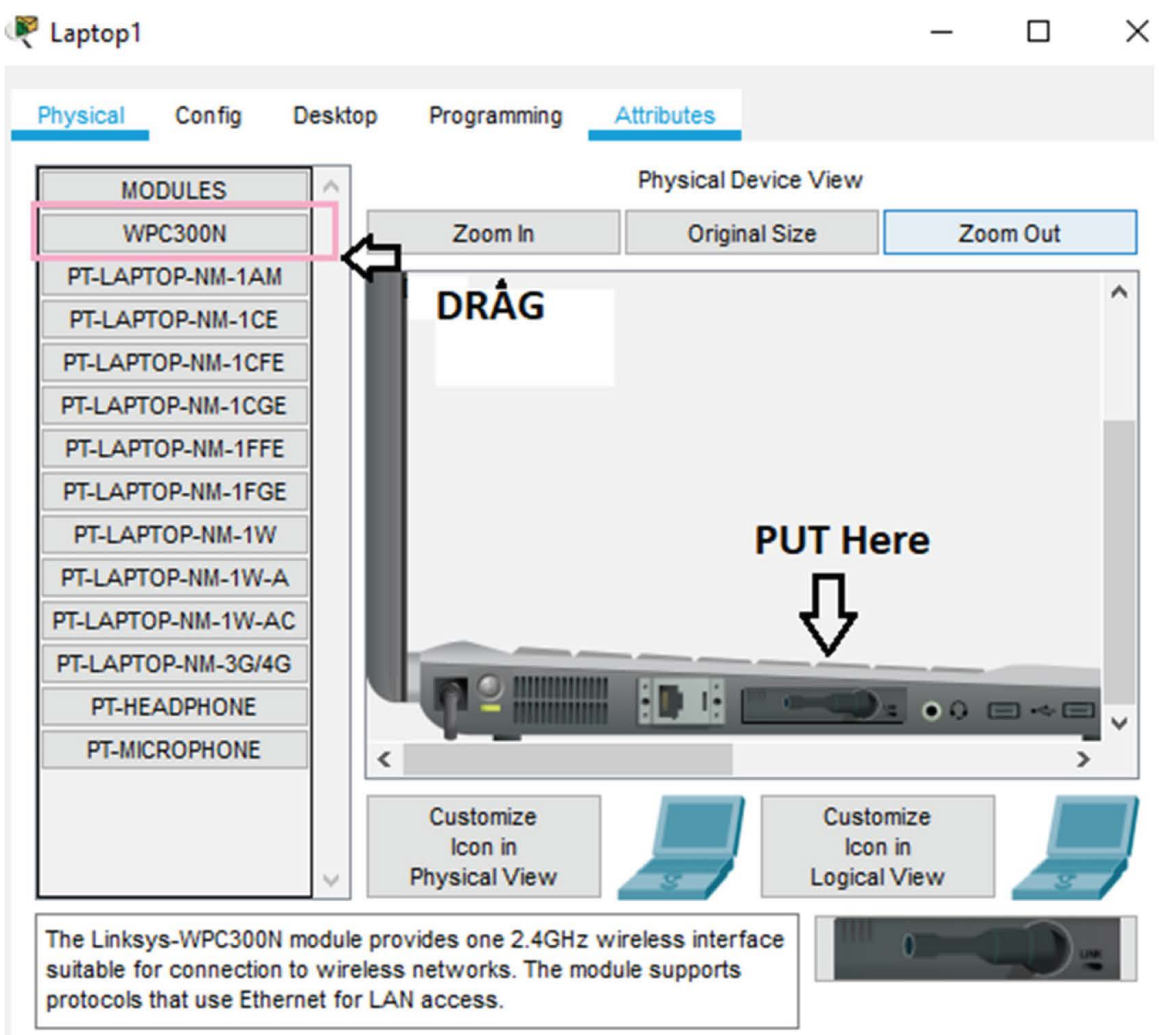
1st step:



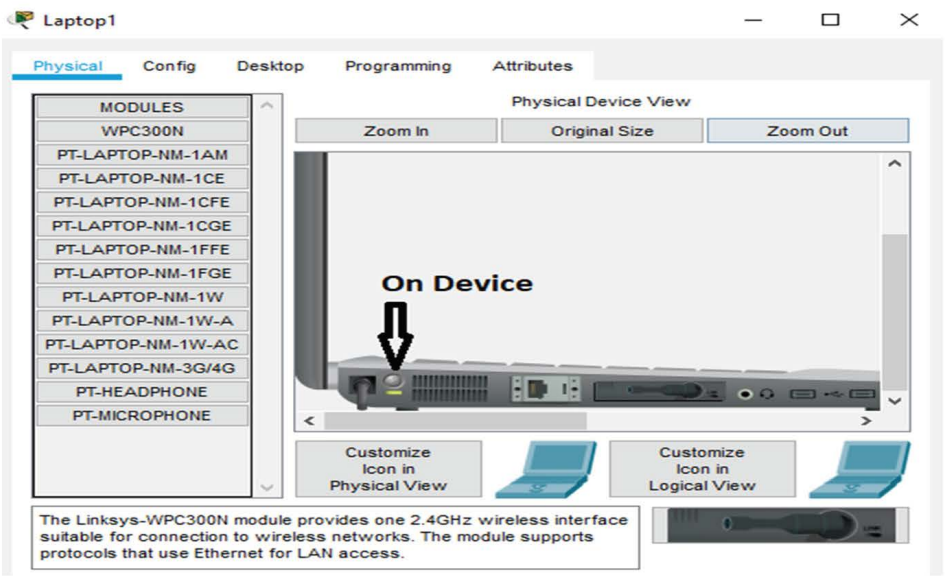
Step 2:
Remove Rj45 Card



Step 3:
Drop WPC300N

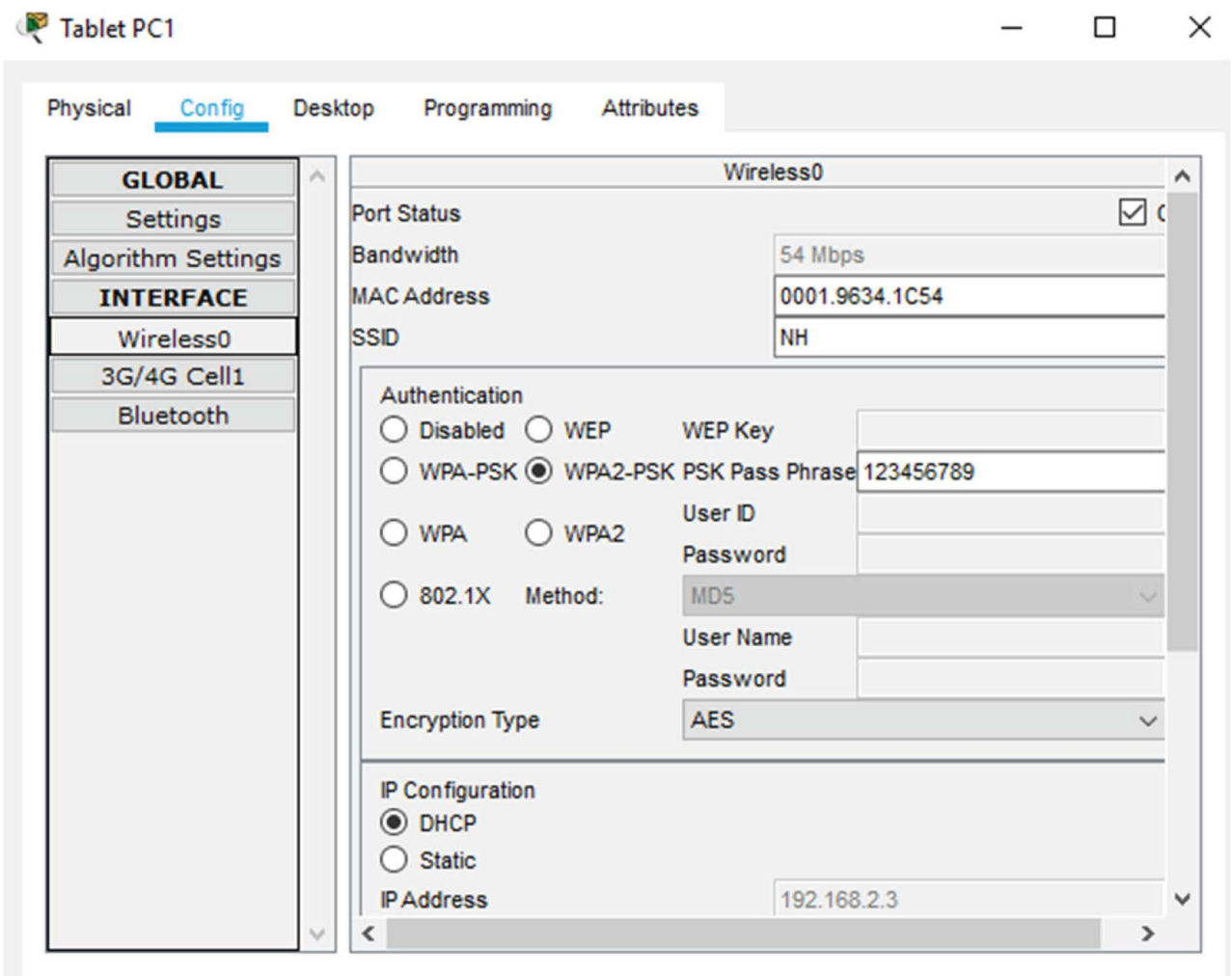


At last Switch on the Device.



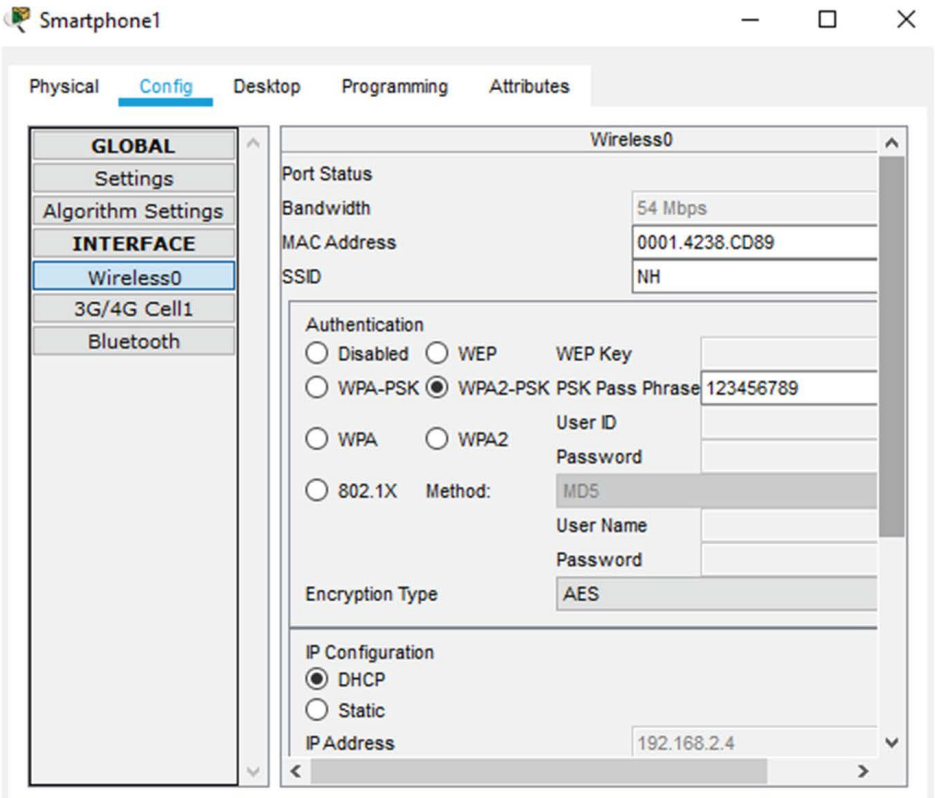
Tablet Configuration:

Config<<<Wireless0<<Change SSID Name<<Set Password



Smart Phone Configuration:

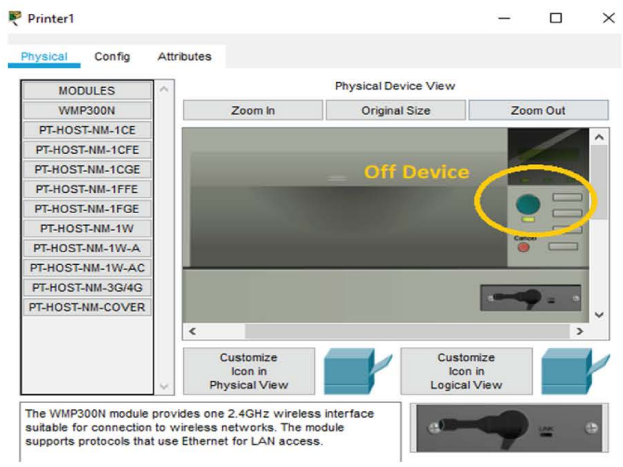
Config<<<Wireless0<<Change SSID Name<<Set Password



Printer Configuration

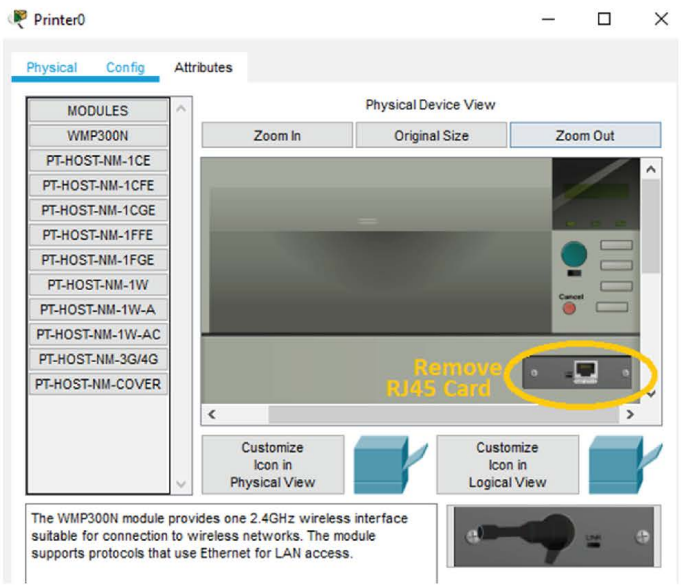
Physically off devices

1st step:



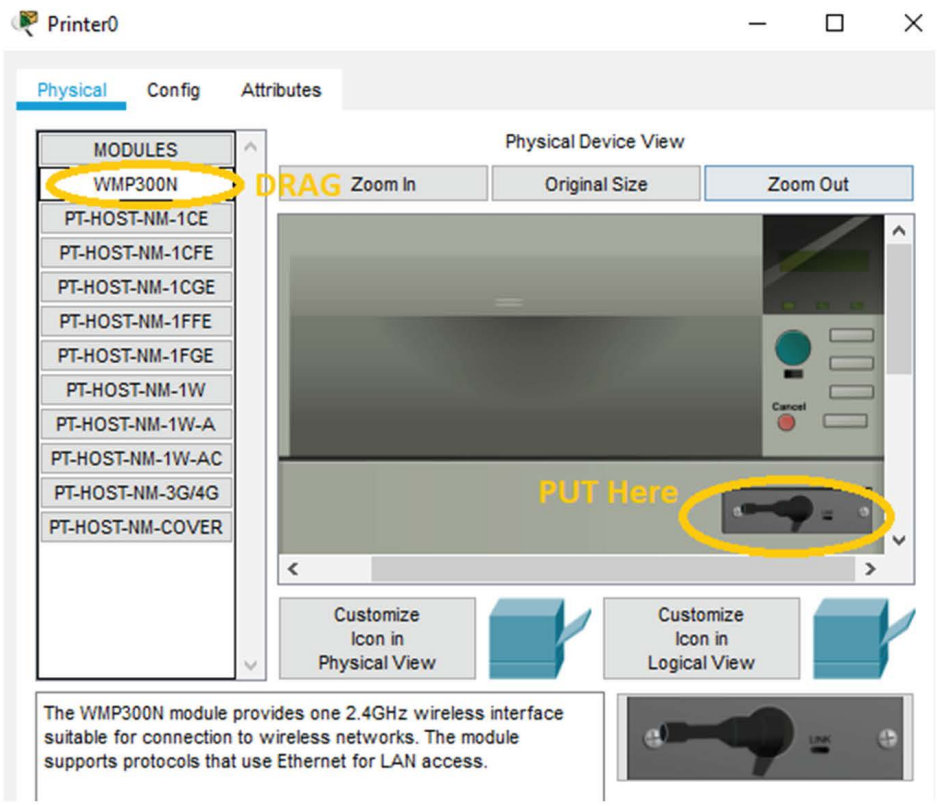
Step 2:

Remove RJ45 Card

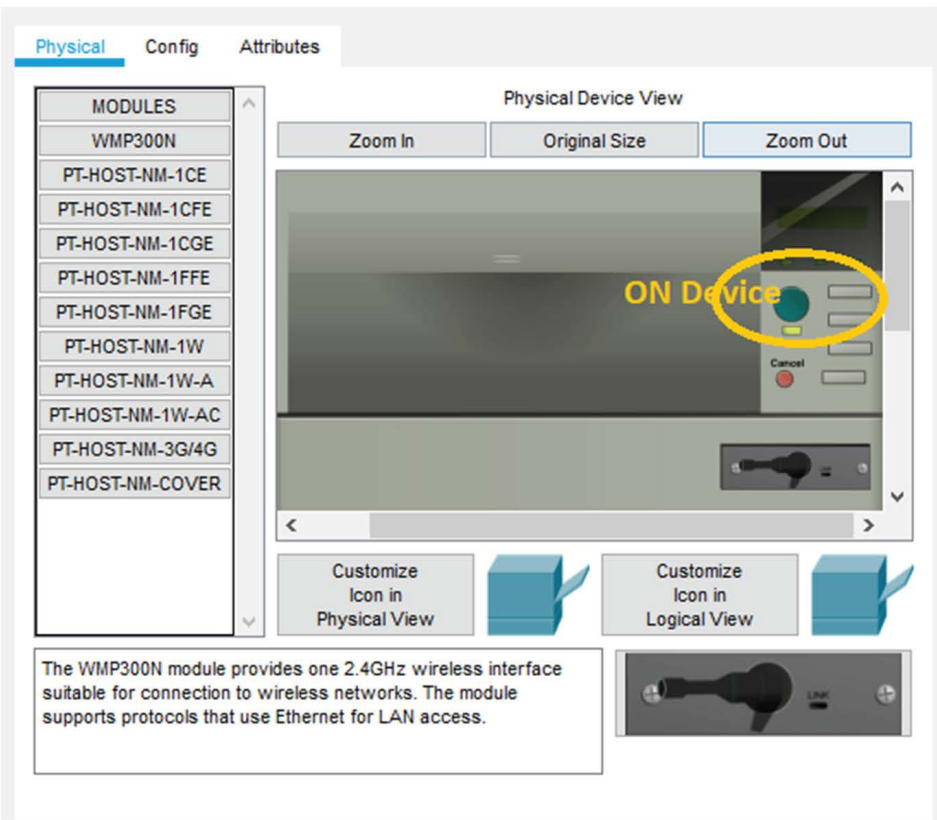


Step 3:

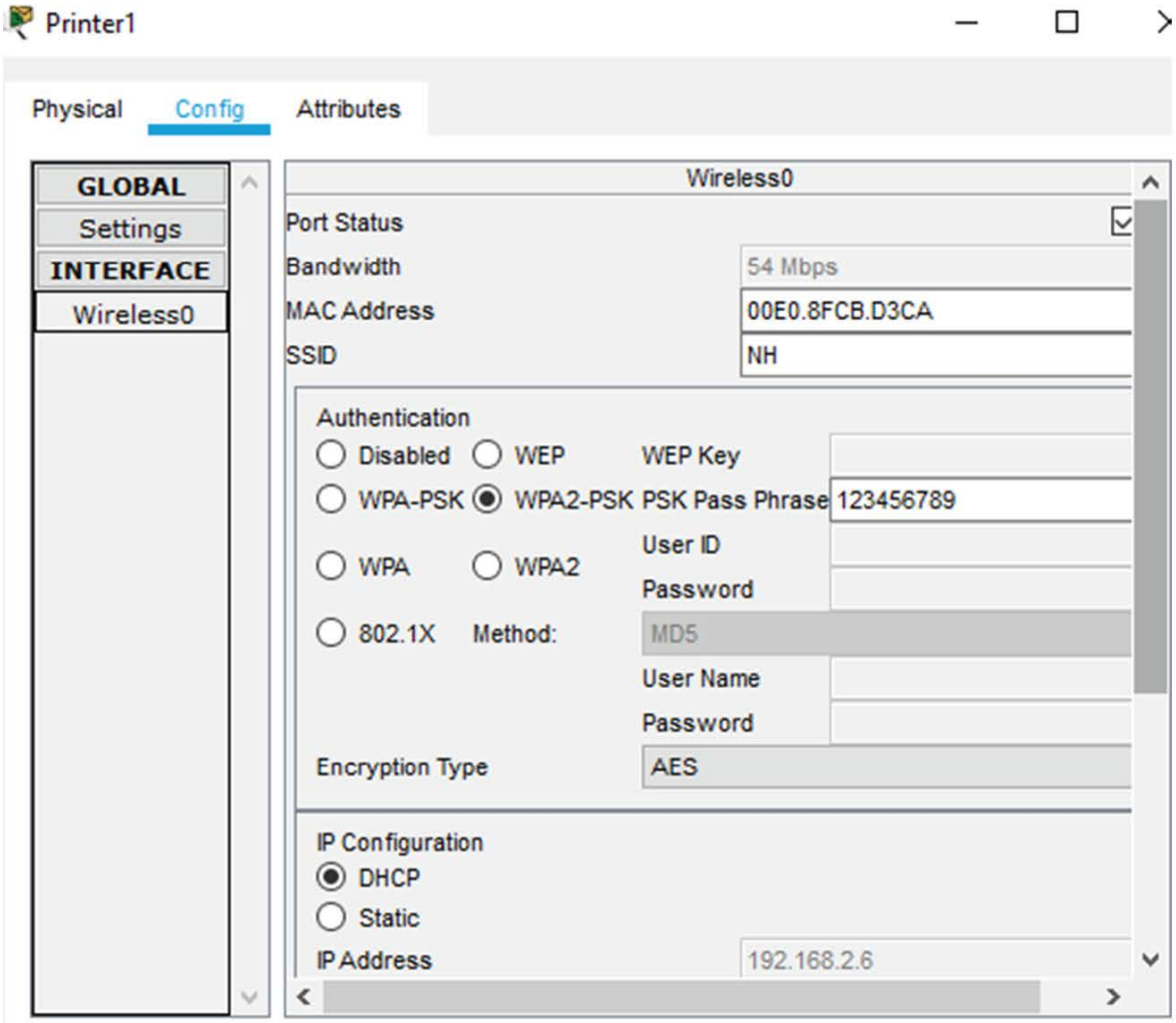
Drop WPC300N



At last Switch on the Device.



Config<<<Wireless0<<Change SSID Name<<Set Password

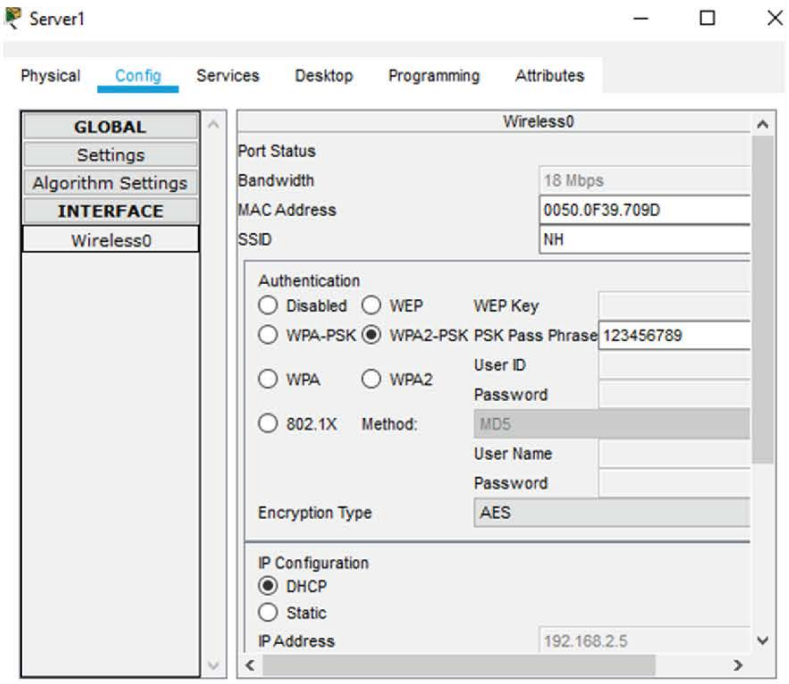


Server Configuration:

Physical<<Off device<<Remove rj45<<Drag and drop WMP300N<<turn on Device.

Config<<<Wireless0<<Change SSID Name<<Set Password

Same process for Server Also.



Laptop1

IP Address	Subnet Mask	Gateway
192.168.2.2	255.255.255.0	192.168.2.1

Tablet PC1

IP Address	Subnet Mask	Gateway
192.168.2.3	255.255.255.0	192.168.2.1

Smartphone 1

IP Address	Subnet Mask	Gateway
192.168.2.4	255.255.255.0	192.168.2.1

Printer 1

IP Address	Subnet Mask	Gateway
192.168.2.5	255.255.255.0	192.168.2.1

Server 1

IP Address	Subnet Mask	Gateway
192.168.2.6	255.255.255.0	192.168.2.1

Task1

Configure the R1, Switch, PC's, Printer, Smartphone, Server, Tablet and AP as per the topology

Task 2

R1

```
interface GigabitEthernet0/0/0
ip address 192.168.2.1 255.255.255.0
no shut
!
```

R1

```
interface GigabitEthernet0/0/1
ip address 192.168.3.1 255.255.255.0
no shut
!
ip dhcp pool NH
 network 192.168.2.0 255.255.255.0
 default-router 192.168.2.1
exit
```

Task 3

```
Router#sh ip dhcp pool

Pool NH :
 Utilization mark (high/low)    : 100 / 0
Subnet size (first/next)       : 0 / 0
Total addresses                : 254
Leased addresses               : 5
Excluded addresses             : 0
```

Pending event : non

1 subnet is currently in the pool

Current index IP address range Leased/Excluded/Total

192.168.2.1 192.168.2.1 - 192.168.2.254 5 / 0 / 254

Task4

Let's check the connectivity from Laptop to Server

```
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.2.6

Pinging 192.168.2.6 with 32 bytes of data:

Reply from 192.168.2.6: bytes=32 time=57ms TTL=128
Reply from 192.168.2.6: bytes=32 time=23ms TTL=128
Reply from 192.168.2.6: bytes=32 time=54ms TTL=128
Reply from 192.168.2.6: bytes=32 time=50ms TTL=128

Ping statistics for 192.168.2.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 23ms, Maximum = 57ms, Average = 46ms
```

Task5

Let's check the connectivity from Tablet to PC

```
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.3.2

Pinging 192.168.3.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.3.2: bytes=32 time=28ms TTL=127
Reply from 192.168.3.2: bytes=32 time=18ms TTL=127
Reply from 192.168.3.2: bytes=32 time=16ms TTL=127

Ping statistics for 192.168.3.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 16ms, Maximum = 28ms, Average = 20ms
```

```
C:\>ping 192.168.2.4

Pinging 192.168.2.4 with 32 bytes of data:

Reply from 192.168.2.4: bytes=32 time=62ms TTL=128
Reply from 192.168.2.4: bytes=32 time=12ms TTL=128
Reply from 192.168.2.4: bytes=32 time=32ms TTL=128
Reply from 192.168.2.4: bytes=32 time=17ms TTL=128

Ping statistics for 192.168.2.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 62ms, Average = 30ms
```

Head Office:

L-149, 1st, 2nd and 3rd Floor, Eshwari Mansion, 5th Main Road, Sector-6 HSR Layout,
Bengaluru, Karnataka 560102, India
Mobile No: **+91-9611027980** | **+91-9354284954**, Email: **info@networkershome.com**