# NETWORK

---

**Network:**

Communication between two devices

**ISP:**

Internet Service Provider

It is bridge for devices and network

**DHCP:** Dynamic Host Configuration Protocol

**DNS:** Domain Name Service

**NIC:**

Network Interface Card

NIC assign IP address

**Note:**

1. Where to Change IP?

   Run → ncpa.cpl

2. Where to see IP details?

   Run → cmd → ipconfig /all

IP (Internet Protocol):

IP (Internet Protocol) is a set of rules that governs how data is sent and received over the internet or a network. It works at Layer 3 (Network Layer) of the OSI model.

It provides a way to identify devices (using IP address) and route data from one device to another across networks.

Key Functions of IP:

1. Addressing – Assigns a unique IP address to each device on the network.
2. Routing – Ensures data is sent through the best path to reach the correct destination.

3. Packetization – Breaks data into packets, each containing source and destination IP addresses.

IPs are assigned in 2 ways:

Static:

User assign IP by manually

Dynamic:

IPs are assigned by automatically.

IPs are 2 versions:

1. IPv4 (Internet Protocol Version 4)
    - 32-bit address, written as 4 numbers(0-255) separated by dots
    - Most common today
      Example: 192.168.1.1
2. IPv6 (Internet Protocol Version 6)
    - Newer, for more address
    - 128-bit address, written in hexadecimal
      Example: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

IP Address Components:

Every IP address has two parts:

- Network ID – Identifies the network/subnet
- Host ID - Identifies the device within that network

Types of IP Address:

| **Type** | **Description** |
|---|---|
| **Public IP** | Used on the internet; globally unique |
| **Private IP** | Used inside private networks (e.g., LAN) |
| **Static IP** | Manually assigned; doesn't change |
| **Dynamic IP** | Automatically assigned by DHCP |
| **Loopback IP** | 127.0.0.1 – used to test your own system |
| **APIPA** | 169.254.x.x – fallback IP if DHCP fails |

How IP works in a Network:

1. You send data to a website
2. IP breaks the data into packets
3. Each packet is labeled with source and destination IPs
4. Packets travel through routers to the destination
5. Destination device reassembles the pakets.

IPv4:

1. It is a Decimal Number
2. It is a 32 bits address
3. It is also known as Logical Address
4. IP address range is 0 – 255

   Ex: 192.168.0.22

| | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|---|
| 192 → | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | → 8 bits |
| 168 → | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | → 8 bits |
| 0 → | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | → 8 bits |
| 21 → | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | → 8 bits |

Total: 32 bits

IPv4 range is divided into 5 classes:

Class A → 0 – 127

Class B → 128 – 191 — LAN & WAN Communication

Class C → 192 – 223

Class D → 224 – 239 ⇨ Multicasting

⇨

Class E → 240 – 255 Research & Development

In IPv4 there are 3 types of communication

1. Unicast: One to One communication
2. Broadcast: One to all
3. Multicast: One to many, one to group, group to group

IPv4 address is divided into 2 portions:

1. Network Portion
2. Host Portion

Portions in classes:

Class A: N.H.H.H → 8 Network bits & 24 Host bits

Class B: N.N.H.H → 16 Network bits & 16 Host bits

Class C: N.N.N.H → 24 Network bits & 8 Host bits

Subnet Mask:

A Subnet Mask is a 32-bit number used with an IP address to divide a large network into smaller sub-networks (subnets).

A Subnet Mask is used in IP networking to divide an IP address into two parts:

1. Network portion – Identifies the specific network
2. Host portion – Identifies individual devices (hosts) within that network

It helps routers and devices understand witch part of an IP address refers to the network and which part refers to the host.

Class A, B and C's default subnet masks:

- Class A: 255.0.0.0
- Class B: 255.255.0.0
- Class c: 255.255.255.0

| *Subnet Mask* | *CIDR Notation* | *Hosts Per Subnet* | *Description* |
|---|---|---|---|
| *255.0.0.0* | */8* | *16,777,214* | *Very large network* |
| *255.255.0.0* | */16* | *65,534* | *Large network* |
| *255.255.255.0* | */24* | *254* | *Most common LAN subnet* |
| *255.255.255.128* | */25* | *126* | *Splits a /24 in half* |
| *255.255.255.252* | */30* | *2* | *Used for point-to-point links* |

Subnetting Benefits:

- Efficient IP address usage
- Improved network performance
- Better security and isolation
- Easier management and scalability
- Controlling Broadcast traffic
- Easy administration

Hosts = $2^n - 2$, where n is the number of host bits (we subtract 2 for network and broadcast addresses).

Subnetting:

- Dividing a single network into multiple networks
- Subnetting is nothing but converting host bits into network bits
- Subnetting can be done based host requirement and network requirement
- If request is host, we have to borrow bits from right to left in host portion
- If request is networks, we have borrow bits from left to right host portion
- Subnetting Methods are:
    - FLSM – Fixed Length Subnet Mask
    - VLSM – Variable length subnet Mask

**1. What is a Subnet?**

A **subnet (sub-network)** is a **smaller network inside a larger IP network**. It helps divide a large network into manageable sections.

✅ **Purpose of Subnetting:**

- Reduces **network congestion**
- Improves **security** and **performance**
- Allows **isolation** of network groups (e.g., HR, IT, Finance)

**How to Calculate Subnets Manually**

Let's say you're given:

📌 IP: 192.168.1.0/26

**Step-by-Step:**

- /26 = 26 bits for network ⇒ **6 bits for host**
- Total hosts = $2^6$ - 2 = **62 hosts** per subnet
- Subnet increment = 256 - 192 = **64** (from the last octet of subnet mask)

**Subnets from 192.168.1.0/26:**

| Subnet | Host Range | Broadcast Address |
|---|---|---|
| **192.168.1.0/26** | 192.168.1.1 – 192.168.1.62 | 192.168.1.63 |
| **192.168.1.64/26** | 192.168.1.65 – 192.168.1.126 | 192.168.1.127 |

You now have 2 subnets, each with 62 usable hosts.

**Subnetting Practice Questions**

Try these:

**Q1:** How many hosts can a /27 subnet support?

**A:**

- 32 - 27 = 5 host bits → $2^5$ - 2 = **30 hosts**

**Q2:** What is the broadcast address of 192.168.10.0/28?

**A:**

- /28 = 16 hosts per subnet ($2^4$ = 16 → 14 usable)
- Subnet ranges: 0, 16, 32, ..., etc.
- Broadcast = one before the next subnet
  → First subnet: 192.168.10.0
  → Next = 192.168.10.16 → So broadcast = **192.168.10.15**

---

**Difference Between Subnet, Subnet Mask, and CIDR**

| Term | Definition |
|---|---|
| **Subnet** | A smaller segment of a network |
| **Subnet Mask** | A 32-bit number used to define network vs host portion |
| **CIDR** | Short notation for subnet mask, like /24 |

They all work together to **define and separate networks** within an organization or the internet.

**Real-life Examples**

**Example 1: Office Network Design**

You have a company with 3 departments:

- HR: 50 devices
- IT: 30 devices
- Finance: 20 devices

Use a /24 network: 192.168.10.0/24

- HR → /26 → 64 addresses
  → 192.168.10.0/26
- IT → /27 → 32 addresses
  → 192.168.10.64/27

- Finance → /27
  → 192.168.10.96/27

---

**Example 2: Point-to-Point Router Link**

You only need **2 usable IPs**. Use a /30 subnet:

- 192.168.1.0/30
    - Host IPs: 192.168.1.1, 192.168.1.2
    - Broadcast: 192.168.1.3

APIPA (Automatic Private IP Addressing):

APIPA is a feature in Windows and some other systems that automatically assigns a private IP address to a computer when DHCP fails (i.e., it can't get an IP from DHCP server).

- Subnet mask: 255.255.0.0
- Reserved for **link-local communication**
- Not routable on the internet or other networks

**Real-Life Example:**

Imagine 2 PCs connected by Ethernet with **no DHCP server**:

- Both will get APIPA addresses, e.g.:
    - PC1: 169.254.12.34
    - PC2: 169.254.88.21
- They **can ping each other** and communicate locally
- But they **can't access the internet or other subnets**

---

**How to Check for APIPA:**

On Windows:

sh

ipconfig

If you see an IP starting with 169.254, it means **DHCP failed** and APIPA was assigned.

| Feature | Details |
|---|---|
| **Full Form** | Automatic Private IP Addressing |
| **Range** | 169.254.0.1 – 169.254.255.254 |
| **Subnet Mask** | 255.255.0.0 |
| **When Used** | DHCP server unavailable |
| **Internet Access** | ❌ No |
| **Communication** | ✅ Local network only |

MAC (Media Access Control):

A MAC address is a unique hardware identifier assigned to network interfaces (like Ethernet cards, Wi-Fi adapter). It is used for Identifies a device on a local network

| Feature | Details |
|---|---|
| **Full Form** | Media Access Control Address |
| **OSI Layer** | Layer 2 (Data Link Layer) |
| **Length** | 48 bits (6 bytes) |
| **Format** | Hexadecimal (e.g., 00:1A:2B:3C:4D:5E) |
| **Assigned By** | Device Manufacturer |
| **Used For** | Identifying a device on a **LAN** |

Example MAC Address:

00:1A:2B:3C:4D:5E

↑ ↑

| └─ Device-specific part (assigned uniquely)

└─ Organizationally Unique Identifier (OUI)

Where MAC addresses are used:

- Ethernet
- Wi-Fi
- Bluetooth
- Any network interface at Layer 2

How MAC works in a Network:

1. You send data to another computer on the same LAN
2. Your device uses the ARP protocol to find the MAC address of the destination IP
3. The data packet is sent to the correct device using its MAC

MAC vs IP Address:

| **Feature** | **MAC Address** | **IP Address** |
|---|---|---|
| **Layer** | Layer 2 (Data Link) | Layer 3 (Network) |
| **Purpose** | Unique device ID on LAN | Logical address for network routing |
| **Format** | Hexadecimal | Decimal (IPv4) or Hex (IPv6) |
| **Example** | 00:1A:2B:3C:4D:5E | 192.168.1.10 |
| **Changes?** | Usually permanent (burned-in) | Can change (dynamic/static) |

ARP (Address Resolution Protocol) is a protocol used to map an IP address to a MAC address in a local network (LAN).

It operates at the Data Link Layer (Layer 2) and is used only within a subnet

⚠ Security Note:

ARP is not secure --- it's vulnerable to ARP spoofing/poisoning (where attackers fake MAC addresses to intercept traffic)

ARP Simulation:

We'll simulate Host A discovering the MAC address of Host B via ARP

Setup (2 PCs + 1 Switch)

| Device | IP Address | MAC Address |
|---|---|---|
| **PC A** | 192.168.1.10 | AA-AA-AA-AA-AA-AA |
| **PC B** | 192.168.1.20 | BB-BB-BB-BB-BB-BB |

Both are connected to the same switch and same subnet.

Step-by-Step ARP Simulation:

🔷 1. PC A wants to ping PC B:

```
ping 192.168.1.20
```

🔷 2. PC A checks its ARP cache:

```
arp -a
```

If there's no entry for 192.168.1.20, it sends an ARP request.

🔷 3. ARP Request Broadcast:

PC A sends a **broadcast** frame on the LAN:

> Who has 192.168.1.20? Tell 192.168.1.10

- Destination MAC: FF:FF:FF:FF:FF:FF (broadcast)
- Source MAC: AA-AA-AA-AA-AA-AA

All devices on the LAN receive this.

---

🔷 **4. PC B responds with ARP Reply:**

Unicast response to PC A:

> I have 192.168.1.20. My MAC is BB-BB-BB-BB-BB-BB

- Destination MAC: AA-AA-AA-AA-AA-AA
- Source MAC: BB-BB-BB-BB-BB-BB

🔷 **5. PC A updates ARP table:**

Now PC A knows:

192.168.1.20 → BB-BB-BB-BB-BB-BB

Subsequent traffic (like ping reply) goes directly using MAC address.

DHCP 4-Step Process (DORA)

**1. DHCP Discover (Client → Broadcast)**

- **Source IP**: 0.0.0.0 (client has no IP)
- **Destination IP**: 255.255.255.255 (broadcast)
- Client says:

"Is there any DHCP server out there?"

**2. DHCP Offer (Server → Broadcast)**

- **Source IP**: DHCP server's IP
- **Destination IP**: 255.255.255.255 (because client still has no IP)
- Server offers:

"Here's an IP address you can use."

**3. DHCP Request (Client → Broadcast)**

- **Source IP**: 0.0.0.0
- **Destination IP**: 255.255.255.255
- Client replies:

"I accept the IP address offered by DHCP server X."

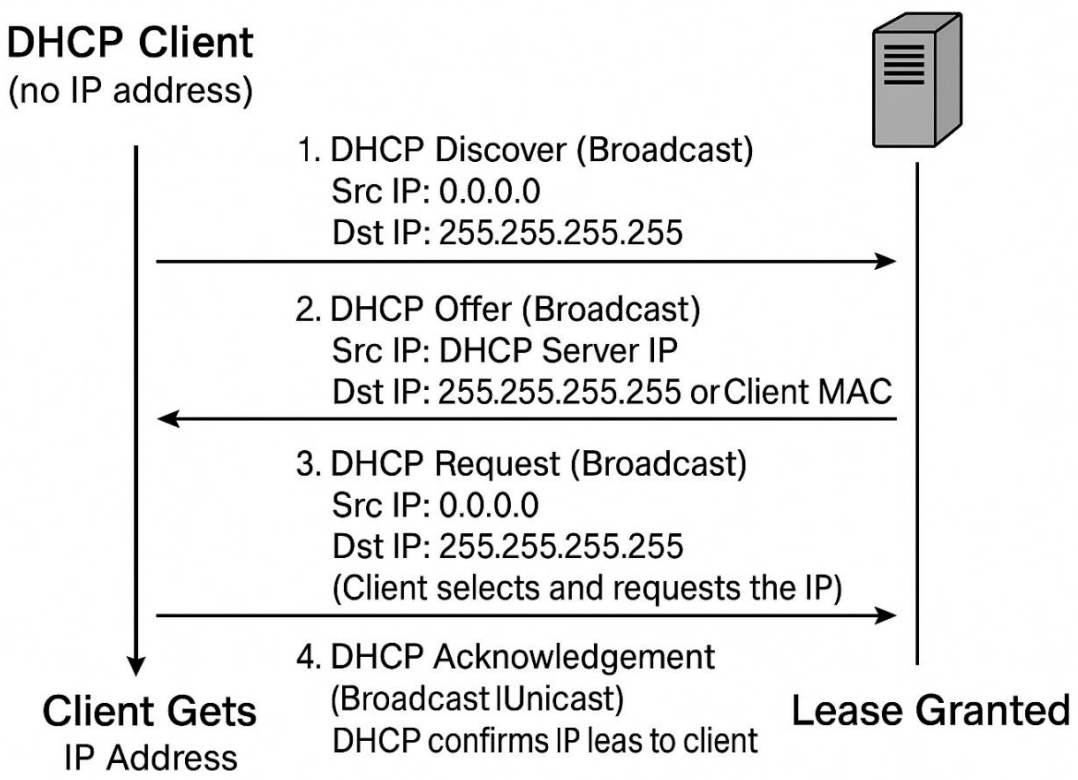
**4. DHCP Acknowledgment (Server → Broadcast or Unicast)**

- Server sends confirmation.
- Now client starts using the assigned IP.

Note:

| Field | Value |
|---|---|
| **Client IP** | 0.0.0.0 (in Discover/Request) |
| **Broadcast IP** | 255.255.255.255 |
| **DHCP Server Port** | UDP **67** |
| **DHCP Client Port** | UDP **68** |

| Step | Message | Src IP | Dest IP | Transport |
|---|---|---|---|---|
| **Discover** | DHCPDISCOVER | 0.0.0.0 | 255.255.255.255 | UDP 68 → 67 |
| **Offer** | DHCPOFFER | Server IP | 255.255.255.255 (or unicast) | UDP 67 → 68 |
| **Request** | DHCPREQUEST | 0.0.0.0 | 255.255.255.255 | UDP 68 → 67 |
| **Ack** | DHCPACK | Server IP | Client IP (unicast) | UDP 67 → 68 |




**Hub (Basic, Outdated Technology):**

- Broadcasts data to all connected devices, regardless of who the actual recipient is.
- Think of it like shouting in a room: everyone hears it, even if it's not for them.

How it works:

- Operates at OSI Layer 1 (Physical Layer)
- No intelligence – it doesn't inspect or filter traffic.
- Only one device can communicate at a time – collisions are common in large networks.

Limitations:

- No MAC address table
- High chance of data collisions.
- Slower performance (especially in large networks)
- Very rarely used today – mostly obsolete.

**Switch (Smarter and Faster):**

- Forwards data only to the specific device (MAC address) it's intended for
- Reduces unnecessary traffic and increases efficiency

How it works:

- Operates at OSI Layer 2 (Data Link Layer)
- Maintains a MAC address table to remember which devices are connected to which ports
- Supports full-duplex communication (simultaneous send/receive)

Advantages:

- Efficient and secure data transfer
- No data collisions (due to full duplex)
- Used in almost all modern networks

| Feature | Hub | Switch |
|---|---|---|
| **OSI Layer** | Layer 1 (Physical) | Layer 2 (Data Link) |
| **Data Direction** | Broadcast to all devices | Unicast to specific device |
| **Speed** | Slower | Faster |
| **Collisions** | Common (half-duplex) | Rare (full-duplex) |
| **MAC Address Table** | ❌ No | ✅ Yes |
| **Usage Today** | Obsolete | Widely used |
| **Cost** | Cheaper (in past) | Affordable & more capable now |

**How does he switch store mac address of directly connected devices**

How a Switch Learns and Stores MAC Addresses

A switch builds a MAC address table (also called a CAM table – Content Addressable Memory table) dynamically using a process called MAC learning.

---

⚙️ Step-by-Step Process:

1. Initial State – Table Empty:

- When the switch starts, its MAC address table is empty.
- It doesn't yet know where any devices are.

---

2. Device Sends Frame:
    - Let’s say Device A (MAC: AA:AA:AA:AA:AA:AA) sends a frame to Device B.
    - The frame enters the switch through Port 1.

---

3. Switch Learns the Source MAC:
    - The switch checks the source MAC address in the frame.
    - It adds an entry in the MAC table:

MAC Address Table:

| MAC Address | Port |
|---|---|
| AA:AA:AA:AA:AA:AA | 1 |

- Now, it knows that any data meant for MAC AA:AA:AA:AA:AA:AA should be sent to Port 1.

---

4. What About the Destination MAC?
    - If the destination MAC is:
        - Known (already in the table): switch sends it directly to the correct port (unicast).
        - Unknown: switch sends the frame to all ports except the source (flooding).
    - When the destination device replies, the switch learns that MAC too and adds it to the table.

5. MAC Address Table Aging:
    - If a device doesn't send data for a while, the switch will remove its MAC from the table (aging timeout, typically 5 minutes).
    - This helps keep the table up to date.

---

📋 Example MAC Table (After Learning):

| MAC Address | Port |
|---|---|
| AA:AA:AA:AA:AA:AA | 1 |
| BB:BB:BB:BB:BB:BB | 2 |
| CC:CC:CC:CC:CC:CC | 3 |

---

🔒 Bonus: Secure Switching

Some switches support port security, which can:

- Limit how many MAC addresses are allowed on a port.
- Stop MAC flooding or spoofing attacks.

OSI Model (Open Systems Interconnection Model)

The OSI model is a conceptual framework used to understand and standardize how data moves trough a network. It divides networking into 7 distinct layers, each with specific functions.

The 7 OSI Layers (Top to Bottom):

| Layer No. | Layer Name | Function Summary | Example Protocols/Devices |
|---|---|---|---|
| 7 | **Application** | User interface, software network access | HTTP, FTP, SMTP, DNS, Telnet |
| 6 | **Presentation** | Data format, encryption, compression | SSL/TLS, JPEG, ASCII, MPEG |
| 5 | **Session** | Establish, manage, terminate sessions | NetBIOS, RPC, PPTP |
| 4 | **Transport** | Reliable delivery, segmentation | TCP, UDP |
| 3 | **Network** | Routing, logical addressing | IP, ICMP, IPsec, Routers |
| 2 | **Data Link** | Physical addressing, error detection | MAC, ARP, Switches, Ethernet, PPP |
| 1 | **Physical** | Transmission of bits over media | Cables, Hubs, Wi-Fi, Fiber, Voltages |

The 7 Layers of the OSI Model (Top to Bottom)

### 🔷 Layer 7: Application Layer

- **Purpose**: Provides network services directly to **user applications**.
- **Functions**:
    - Interacts with software like browsers, email clients.
    - Supports services like **file transfer**, **email**, **web access**, etc.
- **Protocols**:
    - **HTTP/HTTPS** (web browsing)
    - **FTP** (file transfer)
    - **SMTP/POP3/IMAP** (email)
    - **DNS** (domain name resolution)
- **Example**: Typing a website address into Chrome or Firefox.

---

### 🔷 Layer 6: Presentation Layer

- **Purpose**: Translates data between the application and the network.

- **Functions**:
    - **Data formatting** (e.g., from text to binary)
    - **Encryption/Decryption** (e.g., HTTPS uses SSL/TLS)
    - **Compression/Decompression**
- **Example**:
    - When a browser uses SSL to encrypt data before sending.
    - Converting image file formats (e.g., JPEG, PNG).

---

🔷 **Layer 5: Session Layer**

- **Purpose**: Manages sessions (connections) between two systems.
- **Functions**:
    - Opens, maintains, and closes sessions.
    - Synchronizes data exchanges.
    - Handles session recovery after interruption.
- **Example**:
    - When you log in to a remote desktop or streaming site, the session keeps the connection alive during the activity.

---

🔷 **Layer 4: Transport Layer**

- **Purpose**: Provides **end-to-end communication**, ensuring data is delivered **correctly** and **in order**.
- **Functions**:
    - **Segmentation and reassembly** of data
    - **Error checking** and **retransmission**
    - **Flow control**

- **Protocols**:
    - **TCP (Transmission Control Protocol)** – Reliable, ordered (e.g., email, web)
    - **UDP (User Datagram Protocol)** – Fast, connectionless (e.g., video streaming, online gaming)
- **Example**:
    - TCP ensures that all parts of an email arrive correctly.
    - UDP is used for live Zoom calls where speed is more important than perfection.

**TCP (Transmission Control Protocol)** is called a **connection-oriented protocol** because it establishes a reliable connection between sender and receiver **before any actual data is transferred**. Here's why:

---

✅ **Reasons TCP is Connection-Oriented**

**1. Three-Way Handshake (Connection Establishment)**

Before sending data, TCP performs a 3-step process:

- **SYN** – Client sends a connection request.
- **SYN-ACK** – Server acknowledges and responds.
- **ACK** – Client confirms.

➞ This handshake **establishes a connection**, ensuring both sides are ready.

---

**2. Reliable Data Transfer**

TCP:

- Assigns **sequence numbers** to each byte of data.
- **Acknowledges** received data (ACK).
- **Retransmits** lost packets if no ACK is received.
- Ensures **data arrives in order and without duplication**.

### 3. Flow Control

TCP uses **window size** and sliding window mechanisms to ensure that the sender doesn't overwhelm the receiver with too much data.

---

### 4. Congestion Control

TCP adjusts the data sending rate based on network traffic conditions (e.g., via **slow start, congestion avoidance**).

---

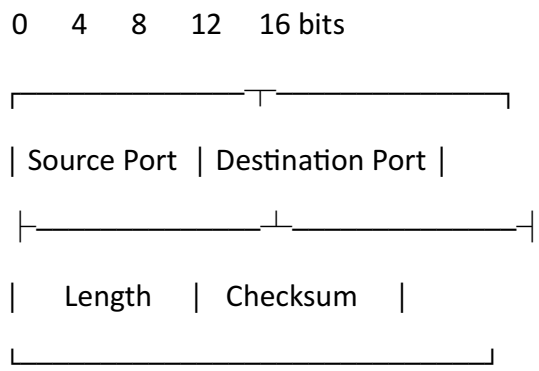
### 5. Connection Termination

After communication, TCP closes the connection using a **4-step FIN/ACK sequence**, ensuring all data is fully transmitted.

UDP:

**UDP** is a **connectionless** transport layer protocol used for fast, lightweight communication. It doesn't establish a connection, provide reliability, or guarantee delivery/order—making it faster but less reliable than TCP.

**UDP Header Structure (8 Bytes Total):**




**Key Features of UDP**

- 🚫 **No connection** → No handshake before sending data
- 🔁 **No reliability** → No retransmission, no guarantee of order
- 🕒 **Low latency** → Ideal for real-time apps

- 🔄 **Broadcast/multicast support**

---

### 🔷 Layer 3: Network Layer

- **Purpose**: Determines how data is **routed** from the source to the destination across networks.
- **Functions**:
  - **Logical addressing** (using IP addresses)
  - **Routing** (finding the best path)
  - **Fragmentation** of data for transmission
- **Protocols**:
  - **IP (IPv4, IPv6)**
  - **ICMP** (for ping)
  - **OSPF, BGP, RIP** (routing protocols)
- **Devices**: **Routers**
- **Example**:
  - When your message goes from your home router through multiple networks to reach a server in another country.

---

### 🔷 Layer 2: Data Link Layer

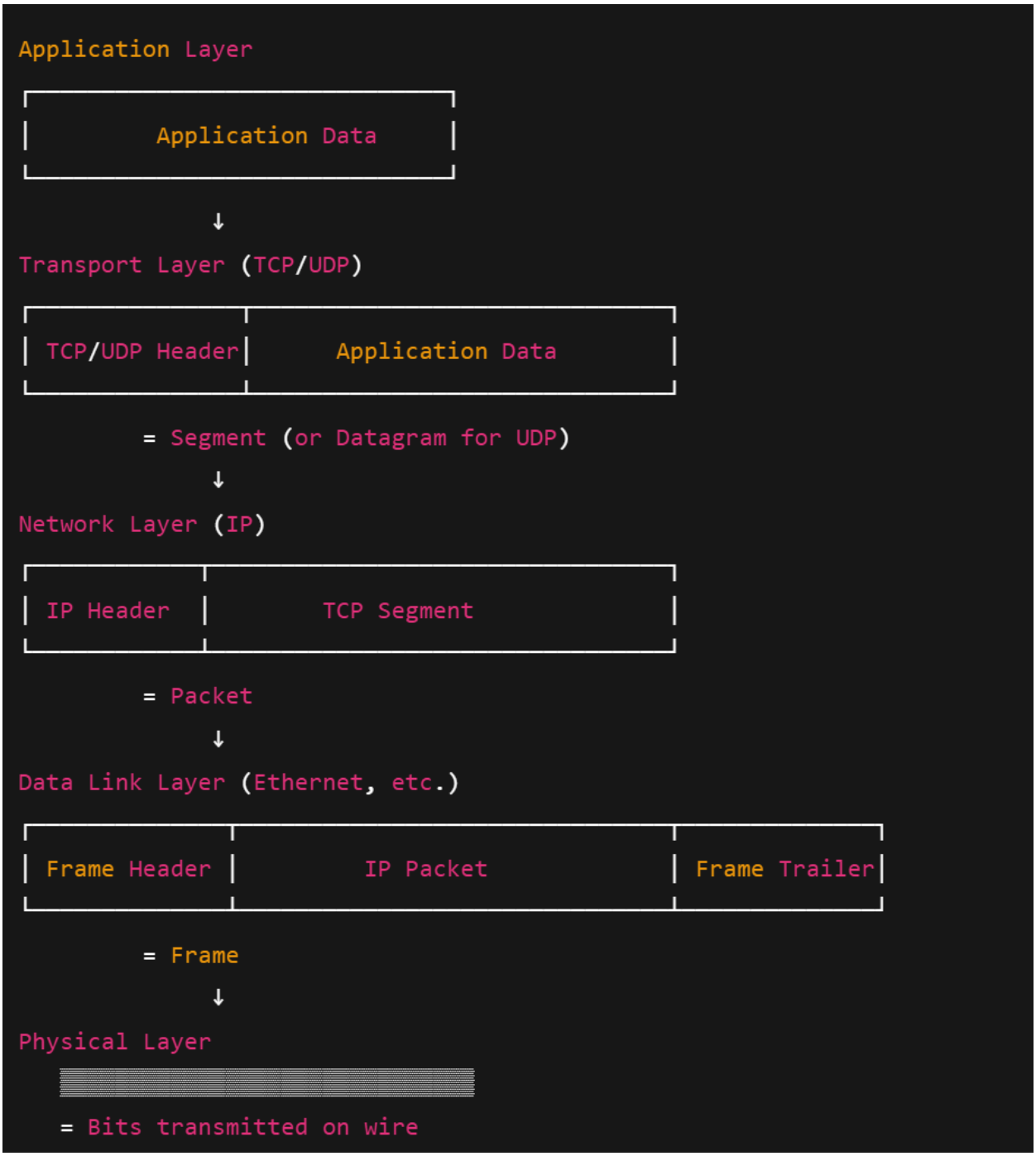
- **Purpose**: Handles communication between devices on the **same network**.
- **Functions**:
  - **MAC addressing** (physical address)
  - **Frame creation**
  - **Error detection** (e.g., CRC checks)

- **Protocols/Tech**:
    - **Ethernet**
    - **ARP**
    - **PPP**
- **Devices**: **Switches**, **bridges**
- **Example**:
    - Sending a frame from your laptop to the switch using your MAC address.

---

🔷 **Layer 1: Physical Layer**

- **Purpose**: Transmits **raw bits** (0s and 1s) over the physical medium.
- **Functions**:
    - Defines the **electrical, optical, or radio signals** used.
    - Includes hardware components (wires, connectors, NICs).
- **Examples of Physical Media**:
    - Ethernet cables
    - Fiber optics
    - Wireless radio signals (Wi-Fi, Bluetooth)
- **Devices**: **Hubs**, **repeaters**, **modems**
- **Example**:
    - The electrical signals that travel over a LAN cable to send a message.


Application Layer
Application Data
Transport Layer (TCP/UDP)
TCP/UDP Header
Application Data
= Segment (or Datagram for UDP)
Network Layer (IP)
IP Header
TCP Segment
= Packet
Data Link Layer (Ethernet, etc.)
Frame Header
IP Packet
Frame Trailer
= Frame
Physical Layer
= Bits transmitted on wire

| Port Range | Range | Description |
|---|---|---|
| **Well-known ports** | 0 – 1023 | Assigned by IANA for common, standard services (e.g., HTTP, FTP, SSH) |
| **Registered ports** | 1024 – 49151 | Assigned to software/apps by IANA (e.g., MySQL, PostgreSQL) |
| **Dynamic/Private** | 49152 – 65535 | Used dynamically for client-side connections; not assigned to any service |

| Purpose | Port(s) |
|---|---|
| **Web (HTTP/HTTPS)** | 80, 443 |
| **File Transfer** | 20, 21, 22, 69 |
| **Email** | 25, 110, 143, 465, 993, 995 |
| **DNS** | 53 |
| **DHCP** | 67 (server), 68 (client) |
| **Remote Access** | 22 (SSH), 23 (Telnet), 3389 (RDP) |
| **Network Services** | 161/162 (SNMP), 514 (Syslog), 137–139 (NetBIOS), 445 (SMB) |
| **Databases** | 3306 (MySQL), 5432 (PostgreSQL), 1433 (MS SQL), 1521 (Oracle) |

Well-Known Ports (0–1023):

| Port | Protocol | Service | Description |
|---|---|---|---|
| **20** | TCP | FTP (Data) | File Transfer Protocol - data channel |
| **21** | TCP | FTP (Control) | File Transfer Protocol - control channel |
| **22** | TCP | SSH | Secure Shell for remote login |
| **23** | TCP | Telnet | Unsecure remote login |
| **25** | TCP | SMTP | Simple Mail Transfer Protocol |
| **53** | UDP/TCP | DNS | Domain Name System |
| **67** | UDP | DHCP (Server) | Dynamic Host Configuration Protocol |
| **68** | UDP | DHCP (Client) | Used by DHCP client |
| **69** | UDP | TFTP | Trivial File Transfer Protocol |
| **80** | TCP | HTTP | Web browsing (insecure) |
| **110** | TCP | POP3 | Post Office Protocol v3 (email) |
| **123** | UDP | NTP | Network Time Protocol |
| **143** | TCP | IMAP | Internet Message Access Protocol |
| **161** | UDP | SNMP | Simple Network Management Protocol |
| **162** | UDP | SNMP Trap | Used for SNMP alerts |
| **179** | TCP | BGP | Border Gateway Protocol |
| **443** | TCP | HTTPS | Secure HTTP over SSL/TLS |
| **445** | TCP | SMB | Windows file sharing (CIFS) |
| **514** | UDP | Syslog | System logging messages |

**Router**

A **router** is a networking device that connects multiple networks together and **forwards data packets** between them based on their destination IP addresses.

---

📌 **Key Functions of a Router:**

1. **Packet Forwarding**:
   Routes packets from one network to another using IP addresses.
2. **Inter-network Communication**:
   Connects devices from different IP subnets (e.g., LAN to WAN).
3. **Path Selection**:
   Uses routing tables and protocols (like OSPF, RIP, BGP) to find the best path.
4. **NAT (Network Address Translation)**:
   Translates private IPs to public IPs, enabling internet access.
5. **Firewall/Security**:
   Many routers include basic firewall features and traffic filtering.

---

🏠 **Home Router Example:**

- **LAN Side**: Connects your devices (phones, laptops) using private IPs like 192.168.x.x.
- **WAN Side**: Connects to the Internet via your ISP using a public IP.

🔁 Difference from Switch:

| **Feature** | **Router** | **Switch** |
|---|---|---|
| **OSI Layer** | Layer 3 (Network) | Layer 2 (Data Link) |
| **Function** | Routes packets across networks | Switches frames within a LAN |
| **IP Knowledge** | Uses IP addresses | Uses MAC addresses |
| **Use Case** | Connects networks (LAN ↔ WAN) | Connects devices in one network |

Ethernet Components:

Interface Types:

- Ethernet – 10 Mbps
- Fast Ethernet – 100Mbps
- Giga Ethernet – 1000 Mbps
- Ten Giga Ethernet – 10000 Mbps

Serial Interface:

- It is also known as WAN interface
- It is use to connect one branch to another branch
- It has 60 pins (old) or 26 pins (latest)

Fast Ethernet:

- It is used to connect LAN
- It is also used to connect WAN (ISP)
- It has 8 pins
- It is a data interface (Transfers the data)

Console Port:

- It is also known as management port
- It is used to access router directly

Copper Cross-Over Cable:

Connect between same devices only

Example:

PC to PC

Switch to Switch

Router to Router

PC to Router

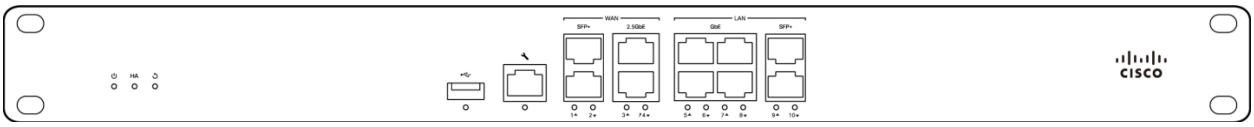
Copper Straight-Through Cable:

Connect between different devices

Example:

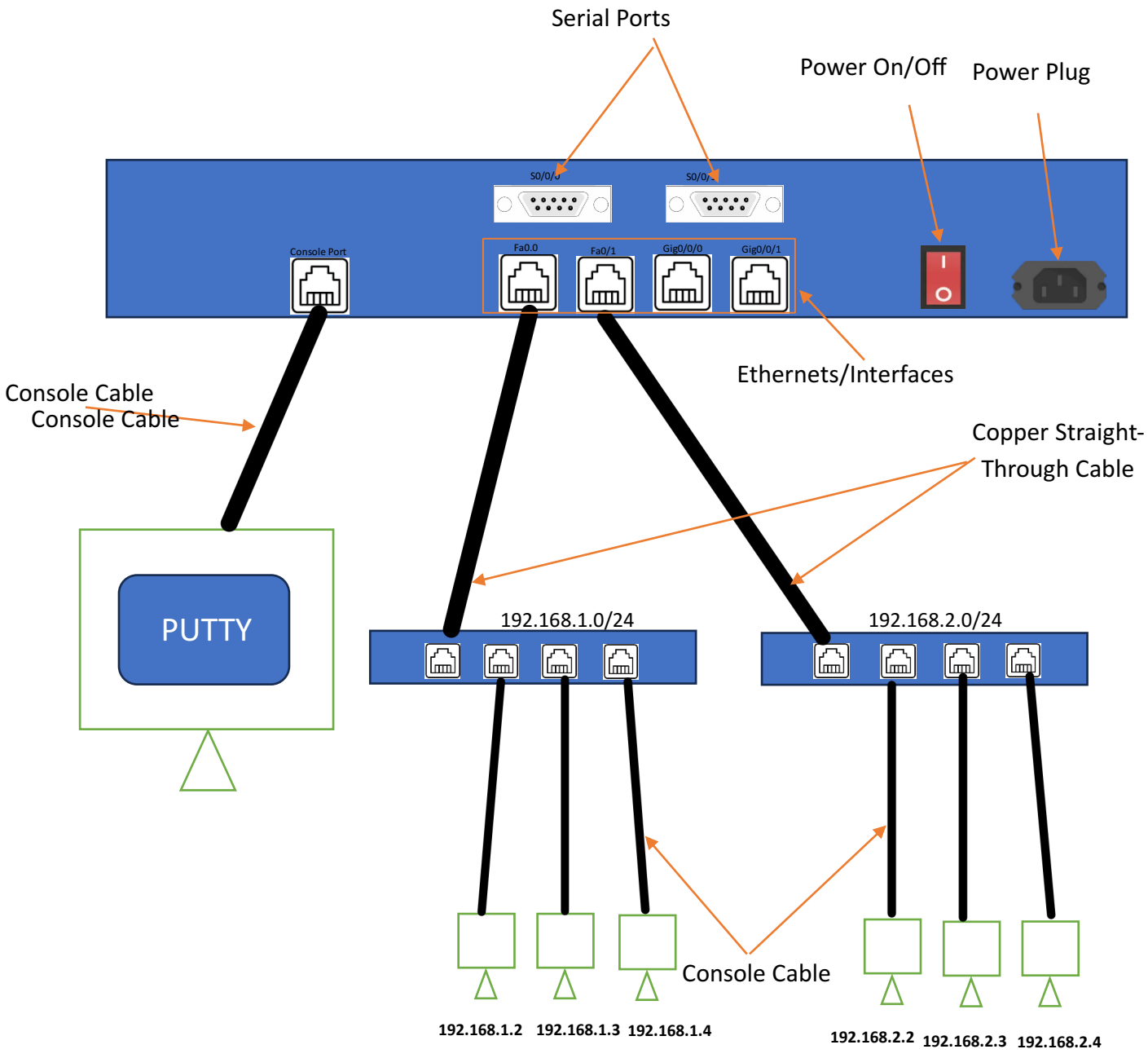
Switch to Router

Switch to PC



**Interfaces**

- 2x SFP+ / 10 Gigabit Ethernet ports dedicated for the WAN uplink
- 1x RJ45 / 2.5 Gigabit Ethernet port dedicated for the WAN uplink
- 1x RJ45 / 2.5 Gigabit Ethernet port with PoE+ capabilities dedicated for the WAN uplink
- 4x RJ45 / 1 Gigabit Ethernet LAN ports
- 2x SFP+ / 10 Gigabit Ethernet LAN ports
- 1x RJ45 management port for local status page access
- 1x USB 3.0 port


Serial Ports
Power On/Off
Power Plug
Console Port
Fa0.0
Fa0/1
Gig0/0/0
Gig0/0/1
Ethernets/Interfaces
Console Cable
Console Cable
Copper Straight-
Through Cable
PUTTY
192.168.1.0/24
192.168.2.0/24
Console Cable
192.168.1.2
192.168.1.3
192.168.1.4
192.168.2.2
192.168.2.3
192.168.2.4

Router> → User Mode

Router# → Privilege mode

Router(config)# → Global Configure Mode

Router(config-if)# → Interface Mode

- Privilege mode is used to execute show commands – verification commands
- Configure mode is used to configure, delete, modify the configuration

Example: Important commands

Router# show ip interface brief

Router# show ip route

Router# show version

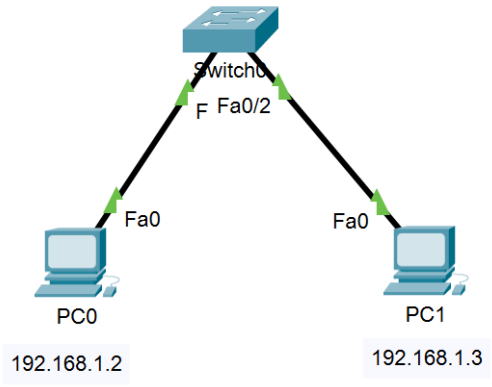
Router# show running-config

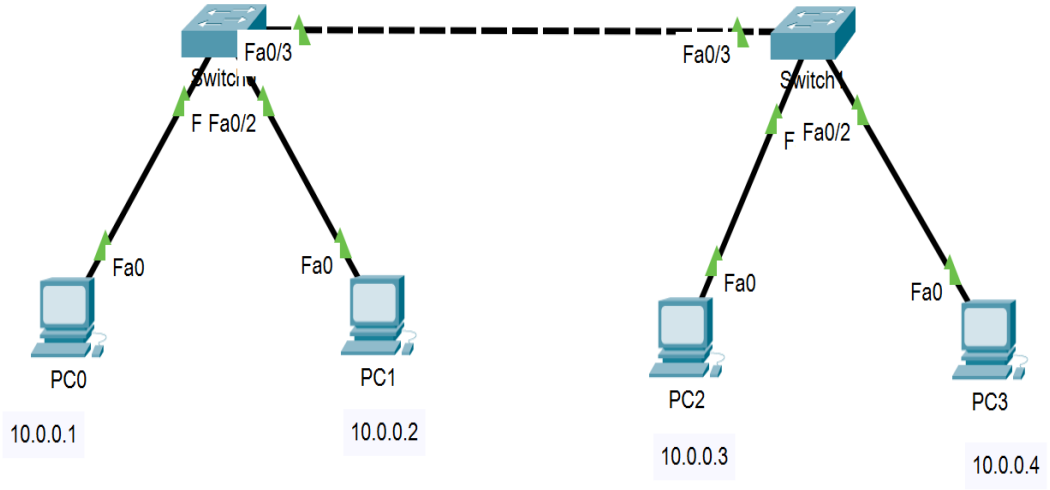
Router# show startup-config

Assigning IP address to fast ethernet f0/0 to router in Cisco Packet Tracer:

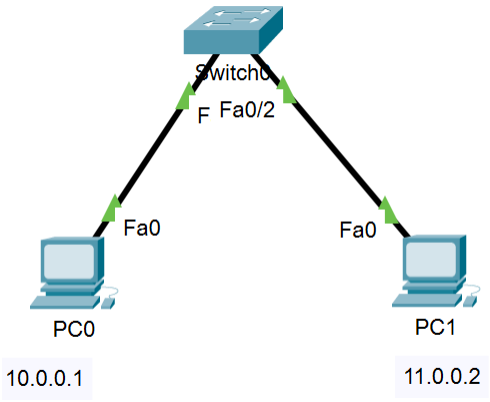
- router>enable
- router#configure terminal
- router(config)# hostname r1
- r1(config)#interface f0/0
- r1(config-if)#ip address 192.168.1.1 255.255.255.0
- r1(config-if)#no shutdown
- r1(config-if)#exit
- r1(config)#exit
- r1#wirte

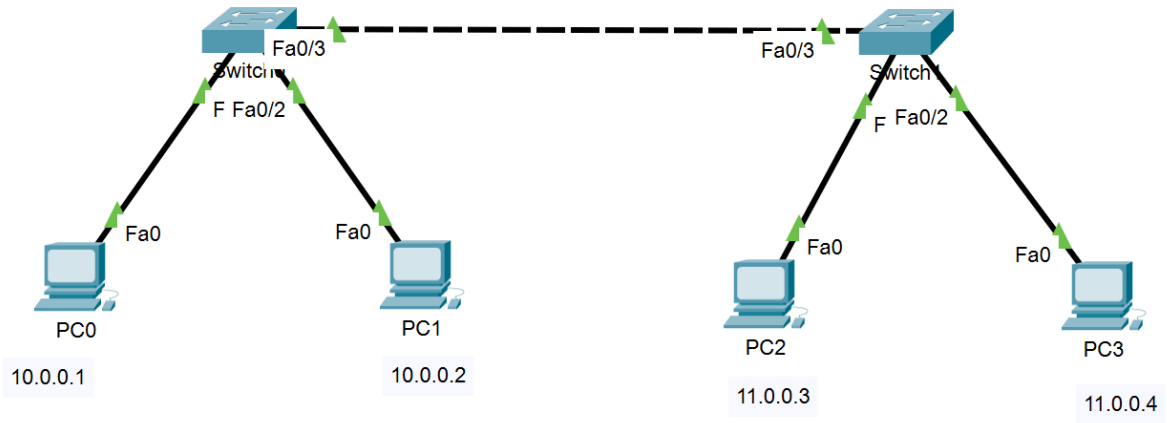
✅ It Works:


Switch0
F Fa0/2
Fa0
Fa0
PC0
PC1
192.168.1.2
192.168.1.3


Fa0/3
Fa0/3
F Fa0/2
F Fa0/2
Fa0
Fa0
Fa0
Fa0
PC0
PC1
PC2
PC3
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4

❌ It not Works:




- Switch can’t use to different network
- Switch only communicate within same network
- But need to communicate different networks then we need to use router

Topology:

Structure of network. It is 2 types, they are:

**1. Physical Topology:**

How devices are **physically connected** (with cables, etc.).

**2. Logical Topology:**

How data **logically flows** within the network, regardless of physical layout.

Router Function:

1. Whenever router receives the packet, it checks the destination IP address in packet
2. Destination IP address will be compares with routing table
3. If no matching route in routing table, the packet would be dropped by router
4. And send “Destination host unreachable” message to sender

Troubleshooting:

If we unable to ping destination of different network:

- First we need to check the PC configuration with IP in cmd ipconfig /all
- If IP is configured with valid IP and mask, then check the connectivity of gateway.
- It gateway connecting is fine, then ping destination of gateway.

Note:

- Ping 8.8.8.8 -n 10 → It sends 10 packets
- Ping 8.8.8.8 -t → It sends the packets continuously

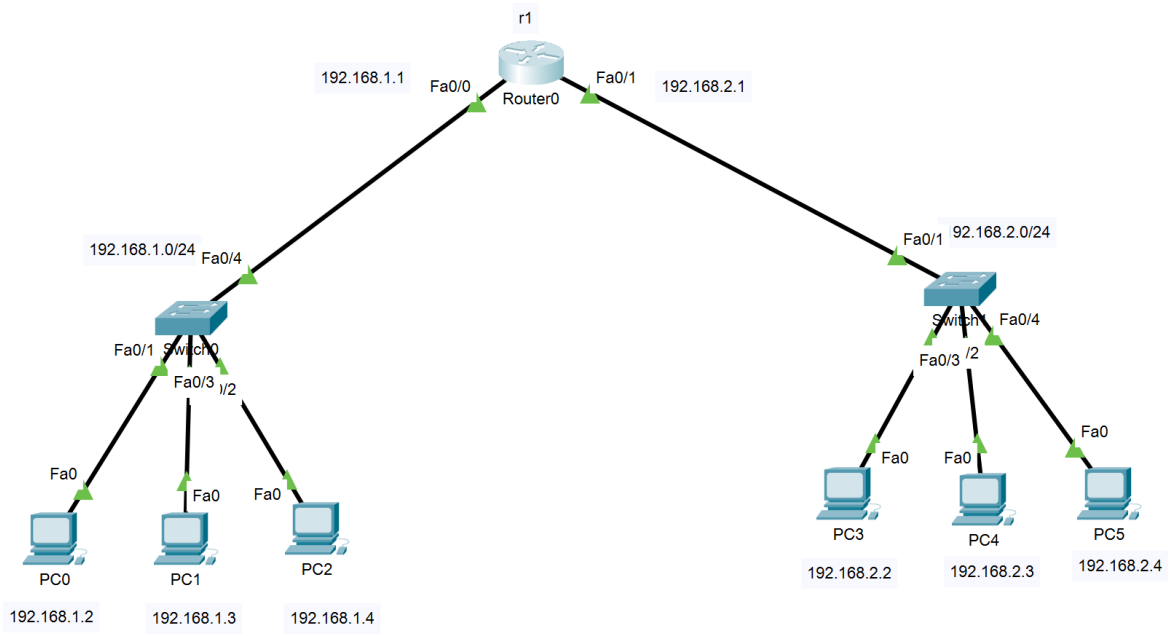
Routing:

- Routing is to find out next hop IP and exit interface from routing table

Switching:

- **Switching** is the process of forwarding data **within a local area network (LAN)**. It occurs at **Layer 2** (Data Link Layer) of the **OSI model** and uses **MAC addresses** to deliver frames between devices.

Topology:




Commands:

```
! On r1


enable
configure terminal
hostname r1
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
```

exit

!

interface Fa0/1

ip address 192.168.2.1 255.255.255.0

no shutdown

exit

**Routing:**

**Routing** is the process of selecting a path for traffic in a network. It is mainly used in **internetworks** (like the Internet) to move **data packets** from **source to destination** across multiple networks.

**Types of Routing**

1. **Static Routing**
    - Manually configured by a network admin.
    - Good for small networks.
    - Doesn't adapt to changes.

    Syntax:

    IP route <Destination Network> <Subnet mask> <Next hop ip address>
2. **Dynamic Routing**
    - Uses routing protocols to automatically update paths.
    - Adapts to network changes (failures, traffic).
    - Examples:
        - **RIP** (Routing Information Protocol)
        - **OSPF** (Open Shortest Path First)
        - **EIGRP** (Enhanced Interior Gateway Routing Protocol)
        - **BGP** (Border Gateway Protocol – used on the Internet)

**Static Routing:**

**Static Routing** is the process of **manually configuring routes** in a router's routing table. It tells the router **exactly which path to use** to reach a destination network, without relying on dynamic routing protocols.

**When to Use Static Routing**

✅ Good for:

- Small networks
- Point-to-point links
- Backup/failover routes
- Default routes (0.0.0.0/0)

❌ Not ideal for:

- Large, dynamic, or constantly changing networks

---

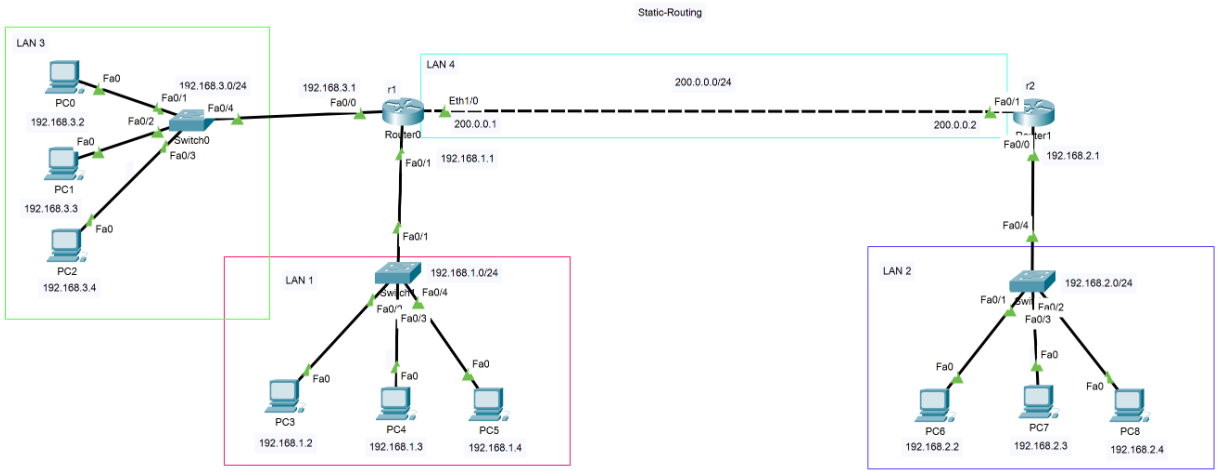
**Advantages**

- Simple to implement
- Predictable and secure
- No bandwidth used by routing protocols
- Less CPU usage

---

⚠️ **Disadvantages**

- No automatic failover if a route goes down
- Hard to maintain in large networks
- Prone to human error in configuration




Commands:

! On R1

Enable

Configure terminal

Hostname r1

!

Interface fa0/1

Ip address 192.168.1.1 255.255.255.0

No shutdown

Exit

!

Interface Fa0/0

Ip address 192.168.3.1 255.255.255.0

No shutdown

Exit

!

Interface Eth1/0

Ip address 200.0.0.1 255.255.255.0

No shutdown

Exit

Ip route 192.168.2.0 255.255.255.0 200.0.0.2

---

! On R2

Enable

Configure terminal

Hostname r2

!

Interface fa0/0

Ip address 192.168.2.1 255.255.255.0

No shutdown

Exit

!

Interface Fa0/1

Ip address 200.0.0.2 255.255.255.0

No shutdown

Exit

!

Ip route 192.168.1.0 255.255.255.0 200.0.0.1

Ip route 192.168.3.0 255.255.255.0 200.0.0.1

**Floating Static Routing**

**Floating Static Route** is a **backup static route** that "floats" below the main route in priority, meaning it will only be used **if the primary route fails**.

**How it works:**

- You assign it a **higher Administrative Distance (AD)** than the primary route.
- If the primary route (with lower AD) fails, the router will then use the floating static route.
- Default Administrative Distance (AD) 1

**Example:**

Let's say:

- You have a main static route with AD = 1.
- You want a backup route to be used only if the main route fails.

```bash
ip route 192.168.2.0 255.255.255.0 10.0.0.3 10
```

This floating route has AD = 10, so it won't be used unless the main one (AD = 1) is gone.

---

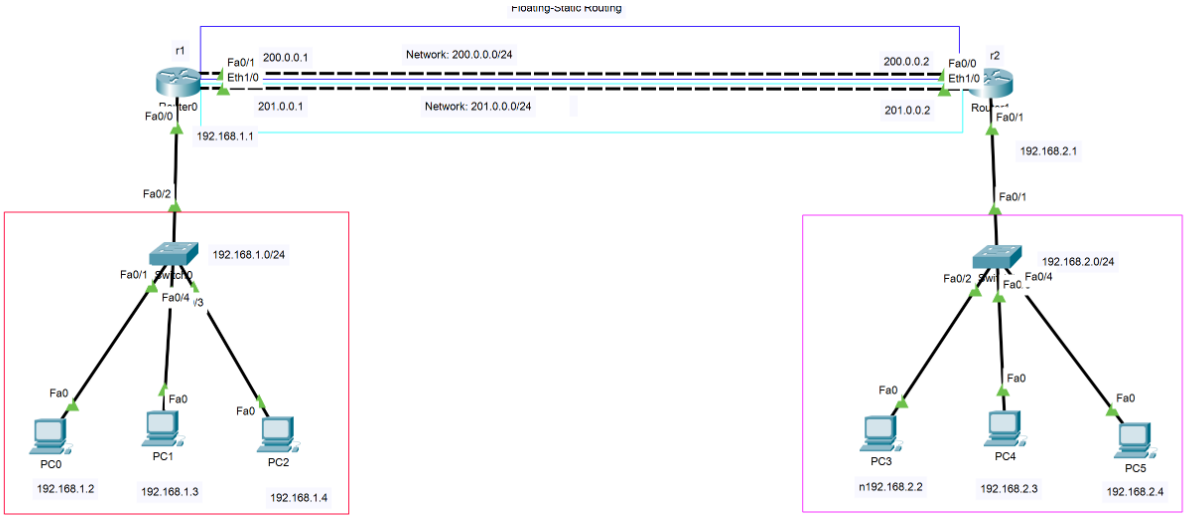
### 🔁 Use Case Example

Imagine two paths to a network:

- Primary: Fast link
- Secondary: Slower or expensive backup

You use:

```
ip route 192.168.2.0 255.255.255.0 10.0.0.2    ← Primary (default AD = 1)
ip route 192.168.2.0 255.255.255.0 10.0.0.3 10 ← Backup (AD = 10)
```




| Part | Meaning |
|---|---|
| **S** | **Static Route** – this route was manually configured by an admin |
| **192.168.2.0/24** | **Destination network** – the route applies to this network |
| **[1/0]** | **[Administrative Distance / Metric]** |
| **via 200.0.0.2** | **Next-hop IP address** – where the packet should be sent |

- **Administrative Distance (AD)** = 1
  → Since it's a static route, default AD is 1 (higher priority than dynamic routes).
- **Metric** = 0
  → The metric is always 0 for static routes

```
! On r1
enable
configure terminal
hostname r1
!
Interface Fa0/0
Ip address 192.168.1.1 255.255.255.0
```

No shutdown

Exit

!

Interface Fa0/1

Ip address 200.0.0.1 255.255.255.0

No shutdown

Exit

!

Interface Eth1/0

Ip address 201.0.0.1 255.255.255.0

No shutdown

Exit

!

Ip route 192.168.2.0 255.255.255.0 200.0.0.2

Ip route 192.168.2.0 255.255.255.0 201.0.0.2 10

- Router 2:

  enable

  configure terminal

  hostname r2

  !

  Interface Fa0/0

  Ip address 200.0.0.2 255.255.255.0

  No shutdown

  Exit

  !

Interface Fa0/1

Ip address 192.168.2.1 255.255.255.0

No shutdown

Exit

!

Interface Eth1/0

Ip address 201.0.0.2 255.255.255.0

No shutdown

Exit

Ip route 192.168.1.0 255.255.255.0 200.0.0.1

Ip route 192.168.1.0 255.255.255.0 201.0.0.1 10

Note:

Removing Ip route

Cmd: no ip route 192.168.2.0 255.255.255.0 200.0.0.2

- Layer 2 information change at every hop
- When packet is sent from one network to another networks
- What is the destination MAC IP pack: Gate MAC address
- Floating static route is used as backup path
- Floating static route is configured with higher administrative distance then primary path
- Floating static route is installed in routing table only when primary path is failed

What are the drawbacks of static routing?

- Each and every remote network has to be manually configured
- If remote networks becomes unavailable, routes can not be know automatically

Dynamic Routing Protocol:

**Dynamic Routing** is a method where routers automatically learn and update routes using **routing protocols**. Unlike **static routing** (manual), dynamic routing adapts to network changes like link failures or new paths.

- To advice networks to neighbors and receives network from networks
- Select the shortest path IP there are multiple routes to send destination

  Metrics:

  1. Hop counts:
     - Number of routes between source and destination
  2. Band Width:
     - Speed of the path
  3. Delay:
     - Timing how much limit takes to reach destination
  4. Reliability
  5. Load
  6. MTU
- If primary path is failed, secondary path will be used automatically
- Trigger updates:
  - Whenever there is a change in network, changes will be immediately send a neighbors
  - Changes are:
    1. Network up/down
    2. Metric Changes

Dynamic Routing Protocol:

- Interior gateway (Protocol IGP)
- Exterior gateway (Protocol EGP)

Interior gateway (Protocol IGP):

Interior gateway protocols are used to exchange routing information within the autonomous system

Interior protocols are three types:

1. Distance vector routing protocol
   - Example: RIP
2. Advanced distance vector routing protocol

- Example: EIGRP

Exterior gateway protocol

Exterior gateway protocols are used to exchange the routing information between Autonomous system

Example: BGP (Border Gateway protocol)

RIP (Routing Information Protocol):

RIP is one of the oldest **dynamic routing protocols**, used to enable routers to **exchange routing information automatically**.

- It is a distance vector routing protocol
- It uses Bellman Ford algorithm
- It uses hop counts as metric
- It supports maximum 15 hop counts
- It sends routing updates every 30 seconds
- It supports equal metric load balancing
- It has two versions
    1. RIPv1
    2. RIPv2
- RIP administrative distance is 120

RIPv1:

- It sends routing updates as broadcasting to 255.255.255.255
- It is a classfull routing protocol
- It does not support VLSM (Variable Length Subnet Mask)

RIPv2:

- It sends routing updates as multicasting to 24.0.0.0
- It is a classless routing protocol
- It support VLSM

What is hopcounts?

Number of routers between source and destination

What is load balancing?

Distributes the traffic through multiple paths

What is classfull routing?

Classfull routing protocols do not exchange subnet mask information in routing updates

Example: RIPv1

What is classless routing?

Classless routing protocols exchange subnet mask information in routing updates.

Example: EIGRP, OSPF, IS-IS, BGP, RIPv2

What is the use of ping?

Ping is used to check the connectivity of system in network

Example: ping 192.168.2.2

What is trace route?

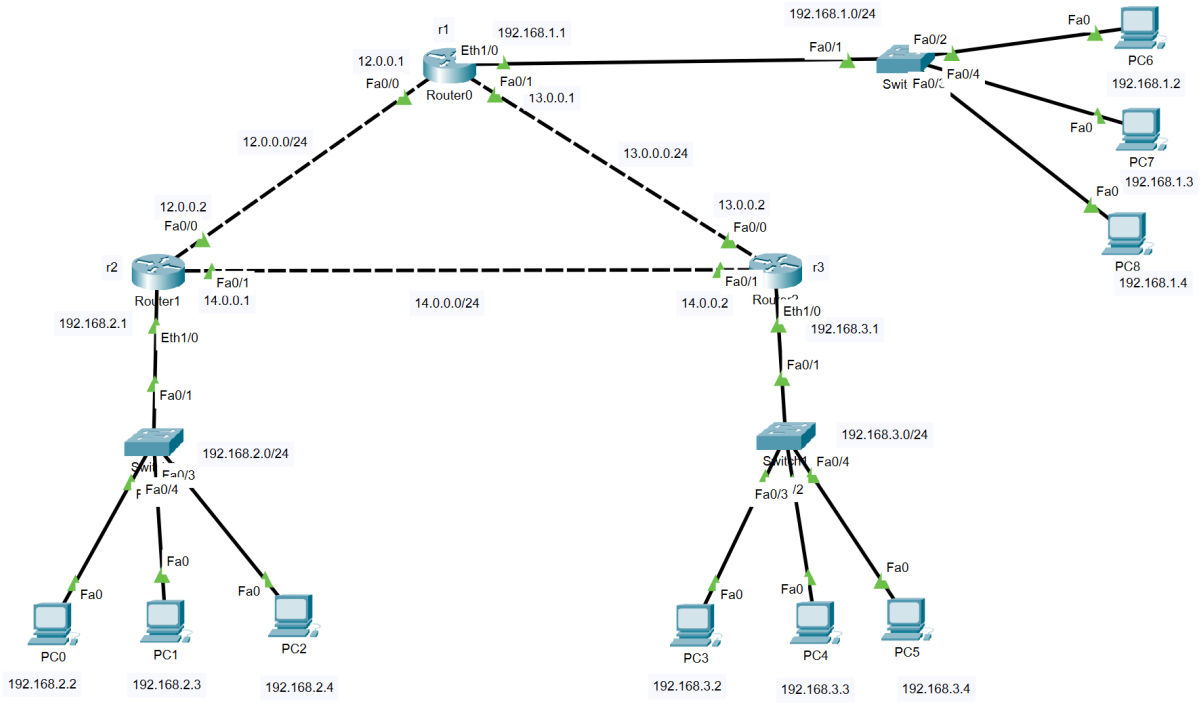
Trace route is used to check the path, which path packets are taking to destination

Example: tracert 192.168.2.2

RIP Drawbacks:

- It sends routing updates every 30 seconds
- It supports only 15 hop counts
- 15 hops max – if further, the network is considered unreachable
- It uses hop counts as metric which means it does not consider bandwidth
- Not suitable for large networks
- Poor loop prevention compared to modern protocols

Reference: LAB-4




```
! on Router 1:
enable
configure terminal
hostname r1
!
Interface Fa0/0
Ip address 12.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 13.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 192.168.1.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip dhcp pool LAN1
Network 192.168.1.0 255.255.255.0
Default-router 192.168.1.1
Dns-server 192.168.1.1
exit
!
Router rip
Version 2
Network 192.168.1.0
Network 12.0.0.0
Network 13.0.0.0
exit
!
```

```
! On Router 2:
enable
configure terminal
hostname r2
!
Interface Fa0/0
Ip address 12.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 14.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 192.168.2.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip dhcp pool LAN2
Network 192.168.2.0 255.255.255.0
Default-router 192.168.2.1
Dns-server 192.168.2.1
Exit
!
Router rip
Version 2
Network 192.168.2.0
Network 12.0.0.0
Network 14.0.0.0
exit
!
```

```
!Router 3:
enable
configure terminal
hostname r3
!
Interface Fa0/0
Ip address 13.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 14.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 192.168.3.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip dhcp pool LAN3
Network 192.168.3.0 255.255.255.0
Default-router 192.168.3.1
Dns-server 192.168.3.1
Exit
!
Router rip
Version 2
Network 192.168.3.0
Network 13.0.0.0
Network 14.0.0.0
exit
!
```

Note:

## ✅ DHCP on a Router – Full Setup & Explanation

In a network, a **DHCP server** automatically assigns IP addresses to client devices.
A **router can act as a DHCP server** to assign:

- IP address
- Subnet mask
- Default gateway
- DNS server

1. Assign IP to Router Interface

```
Router> enable
Router# config t
Router(config)# interface fa0/0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
```

2. Configure DHCP Pool on Router

```
Router(config)# ip dhcp pool LAN1
Router(dhcp-config)# network 192.168.1.0 255.255.255.0
Router(dhcp-config)# default-router 192.168.1.1
Router(dhcp-config)# dns-server 192.168.1.1
Router(dhcp-config)# exit
```

EIGRP (Enhanced Interior Gateway Routing Protocol):

- It was a Cisco priority protocol till 2012
- It uses DUAL Algorithm
- It does not send periodic updates
- It supports 100 hopcounts by default, maximum 255

- It supports equal metric and un equal metric load balancing
- It uses Hello packets for discovering neighbors
- Hello packets are sent for every 5 seconds
- Hold time is 15 seconds
- EIGRP maintains three types of tables:
    1. Neighbor Table:
        - Maintains neighbors' information
    2. Topology Table:
        - Maintains primary and secondary routes to destination
    3. Routing Table:
        - Maintains only primary path
- EIGRP has five metric components. But it uses only two metric components (Bandwidth, Delay)
    1. Bandwidth (Default)
    2. Delay (Default)
    3. Reliability
    4. Load
    5. MTU
- EIGRP does not consider hop counts when selecting shortest path to destination

Note:

Hello packets are used:

1. To form Neighbor Relationship
2. As keep alive (to check whether neighbors are alive or not)

- If Hello packets are not getting within the 5 seconds, it will waiting for 15 seconds after 15 seconds if not getting, neighbor will be declared as DEAD and remove neighbors in neighbors' table.
- EIGRP routing domain can divided into Autonomous System
- Autonomous system number is 1-65535
- Neighbor should have same as number otherwise, neighbors will not be established

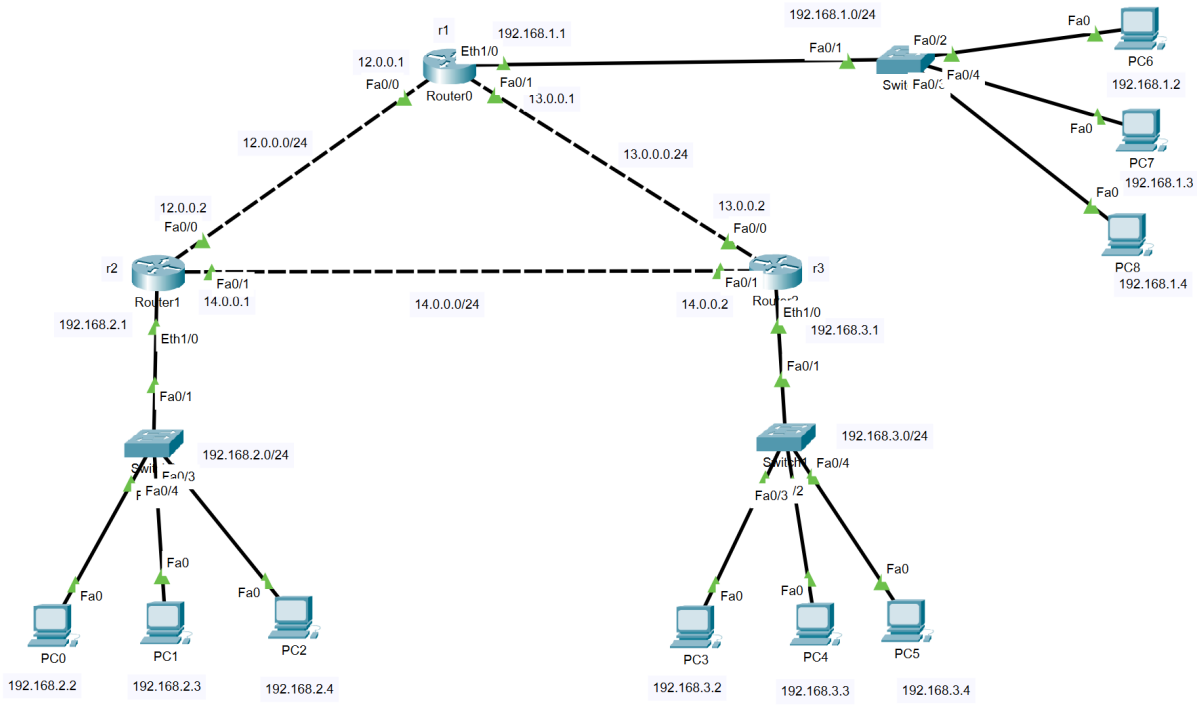
Syntax:

Router eigrp 100

Network <Connected Network>

Here, 100 is Autonomous System number – the AS number can any between 1-65535

Reference: LAB-5



```
! on Router 1:
enable
configure terminal
hostname r1
!
Interface Fa0/0
Ip address 12.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 13.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 192.168.1.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip dhcp pool LAN1
Network 192.168.1.0 255.255.255.0
Default-router 192.168.1.1
Dns-server 192.168.1.1
exit
!
Router eigrp 100
Network 192.168.1.0
Network 12.0.0.0
Network 13.0.0.0
exit
!
```

```
! On Router 2:
enable
configure terminal
hostname r2
!
Interface Fa0/0
Ip address 12.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 14.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 192.168.2.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip dhcp pool LAN2
Network 192.168.2.0 255.255.255.0
Default-router 192.168.2.1
Dns-server 192.168.2.1
Exit
!
Router eigrp 100
Network 192.168.2.0
Network 12.0.0.0
Network 14.0.0.0
exit
!
```

```
!Router 3:
enable
configure terminal
hostname r3
!
Interface Fa0/0
Ip address 13.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 14.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 192.168.3.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip dhcp pool LAN3
Network 192.168.3.0 255.255.255.0
Default-router 192.168.3.1
Dns-server 192.168.3.1
Exit
!
Router eigrp 100
Network 192.168.3.0
Network 13.0.0.0
Network 14.0.0.0
exit
!
```

| EIGRP Metrics | Fast Ethernet | Serial |
|---|---|---|
| **Bandwidth** | 100,000 | 1544 Kbps |
| **Delay** | 100 Ms | 20,000 Ms |
| **Reliability** | 255 | 255 |
| **Load** | 1 | 1 |
| **MTU** | 1500 bytes | 1500 bytes |

**Metric Formula:**

$$\left( \frac{10^7}{\text{Min Bandwidth}} + \frac{\text{Total Delay}}{10} \right) \times 256$$

**Important Router Commands:**

- Show ip eigrp neighbors
    - To see the neighbor table
- Show ip eigrp topology
    - To see primary and secondary routes
- Show ip route eigrp
    - To see the eigrp routes in routing table
- Show ip eigrp traffic
    - To check the how many hello packets are sent
- Show ip protocols
    - display dynamic routing protocol information
- show ip interface brief
    - Display all interfaces
- Show ip interface | include up
    - Display only up state ip interfaces
- Show ip interface | exclude up
    - Display exclude up state ip interfaces
- Show ip interface brief | section up
    - Shows the **section** (entire block of output) starting from the first line that **contains up**
- show ip interface brief | begin up
    - Displays output **starting from the first line that includes the word up**, and **continues to the end**
- show run | include ip address
    - Shows only lines from the running configuration that contain the phrase ip address
- show interface | include line
    - Shows only lines that include the word line
- show ip route | include O
    - Filters and displays only routes learned through the **OSPF** (Open Shortest Path First) protocol

Here, '|' is called pipe

- Show ip route
    - To see routing table
- Show ip route | begin gateway
    - To see only ip in routing table

- ⇨ RIP is a layer 7 protocol because it uses UDP as a transport protocol
- ⇨ RIP port number is 520
- ⇨ EIGRP is a layer 3 protocol because it uses IP as a transport protocol
- ⇨ EIGRP protocol number is 88
- ⇨ ICMP is the management protocol
- ⇨ Ping, trace uses ICMP messages
- ⇨ ICMP is layer 3 protocol
- ⇨ ARP is layer 2 protocol

- When pinging output is unreachable means the router don't the destination networks (remote networks). The route only know directly connected network
- We connect remote networks by static (manually) and Dynamic (RIP, EIGRP)
- Routing protocols are change the networks those who those neighbors

Did You Know ?

| Command | Description |
|---|---|
| show ip route | See all routes and their metrics |
| show ip route <network> | View metric for a specific route |
| show ip ospf interface | View OSPF cost per interface |
| show ip eigrp topology | View EIGRP metrics |
| show ip rip database | View RIP hop counts (metrics) |

**Metric:**

- Helps the router choose the **best path** when multiple paths are available.
- Different routing protocols use **different methods** to calculate metric.

Metric by Routing Protocol:

| Routing Protocol | Metric Type Used | Meaning |
|---|---|---|
| **RIP** | Hop Count | Number of routers the packet must cross |
| **OSPF** | Cost (based on bandwidth) | Lower bandwidth = higher cost |
| **EIGRP** | Composite metric (bandwidth, delay, etc.) | Complex formula considering link quality |
| **BGP** | Attributes (e.g., AS path, local-pref, MED) | Policy-based path selection |
| **Static Route** | Manual metric (optional) | Used when multiple static routes exist |

**EIGRP Terminology:**

Successor:

Next hop router in best path

Successor route:

Primary path which has least/lowest metric to destination

Feasible successor route:

Backup path or secondary path which has higher metric

Feasible Distance:

Total metric from local router to the destination network

If the FD is least that path is become primary path

Advertise Distance: (AD/RD):

Metric from neighbor to destination

Feasibility condition (FC):

Advertise Distance of backup path (Feasible successor) should be less than FD of primary path (successor route)

Example:

```
Router# show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(1)/ID(10.1.1.1)

P 192.168.1.0/24, 1 successors, FD is 28160
        via 10.0.0.2 (28160/2169856), Serial0/0
```

**How to read this:**

- P = Passive (route is stable, ready to use)
- 1 successors = One best route (successor)
- FD is 28160 = Total metric to reach destination
- via 10.0.0.2 (28160/2169856):
    - 28160 = Feasible Distance (your total cost)
    - 2169856 = Reported Distance (cost reported by neighbor)

If you see **"A" = Active**, that means the router is currently **searching** for a route (not stable).

Difference between FD and RD:

Feasible Distance (FD) - Total cost from this router to the destination

Reported Distance (RD) - Cost from the **neighbor** to the destination

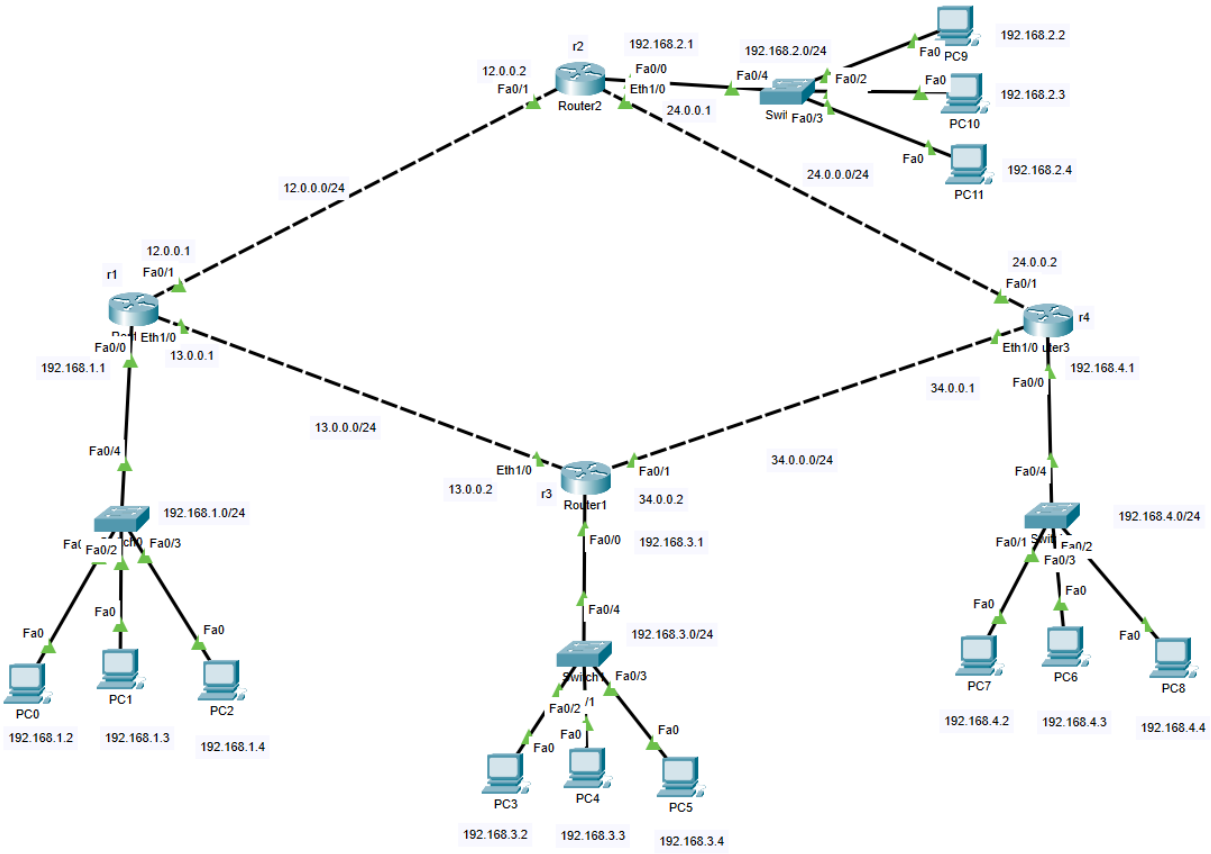
**EIGRP Metric Formula (Default):**

With default **K-values** (1 and 3 enabled), the metric is calculated using **bandwidth** and **delay** only:

$$\textbf{Metric} = [(10^7 / \text{min bandwidth}) + \text{total delay}] \times 256$$

| Term | Meaning |
|---|---|
| $10^7$ | Constant |
| min bandwidth | Minimum bandwidth along the path (in Kbps) |

| total delay | Cumulative delay (in **tens of microseconds**) |
|---|---|
| × 256 | EIGRP scales final value |




```
!Router 1:
enable
configure terminal
hostname r1
!
Interface Fa0/0
Ip address 192.168.1.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 12.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 13.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN1
Network 192.168.1.0 255.255.255.0
Default-router 192.168.1.1
Dns-server 192.168.1.1
Exit
!
Router eigrp 100
Network 192.168.1.0
Network 12.0.0.0
Network 13.0.0.0
exit
!
```

```
! On Router 2:
enable
configure terminal
hostname r2
!
Interface Fa0/0
Ip address 192.168.2.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 12.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 24.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN2
Network 192.168.2.0 255.255.255.0
Default-router 192.168.2.1
Dns-server 192.168.2.1
Exit
!
Router eigrp 100
Network 192.168.2.0
Network 12.0.0.0
Network 24.0.0.0
exit
!
```

```
! On Router 3:
enable
configure terminal
hostname r3
!
Interface Fa0/0
Ip address 192.168.3.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 34.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 13.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN3
Network 192.168.3.0 255.255.255.0
Default-router 192.168.3.1
Dns-server 192.168.3.1
Exit
!
Router eigrp 100
Network 192.168.3.0
Network 13.0.0.0
Network 34.0.0.0
exit
!
```

```
! On Router 4:
enable
configure terminal
hostname r4
!
Interface Fa0/0
Ip address 192.168.4.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 24.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 34.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN4
Network 192.168.4.0 255.255.255.0
Default-router 192.168.4.1
Dns-server 192.168.4.1
Exit
!
Router eigrp 100
Network 192.168.4.0
Network 24.0.0.0
Network 34.0.0.0
exit
!
```

steps for calculating path metric:

1. Configure eigrp
2. Run command for details of interface
    a. Show interface fa0/1
3. Now calculate the metric
4. Formula: Metric = [(10^7 / Bandwidth) + Delay] × 256

**EIGRP Packets:**

- Hello Packets
- Update Packets
- Query Packets
- Reply Packets
- Ack Packets

**Hello packets:**

- Hello packets are used to Discover Neighbor and detect neighbor failures
- Hello packets are Multicast packets and sent to 224.0.0.10

**Update packets:**

- Update packet is used to Routing Information
- Update packets are Unicast packets
- Update packet require ack from neighbors

**Query packets:**

- Query packets are sent when primary path is failed and if no secondary path in Topology Table

**Reply packets:**

- Routers are send reply packets to query packets

**Ack packets:**

- When the router receive update packet it send ack

**what is SIA?**

If router is not getting reply packet for query, it will be waiting for 180 seconds. After 180 seconds that route will be SIA (Stuck In Active)

Note:

RIP is UDP based protocol, it is connectionless protocol

**OSPF (Open Shortest Path First):**

- ⇨ It is a Link State routing protocol
- ⇨ It uses SPF Algorithm
- ⇨ It supports unlimited hop counts
- ⇨ It supports only equal metric Load Balancing
- ⇨ It is a standard protocol
- ⇨ It uses Hello packets for neighbor discovery
- ⇨ Hello packets are sent every 10 seconds
- ⇨ Hello packets are sent to 224.0.0.5
- ⇨ Dead interval is 40 seconds
- ⇨ OSPF maintains 3 types of tables
    1. Neighbor Table
    2. Database Table
    3. Routing Table
- ⇨ OSPF sends routing updates every 30 minutes
- ⇨ OSPF routing domain can be divided into area
- ⇨ Areas means logical group of OSPF networks
- ⇨ Areas are two types
    1. Backbone Area
    2. Non-Backbone Area
- ⇨ Backbone Area is 0
- ⇨ Non-Backbone Area range is 1-65535

**Wildcard Mask:**

- It is also known as inverse mark
- It is used to find out number of host in network
  Example:
  Network: 192.168.1.0/24
  Subnet mask: 255.255.255.0
  Default Subnet mask: 255.255.255.255

255.255.255.255

(-) 255.255.255.0

_____

0.0.0.255 ----------→ Wildcard Mask

_____

**Process ID:**

Process ID is used to enable the OSPF instance and process ID range is 1-65535

**Router ID:**

- It is used to identify the OSPF router in OSPF routing domain
- It is a 32-bit address
- Any IPv4 address can be configured as router ID
- It can be configured either statically or dynamically
- It should be unique

**Router ID selection process:**

- OSPF always prefers manual router -D
- If normal router-ID is configured, Loopback IP address will be taken as router-ID
- If no loopback and manual router-ID, OSPF always takes highest IP address of any active physical interface
- Loopback interface is a Logical interface
- We can configure number of Loopbacks in router
- Loopback interface configuration:

  Interface loopback 0

  Ip address 1.1.1.1 255.255.255.255.255

  Exit

**OSPF Syntax:**

**Route ospf <Process ID>**

**Router-id <Unique IP Address>**

**Network <Connected network ID> <Wildcard Mask> Area <Area Number>**

Here,

1 is Process ID

Area Number – is same on other routers

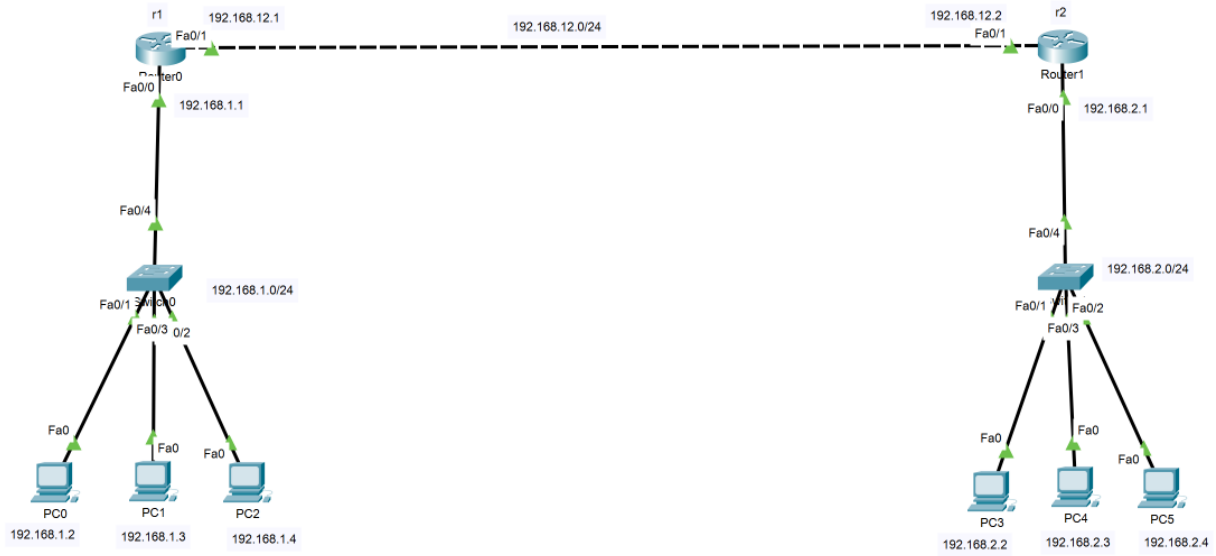
Note:

Commands:

Show ip protocols

Show ip eigrp neighbors

Show ip ospf neighbors




```
! On Router 1:
enable
configure terminal
hostname r1
!
Interface Fa0/0
Ip address 192.168.1.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 192.168.12.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN1
Network 192.168.1.0 255.255.255.0
Default-router 192.168.1.1
Dns-server 192.168.1.1
Exit
!
Router ospf 1
router-id 1.1.1.1
Network 192.168.1.0 0.0.0.255 area 0
Network 192.168.12.0 0.0.0.255 area 0
exit
!
```

```
! On Router 2:
enable
configure terminal
hostname r2
!
Interface Fa0/0
Ip address 192.168.2.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/1
Ip address 192.268.12.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN2
Network 192.168.2.0 255.255.255.0
Default-router 192.168.2.1
Dns-server 192.168.2.1
Exit
!
Router ospf 1
router-id 2.2.2.2
Network 192.168.2.0 0.0.0.255 area 0
Network 192.168.12.0 0.0.0.255 area 0
exit
!
```

**OSPF Neighbors States:**

- Down
- Init
- 2 way
- Extract
- Exchange
- Loading
- Full

**Down:**

OSPF router is both receiving Hello packets from its neighbor but it can send Hello packets

**Init:**

Router can receive Hello packets from neighbor but it can not see it's router id in received hello packets

**2 Way:**

Both routers can see it's router id in receiver hello packets

Or

In this state Bi-Directional communication established and OSPF perform DR/BDR elections

**Extract:**

In this state router exchange database description packets to neighbors. And also OSPF perform Master and slave elections. Master router responsibility is to generate ISN (Initial Sequence number) for exchanging database.

**Exchange:**

In this state routers can send request to neighbors for database from neighbors

**Loading:**

In this state routers send update packets to neighbors who has sent request for updates

**Full:**

In this full communication will be established

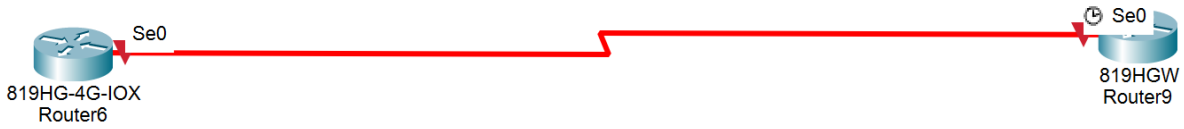
**What is difference between neighbor and adjacency neighbors?**

- Neighbor means only exchange Hello packets
- Adjacency neighbors means they can exchange Hello packets and Database packets.

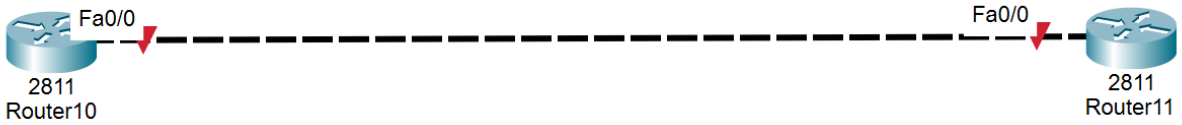
**OSPF Network Types:**

- Point-to-Point (Default on serial interface)
- Broadcast Networks (Default on Ethernet interface)
- Point-to-Multipoint (optional)
- Non broadcast (optional)

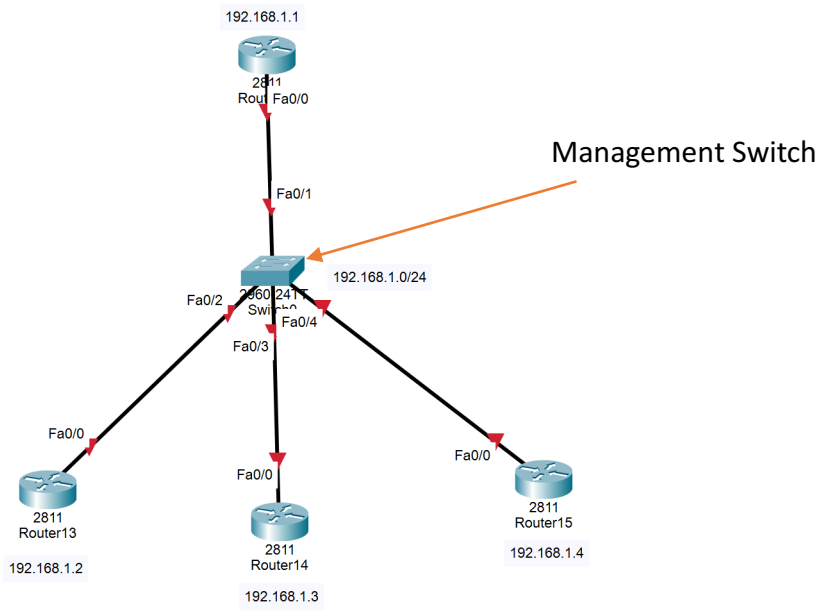
**Point-to-Point:**




**Broadcast (Ethernet):**




**Broadcast Network:**



**DR/BDR Election Process:**

**DR (Designated Router)** and **BDR (Backup Designated Router)** in OSPF

OSPF DR/BDR Explanation

| Term | Meaning |
|---|---|
| **DR** | **Designated Router**: The main router elected on a broadcast or multi-access network (like Ethernet) to reduce OSPF traffic by acting as a hub. |
| **BDR** | **Backup Designated Router**: The standby router that takes over if the DR fails. Maintains the same adjacencies as the DR. |
| **Purpose** | DR/BDR help reduce the number of OSPF adjacencies and LSAs (Link-State Advertisements) on networks with multiple routers. |
| **Election** | Elected based on: 1. Highest **OSPF priority** (default is 1; 0 means not eligible) 2. If tied, highest **Router ID** is used. |
| **Non-DR/BDR Routers** | Called **DROthers**. They form adjacencies only with DR and BDR, not with each other. |

OSPF performs DR/BDR elections on Broadcast Networks:

1. DR is elected based on highest interface priority

2. Default priority is 1

3. If priority is equal on all routers, then it use router-ID

4. The router which has highest router-ID become DR

5. The router which has second highest router-ID become BDR

6. Remaining all other routers become DRothers

**OSPF use two multicast address:**

- **224.0.0.5**
  - ⇨ Is used to exchange Hello packets
- **224.0.0.6**
  - ⇨ Is used to exchange Database to DR and BDR

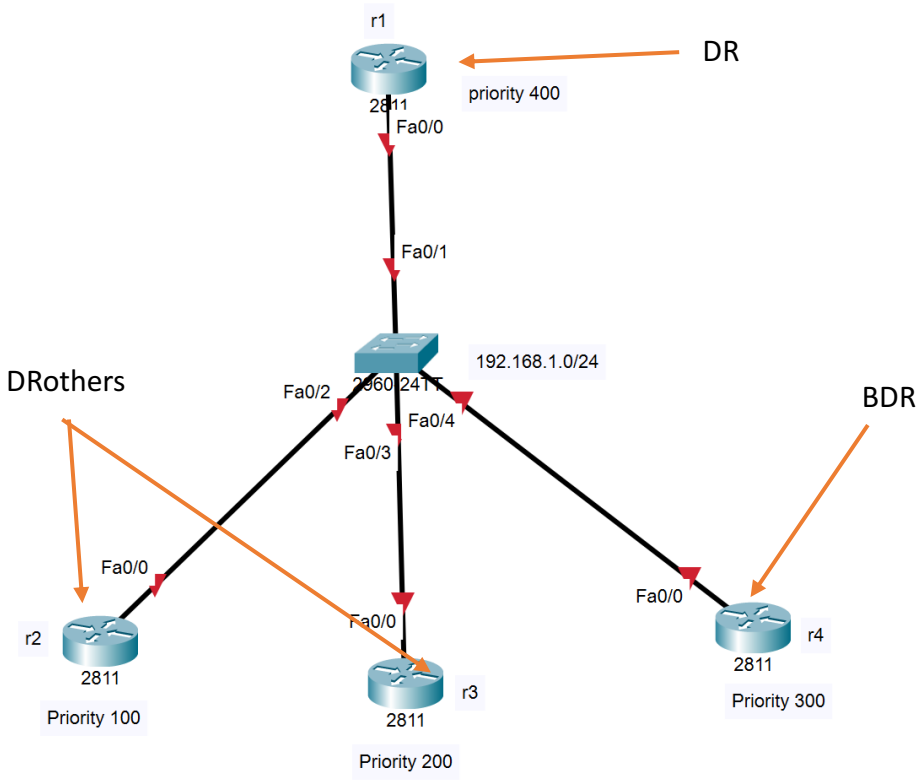
**Where DR/BDR Are Used:**

- **Only on broadcast (e.g., Ethernet) and non-broadcast multi-access (NBMA) networks (e.g., Frame Relay).**
- **Not used on point-to-point or point-to-multipoint links.**

---

✅ **Example Scenario:**

- **3 routers on a switch: R1, R2, R3**
- **OSPF Priority:**
  - **R1: 1**
  - **R2: 100**
  - **R3: 1**
- **Result:**
  - **R2 becomes DR (highest priority)**
  - **R1 or R3 becomes BDR (highest Router ID among remaining)**
  - **The other is DROther**



- **OSPF** behaves in different ways based on interface
    - Serial → PTP (Point-To-Point)
    - Ethernet → Broadcast

**2 Way:**

Between Drother and Drother is a 2 way communication

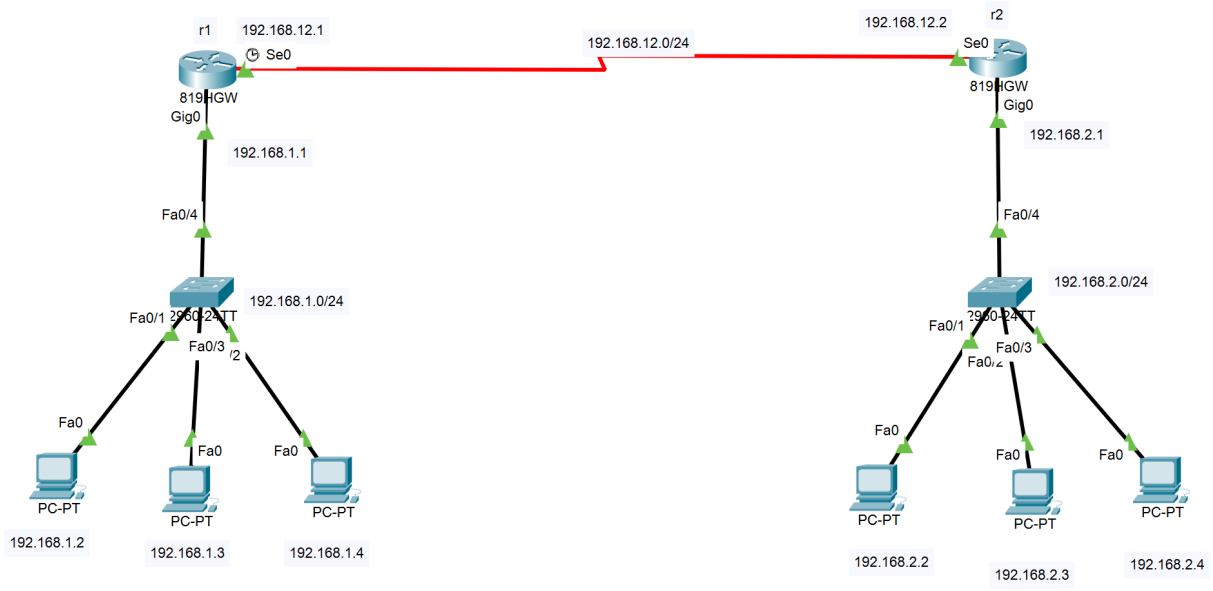
**Full:**

Exchange Hello packets & database data packets

Example:

Interface f0/0

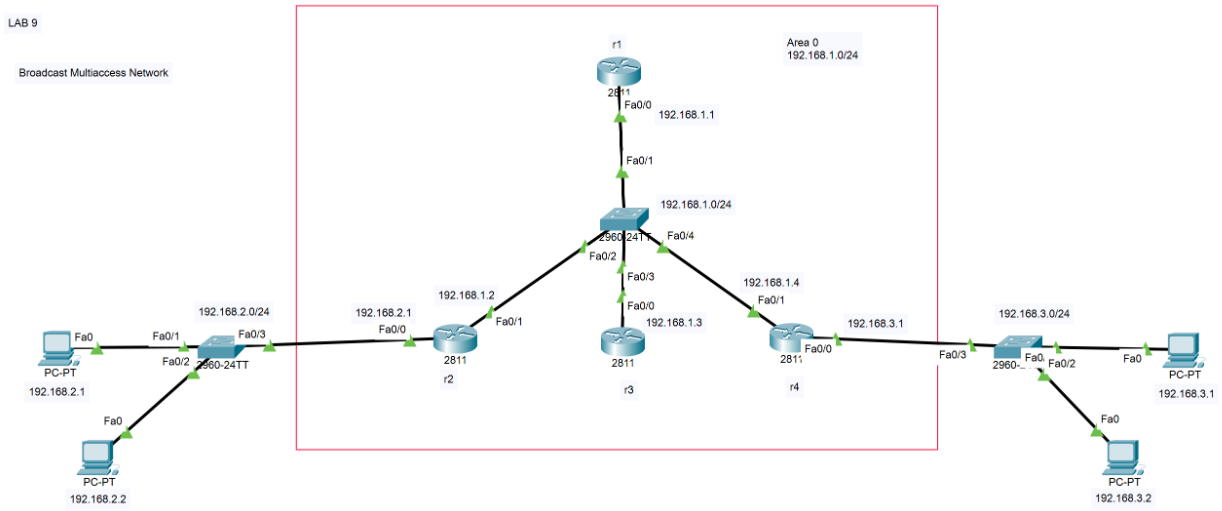
Ip ospf priority 0 → 0 means DRothers




```
! On Router 1:
enable
configure terminal
hostname r1
!
Interface Gig0
Ip address 192.168.1.1 255.255.255.0
No shutdown
Exit
!
Interface Se0
Ip address 192.168.12.1 255.255.255.0
No shutdown
Exit
!
Interface Gig0
Ip dhcp pool LAN1
Network 192.168.1.0 255.255.255.0
Default-router 192.168.1.1
Dns-server 192.168.1.1
Exit
!
Router ospf 1
Router-id 1.1.1.1
Network 192.168.1.0 0.0.0.255 area 0
Network 192.168.12.0 0.0.0.255 area 0
exit
!
```

```
! On Router 2:
enable
configure terminal
hostname r2
!
Interface Gig0
Ip address 192.168.2.1 255.255.255.0
No shutdown
Exit
!
Interface Se0
Ip address 192.168.12.2 255.255.255.0
No shutdown
Exit
!
Interface Gig0
Ip dhcp pool LAN2
Network 192.168.2.0 255.255.255.0
Default-router 192.168.2.1
Dns-server 192.168.2.1
Exit
!
Router ospf 1
Router-id 2.2.2.2
Network 192.168.2.0 0.0.0.255 area 0
Network 192.168.12.0 0.0.0.255 area 0
exit
!
```

## Broadcast Multiaccess Network:




```
! On R1

enable
configure terminal
hostname r1
!
interface Fa0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
!
router ospf 1
router-id 11.11.11.111
network 192.168.1.0 0.0.0.255 area 0
exit
!
```

```
! On R2

enable
configure terminal
hostname r2
!
interface Fa0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 192.168.1.2 255.255.255.0
no shutdown
exit
!
router ospf 1
router-id 2.2.2.2
network 192.168.1.0 0.0.0.255 area 0
networ 192.168.2.0 0.0.0.255 area 0
exit
!
!
interface Fa0/0
ip dhcp pool LAN2
network 192.168.2.0 255.255.255.0
default-router 192.168.2.1
exit
!
```

```
! On R3

enable
configure terminal
hostname r3
!
interface Fa0/0
ip address 192.168.1.3 255.255.255.0
no shutdown
exit
!
router ospf 1
router-id 3.3.3.3
network 192.168.1.0 0.0.0.255 area 0
exit
!
```

```
! On R4

enable
configure terminal
hostname r4
!
interface Fa0/0
ip address 192.168.3.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 192.168.1.4 255.255.255.0
no shutdown
exit
!
router ospf 1
router-id 4.4.4.4
network 192.168.1.0 0.0.0.255 area 0
networ 192.168.3.0 0.0.0.255 area 0
exit
!
interface Fa0/0
ip dhcp pool LAN3
network 192.168.3.0 255.255.255.0
default-router 192.168.3.1
exit
!
exit
write
```

Did You Know ?

In Organization:

Operation team → Ticket

Build team → Migrate/ new development

Tickets:

- Service request (SR)
- Incident
- Change management (CR)

SLA → Service Level Agreement

Incident:

| Tickets | Response Time | Resolution Time | Outage |
|---|---|---|---|
| P1 | 5 Minutes | 1 Hour | |
| P2 | 30 Minutes | 4 Hours | |
| P3 | 4 Hours | 8 Hours | |
| P4 | 8 Hours | 24 Hours | |

Tools:

- Ticketing Tools:
  - ServiceNow, Myshift
- Monitoring Tools:
  - SolarWinds Network performance Monitor, Auvik

# ACL (Access Control Unit)

**Access Control List**: A set of rules used to **permit or deny traffic** on a router or switch, based on source/destination IP, protocol, or port. It's like a firewall rule on an interface.

**Purpose of ACLs:**

- Control traffic **into or out of** a network.
- Filter packets based on criteria like IP address, protocol, or port.
- Improve **security**, manage **bandwidth**, or limit **access** to sensitive areas.

- ACL is used to control incoming and outgoing traffic

**What is an ACL?**

An Access Control List (ACL) is a set of rules used on routers or firewalls to filter traffic based on specific conditions such as:

- Source IP address
- Destination IP address
- Protocol (TCP/UDP/ICMP)
- Port numbers
- etc.

ACLs are used to permit or deny traffic.

**Types of ACLs:**

| Type | Number Range | Features |
|---|---|---|
| **Standard ACL** | **1–99, 1300–1999** | **Filters traffic only by source IP address.** |
| **Extended ACL** | **100–199, 2000–2699** | **Filters traffic by source, destination, protocol, port, etc.** |
| **Named ACL** | **Name-based (string)** | **Same as numbered but easier to manage with names instead of numbers.** |

**Standard ACL Syntax (Numbered):**

access-list <ACL no> <Action> <Source IP> <Wildcard Mask>

**Standard ACL Syntax (Numbered):**

Access-list standard block-<Network IP>

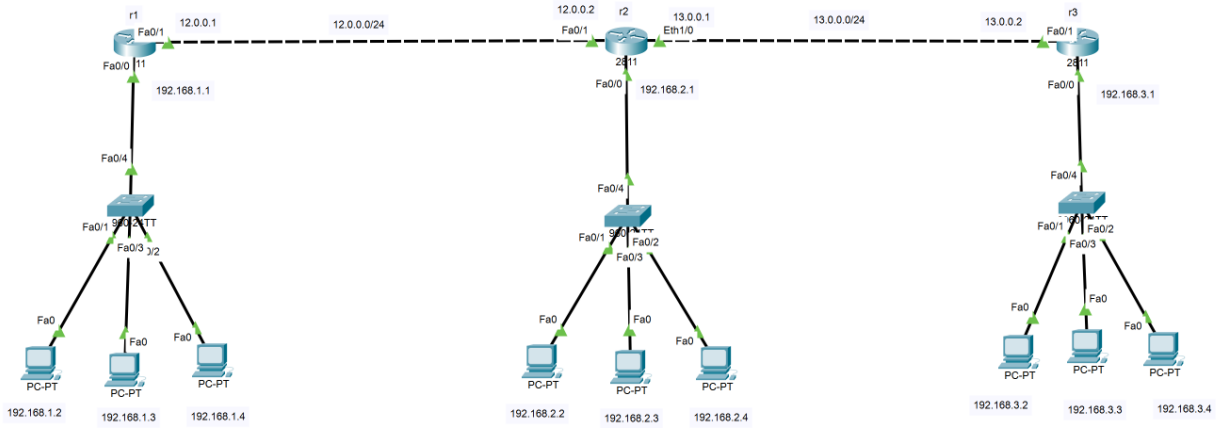
Deny <Network IP> <Wildcard Mask>

Permit any

Interface <Interface Name>

**Numbered ACL:**

Syntax: access-list <ACL no> <Action> <Source IP> <Wildcard Mask>



```
! On r1
Enable
Configure terminal
Hostname r1
!
Interface Fa0/1
Ip address 12.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.1.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN1
Network 192.168.1.0 255.255.255.0
Default-router 192.168.1.1
Exit
!
Router ospf 1
Router-id 1.1.1.1
Network 192.168.1.0 0.0.0.255 area 0
Network 12.0.0.0 0.0.0.255 area 0
Exit
!
! On r1 Configuration mode
Access-list 10 deny 192.168.3.0 0.0.0.255
Access-list 10 permit any
! Apply ACL in interface Fa0/0
Interface Fa0/0
Access-group 10 out
exit
!
exit
!
write
```

```
! On r2
Enable
Configure terminal
Hostname r2
!
Interface Fa0/1
Ip address 12.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.2.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 13.0.0.1 255.255.255.0
No shutdown
Exit
!
! On r2
Interface Fa0/0
Ip dhcp pool LAN2
Network 192.168.2.0 255.255.255.0
Default-router 192.168.2.1
Exit
!
Router ospf 1
Router-id 2.2.2.2
Network 192.168.2.0 0.0.0.255 area 0
Network 12.0.0.0 0.0.0.255 area 0
Network 13.0.0.0 0.0.0.255 area 0
Exit
!
!
exit
!
write
```

```
! On r3
Enable
Configure terminal
Hostname r3
!
Interface Fa0/1
Ip address 13.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.3.1 255.255.255.0
No shutdown
Exit
!
! On r3
Interface Fa0/0
Ip dhcp pool LAN3
Network 192.168.3.0 255.255.255.0
Default-router 192.168.3.1
Exit
!
Router ospf 1
Router-id 3.3.3.3
Network 192.168.3.0 0.0.0.255 area 0
Network 13.0.0.0 0.0.0.255 area 0
Exit
!
!
exit
!
write
```

Note:

- we block the 192.168.3.0/24 entire network in r1
- If we need to block the any particular system. Then,

  Access-list 10 deny 192.168.3.2
  Access-list 10 permit any
  Interface Fa0/0
  Access-group 10 out

  Here, we mentioned a particular system IP address to block the traffic. For that the r1 router block the 192.168.3.2 traffic, and allow other all system traffic including 192.168.3.0/24 network except 192.168.3.2 traffic.

**Standard Named ACL:**

- It provides layer 3 security (IP protocols)
- It allows based on source IP address
- It is always configured close to destination of packet
- Number range is 1-99

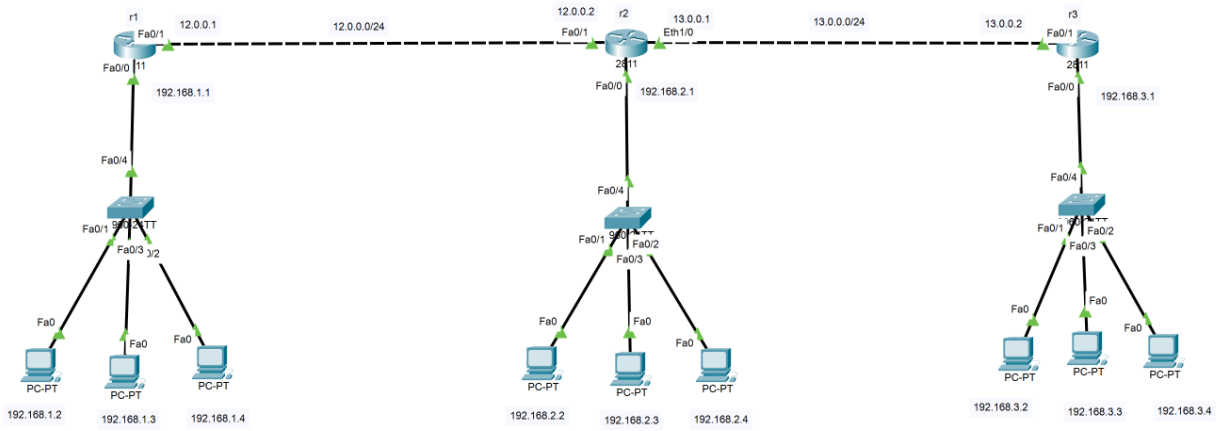
Syntax:

Access-list standard block-<Network IP>

Deny <Network IP> <Wildcard Mask>

Permit any

Interface <Interface Name>



```
! On r1
Enable
Configure terminal
Hostname r1
!
Interface Fa0/1
Ip address 12.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.1.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN1
Network 192.168.1.0 255.255.255.0
Default-router 192.168.1.1
Exit
!
Router ospf 1
Router-id 1.1.1.1
Network 192.168.1.0 0.0.0.255 area 0
Network 12.0.0.0 0.0.0.255 area 0
Exit
!
exit
!
write
```

```
! On r2
Enable
Configure terminal
Hostname r2
!
Interface Fa0/1
Ip address 12.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.2.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 13.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN2
Network 192.168.2.0 255.255.255.0
Default-router 192.168.2.1
Exit
!
Router ospf 1
Router-id 2.2.2.2
Network 192.168.2.0 0.0.0.255 area 0
Network 12.0.0.0 0.0.0.255 area 0
Network 13.0.0.0 0.0.0.255 area 0
Exit
!
exit
!
write
```

```
! On r3
Enable
Configure terminal
Host namer3
!
Interface Fa0/1
Ip address 13.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.3.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN3
Network 192.168.3.0 255.255.255.0
Default-router 192.168.3.1
Exit
!
Router ospf 1
Router-id 3.3.3.3
Network 192.168.3.0 0.0.0.255 area 0
Network 13.0.0.0 0.0.0.255 area 0
Exit
!
ip access-list standard block-192.168.2.0
Deny 192.168.2.0 0.0.0.255
Permit any
exit
!
! Apply ACL in interface Fa0/0
Interface Fa0/0
Ip access-group block-192.168.2.0 out
Exit
!
exit
!
write
```

**Extended Numbered ACL:**

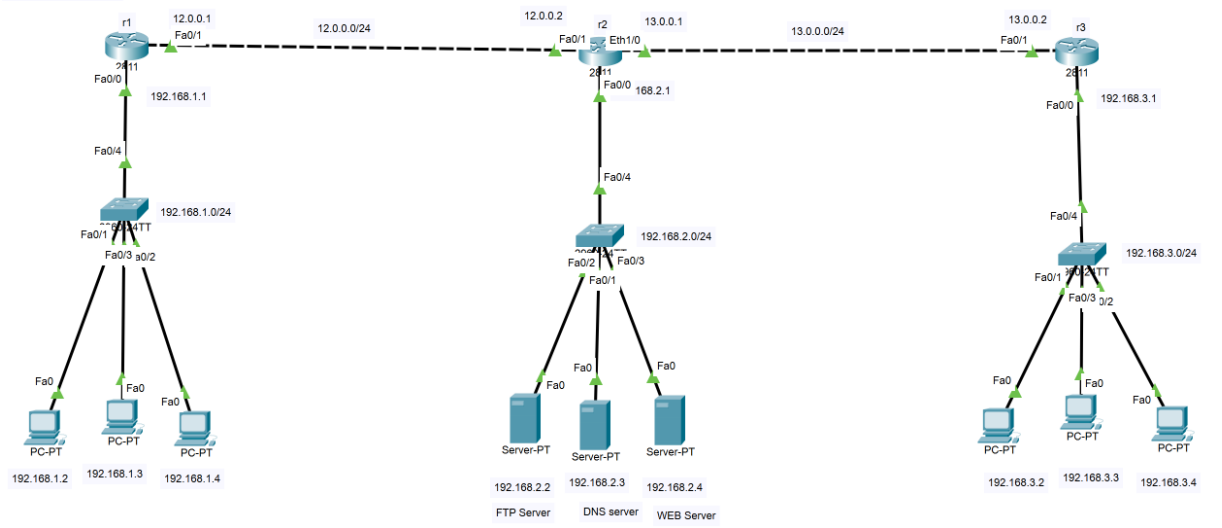
It filters the traffic based on source IP, destination IP and destination port number.

It always configured close to source of IP packet

Number range is 100 – 199

Syntax:

Access-list <ACL Number> <Action> <Protocol> <Source IP> <Wildcard Mask> <Destination IP> <Wildcard Mask> <Operator> <Port Number>




```
! On r1
Enable
Configure terminal
Hostname r1
!
Interface Fa0/1
Ip address 12.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.1.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN1
Network 192.168.1.0 255.255.255.0
Default-router 192.168.1.1
Exit
!
Router ospf 1
Router-id 1.1.1.1
Network 192.168.1.0 0.0.0.255 area 0
Network 12.0.0.0 0.0.0.255 area 0
Exit
!
```

```
! On r2
Enable
Configure terminal
Hostname r2
!
Interface Fa0/1
Ip address 12.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.2.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 13.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN2
Network 192.168.2.0 255.255.255.0
Default-router 192.168.2.1
Exit
!
Router ospf 1
Router-id 2.2.2.2
Network 192.168.2.0 0.0.0.255 area 0
Network 12.0.0.0 0.0.0.255 area 0
Network 13.0.0.0 0.0.0.255 area 0
Exit
!
```

```
! On r3
Enable
Configure terminal
Host namer3
!
Interface Fa0/1
Ip address 13.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.3.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN3
Network 192.168.3.0 255.255.255.0
Default-router 192.168.3.1
Exit
!
Router ospf 1
Router-id 3.3.3.3
Network 192.168.3.0 0.0.0.255 area 0
Network 13.0.0.0 0.0.0.255 area 0
Exit
!
! Block FTP (ports 20 and 21) from 192.168.3.0/24 to anywhere
        access-list 100 deny tcp host 192.168.3.2 any range 20 21
        ! Allow all other traffic
        access-list 100 permit ip any any
        ! Apply ACL inbound on Fa0/0
        Interface FastEthernet0/0
        ip access-group 100 in
```

```
! On FTP Server
! On Services
! Select FTP
! Select ON
! enter username: user1
! entner password: pass1
! Select read option
! Click on ADD
```

Command to see ACL List: show access-lists

---

Note:

Why we use ftp 192.168.3.2 instead of tcp

Here's why:

- **FTP uses TCP ports 21 (control) and 20 (data)**.
- When you run ftp 192.168.2.2 from a PC in Packet Tracer, the PC tries to open a TCP connection to **port 21** on the server.

- If your ACL is correct, one of two things will happen:
    - **Allowed** → You'll see Connected to 192.168.2.2 and then a login prompt.
    - **Blocked** → You'll see Connection timed out or no response.
- Simply pinging (ping 192.168.2.2) won't help here — ping uses **ICMP**, not TCP.

So, the ftp command is just a quick way to **trigger TCP port 21 traffic** so you can see if your ACL rule is really matching and blocking only the intended host.

---

**How to block entire network?**

If we need to block the entire network 192.168.3.0/24 then,

Syntax is:

```
access-list 100 deny tcp 192.168.3.0 0.0.0.255 any range 20 21

access-list 100 permit ip any any

interface FastEthernet0/0

ip access-group 100 in
```

---

**How to block ping?**

**1) Block all pings from the entire subnet 192.168.3.0/24**

Blocks any ICMP (including echo) originating from that subnet to anywhere.

Syntax:

```
access-list 110 deny icmp 192.168.3.0 0.0.0.255 any

access-list 110 permit ip any any


interface FastEthernet0/0

ip access-group 110 in
```

**2) Block only echo requests (ping) from the entire subnet**

More precise — allows other ICMP types but stops ping requests.

Syntax:

```
access-list 110 deny icmp 192.168.3.0 0.0.0.255 any echo
access-list 110 permit ip any any

interface FastEthernet0/0
 ip access-group 110 in
```

**3) Block ping from one host (192.168.3.2) to anywhere**

If you want to block just that PC from pinging anything:

Syntax:

```
access-list 110 deny icmp host 192.168.3.2 any echo
access-list 110 permit ip any any

interface FastEthernet0/0
 ip access-group 110 in
```

**4) Block ping to a specific server (e.g., 192.168.2.2)**

If you want to prevent *anyone* from pinging that server (apply inbound on the router interface facing the server, or on the router upstream):

Syntax:

```
access-list 110 deny icmp any host 192.168.2.2 echo
access-list 110 permit ip any any

! apply on interface facing the server (or on the upstream interface)
interface FastEthernet0/x
ip access-group 110 in
```

**Extended Named ACL:**

- Use a **custom name** instead of a number.
- Can filter **traffic based on multiple criteria** — source IP, destination IP, protocol (TCP, UDP, ICMP, etc.), and port numbers.
- Support more descriptive management (you can edit specific lines instead of re-creating the ACL).

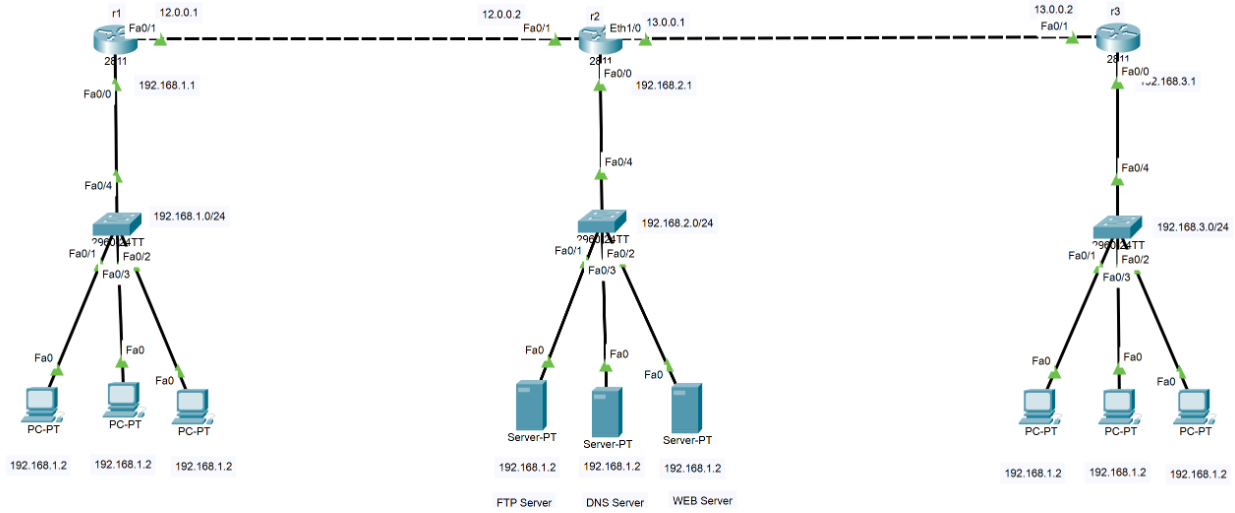
**Why use Named ACLs instead of Numbered ACLs?**

- Easier to remember (BLOCK_HTTP is more descriptive than 101).
- Can insert or delete individual statements without removing the whole ACL.
- Better for large configurations or frequent changes.

Syntax:

Router(config)# ip access-list extended <ACL_NAME>

Router(config-ext-nacl)# permit|deny protocol source source_wildcard destination destination_wildcard [eq port]



```
! On r1
Enable
Configure terminal
Hostname r1
!
Interface Fa0/1
Ip address 12.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.1.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN1
Network 192.168.1.0 255.255.255.0
Default-router 192.168.1.1
Exit
!
Router ospf 1
Router-id 1.1.1.1
Network 192.168.1.0 0.0.0.255 area 0
Network 12.0.0.0 0.0.0.255 area 0
Exit
!
```

```
! On r2
Enable
Configure terminal
Hostname r2
!
Interface Fa0/1
Ip address 12.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.2.1 255.255.255.0
No shutdown
Exit
!
Interface Eth1/0
Ip address 13.0.0.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN2
Network 192.168.2.0 255.255.255.0
Default-router 192.168.2.1
Exit
!
Router ospf 1
Router-id 2.2.2.2
Network 192.168.2.0 0.0.0.255 area 0
Network 12.0.0.0 0.0.0.255 area 0
Network 13.0.0.0 0.0.0.255 area 0
Exit
!
```

```
! On r3
Enable
Configure terminal
Host namer3
!
Interface Fa0/1
Ip address 13.0.0.2 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip address 192.168.3.1 255.255.255.0
No shutdown
Exit
!
Interface Fa0/0
Ip dhcp pool LAN3
Network 192.168.3.0 255.255.255.0
Default-router 192.168.3.1
Exit
!
Router ospf 1
Router-id 3.3.3.3
Network 192.168.3.0 0.0.0.255 area 0
Network 13.0.0.0 0.0.0.255 area 0
Exit
!
ip access-list extended icmp-blocking
deny icmp host 192.168.3.2 host 13.0.0.1 echo
permit ip any any
interface Fa0/0
ip access-group icmp-blocking in
```

**Block the IP:**

ip access-list extended ip-blocking

deny ip host 192.168.3.2 host 192.168.2.2

permit ip any any

interface Fa0/0

ip access-group ip-blocking in

**Block the FTP:**

ip access-list extended ftp-blocking

deny tcp 192.168.3.0 0.0.0.255 host 192.168.3.2 equal ftp

permit ip any any

interface Fa0/0

ip access-group ftp-blocking in

**ACL COMMANDS:**

show access-lists

show ip access-list

**Default Routing:**

**Default Routing** is a routing method used when a router doesn't have a specific route in its routing table for a destination network.
Instead of dropping the packet, the router forwards it to a **default route** — often called the **gateway of last resort**.

---

**1. Definition**

A **default route** is a route that matches **all possible destinations** that are not explicitly listed in the routing table.
It is typically represented as:

IPv4: 0.0.0.0/0

IPv6: ::/0

---

**2. Purpose**

- Used when you want all unknown traffic to go to a single next hop.
- Common in:
    - **Small networks** with only one way out to the internet.
    - **Stub networks** (only one router to connect outside).

---

**3. Syntax in Cisco IOS:**

Router(config)# ip route 0.0.0.0 0.0.0.0 <next-hop-ip>

**Example:**

Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.1.1

This means:
If no match in routing table → send to **192.168.1.1**.

### 4. How It Works

When a packet arrives:

1. Router checks the **destination IP** against its routing table.
2. If no more specific route is found, it uses the **default route**.
3. Packet is sent to the **next-hop IP** or **exit interface** defined in the default route.

---

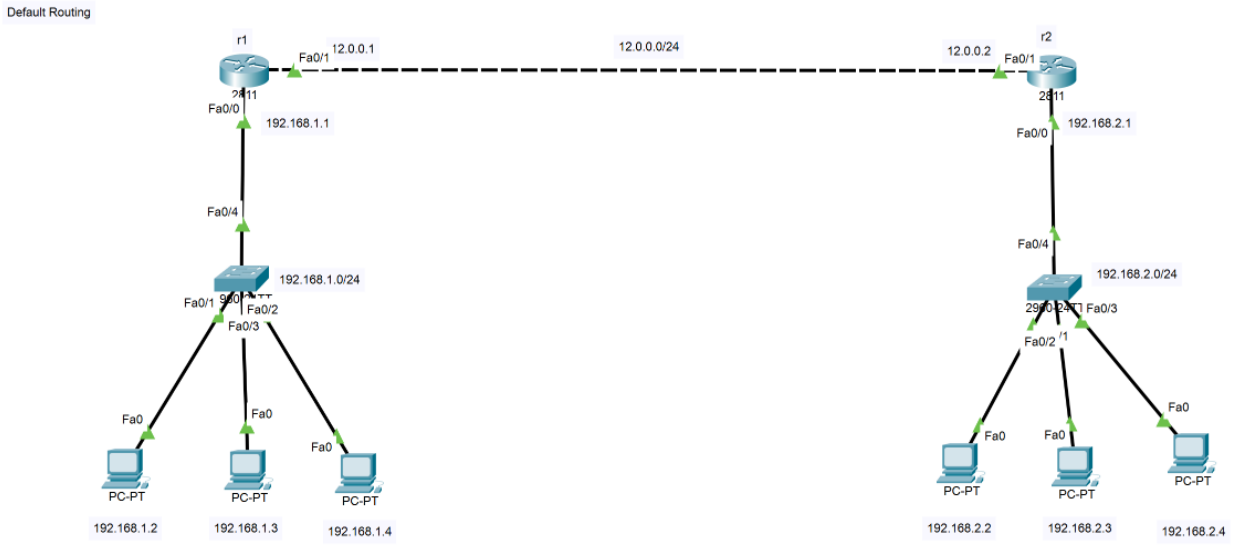
### 5. To see Default Routing table

Command is: show ip route

---

### 6. Real-World Example

**Home Router**:
Your home Wi-Fi router doesn't know routes for every IP in the world.
So, it has a **default route** pointing to your ISP's router.



```
! On r1
enable
configure terminal
hostname r1
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.1 255.255.255.0
no shutdown
exit
!
! On r1 configure mode
interface Fa0/0
ip dhcp pool LAN1
network 192.168.1.0 255.255.255.0
default-router 192.168.1.1
exit
!
ip route 0.0.0.0 0.0.0.0 12.0.0.2
```

```
! On r2
enable
configure terminal
hostname r2
!
interface Fa0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.2 255.255.255.0
no shutdown
exit
!
! on r2 configure mode
interface Fa0/0
ip dhcp pool LAN2
network 192.168.2.0 255.255.255.0
default-router 192.168.2.1
exit
!
ip route 0.0.0.0 0.0.0.0 12.0.0.1
```

## NAT (NETWORK ADDRESS TRANLATION)

In networking, **NAT (Network Address Translation)** is a process where a router or firewall modifies IP address information in packet headers as they pass through it.
It's mainly used to allow multiple devices on a private network to share a single public IP address.

---

**Why NAT is Used**

1. **IP Address Conservation** – Saves public IPv4 addresses by using private addresses internally.
2. **Security** – Hides internal network structure from the internet.

3. **Connectivity** – Lets private devices access the internet without needing unique public IPs.

---

**Types of NAT**

| Type | Description | Example |
|---|---|---|
| **Static NAT** | One-to-one mapping between a private IP and a public IP. | Internal server hosting a website. |
| **Dynamic NAT** | Maps private IPs to any available public IP from a pool. | Rare in-home networks. |
| **PAT (Port Address Translation) / Overloading** | Multiple private IPs share a single public IP, differentiated by port numbers. | Home internet routers. |

---

**How It Works (Example: PAT)**

1. Device with private IP 192.168.1.10 sends a request to the internet.
2. NAT router changes source IP to its public IP (e.g., 203.0.113.5) and assigns a unique source port.
3. Response from the internet comes back to the router.
4. Router looks up the port mapping and sends the packet to the correct internal device.

**1. NAT translates one IP to another IP**

That's the core idea. NAT rewrites the source or destination IP address (sometimes ports too) in packet headers as they pass through a NAT device

---

**2. Types of translations by address type**

| Translation | Example Scenario | Notes |
|---|---|---|
| **Private → Private** | Two different internal subnets in the same company that need to communicate but have different addressing. | Often done in lab or inter-VLAN NAT setups. |
| **Private → Public** | A home PC (192.168.1.10) accessing the internet via ISP public IP (203.0.113.5). | Most common form (PAT/overloading). |
| **Public → Public** | ISP load balancing or when you want to hide the real public IP of a server behind another public IP. | Used in data centers and proxy services. |
| **Public → Private** | Allowing internet users to access an internal server (e.g., web server at 192.168.1.100 via public IP 198.51.100.20). | Achieved via **Static NAT** or **port forwarding**. |

---

**3. NAT for overlapping networks**

- Problem: Two networks use the same IP range (e.g., both use 192.168.1.0/24) and need to talk.
- Solution: NAT can translate one side's address into a non-conflicting range before forwarding packets, allowing communication.
- This is called NAT for overlapping networks or NAT with double translation
  Example:

  Site A: 192.168.1.0/24
  Site B: 192.168.1.0/24

  Router at Site A translates:
  192.168.1.x → 10.10.10.x
  before sending to Site B.

---

**Quick analogy**

Think of NAT like a receptionist at an office:

- You call the office (public number), receptionist routes it to the right desk (private number).

- When an employee calls you back, receptionist dials out from the office main line so you never see the real desk phone number.
- They can even give people in **two different offices using the same extension numbers** a way to talk without confusion — by remapping the numbers.

**PAT (Port Address Translation)**

**Port Address Translation** is a type of NAT (Network Address Translation) that lets multiple private IP addresses share a single public IP address by differentiating each connection using port numbers. Because of this, it's often called NAT Overload.

**Why PAT is used?**

- To save public IPv4 addresses (since they're limited)
- To allow many devices on a LAN to access the internet at the same time
- To provide a basic security layer (internal Ips are hidden from the public network)

**How PAT works (Step-by-Step):**

**1.** Private device sends a request

Example: 192.168.1.10 sends a request to a website.

**2. PAT** changes the source IP from the private address to the router's public IP

**3. PAT** changes the source port to a unique port number to track the session

**4.** Router keeps a translation table mapping:

Private IP: Port → Public IP: Unique Port

**5.** Response comes back to the public IP and port.

**6.** Router looks up the port in its table and sends the packet to the correct private device

**Example**

| Private IP & Port | Public IP & Port |
|---|---|
| **192.168.1.10:1025 →** | 203.0.113.5:30001 |
| **192.168.1.11:1026 →** | 203.0.113.5:30002 |
| **192.168.1.12:1027 →** | 203.0.113.5:30003 |

Here, **one public IP** (203.0.113.5) is shared by **three devices** because PAT uses **different ports**.

**Key Points**

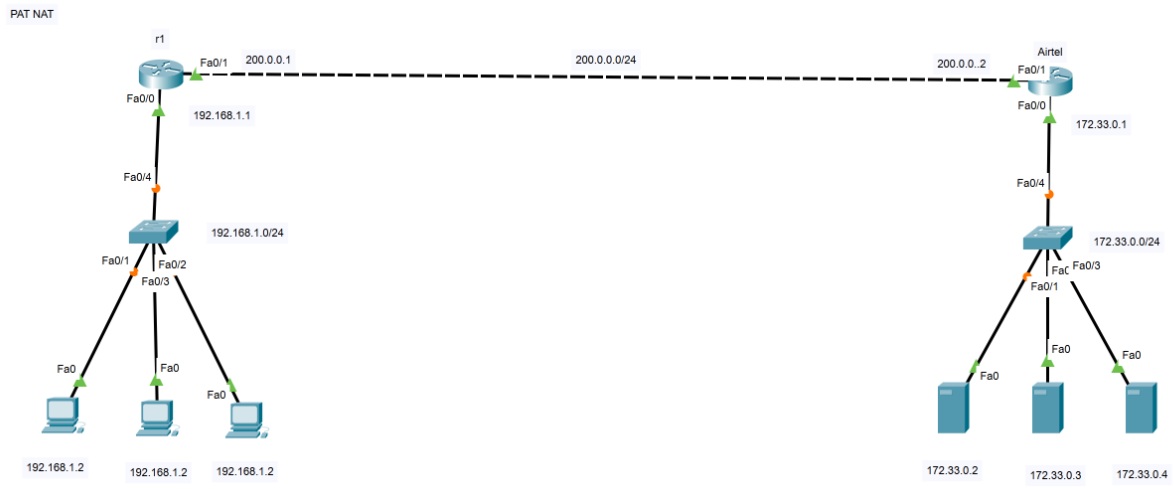
- **Type of NAT**: Many-to-One mapping.
- **Also called**: NAT Overload.
- **Port range**: 0–65535 (TCP/UDP).
- **Common in**: Home routers, corporate gateways.
- **Limit**: Number of simultaneous connections is limited by available ports.

PAT:

- It translates many private IPs to one public IP
- It is used for outbound communication
- It translates source IP

Note:

Private Ips are cannot access the Internet. Only public Ips are access the Internet. So NAT converts the private Ips to public Ips



```
! On r1
enable
configure terminal
hostname r1
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 200.0.0.1 255.255.255.0
no shutdown
exit
!
ip route 0.0.0.0 0.0.0.0 200.0.0.2
!
! PAT configuration
access-list 10 permit 192.168.1.0 0.0.0.255
! To identify the private network which needs to translate
ip nat inside source list 10 interface Fa0/1 overload
! to identify the public IP
!
interface Fa0/0
ip nat inside
exit
!
interface Fa0/1
ip nat outside
exit
!
```

```
! On Airtel
enable
configure terminal
hostname Airtel
!
interface Fa0/0
ip address 172.33.0.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 200.0.0.2 255.255.255.0
no shutdown
exit
!
! PAT configuration
access-list 10 permit 172.33.0.0 0.0.0.255
! To identify the private network which needs to translate
ip nat inside source list 10 interface Fa0/1 overload
! to identify the public IP
!
interface Fa0/0
ip nat inside
exit
!
interface Fa0/1
ip nat outside
exit
!
```

**Static NAT**

**Static NAT (Static Network Address Translation)** is a one-to-one mapping between a **private IP address** and a **public IP address** (or another IP in a different network).
It is "static" because the mapping **never changes** — every time that internal device sends or receives traffic, it always uses the same public IP.

---

**Key Points**

- **One-to-one mapping**: Each internal (private) IP has a dedicated external (public) IP.
- **Permanent mapping**: The mapping is configured manually and doesn't change automatically.
- **Used for**:
    - Hosting servers (web, mail, FTP) inside a private network that need to be accessible from the internet.
    - Devices that require a **consistent public IP** for security or licensing reasons.

- **No Port Overloading**: Unlike PAT, each mapping uses the full IP without port translation.
- Works both **incoming** and **outgoing**.

---

**Example**

Imagine a company with:

- **Private IP**: 192.168.1.10 (web server inside LAN)
- **Public IP**: 203.0.113.5

Static NAT mapping:

192.168.1.10 <-> 203.0.113.5

- Any internet user accessing 203.0.113.5 will reach the internal server 192.168.1.10.
- When the server sends a reply, it always appears to come from 203.0.113.5.

**Cisco Router Configuration Example**

```
Router(config)# interface fa0/0
Router(config-if)# ip address 203.0.113.1 255.255.255.0
Router(config-if)# ip nat outside

Router(config)# interface fa0/1
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# ip nat inside

Router(config)# ip nat inside source static 192.168.1.10 203.0.113.5
```

**This makes the internal server always reachable via 203.0.113.5.**

**Static NAT**

- One to one translation
- Is used for inbound communication (out to in)
- It translates destination IP in IP packet

- Mainly used when external users want to access company web servers.

**Advantages**

- Predictable — same public IP always used.
- Good for servers that must be reachable at a fixed address.

**Disadvantages**

- Wastes public IPs (needs one per device).
- Less flexible than dynamic NAT or PAT.

---

**Quick comparison table** for **Static NAT vs Dynamic NAT vs PAT**

| **Feature** | **Static NAT** | **Dynamic NAT** | **PAT (Port Address Translation)** |
|---|---|---|---|
| **Mapping** | One-to-one (fixed) | One-to-one (changes dynamically) | Many-to-one (multiple private IPs share one public IP using ports) |
| **Public IP Requirement** | One public IP per private IP | A pool of public IPs needed | Only **one** public IP needed for many devices |
| **IP Consistency** | Always same public IP | Changes each session | Changes each session (port numbers change) |
| **Common Use** | Servers that must be reachable (web, mail, FTP) | Temporary internet access for internal devices | Most common in home/office internet connections |
| **Advantages** | Predictable, reliable | Saves some public IPs compared to static | Best for saving public IPs, cost-effective |
| **Disadvantages** | Wastes public IPs | Still needs multiple public IPs | Harder to trace individual users from public side |

| | | | |
|---|---|---|---|
| **Example Mapping** | 192.168.1.10 ↔ 203.0.113.5 | 192.168.1.10 ↔ 203.0.113.5 (this time), next time might be 203.0.113.6 | 192.168.1.10, 192.168.1.11 → 203.0.113.5:1024, 203.0.113.5:1025 |

**Hub-and-Spoke Topology**

Hub-and-Spoke is a **network design** where one central device (**hub**) connects to multiple remote sites (**spokes**).
All communication between spokes must pass through the hub.

---

**How It Works**

- **Hub**: Central point of communication — often a router, switch, or VPN concentrator.
- **Spokes**: Remote locations or branches that connect only to the hub, not directly to each other.
- If **Spoke A** wants to talk to **Spoke B**, the traffic goes:

  Spoke A → Hub → Spoke B
- Often used in WANs (Wide Area Networks) and MPLS or VPN setups.

---

**Advantages**

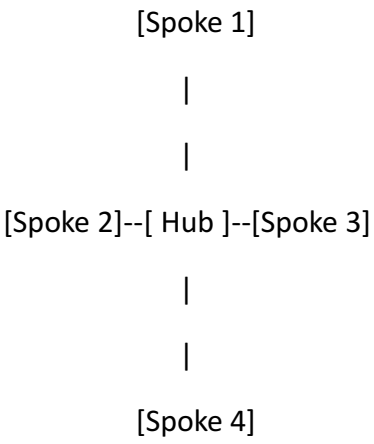
- **Centralized control** — easy to manage, configure, and secure.
- **Lower cost** — fewer total links needed compared to a full mesh.
- **Scalability** — adding a new spoke doesn't require connections to all other spokes.

---

**Disadvantages**

- **Single point of failure** — if the hub fails, the whole network is down.
- **Potential bottleneck** — all traffic flows through the hub.
- **Increased latency** — spoke-to-spoke traffic must pass through hub.

**Diagram:**



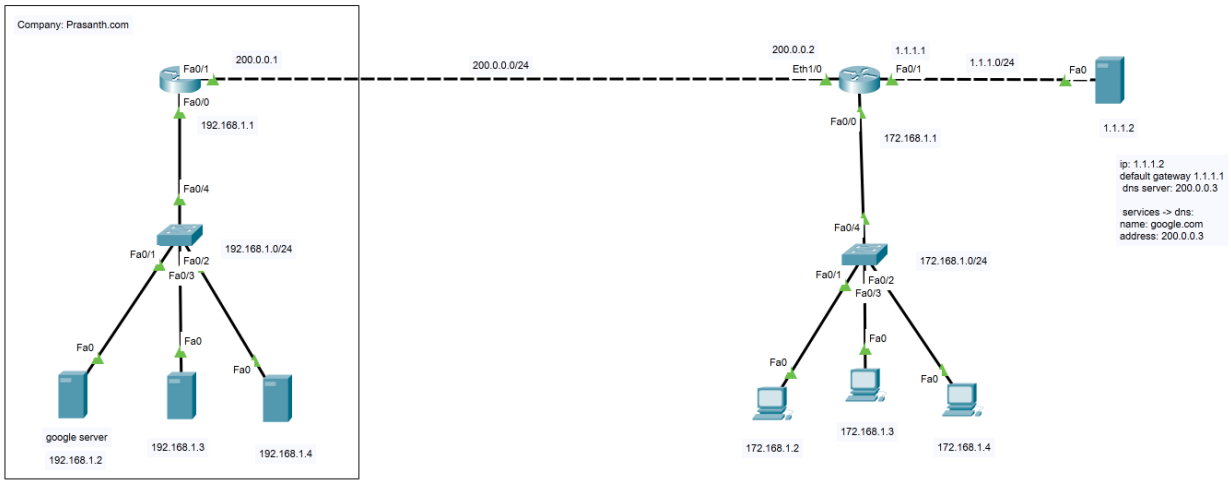

**Real-world examples**

- **VPN Hub-and-Spoke**: One data center (hub) connects to multiple branch offices (spokes) via secure tunnels.
- **MPLS networks**: Service provider routes all branch traffic through the central HQ.

- **Airline flight routes**: Big airports are hubs; smaller airports are spokes.

**Static NAT:**

- One to one translation
- It used for inbound communication (out to in)
- It translates destination IP in IP packet
- Mainly used when external users want to access company web servers




```
! Edge-Router

enable
configure terminal
hostname Edge-Router
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 200.0.0.1 255.255.255.0
no shutdown
exit
!
ip route 0.0.0.0 0.0.0.0 200.0.0.2
!
interface fa0/0
ip nat inside
exit
!
interface fa0/1
ip nat outside
exit
!
ip nat inside source static 192.168.1.2 200.0.0.3
ip nat inside source static 192.168.1.3 200.0.0.4
ip nat inside source static 192.168.1.4 200.0.0.5
```

```
! R2

enable
configure terminal
hostname Edge-Router
!
interface Fa0/1
ip address 172.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/0
ip address 200.0.0.2 255.255.255.0
no shutdown
exit
!
interface Eth1/0
ip address 1.1.1.1 255.255.255.0
no shutdown
exit
```

**Note:**

ISP block the private Ips

How to verify NAT (or) how to check the NAT table in router>?

#show ip nat translations

How to check the NAT configure?

#show ip nat statistics

---

**Dynamic NAT (Network Address Translation)**

Dynamic NAT automatically translates a group of **private IP addresses** into a pool of **public IP addresses**. Unlike Static NAT (1-to-1 mapping), Dynamic NAT assigns addresses **on demand** from the available pool.

**Key Points**

1. **Address Pool** – A predefined range of public IPs is configured.
2. **Mapping** – When an internal host initiates communication to the internet, the router/firewall picks the **next available public IP** from the pool.
3. **Temporary Binding** – The mapping exists only for the duration of the session. Once the connection ends, the public IP is released back into the pool.
4. **Limitation** – If the number of internal users exceeds the pool of public IPs, extra users cannot access external networks.

**Example**

- Private network: 192.168.1.0/24
- Public pool: 200.1.1.10 – 200.1.1.20

If three internal hosts connect:

- 192.168.1.2 → 200.1.1.10
- 192.168.1.3 → 200.1.1.11

- 192.168.1.4 → 200.1.1.12

  When 192.168.1.2 finishes, 200.1.1.10 becomes available again.

  **Cisco Configuration Example**

  Router(config)# ip nat pool MYPOOL 200.1.1.10 200.1.1.20 netmask 255.255.255.0

  Router(config)# access-list 1 permit 192.168.1.0 0.0.0.255

  Router(config)# ip nat inside source list 1 pool MYPOOL

  Router(config)# interface g0/0

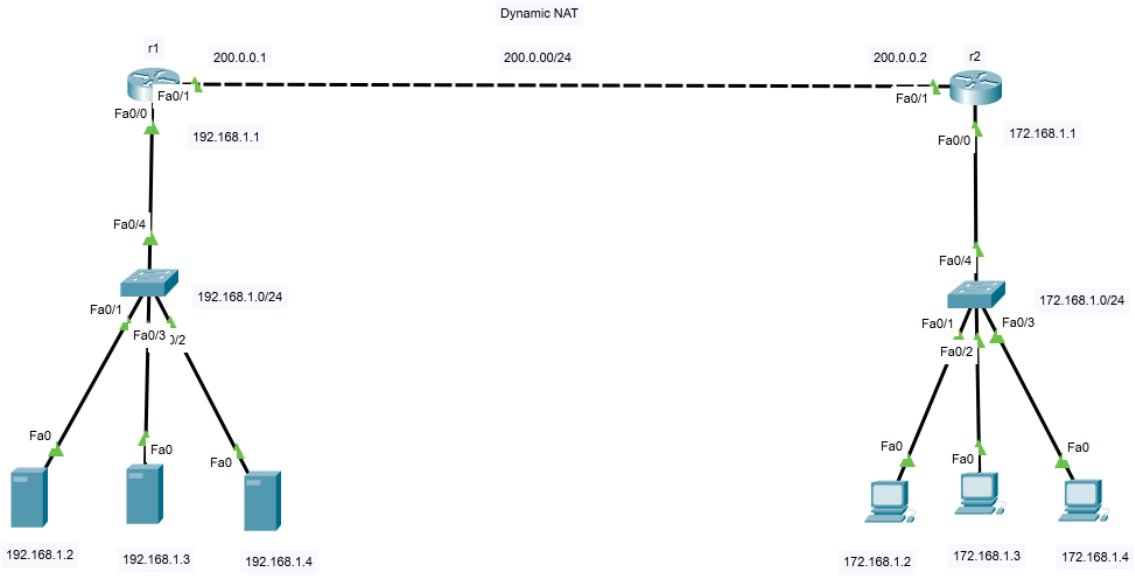
  Router(config-if)# ip nat inside

  Router(config)# interface g0/1

  Router(config-if)# ip nat outside

**Difference from PAT**

- **Dynamic NAT:** One private IP → one unique public IP (from pool).
- **PAT (Port Address Translation):** Many private IPs share **one public IP** using different port numbers.



```
! on R1
enable
configure terminal
hostname r1
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 200.0.0.1 255.255.255.0
no shutdown
exit
!
ip route 0.0.0.0 0.0.0.0 200.0.0.2
!
interface Fa0/0
ip nat inside
exit
!
interface Fa0/1
ip nat outside
exit
!
ip nat pool MYPOOL 200.0.0.10 200.0.0.13 netmask 255.255.255.0
access-list 1 permit 192.168.1.0 0.0.0.255
ip nat inside source list 1 pool MYPOOL
```

```
! On r2
enable
configure terminal
hostname r2
!
interface Fa0/0
ip address 172.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 200.0.0.2 255.255.255.0
no shutdown
exit
!
ip route 0.0.0.0 0.0.0.0 200.0.0.1
ip route 200.0.0.0 255.255.255.0 200.0.0.1
```

# FHRP (First Hop Redundancy Protocol)

FHRP is a group of protocols that provide **gateway redundancy** for end devices on a LAN. Instead of configuring hosts with a single default gateway (which becomes a single point of failure), FHRPs allow multiple routers to work together and present a **virtual default gateway IP**. If the active router fails, another router automatically takes over, ensuring uninterrupted connectivity.

## Main Types of FHRP

1. **HSRP (Hot Standby Router Protocol)** – Cisco proprietary.
    - One active router, one standby, others in listen state.
    - Uses a virtual IP and MAC address.
2. **VRRP (Virtual Router Redundancy Protocol)** – Open standard (RFC 5798).
    - Similar to HSRP but supports faster failover.
    - One master router, others act as backups.
3. **GLBP (Gateway Load Balancing Protocol)** – Cisco proprietary.
    - Provides redundancy plus load balancing across multiple routers.
    - Uses an AVG (Active Virtual Gateway) and multiple AVFs (Active Virtual Forwarders).

## Why FHRP is Important

- Eliminates default gateway single point of failure.
- Provides high availability in enterprise LANs.
- Improves resilience during router failures or maintenance.

## Example

- Hosts in VLAN 10 use 192.168.10.1 as the default gateway.
- Two routers share this virtual IP through FHRP.
- If Router1 (active) fails, Router2 (standby) takes over instantly.

Recommendation: Learn configuration for HSRP, VRRP, and GLBP.

# HSRP (Hot Standby Router Protocol)

HSRP is a **Cisco-proprietary FHRP** that provides **default gateway redundancy**. Multiple routers form an HSRP group and share a **virtual IP and MAC address**. End devices use this virtual IP as their default gateway.

If the **active router** fails, the **standby router** automatically takes over, ensuring uninterrupted connectivity.

**HSRP States**

1. **Initial** – HSRP not running.
2. **Learn** – Router has not learned the virtual IP yet.
3. **Listen** – Knows IP but not active/standby.
4. **Speak** – Sending/receiving hello messages.
5. **Standby** – Backup router, ready to take over.
6. **Active** – Currently forwarding traffic for the virtual IP.

**Key Parameters**

- **Hello time**: 3 sec (default).
- **Hold time**: 10 sec (default).
- **Preemption**: Optional, allows a higher-priority router to take over active role.
- **Priority**: 0–255 (default 100, higher = more preferred).

**Basic Configuration (Example)**

Two routers sharing gateway 192.168.1.1 for VLAN 10:

**Router1 (Active)**

```
interface g0/0
 ip address 192.168.1.2 255.255.255.0
 standby 1 ip 192.168.1.1
 standby 1 priority 110
```

standby 1 preempt

**Router2 (Standby)**

interface g0/0

ip address 192.168.1.3 255.255.255.0

standby 1 ip 192.168.1.1

standby 1 priority 100

standby 1 preempt

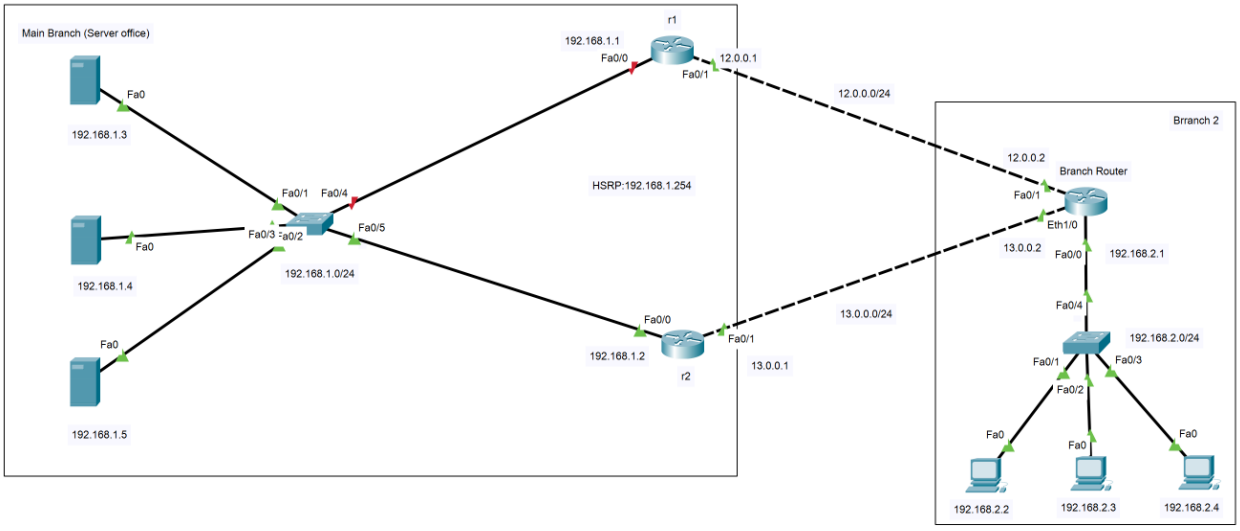
**Verification Commands**

- show standby → Shows HSRP status, active/standby routers.
- debug standby → Real-time HSRP messages.

**Advantages**

- Provides gateway redundancy.
- Automatic failover.
- Transparent to end devices (they only see virtual IP).

Recommendation: Memorize HSRP states and timers for exams.



```
! On r1
enable
configure terminal
hostname r1
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.1 255.255.255.0
no shutdown
exit
!
interface Fa0/0
standby 1 priority 150
standby 1 ip 192.168.1.254
standby 1 preempt
exit
!
route eigrp 100
network 12.0.0.0
network 192.168.1.0
exit
!
```

```
! On r2
enable
configure terminal
hostname r2
!
interface Fa0/0
ip address 192.168.1.2 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 13.0.0.1 255.255.255.0
no shutdown
exit
!
interface Fa0/0
standby 1 priority 140
standby 1 ip 192.168.1.254
standby 1 preempt
exit
!
route eigrp 100
network 13.0.0.0
network 192.168.1.0
exit
!
```

```
! On Branch Router
enable
configure terminal
hostname Branch Router
!
interface Fa0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.2 255.255.255.0
no shutdown
exit
!
interface Eth1/0
ip address 13.0.0.2 255.255.255.0
no shutdown
exit
!
route eigrp 100
network 192.168.2.0
network 12.0.0.0
network 13.0.0.0
exit
!
```

**VRRP (Virtual Router Redundancy Protocol)**

VRRP is an **open standard FHRP (RFC 5798)** that provides **default gateway redundancy**. Like HSRP, it allows multiple routers to share a **virtual IP and MAC address**, ensuring that if the master router fails, another router takes over without affecting end devices.

**Key Concepts**

1. **Virtual IP** – The IP address shared by the VRRP group (used as the hosts' default gateway).
2. **Master Router** – The router actively forwarding traffic for the virtual IP.
3. **Backup Routers** – Monitor the master; if it fails, one of them takes over.
4. **Virtual MAC Address** – Automatically assigned (00-00-5E-00-01-XX, where XX = VRID).

**VRRP Characteristics**

- **Standardized** → Works across vendors (unlike Cisco-only HSRP/GLBP).
- **Preemption** → Enabled by default (higher priority router takes over automatically).
- **Priority values**: 1–254 (default 100).
- **Virtual Router ID (VRID)**: 1–255 (identifies the group).
- **Timers**: Advertisement interval = 1s (default).

**Basic Configuration Example**

Two routers share the gateway 192.168.1.254:

**Router1 (Master preferred)**

```
interface g0/0
 ip address 192.168.1.1 255.255.255.0
 vrrp 10 ip 192.168.1.254
 vrrp 10 priority 120
```

**Router2 (Backup)**

```
interface g0/0
 ip address 192.168.1.2 255.255.255.0
```

vrrp 10 ip 192.168.1.254

vrrp 10 priority 100

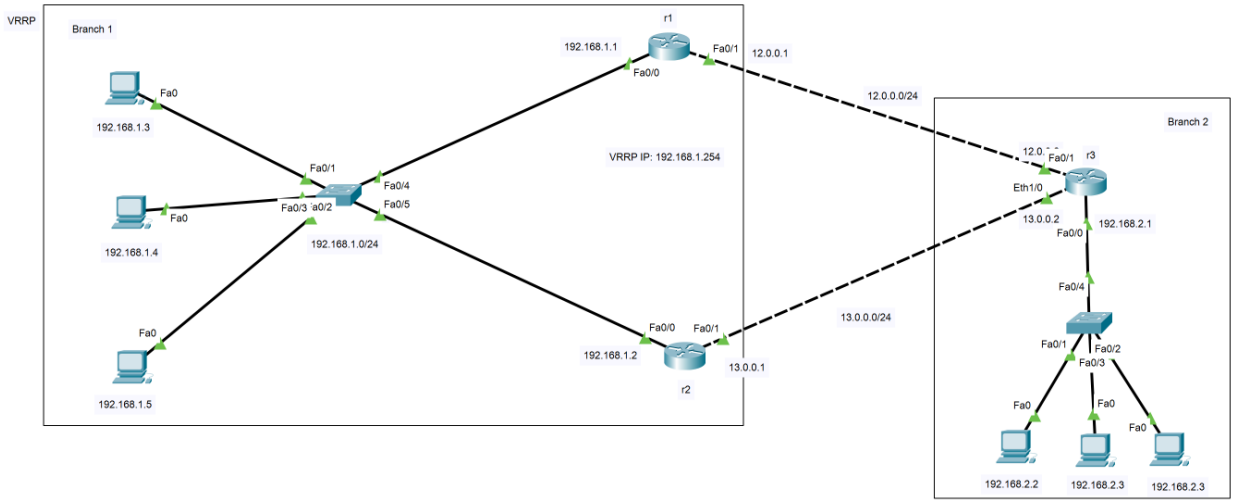
**Verification Commands**

- show vrrp → Displays VRRP status (master/backup, virtual IP, timers).
- debug vrrp → Real-time protocol events.

**Comparison with HSRP**

- **HSRP** → Cisco proprietary, preemption optional.
- **VRRP** → Open standard, preemption enabled by default.
- **HSRP virtual MAC**: 0000.0C07.ACxx
- **VRRP virtual MAC**: 0000.5E00.01xx

Recommendation: Use VRRP in multi-vendor networks for interoperability.




```
! on r1
enable
configure terminal
hostname r1
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.1 255.255.255.0
no shutdown
exit
!
interface Fa0/0
vrrp 1 ip 192.168.1.254
vrrp 1 priority 150
exit
!
router eigrp 100
network 192.168.1.0
network 12.0.0.0
exit
!
```

```
! on r2
enable
configure terminal
hostname r2
!
interface Fa0/0
ip address 192.168.1.2 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 13.0.0.1 255.255.255.0
no shutdown
exit
!
interface Fa0/0
vrrp 1 ip 192.168.1.254
vrrp 1 priority 140
exit
!
router eigrp 100
network 192.168.1.0
network 13.0.0.0
exit
!
```

```
! On r3
enable
configure terminal
hostname r3
!
interface Fa0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.2 255.255.255.0
no shutdown
exit
!
interface Eth1/0
ip address 13.0.0.2 255.255.255.0
no shutdown
exit
!
router eigrp 100
network 192.168.2.0
network 12.0.0.0
network 13.0.0.0
```

**Details**

- **Packet Tracer IOS**: Cisco Packet Tracer does **not** implement VRRP or GLBP. It only supports **HSRP**.
- **Cisco IOS (real hardware or Dynamips in GNS3/EVE-NG)**: VRRP is available if the IOS feature set supports it (e.g., IP Services, Advanced IP Services).

That’s why your router eigrp 100 worked, but vrrp was rejected.

## GLBP (Gateway Load Balancing Protocol)

GLBP is a **Cisco-proprietary FHRP** that provides both **gateway redundancy** and **load balancing** across multiple routers in the same group. Unlike HSRP and VRRP (which have only one active gateway), GLBP allows **all routers to actively forward traffic** while still maintaining redundancy.

### Key Concepts

1. **AVG (Active Virtual Gateway)**
    - Elected from the GLBP group.
    - Assigns virtual MAC addresses to each participating router.
2. **AVF (Active Virtual Forwarder)**
    - Each router that forwards traffic for one of the virtual MAC addresses.
    - Ensures multiple routers can share the load.
3. **Virtual IP**
    - Common gateway IP used by all hosts in the subnet.
4. **Load Balancing Methods**
    - **Round-robin** (default) → Each host gets a different virtual MAC in rotation.
    - **Weighted** → Traffic distributed based on router weight values.
    - **Host-dependent** → A host always uses the same router once assigned.

### GLBP Characteristics

- Proprietary (Cisco-only).
- Supports up to **4 AVFs per group** (others in listen state).
- Virtual MAC format: 0007.b400.xxyy.
- Preemption optional.

### Basic Configuration Example

Two routers in the same LAN (192.168.1.0/24) sharing gateway 192.168.1.254:

**Router1 (will likely become AVG)**

```
interface g0/0
```

ip address 192.168.1.1 255.255.255.0

glbp 1 ip 192.168.1.254

glbp 1 priority 120

glbp 1 preempt

**Router2**

interface g0/0

ip address 192.168.1.2 255.255.255.0

glbp 1 ip 192.168.1.254

glbp 1 priority 110

glbp 1 preempt

**Verification Commands**

- show glbp → Displays AVG, AVFs, and load-balancing method.
- debug glbp → Shows protocol activity.

**Comparison with HSRP/VRRP**

- **HSRP/VRRP**: Only one router actively forwards traffic (others wait).
- **GLBP**: All routers can forward traffic (better bandwidth use).

Recommendation: Use GLBP when both redundancy and load balancing are needed in a Cisco-only environment.

You want to configure **SSH on a Cisco Router** so you can access it securely instead of using Telnet

### 🔷 Step 1: Set Router Hostname & Domain Name

SSH requires a hostname and domain name for RSA key generation.

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
R1(config)# ip domain-name mylab.local
```

### 🔷 Step 2: Generate RSA Keys

RSA keys are required for SSH encryption.

```
R1(config)# crypto key generate rsa
The name for the keys will be: R1.mylab.local
How many bits in the modulus [512]: 1024
```

Use **1024** or **2048 bits** (more secure than 512).

### 🔷 Step 3: Create Local User for SSH Login

```
R1(config)# username admin privilege 15 secret cisco123
```

### 🔷 Step 4: Enable SSH & Set VTY Line Password

```
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# exit
```

### 🔷 Step 5: Enable SSH Version 2 (more secure)

```
R1(config)# ip ssh version 2
```

### 🔷 Step 6: Save the Configuration

```
R1# write memory
```

Or

R1# copy running-config startup-config

### 🔷 Step 7: Test SSH from a PC/Host

From your PC/Client (Command Prompt / Terminal):

ssh admin@192.168.1.1

(Replace 192.168.1.1 with router's IP address)

✅ Now your router only allows **SSH** on VTY lines (no Telnet).

## TELNET


In Cisco devices, line vty 0 4 configures virtual terminal (VTY) lines for remote access, specifically allowing up to five simultaneous connections (lines 0,1,2,3, and 4) via protocols like Telnet or SSH.

This means five different users or sessions can remotely access the devices's command-line interface (CLI) at the same time.

```
! On R1

enable
configure terminal
hostname R1
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 200.0.0.1 255.255.255.0
no shutdown
exit
!
description "Giving Router username, password and login permissions"
username aaa privilege 15 password 123
line vty 0 4
login local
!
description "Login to Router 2 and make interface fa0/1 to up state"
telnet 200.0.0.2
description "Enter username and password"
bbb
321
! on R2
enable
configure terminal
!
interface fa0/1
no shutdown
exit
exit
exit
! on R1
```

```
! On R2

enable
configure terminal
hostname R2
!
interface Fa0/0
ip address 200.0.0.2 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 192.168.2.1 255.255.255.0
description "For testing, leaving this interface in shutdown state"
exit
!
description "Giving Router username, password and login permissions"
username bbb privilege 15 password 321
line vty 0 4
login local
```

**WAN (Wide Area Network) technologies** are methods and protocols that connect devices across **large geographic areas** (cities, countries, or globally). These technologies allow enterprises, ISPs, and data centers to interconnect their networks.

### 🔷 1. Traditional WAN Technologies

Older but still found in many networks:

- **PSTN (Public Switched Telephone Network)** – Legacy analog telephone system, used for dial-up.
- **ISDN (Integrated Services Digital Network)** – Digital version of telephone network for voice and data.
- **Leased Lines** – Dedicated point-to-point private circuits (e.g., T1, E1).
- **Frame Relay** – Packet-switched WAN tech, faster than leased lines.
- **ATM (Asynchronous Transfer Mode)** – Cell-switching tech using 53-byte cells.
- **X.25** – Early packet-switched network, very slow, replaced by Frame Relay.

### 🔷 2. Modern WAN Technologies

- **MPLS (Multiprotocol Label Switching)**
    - Uses labels instead of IP routing to forward packets.
    - Provides QoS and supports VPNs.
    - Popular in enterprise WANs.
- **Metro Ethernet**
    - Extends Ethernet over a wide area (used by ISPs and carriers).
    - Cost-effective and supports high bandwidth.
- **DWDM (Dense Wavelength Division Multiplexing)**
    - Optical fiber technology, multiplexes multiple signals.
    - Used in backbone ISPs.

### 🔷 3. Broadband & Consumer WAN

- **DSL (Digital Subscriber Line)** – Uses telephone lines for internet.
- **Cable Broadband** – Uses coaxial TV cables.

- **Fiber-to-the-Home (FTTH)** – High-speed optical internet.
- **Satellite Internet** – Long distance, higher latency.

🔷 **4. Wireless & Cellular WAN**

- **3G / 4G / 5G** – Mobile data WANs.
- **Microwave Links** – Point-to-point wireless WAN.
- **VSAT (Very Small Aperture Terminal)** – Satellite-based WAN for remote areas.

🔷 **5. VPN & Secure WAN**

- **Site-to-Site VPN (IPsec VPN)** – Connects branch offices securely.
- **Remote Access VPN** – For users connecting securely over public internet.
- **SSL VPN** – VPN over HTTPS.

🔷 **6. Software-Defined WAN (SD-WAN)**

- New-generation WAN technology.
- Uses software to intelligently route traffic over multiple links (MPLS, Internet, LTE).
- Improves **performance, cost efficiency, and security**.
- Popular in cloud-first enterprises.

✅ **In short:**

- **Legacy WANs** → Leased lines, ISDN, Frame Relay, ATM.
- **Modern WANs** → MPLS, Metro Ethernet, Broadband, 4G/5G, SD-WAN.

# VPN (Virtual Private Network)

A **VPN (Virtual Private Network)** creates a **secure, encrypted connection (tunnel)** between two endpoints over an **untrusted network (like the Internet)**.
It hides your traffic from attackers, ISPs, and allows **remote/private communication**.

## 🔷 Types of VPN

### 1. Site-to-Site VPN

- Connects **entire branch offices** to HQ.
- Usually uses **IPsec** for encryption.
- Example: A company with offices in New York and London links them securely over the internet.
- **Used in enterprises.**

### 2. Remote Access VPN

- For **individual users** working remotely.
- User → VPN Client → VPN Gateway (Company Router/Firewall).
- Example: Employees connecting from home.

### 3. SSL VPN

- Uses **SSL/TLS (HTTPS)** instead of IPsec.
- Works via a web browser (no special software needed).
- Good for secure remote access to applications.

### 4. MPLS VPN

- Provided by ISPs using MPLS backbone.
- Not encrypted by default (depends on ISP security).
- Used for corporate WANs.

### 5. GRE + IPsec VPN

- **GRE (Generic Routing Encapsulation)** allows routing protocols across VPN.
- GRE provides encapsulation, IPsec provides encryption.

## 🔷 VPN Protocols

- **IPsec** → Encrypts at network layer (very secure, common in Site-to-Site).
- **SSL/TLS** → Encrypts at transport layer (common in Remote Access).
- **L2TP (Layer 2 Tunneling Protocol)** → Often combined with IPsec.
- **PPTP (Point-to-Point Tunneling Protocol)** → Old, insecure, not recommended.
- **IKEv2 (Internet Key Exchange v2)** → Secure, fast, supports mobility.
- **OpenVPN** → Open-source, uses SSL/TLS.

## 🔷 How VPN Works (Step by Step)

1. **Encapsulation** – Data is wrapped inside a VPN packet.
2. **Encryption** – Data is encrypted with keys (AES, DES, 3DES).
3. **Tunneling** – Packet travels securely over public internet.
4. **Decryption** – Receiver removes encryption & gets original data.

## 🔷 Benefits of VPN

- ✅ Security (encryption protects data from sniffing)
- ✅ Remote Access (work from anywhere)
- ✅ Privacy (hides your real IP)
- ✅ Bypasses restrictions (geo-blocking, censorship)
- ✅ Cost-effective (no need for dedicated leased lines)

## 🔷 Real-World Example

- An employee connects from home → VPN client → Encrypted tunnel → Company firewall/router → Accesses servers just like in office.
- A bank uses **Site-to-Site VPN** between its branches for secure transaction data.

## 🔷 What is a Tunnel VPN?

A **VPN Tunnel** is the **virtual encrypted path** created between two endpoints across an untrusted network (like the Internet).
All traffic between them is **encapsulated and encrypted** inside this tunnel, making it private.

Think of it like:
📦 Data → 🚚 Encapsulation → 🔒 Encryption → 🌐 Internet → 🔓 Decryption → 📦 Data delivered.

## 🔷 Tunneling Protocols

Several protocols can be used to build VPN tunnels:

1. **IPsec (Internet Protocol Security)**
    - Most common for Site-to-Site VPNs.
    - Works at **Layer 3 (Network Layer)**.
    - Provides **encryption, integrity, and authentication**.
    - Modes:
        - **Tunnel Mode** – Encrypts the whole IP packet (common for Site-to-Site).
        - **Transport Mode** – Encrypts only the payload, keeps IP header (common for host-to-host).
2. **GRE (Generic Routing Encapsulation)**
    - Encapsulates many types of traffic (multicast, routing protocols).
    - But GRE alone has **no encryption** → Usually combined with IPsec (GRE over IPsec).
3. **SSL/TLS VPN**
    - Works at **Layer 4/7** (Transport/Application).
    - Common for Remote Access VPNs.
    - Uses HTTPS port (443), good for firewall traversal.
4. **L2TP (Layer 2 Tunneling Protocol)**
    - Often used with IPsec (L2TP/IPsec).

- Provides tunneling, IPsec provides encryption.

5. **PPTP (Point-to-Point Tunneling Protocol)**
    - Old, insecure, not recommended anymore.

🔷 **Types of Tunnel VPNs**

- **Site-to-Site Tunnel VPN**
    - Used between branch offices or HQ ↔ branch.
    - Example: Two Cisco routers creating an IPsec tunnel.
- **Remote Access Tunnel VPN**
    - A remote user creates an **encrypted tunnel** from their laptop/PC to the company VPN gateway.

🔷 Example Topology

Branch Router (LAN 10.1.1.0/24) ←→ Internet ←→ HQ Router (LAN 10.2.2.0/24)

_____ Encrypted IPsec Tunnel _____/

🔷 **Verification (Cisco Commands)**

On Cisco routers/firewalls you can check tunnel status:

```
show crypto isakmp sa      ! Shows ISAKMP Phase 1 status
show crypto ipsec sa       ! Shows IPsec Phase 2 status
```

✅ **In short:**

- Tunnel VPN = secure encrypted “tube” over Internet.
- Protocols: **IPsec, GRE, SSL, L2TP, PPTP**.
- Types: **Site-to-Site** and **Remote Access**.

# GRE (Generic Routing Encapsulation)

## 🔷 What is GRE?

- GRE is a **tunneling protocol** developed by Cisco.
- It encapsulates many types of network layer protocols (IPv4, IPv6, multicast, routing protocols) inside IP tunnels.
- This allows communication between networks over an IP-only infrastructure.

## 🔷 Why GRE is used?

- **Supports multiple protocols** (unlike plain IP tunnels, GRE can carry IPX, IPv6, multicast, etc.).
- **Routing protocols** (like OSPF, EIGRP, BGP) can work across tunnels.
- Used in **site-to-site VPNs** (often combined with IPSec for encryption since GRE itself has no encryption).
- Useful for **lab testing** or **linking disjoint networks**.

## 🔷 GRE Tunnel Structure

A GRE tunnel encapsulates the original packet:

[New IP Header] → [GRE Header] → [Original Packet]

- **Outer IP header**: Source = tunnel source IP, Destination = tunnel destination IP
- **GRE header**: Identifies GRE
- **Payload**: Original packet (IPv4, IPv6, multicast, etc.)

## 🔷 GRE Tunnel Configuration (Cisco Example)

Let's say we want to connect **R1 (10.1.1.1)** to **R2 (20.1.1.1)**:

**On R1:**

```
interface Tunnel0
 ip address 192.168.100.1 255.255.255.0
```

tunnel source 10.1.1.1

tunnel destination 20.1.1.1

On R2:

interface Tunnel0

ip address 192.168.100.2 255.255.255.0

tunnel source 20.1.1.1

tunnel destination 10.1.1.1

Now R1 and R2 can talk over **192.168.100.x** network as if they were directly connected.

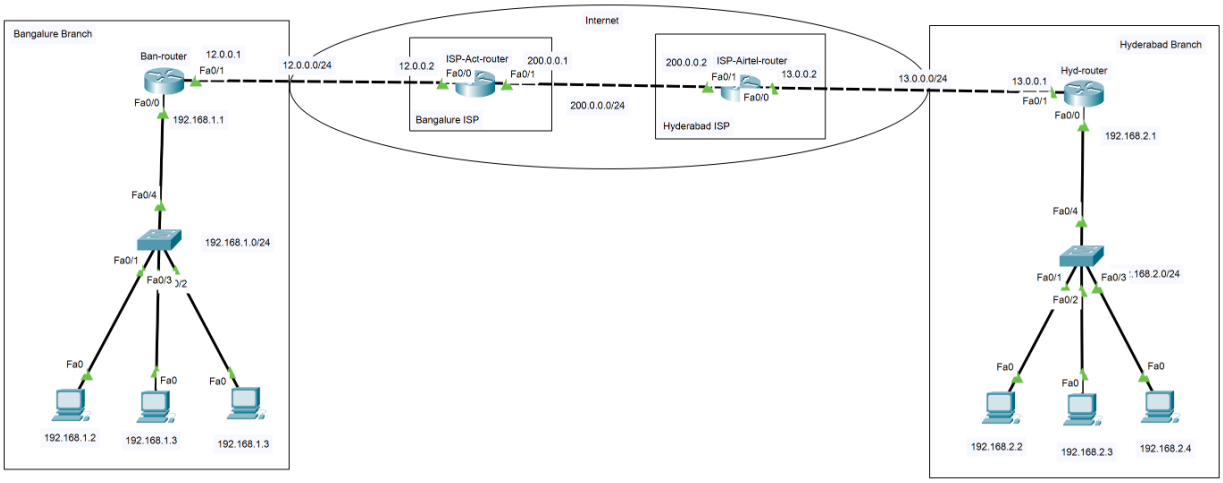
## 🔷 GRE Pros and Cons

### ✅ Advantages:

- Multiprotocol support (IPv4, IPv6, multicast).
- Works with dynamic routing protocols.
- Simple to configure.

### ❌ Disadvantages:

- **No encryption/authentication** (use GRE + IPSec for secure VPN).
- **Adds overhead** (extra headers reduce MTU).
- **Not NAT-friendly** (need adjustments for NAT traversal).



```
! On Ban-router
enable
configure terminal
hostname Ban-router
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.1 255.255.255.0
no shutdown
exit
!
ip route 0.0.0.0 0.0.0.0 12.0.0.2
!
interface tunnel 0
ip address 10.0.0.1 255.255.255.0
tunnel source Fa0/1
tunnel destination 13.0.0.1
exit
!
router eigrp 100
network 10.0.0.0
network 192.168.1.0
exit
!
```

```
! On ISP-Act-router
enable
configure terminal
hostname ISP-Act-router
!
interface Fa0/0
ip address 12.0.0.2 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 200.0.0.1 255.255.255.0
no shutdown
exit
!
ip route 13.0.0.0 255.255.255.0 200.0.0.2
!
```

```
! On ISP-Airtel router
enable
configure terminal
hostname ISP-Airtel-router
!
interface Fa0/1
ip address 200.0.0.2 255.255.255.0
no shutdown
exit
!
interface Fa0/0
ip address 13.0.0.2 255.255.255.0
no shutdown
exit
!
ip route 12.0.0.0 255.255.255.0 200.0.0.1
!
```

```
! On Hyd-router
enable
configure terminal
hostname Hyd-router
!
interface Fa0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 13.0.0.1 255.255.255.0
no shutdown
exit
!
ip route 0.0.0.0 0.0.0.0 13.0.0.02
!
interface tunnel 0
ip address 10.0.0.2 255.255.255.0
tunnel source Fa0/1
tunnel destination 12.0.0.1
exit
!
router eigrp 100
network 10.0.0.0
network 192.168.2.0
exit
!
```

# Virtual Local Area Network

## 🔷 What is a VLAN?

- A **VLAN** is a **logical segmentation** of a physical network.
- It allows devices in the same VLAN to communicate **as if they were in the same LAN**, even if they are on different switches.
- Devices in **different VLANs cannot communicate** unless routed (via a router or Layer 3 switch).

## 🔷 Why use VLANs?

1. **Segmentation** – divide networks (e.g., HR, Finance, Students, Servers).
2. **Security** – isolate sensitive traffic.
3. **Performance** – reduce broadcast domains.
4. **Flexibility** – devices grouped logically, not physically.

## 🔷 VLAN Types

1. **Default VLAN** – VLAN 1 (exists by default).
2. **Data VLAN** – used for user-generated traffic.
3. **Voice VLAN** – dedicated for VoIP traffic.
4. **Management VLAN** – for switch management (Telnet/SSH, SNMP).
5. **Native VLAN** – VLAN assigned to untagged frames on trunk links (default = VLAN 1).

## 🔷 VLAN Configuration (Cisco Switch Example)

### 1️⃣ Create VLANs

```
Switch> enable
Switch# configure terminal
Switch(config)# vlan 10
Switch(config-vlan)# name HR
Switch(config-vlan)# exit
Switch(config)# vlan 20
Switch(config-vlan)# name Finance
```

### 2️⃣ Assign VLAN to Interfaces

```
Switch(config)# interface fastEthernet0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10

Switch(config)# interface fastEthernet0/2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 20
```

### 3️⃣ Configure Trunk (between switches)

```
Switch(config)# interface fastEthernet0/24
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan 10,20
```

## 🔷 VLAN Communication

- Devices in the **same VLAN** can talk directly.
- Devices in **different VLANs** need **Router-on-a-Stick** or a **Layer 3 Switch** for **Inter-VLAN Routing**.

## 🔷 Verification Commands

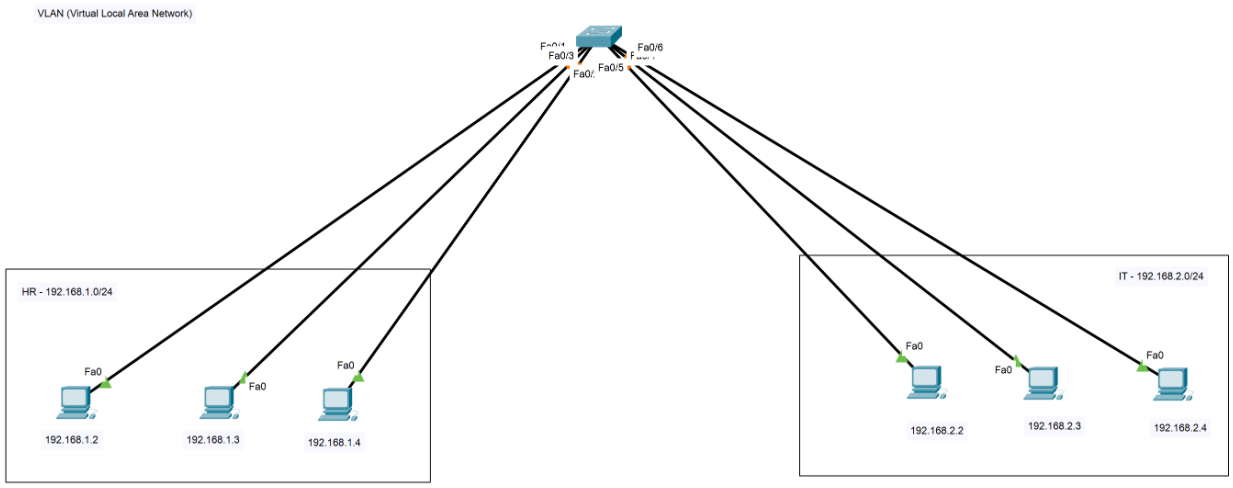
Switch# show vlan brief ← shows VLANs and assigned ports

Switch# show interfaces trunk ← shows trunk status

⚡ Example:

- PC1 (Fa0/1 → VLAN 10) → 192.168.10.10
- PC2 (Fa0/2 → VLAN 20) → 192.168.20.10

  👉 They **cannot ping** each other unless a router or L3 switch does inter-VLAN routing.




```
! On swithc
enable
configure terminal
hostname switch1
!
vlan 2
name HR
exit
!
vlan 3
name IT
exit
!
interface range FastEthernet0/1-3
switchport mode access
switchport access vlan 2
exit
!
interface range FastEthernet0/4-6
switchport mode access
switchport access vlan 3
exit
!
```

Note:

#show mac-address-table

Switch ports are 2 types

1. Access port
2. Trunk port

- Access ports are used to connect end devices laptop, desktop, servers, printer, etc
- Access ports can carry traffic for single vlans

- Trunk ports are used to extend vlans
- Trunk ports can carry traffic for multiple vlans
- Trunks port is used to between switch to swich, switch to router and switch to server

# Trunk Port

## 🔷 What is a Trunk Port?

- A **trunk port** is a switch port that **carries traffic for multiple VLANs** across a single physical link.
- Unlike an **access port** (which belongs to only one VLAN), a trunk port can transport traffic from **many VLANs at the same time** using VLAN tags.

---

## 🔷 Why do we need Trunk Ports?

- To allow devices in the same VLAN but connected to different switches to communicate.
- To reduce the number of physical connections required between switches.
- To carry VLAN information to routers, firewalls, servers, and other switches.

---

## 🔷 How Trunk Ports Work

- **Tagging:** Frames are tagged with a VLAN ID using **IEEE 802.1Q encapsulation**.
    - The **VLAN tag** (4-byte header) is added to Ethernet frames.
    - One VLAN (the **native VLAN**) is usually left **untagged**.
- Both switches must agree on trunking protocol and VLAN configuration.

---

## 🔷 Key Trunk Port Concepts

1. **Trunking Protocols**
    - **802.1Q** → Industry standard, used in Cisco & non-Cisco.
    - **ISL (Inter-Switch Link)** → Cisco proprietary (legacy, rarely used now).
2. **Native VLAN**
    - The VLAN that passes traffic **without tags** on a trunk.
    - Default is VLAN 1 (but usually changed for security).
3. **Allowed VLANs**

- You can restrict which VLANs are allowed to traverse the trunk.
- Example: Only VLANs 10, 20, 30 can be allowed on a trunk.

---

🔷 **Cisco Switch Trunk Port Configuration**

```
Switch(config)# interface fa0/1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk encapsulation dot1q   # (if required on older models)
Switch(config-if)# switchport trunk allowed vlan 10,20,30
Switch(config-if)# switchport trunk native vlan 99
```

🔷 **Where Trunk Ports Are Used**

- Between **switch-to-switch** connections.
- Between **switch and router** (Router-on-a-stick setup).
- Between **switch and firewall** (when handling multiple VLANs).
- Between **switch and hypervisor host** (VMware/Hyper-V using VLAN tagging).

Trunk protocols (802.1Q and ISL):

🔷 **1. IEEE 802.1Q (dot1q)**

- **Standard:** Open standard (IEEE), supported by Cisco and non-Cisco devices.
- **Encapsulation:** Inserts a **4-byte tag** inside the Ethernet frame header after the source MAC address.
- **VLAN ID:** Uses **12 bits** → supports up to **4096 VLANs (1–4094 usable)**.
- **Native VLAN:** Frames belonging to the native VLAN are **not tagged**.
- **Widely Used:** This is the **current industry standard**.

📌 Example frame structure (with tag):

[Dest MAC | Src MAC | 802.1Q Tag (4 bytes: TPID, Priority, VLAN ID) | EtherType | Payload | FCS]

🔷 **2. ISL (Inter-Switch Link)**

- **Standard:** Cisco **proprietary protocol** (not supported by other vendors).
- **Encapsulation:** Encapsulates the **entire Ethernet frame** with a **26-byte ISL header + 4-byte CRC trailer**.
- **VLAN ID:** Supports up to **1000 VLANs**.
- **Native VLAN:** Does **not** have a native VLAN (all traffic is tagged).
- **Legacy:** Rarely used today, replaced by 802.1Q.

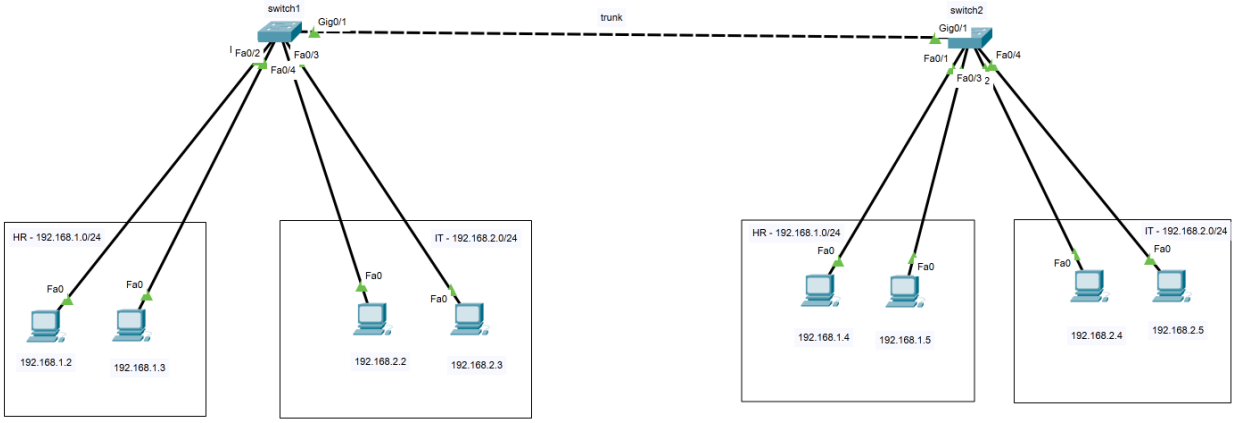
📌 Example frame structure:

[ISL Header (26 bytes) | Original Ethernet Frame | ISL CRC (4 bytes)]

🔷 Key Differences Between 802.1Q and ISL

| **Feature** | **802.1Q (dot1q)** | **ISL (Cisco proprietary)** |
|---|---|---|
| **Standard** | IEEE (open standard) | Cisco proprietary (legacy) |
| **Encapsulation** | 4-byte tag inside frame header | Entire frame encapsulated |
| **VLAN IDs supported** | 4096 (12-bit field) | ~1000 |
| **Native VLAN** | Supported (untagged) | Not supported (all tagged) |
| **Overhead** | 4 bytes | 30 bytes (26 + 4) |
| **Usage today** | Widely used, industry standard | Rarely used, obsolete |

✅ In modern networks, **802.1Q** is the only trunking protocol you'll see.
Cisco officially deprecated **ISL** years ago.


switch1
Gig0/1
trunk
switch2
Gig0/1
Fa0/2
Fa0/3
Fa0/4
Fa0/1
Fa0/3
Fa0/4
HR - 192.168.1.0/24
IT - 192.168.2.0/24
HR - 192.168.1.0/24
IT - 192.168.2.0/24
Fa0
192.168.1.2
192.168.1.3
192.168.2.2
192.168.2.3
192.168.1.4
192.168.1.5
192.168.2.4
192.168.2.5

```
! On switch1
enable
configure terminal
hostname switch1
!
! Configuring vlans
vlan 2
name HR
exit
!
vlan 3
name IT
exit
!
! Configuring vlan interfaces
interface range FastEthernet0/1-2
switchport mode access
switchport access vlan 2
exit
!
interface range FastEthernet0/3-4
switchport mode access
switchport access vlan 3
exit
!
! configuring trunk to gigbitethernet
interface GigabitEthernet0/1
switchport mode trunk
! allow vlan 2 only to trunk
switchport trunk allowed vlan 2
exit
!
```

Note:
By default trunk allow all vlans

what is difference between tagged and untagged traffic?

- Tagged traffic means whenever a switch forwards frame on trunk interface it will add Dot1q header
- Untagged traffic means whenever a switch forwards native VLAN traffic on trunk interface it will not add Dot1q header.

🔷 **Tagged Traffic**

- **Definition:** Ethernet frames that **carry VLAN information** inside them.
- **How:** A **4-byte VLAN tag (802.1Q header)** is inserted into the frame.
- **Use:** Sent **over trunk ports** to distinguish which VLAN the frame belongs to.
- **Example:** If a switch sends a frame from VLAN 10 to another switch over a trunk, it will **tag the frame with VLAN ID 10**.

📌 Tagged = "This frame belongs to VLAN X."

🔷 **Untagged Traffic**

- **Definition:** Ethernet frames that **do not contain VLAN information**.
- **Use:**
    - Sent **over access ports** (end devices don't understand VLAN tags).
    - Or, on trunk ports as part of the **native VLAN** (by default VLAN 1, but usually changed).
- **Example:** A PC connected to an access port in VLAN 20 sends traffic → the switch forwards it **without a VLAN tag**. The receiving switchport already knows it belongs to VLAN 20.

📌 Untagged = "Belongs to the port's access VLAN (or native VLAN on trunks)."

🔷 **Key Differences**

| Feature | Tagged Traffic | Untagged Traffic |
|---|---|---|
| **VLAN Info** | Has VLAN ID inside frame (802.1Q tag) | No VLAN ID in frame |
| **Typically Seen On** | **Trunk ports** | **Access ports** (and native VLAN on trunks) |
| **Device Understanding** | Switches/routers can interpret VLAN ID | End devices (PCs, printers, etc.) expect this |
| **Example** | Frame marked as "VLAN 10" | Plain Ethernet frame (default VLAN) |

✅ **In short:**

- **Tagged = traffic with VLAN ID → for trunks (switch-to-switch, switch-router).**
- **Untagged = traffic without VLAN ID → for access ports or native VLAN.**

How to change allow specific vlan in trunk interface

In switch:

Enable

Configure terminal

!

Interface GigabitEthernet0/1

Switchport trunk allowed vlan 2 //allows vlan 2 traffic only

(or)

Switchport trunk allowed vlan 3 // allows vlan 3 traffic only

(or)

Switchport trunk allowed vlan 2, 3 // allows vlan 2,3

exit

How to remove VLAN on trunk

```
Interface GigbitEthernet0/1
Switchport trunk allowed vlan remove 5
Exit
```

# STP (Spanning Tree Protocol)

## 🔷 What is STP?

**Spanning Tree Protocol (STP)** is a **Layer 2 protocol** (IEEE 802.1D) that prevents **loops** in a switched Ethernet network by creating a **loop-free logical topology**.

## 🔷 Why do we need STP?

- In a switched network with **redundant links**, **loops** can form.
- Loops cause:
    - **Broadcast storms** (frames keep circulating forever).
    - **MAC table instability** (switches keep updating MAC entries).
    - **Multiple frame copies** (end devices receive duplicates).
- STP prevents this by **blocking some redundant links**, leaving only **one active path** between switches.

## 🔷 How STP Works

1. **Root Bridge Election**
    - One switch is elected the **root bridge** (lowest Bridge ID = Priority + MAC).
2. **Path Selection**
    - STP calculates the **shortest path to the root** using **path cost** (based on port speed).
3. **Port Roles**
    - **Root Port (RP):** Best path to the root bridge (one per switch, except root).
    - **Designated Port (DP):** Forwarding port for a LAN segment.
    - **Blocked Port (BP):** Redundant port placed in blocking state to prevent loops.
4. **Port States (Classic STP)**
    - **Blocking → Listening → Learning → Forwarding**
    - Takes ~30–50 seconds to converge.

🔷 **STP Variants**

- **STP (802.1D):** Original, slow convergence (~50s).
- **RSTP (802.1w):** Rapid STP, faster convergence (~<10s).
- **MSTP (802.1s):** Multiple STP, supports multiple VLANs mapped to instances.
- **PVST+ (Cisco):** Per-VLAN STP (separate STP instance per VLAN).
- **Rapid-PVST+ (Cisco):** Cisco's fast version of PVST.

🔷 **Example (2 Switches + Redundant Links)**

```
SW1 ---------------- SW2
 |                    |
 |                    |
(Link A)           (Link B)
```

- Without STP → loop forms between SW1 & SW2.
- With STP → one link (say Link B) is **blocked**, traffic flows only over Link A.
- If Link A fails → STP **unblocks** Link B, ensuring redundancy.

🔷 **Verify STP on Cisco**

```
show spanning-tree
```

You'll see:

- Root bridge info
- Port roles (Root, Designated, Alternate/Blocked)
- STP timers and costs

✅ **In short:** STP is the "traffic cop" at Layer 2, preventing loops while still allowing redundancy.

**STP Elections** — this is where switches decide **who becomes the Root Bridge** and which ports take on which roles. Let's go step by step:

### 🔷 Step 1: Root Bridge Election

- Each switch has a **Bridge ID (BID)** = **Bridge Priority (default 32768) + Switch MAC address**.
- The switch with the **lowest Bridge ID** becomes the **Root Bridge**.
- If priorities are the same → the switch with the **lowest MAC address** wins.

📌 Example:

- SW1 → Priority 32768, MAC 00:11:22:33:44:55
- SW2 → Priority 32768, MAC 00:11:22:33:44:11
  👉 SW2 becomes Root Bridge (lower MAC).

### 🔷 Step 2: Root Port (RP) Election

- On **non-root switches**, the **port with the lowest path cost** to the root becomes the **Root Port**.
- **Path cost** is based on link speed (shorter = better):
  - 10 Mbps → 100 cost
  - 100 Mbps → 19 cost
  - 1 Gbps → 4 cost
  - 10 Gbps → 2 cost

📌 Example:
If SW3 has 2 paths to the root:

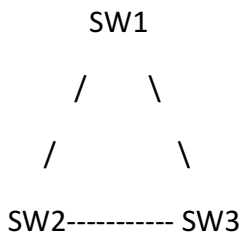
- Path A (SW3→SW2) = 19 cost
- Path B (SW3→SW1→SW2) = 19 + 19 = 38 cost
  👉 SW3 chooses Path A (lower cost), so its port facing SW2 is Root Port.

### 🔷 Step 3: Designated Port (DP) Election

- On each network segment, the **switch with the lowest path cost to the root** has its port become the **Designated Port** (forwarding).
- Other ports on that segment go into **Blocking/Alternate** state.

**STP election example with 3 switches**.

🔷 Topology




- All three are connected (triangle topology → redundant links).

🔷 **Step 1: Root Bridge Election**

Each switch advertises its **Bridge ID (BID)** = Priority + MAC.

Assume:

- SW1 → Priority 32768, MAC 00:11:11:11:11:11
- SW2 → Priority 32768, MAC 00:22:22:22:22:22
- SW3 → Priority 32768, MAC 00:33:33:33:33:33

👉 **SW1 wins Root Bridge** (lowest MAC, since priorities equal).

🔷 **Step 2: Root Ports (on Non-root Switches)**

- SW2:
  - Path to Root = Direct link SW2 → SW1
  - Cost = 19 (FastEthernet 100 Mbps)
    👉 SW2's port facing SW1 = **Root Port**.
- SW3:
  - Path to Root = Direct link SW3 → SW1
  - Cost = 19
    👉 SW3's port facing SW1 = **Root Port**.

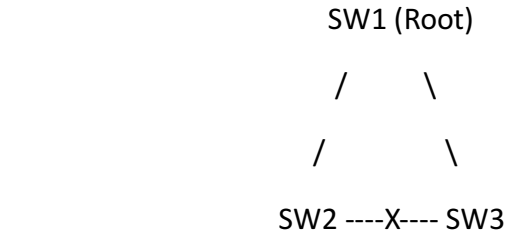
🔷 **Step 3: Designated Ports**

- On each **segment**, the port closest to the Root is **Designated**.
- SW1 (Root):
  - Its ports to SW2 and SW3 are always **Designated Ports**.
- SW2 ↔ SW3 segment:
  - SW2 path to Root = 19
  - SW3 path to Root = 19
  - Tie-breaker = lower Bridge ID → SW2 wins.
    👉 SW2's port = **Designated Port**, SW3's port = **Blocked (Alternate Port)**.

🔷 **Step 4: Port Roles Summary**

- **SW1 (Root Bridge)**
  - Ports → **Designated** (to SW2, SW3).
- **SW2**
  - Port to SW1 → **Root Port**
  - Port to SW3 → **Designated Port**
- **SW3**
  - Port to SW1 → **Root Port**
  - Port to SW2 → **Blocked Port**

✅ **Final Logical Topology (loop-free path)**




(X = blocked port on SW3 toward SW2)

STP (Spanning Tree Protocol)

- Is a layer 2 protocol
- Is used to stop layer 2 loops
- STP performs root bridge, root port and designated port elections to stop layer 2 loops

**Elections:**

**1. Root Bridge:**

- Is elected based on lowest switch priority
- Default priority is 32768
- If priority is equal on all switches. It will use lowest MAC-address
- The switch which has lowest MAC-address becomes root-bridge

**2. Root port:**

- Root port is elected only on non-root switches
- Is elected based on lowest path cost
- The port which has lowest path cost becomes root port
- If path cost is equal on both ports then it use lowest MAC-address of up-stream root port
- If up-stream MAC-address is same then it use up-stream switch lowest port priority as a tie breaker

**3. Designated Port:**

- On every segment one port should be elected as designated port
- The port on switch, which has lowest path cost to root bridge becomes designated port
- If path cost is equal on both switches
- It select port on switch, which has lowest MAC-address

**STP port rules:**

- Root port
- Designated port
- Block port

**STP port states:**

- Blocking
- Listening (15 seconds)
- Learning (15 seconds)

- Forwarding (20 seconds)
- Alternating

**Note:**

#show version // to see base ethernet MAC address

#show spanning-tree // to see spanning tree details

#show interface status

**To configure switch priority:**

#spanning-tree VLAN 1 priority 4092

(or)

#spanning-tree VLAN 1 root primary

**If decrease the priority based on remaining switch priorities:**

**Ex:**

32768

(-) 8192

---

24576

(+) 1 // default vlan0

---

24577

---

🔷 **Standard Path Costs (IEEE)**

**Old STP Cost Values (IEEE 802.1D, “short” cost method):**

| **Link Speed** | **Path Cost** |
|---|---|
| 10 Mbps (Ethernet) | 100 |

| Link Speed | Path Cost |
|---|---|
| 100 Mbps (FastEthernet) | 19 |
| 1 Gbps (Gigabit Ethernet) | 4 |
| 10 Gbps | 2 |

**New STP Cost Values (IEEE 802.1t, "long" cost method – to handle faster speeds):**

| Link Speed | Path Cost |
|---|---|
| 10 Mbps | 2,000,000 |
| 100 Mbps | 200,000 |
| 1 Gbps | 20,000 |
| 10 Gbps | 2,000 |
| 100 Gbps | 200 |
| 1 Tbps | 20 |

👉 Modern Cisco switches usually use the **long method** (so STP can scale with higher speeds), but you can configure which method is used.

🔷 Cisco Command to Check

```
Switch# show spanning-tree
```

Shows the **cost** of each Root Port/Designated Port.

```
Switch(config)# spanning-tree pathcost method short
Switch(config)# spanning-tree pathcost method long
```

- Changes between short/long calculation.

🔷 **Example**

If SW2 has two paths to the Root Bridge:

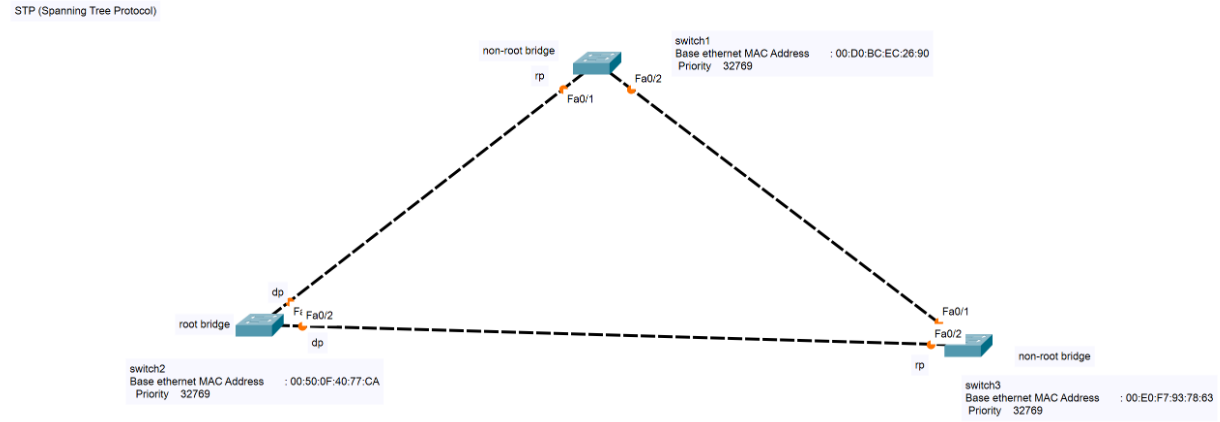
- Path A: 100 Mbps link (cost 19)
- Path B: 2 × 100 Mbps links (EtherChannel, seen as 200 Mbps → cost 9)

👉 STP will select **Path B** (lower cost).

✅ **In short:**

- **Ethernet (10 Mbps)** = cost 100
- **FastEthernet (100 Mbps)** = cost 19
- **GigabitEthernet (1 Gbps)** = cost 4
- **10 Gbps** = cost 2

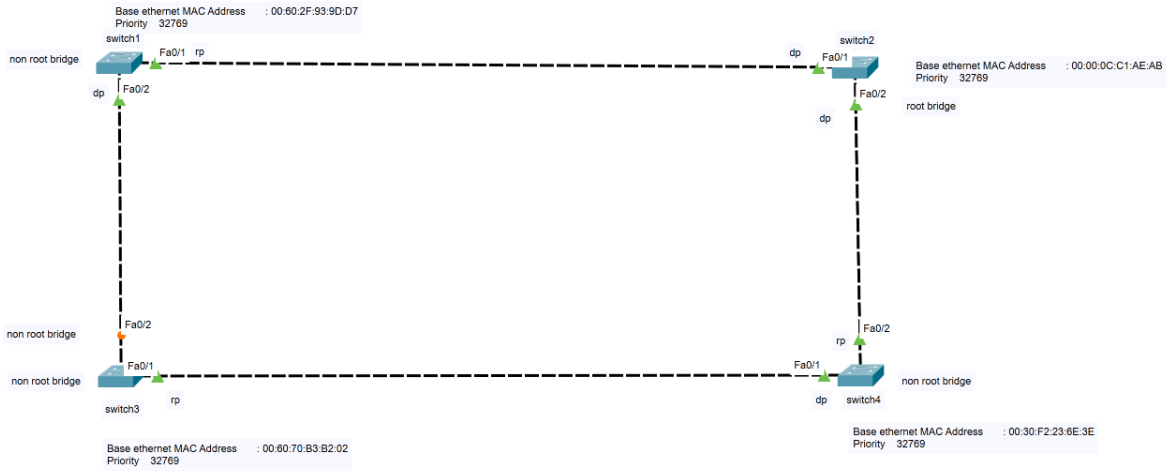
(using short method; long method uses bigger numbers but same logic).




Election1:
Root Bridge:
Condition 1:
Is elected based on lowest switch priority
Default all switches priority is 32768 +1
so all switches priority is equal.
Condition 2:
If priority is equal on all switches, It will use lowest MAC-address
Which means which switch has lowest MAC-address, that switch become Root Bridge
switch2
Base ethernet MAC Address : 00:50:0F:40:77:CA
remaining all swithces become non-root bridges

Election 2:
Root port:
Condition 1:
Root port is elected only on non-root bridge switches
Condition 2:
RP is elected based on lowest path cost
Here, takes all are Fastethernets path cost is 19
So all paths have same cost
Condition 3:
If path cost is equal on both ports then, in switch which port has facing root bridge that
Become root port
In switch 1:
Fa0/1 is RP
In switch 3:
Fa0/2 is RP

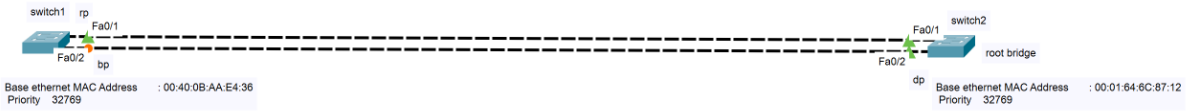
Election 3:Designated Port:
Condition 1:
On root bridge all ports are dp
Condition 2:
On every segment one port should be elected as designated port
Condition 3:
The port on switch, which has lowest path cost to root bridge becomes designated port
Condition 4:
If path cost is equal on both switches. it select port on switch, which has lowest MAC-address. That switch port become dp




Election 1:
Root Bridge:
Condition 1:
Is elected based on lowest switch priority
Default all switches priority is 32768 +1
so all switches priority is equal.
Condition 2:
If priority is equal on all switches, It will use lowest MAC-address
Which means which switch has lowest MAC-address, that switch become Root Bridge
switch2
Base ethernet MAC Address : 00:00:0C:C1:AE:AB
remaining all swithces become non-root bridges

Election 2:
Root port:
Condition 1:
Root port is elected only on non-root bridge switches
Condition 2:
RP is elected based on lowest path cost
Here, takes all are Fastethernets path cost is 19
So all paths have same cost
Condition 3:
If path cost is equal on both ports then, in switch which port has facing root bridge that
Become root port
In switch 1:
Fa0/1 is RP
In switch 4:
Fa0/2 is RP
In switch 3:
SW3: has two possible paths to SW1:
SW3 → SW1 → SW2 = cost 19 + 19 = 38
SW3 → SW4 → SW2 = cost 19 + 19 = 38
Tie (equal cost),
Condition 4:
If path cost same then choose port towards lowest mac address port
so STP breaks tie using Bridge IDs: SW1 (00:60:2F:93:9D:D7) vs SW4 (00:30:F2:23:6E:3E), SW4 wins (lower BID).
👉 SW3 chooses port toward SW4 as Root Port.

Election 3:
Designated Port:
Condition 1:
On root bridge all ports are dp
All swith2 ports are dp
Condition 2:
Sw1 – sw2:
Sw1 Fa0/1: rp
Sw2 Fa0/1: dp
Sw2-sw4:
Sw2 Fa0/2: dp
Sw4 Fa0/2: rp
Sw3-sw4:
Sw3 Fa0/1: rp
Sw4 Fa0/1:
Sw3 Fa0/1 is rp and sw4 Fa0/2 is rp so sw4 Fa0/1 is automatically dp
Sw1-sw3:
Sw1 Fa0/2:
Path cost towards root bridge:
Path cost: 19
Sw3 Fa0/2:
Path cost towards root bridge:
Path cost: 38
Least path cost switch port become dp
Sw1 Fa0/2 is dp
Sw3 Fa0/2 is Block port




Election 1:
Root Bridge:
swithc2 mac-address is lowest
so switch2 is become root bridge

Election 2:
Root port:
Condition 1:
Switch1 is not the root. Its cost to reach the root (switch2) through either link = 19 (FastEthernet)
So both Fa0/1 and Fa0/2 on switch1 have equal cost (19) to the root
Condition 2:
A non-root switch must pick on root port (rp) = the port with the lowest cost to the root
Here both have same cost =19
Tie-breaker:
1. Lowest upstream bridge ID
Both ports connect to the same root bridge (swithc2), so equal
2. Lowest port ID (on the root side)
Each switch port has a Port ID = (Port Priority [default 128] + Port Number)
On switch2 (the root), the port IDs are:
Fa0/1 à port ID = 128.1
Fa0/2 à port ID = 128.2
Since 128.1 3. Therefore, switch1 chooses its Fa0/1 as the Root Port (RP)
The other port (Fa0/2) is not needed for reaching the root, so it goes into Blocking state.

Eletion 3:
on root bridge all ports are dp

**How to change port priority?**

Interface interface_name

Spanning-tree VLAN 1 port priority 64

Example:

Interface Fa0/20

Spanning-tree vlan 1 port priority 64

# EtherChannel

## 🔷 What is EtherChannel?

EtherChannel is a **link aggregation technology** used in Cisco networking.
It **bundles multiple physical Ethernet links** into a single **logical link** between switches, routers, or servers.

This provides:

1. **Higher Bandwidth** (sum of all bundled links).
2. **Redundancy** (if one link fails, traffic continues over others).
3. **Load Balancing** (traffic is distributed across links).

## 🔷 Key Points

- Up to **8 physical links** can be bundled into one EtherChannel.
- Seen as **one logical interface** (Port-Channel).
- STP (Spanning Tree Protocol) treats the bundle as **one link**, avoiding blocking of extra links.

## 🔷 EtherChannel Protocols

Two main negotiation protocols are used:

- **PAgP (Port Aggregation Protocol – Cisco proprietary)**
    - Modes: **Auto, Desirable**
    - Connection:

| | Auto | Desirable |
|---|---|---|
| **Auto** | ❌ | ✅ |
| **Desirable** | ✅ | ✅ |

With **PAgP (Port Aggregation Protocol – Cisco proprietary)**:

- You can bundle a **maximum of 8 physical interfaces** into a single EtherChannel.
- Unlike LACP, **PAgP does NOT support standby ports**.

- That means: if you try to configure more than 8 ports, the extras will just stay inactive or error out — they won't sit in a standby pool.

- **LACP (Link Aggregation Control Protocol – IEEE 802.3ad)**
  - Modes: **Active, Passive**
  - Connection:

| | **Active** | **Passive** |
|---|---|---|
| **Active** | ✅ | ✅ |
| **Passive** | ✅ | ❌ |

- You can **configure up to 16 physical interfaces** in one EtherChannel (Port-Channel).
- Out of those:
  - **8 interfaces are active** (actually carrying traffic).
  - **8 interfaces are standby** (waiting to take over if one of the active links fails).

So in practice:

- **Max active links = 8**
- **Extra links (up to 8) = standby**

✅ Example:
If you bundle **12 ports** in an LACP group:

- 8 ports → used (active)
- 4 ports → standby (automatically activated if any active port goes down)

Quick comparison:

| **Feature** | **LACP (802.3ad)** | **PAgP (Cisco)** |
|---|---|---|
| **Max ports in bundle** | 16 (8 active + 8 standby) | 8 (all active) |
| **Vendor** | Open standard (IEEE) | Cisco proprietary |
| **Modes** | Active / Passive | Desirable / Auto |
| **Standby support** | ✅ Yes (up to 8) | ❌ No |

✅ So:

- PAgP → up to 8 active links, no standby.
- LACP → up to 16 (8 active + 8 standby).

🔷 **Configuration Example (Cisco Switch)**

**Step 1: Select interfaces**

```
Switch(config)# interface range fa0/1 – 2
```

**Step 2: Configure channel group**

- Using **PAgP**

```
Switch(config-if-range)# channel-group 1 mode desirable
```

- Using **LACP**

```
Switch(config-if-range)# channel-group 1 mode active
```

- Manual (On)

```
Switch(config-if-range)# channel-group 1 mode on
```

**Step 3: Configure Port-Channel**

```
Switch(config)# interface port-channel 1

Switch(config-if)# switchport mode trunk
```

🔷 **EtherChannel Load Balancing**

Traffic can be balanced based on:

- Source MAC / Destination MAC
- Source IP / Destination IP
- Source-Destination IP/MAC pair

**Check or set:**

```
Switch# show etherchannel load-balance

Switch(config)# port-channel load-balance src-dst-ip
```

## 🔷 Verification Commands

Switch# show etherchannel summary

Switch# show running-config

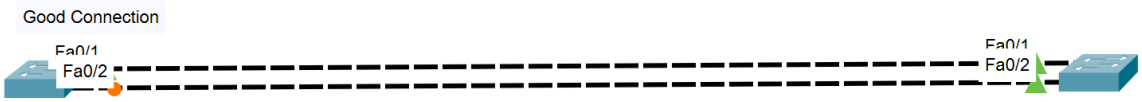
Switch# show interfaces port-channel 1

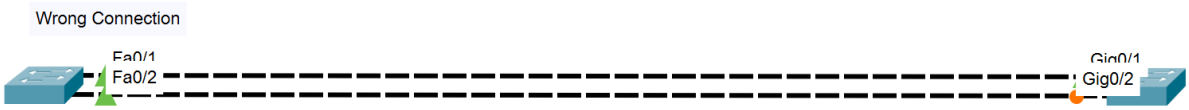
✅ In short: **EtherChannel = multiple links acting as one big pipe, with redundancy & load balancing.**

**Requirements:**

Interface type, speed, mode should be same both sides.

Example:




# Inter-VLAN Routing

## What is Inter-VLAN Routing?

- By default, **hosts in different VLANs cannot communicate** with each other (Layer 2 isolation).
- To allow communication **between VLANs**, we need a **Layer 3 device** (Router or L3 Switch).
  👉 This process is called **Inter-VLAN Routing**.

## 🛠 Methods of Inter-VLAN Routing

### 1. Router-on-a-Stick (ROAS)

- A single router interface is used.
- It is configured as a **trunk** port.
- Router creates **sub-interfaces**, each with an IP address for the VLAN.
- Acts as the **default gateway** for hosts in that VLAN.

🔷 Example:

```
# On Switch
Switch(config)#interface fa0/1
Switch(config-if)#switchport mode trunk

# On Router
Router(config)#interface g0/0
Router(config-if)#no shutdown

Router(config)#interface g0/0.10
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0

Router(config)#interface g0/0.20
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
```

✅ Now PCs in VLAN 10 (192.168.10.0/24) can talk to VLAN 20 (192.168.20.0/24).

**2. Layer 3 Switch (SVI – Switched Virtual Interface)**

- No router needed, the L3 switch itself does routing.
- Create a **VLAN interface (SVI)** and assign an IP.
- Enable **IP routing**.

🔷 Example:

```
Switch(config)#ip routing
Switch(config)#interface vlan 10
Switch(config-if)#ip address 192.168.10.1 255.255.255.0
Switch(config-if)#no shutdown

Switch(config)#interface vlan 20
Switch(config-if)#ip address 192.168.20.1 255.255.255.0
Switch(config-if)#no shutdown
```

✅ Now VLANs communicate directly via the L3 switch.

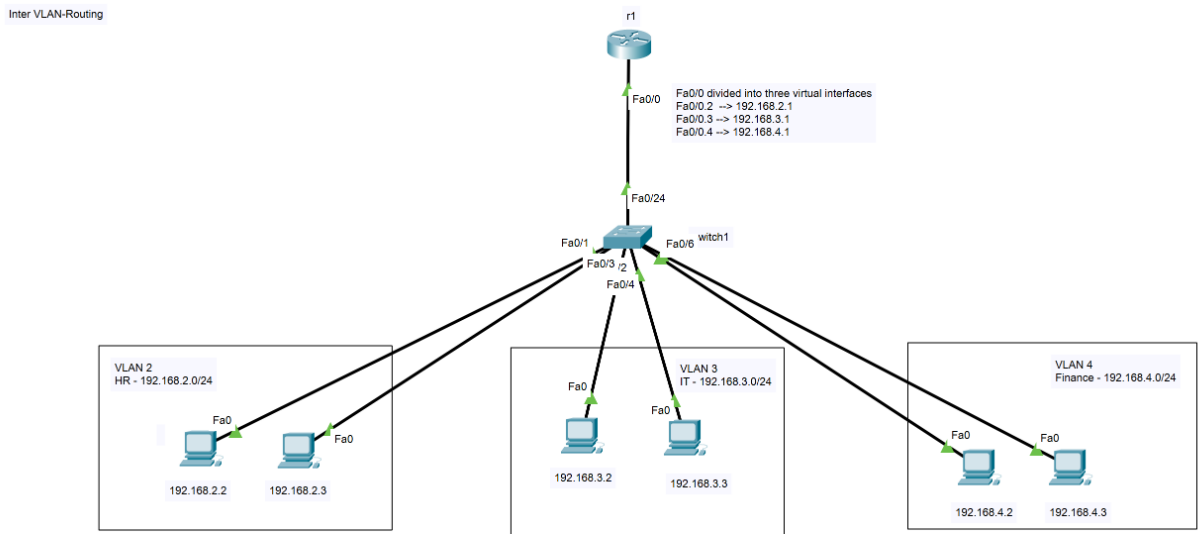
**3. Legacy Method (Multiple Router Interfaces)**

- Each VLAN connected to a separate router physical interface.
- Not scalable → rarely used.

📊 **Summary**

- **Router-on-a-Stick** = Good for small networks.
- **L3 Switch (SVI)** = Scalable, faster (hardware routing).
- **Multiple Router Interfaces** = Outdated.

# 1. Router-on-a-Stick (ROAS)




```
! On r1
enable
configure terminal
hostname r1
!
! Only physical interface make up state
interface Fa0/0
no shutdown
exit
!
! create 3 virtual interfaces
interface Fa0/0.2
encapsulation dot1Q 2
ip address 192.168.2.1 255.255.255.0
exit
!
interface Fa0/0.3
encapsulation dot1Q 3
ip address 192.168.3.1 255.255.255.0
exit
!
interface Fa0/0.4
encapsulation dot1Q 4
ip address 192.168.4.1 255.255.255.0
exit
!
```

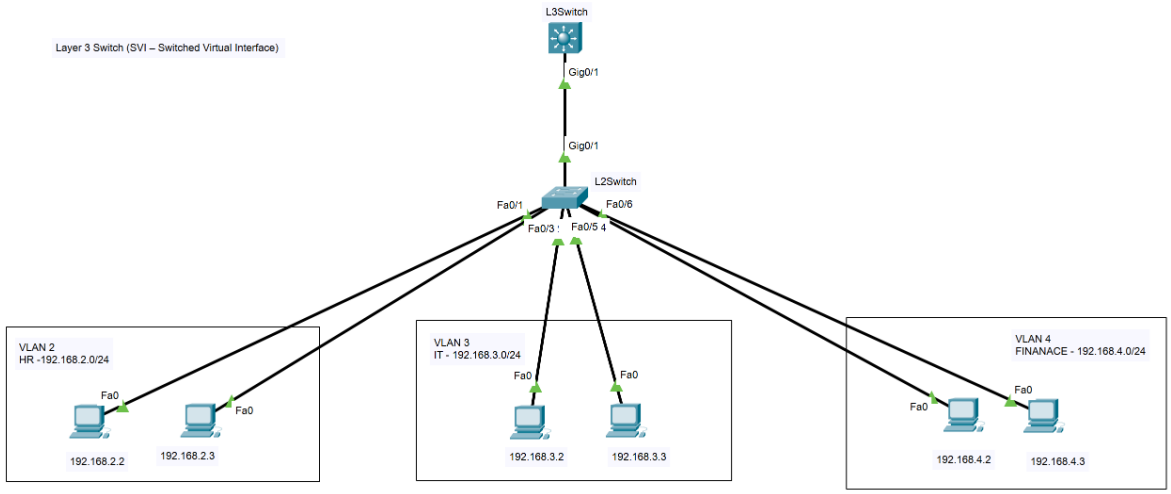
```
! On switch1
enable
configure terminal
hostname switch1
!
! Create VLANs for 3 depts
vlan 2
name HR
exit
!
vlan 3
name IT
exit
!
vlan 4
name Finance
exit
!
! Configuring interfaces to vlans
interface range FastEthernet0/1-2
switchport mode access
switchport access vlan 2
exit
!
interface range FastEthernet0/3-4
switchport mode access
switchport access vlan 3
exit
!
interface range FastEthernet0/5-6
switchport mode access
switchport access vlan 4
exit
!
! configuring trunk to Fa0/24 for vlan communications
interface Fa0/24
switchport mode trunk
switchport trunk allowed vlan 2,3,4
exit
!
```

Here,
One department can communicate with other departments because we use trunk

IF: we need to allow only communication between HR and Finanace department

```
! on swith1
interface Fa0/24
switchport mode trunk
switchport trunk allowed vlan 2, 4
exit
!
```

## 2. Layer 3 Switch (SVI – Switched Virtual Interface)




```
! On L3Switch
enable
configure terminal
hostname L3Switch
!
vlan 2
name HR
exit
!
vlan 3
name IT
exit
!
vlan 4
name Finance
exit
!
interface vlan 2
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
!
interface vlan 3
ip address 192.168.3.1 255.255.255.0
no shutdown
exit
!
interface vlan 4
ip address 192.168.4.1 255.255.255.0
no shutdown
exit
!
 interface Gig0/1
switchport trunk encapsulation dot1q
switchport mode trunk
exit
!
ip routing
!
```

```
! On L2Switch
enable
configure terminal
!
vlan 2
name HR
exit
!
vlan 3
name IT
exit
!
vlan 4
name Finance
exit
!
interface range Fa0/1-2
switchport mode access
switchport access vlan 2
exit
!
interface range Fa0/3-4
switchport mode access
switchport access vlan 3
exit
!
interface range Fa0/5-6
switchport mode access
switchport access vlan 4
exit
!
interface Gig0/1
switchport mode trunk
switchport trunk allowed vlans 3,4
exit
!
```

# BGP (Border Gateway Protocol)

## 🔷 What is BGP?

**BGP (Border Gateway Protocol)** is a **routing protocol** used to exchange routing information between different autonomous systems (AS) on the internet. It's called a **Path Vector Protocol**.

- Current version: **BGP-4**
- Defined in: **RFC 4271**
- It's the protocol that **makes the internet work**, deciding how packets travel between ISPs, data centers, and enterprises.

---

## 🔷 Key Features

1. **Inter-domain Routing** → Connects different ASes.
2. **Path Vector** → Uses AS-Path attribute to avoid loops.
3. **Policy-based Routing** → Routing decisions based on policies, not just metrics.
4. **Scalable** → Can handle hundreds of thousands of routes (global internet BGP tables are ~1M+ routes).

---

## 🔷 BGP Types

- **eBGP (External BGP)** → Runs between different ASes (e.g., ISP to ISP).
- **iBGP (Internal BGP)** → Runs within the same AS (e.g., inside an ISP's network).

---

## 🔷 BGP Attributes (used to choose best path)

1. **Weight** (Cisco-specific, local to router)
2. **Local Preference** (higher preferred, within AS)
3. **AS-Path** (shorter preferred, prevents loops)
4. **Origin** (IGP > EGP > Incomplete)
5. **MED (Multi-Exit Discriminator)** (lower preferred, between ASes)
6. **eBGP over iBGP**

7. **IGP metric to next-hop** (shorter is preferred)

---

🔷 **BGP Message Types**

1. **OPEN** → Establishes BGP session.
2. **UPDATE** → Advertises or withdraws routes.
3. **KEEPALIVE** → Keeps session alive.
4. **NOTIFICATION** → Error messages, closes session.

---

🔷 **BGP States**

1. Idle
2. Connect
3. Active
4. OpenSent
5. OpenConfirm
6. Established ✅ (routes exchanged here)

---

🔷 **BGP Configuration (Syntax):**

router bgp <Autonomous System Number>

neighbor <Next Hop IP address> remote-as <Neighbor Autonomous System Number>

network <Connected Network> <Subnet Mask>

exit

example:

router bgp 65001

neighbor 192.0.2.2 remote-as 65002

network 10.1.1.0 mask 255.255.255.0

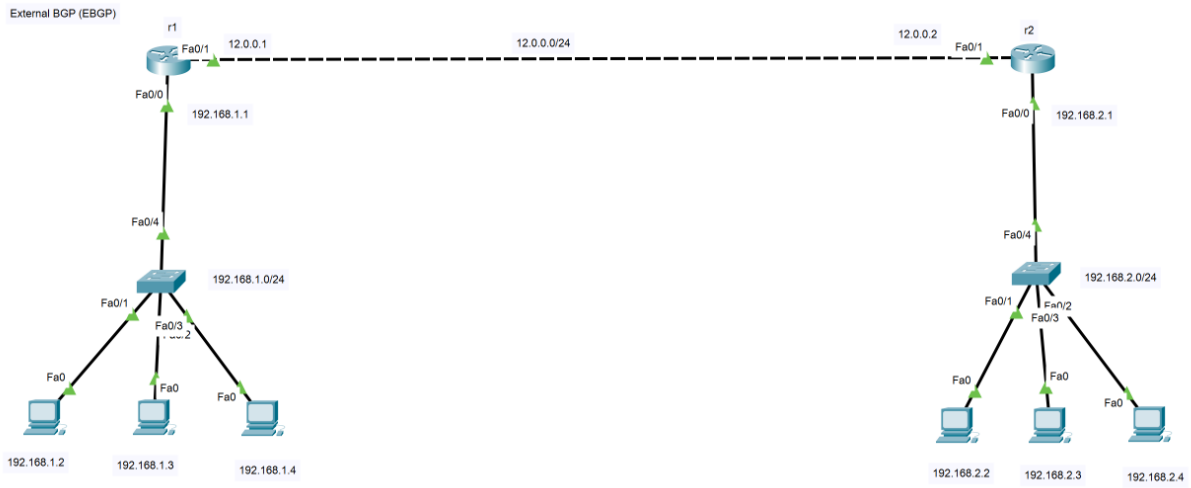
Verify BGP session:

R1# show ip bgp summary

R2# show ip bgp summary

Verify routes learned:

R1# show ip route




```
! On r1
enable
configure terminal
hostname r1
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.1 255.255.255.0
no shutdown
exit
!
router bgp 65001
neighbor 12.0.0.2 remote-as 65002
network 192.168.1.0 mask 255.255.255.0
network 12.0.0.0 mask 255.255.255.0
exit
!
```

```
! On r2
enable
configure terminal
hostname r2
!
interface Fa0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.2 255.255.255.0
no shutdown
exit
!
!configuring bgp
router bgp 65002
neighbor 12.0.0.1 remote-as 65001
network 192.168.2.0 mask 255.255.255.0
network 12.0.0.0 mask 255.255.255.0
exit
!
```

```
! On r1
enable
configure terminal
hostname r1
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.1 255.255.255.0
no shutdown
exit
!
router bgp 65001
neighbor 12.0.0.2 remote-as 65001
network 192.168.1.0 mask 255.255.255.0
network 12.0.0.0 mask 255.255.255.0
exit
!
```

```
! On r2
enable
configure terminal
hostname r2
!
interface Fa0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.2 255.255.255.0
no shutdown
exit
!
!configuring bgp
router bgp 65001
neighbor 12.0.0.1 remote-as 65001
network 192.168.2.0 mask 255.255.255.0
network 12.0.0.0 mask 255.255.255.0
exit
!
```

# IPSec (Internet Protocol Security)

## 🔷 What is IPSec VPN?

**IPSec (Internet Protocol Security)** is a suite of protocols that provides **secure communication over IP networks**.
When combined with VPN (Virtual Private Network), it allows two networks (or devices) to securely exchange data across an untrusted network (like the internet).

👉 Think of it as building an **encrypted tunnel** between two routers so LANs behind them can communicate securely.

## 🔷 Why Use IPSec VPN?

- **Confidentiality** → Encrypts traffic (nobody can read your data).
- **Integrity** → Protects against tampering (hashing).
- **Authentication** → Verifies who is on the other side.
- **Anti-replay protection** → Prevents attackers from reusing old packets.

## 🔷 How IPSec VPN Works (Two Phases)

### 1. IKE Phase 1 (Key Exchange & Secure Channel)

- Protocol: **ISAKMP (Internet Security Association and Key Management Protocol)**.
- Purpose: Establish a **secure, authenticated channel** between VPN peers.
- Negotiates:
    - Encryption (AES, DES, 3DES…)
    - Hashing (SHA, MD5…)
    - Authentication (pre-shared key or certificates)
    - Diffie-Hellman group (key exchange)
    - Lifetime
- Result: Creates an **ISAKMP Security Association (SA)**.

👉 Output: a **secure channel** for negotiating IPSec parameters.

**2. IKE Phase 2 (IPSec Tunnel Negotiation)**

- Uses the secure channel from Phase 1.
- Negotiates IPSec parameters:
    - Protocol: **ESP (Encapsulating Security Payload)** or **AH (Authentication Header)**
        - **ESP** = encryption + authentication (most common)
        - **AH** = authentication only (rarely used)
    - Encryption & hashing algorithms
- Defines which traffic to protect (via Access Control List).
- Result: Creates an **IPSec Security Association (SA)**.

👉 Output: a **secure tunnel** where data packets are encrypted and authenticated.

🔷 **Modes of IPSec**

1. **Transport Mode**
    - Encrypts only the payload of the packet.
    - Header remains in clear text.
    - Used in host-to-host or end-to-end connections.
2. **Tunnel Mode**
    - Encrypts the entire packet (payload + header).
    - Wraps it in a new IP header.
    - Most common for **site-to-site VPNs**.

🔷 **IPSec VPN Types**

1. **Site-to-Site VPN**
    - Connects two networks securely over WAN/internet.
    - Example: Office A ↔ Office B
    - Implemented on **routers/firewalls**.

2. **Remote Access VPN**
    - Connects individual remote users securely to a central office network.
    - Example: Work-from-home employees using VPN client software.

🔷 **Example Flow in Your Topology**

- LAN A (192.168.1.0/24) ↔ LAN B (192.168.2.0/24)
- Routers R1 & R2 form an **IPSec tunnel** over the WAN (12.0.0.0/24).
- When PC1 pings PC2:
    1. R1 checks ACL → traffic matches VPN policy.
    2. R1 encrypts packet using IPSec (ESP).
    3. Encrypted packet travels across WAN to R2.
    4. R2 decrypts and delivers original packet to PC2.

So the data looks like **scrambled garbage** on the WAN, but arrives cleanly at the destination.

🔷 **Verification (Cisco IOS)**

- **IKE Phase 1**:

  show crypto isakmp sa

  Should show **QM_IDLE** (means Phase 1 successful).
- **IKE Phase 2**:
  show crypto ipsec sa

  Shows **encaps/decaps counters** (packets being encrypted/decrypted).

✅ In summary:
**IPSec VPN = secure encrypted tunnel (over untrusted networks) between two peers, established in two phases (IKE Phase 1 and Phase 2), protecting LAN-to-LAN or host-to-network traffic.**

**Example:**

**build an IPSec VPN (Site-to-Site VPN) between your two routers (R1 ↔ R2) so that LANs 192.168.1.0/24 and 192.168.2.0/24 can securely communicate over the WAN (12.0.0.0/24).**

🔷 **IPSec VPN – Step by Step (Cisco Routers)**

**We'll configure a site-to-site IPSec VPN.**

**1. Topology Recap**

- **R1**
  - **LAN: 192.168.1.0/24**
  - **WAN: 12.0.0.1**
- **R2**
  - **LAN: 192.168.2.0/24**
  - **WAN: 12.0.0.2**

**2. Access-Lists (Define Interesting Traffic)**

**On R1:**

```
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
```

On **R2**:

```
access-list 101 permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
```

**3. IKE Phase 1 – ISAKMP Policy**

On **R1**:

```
crypto isakmp policy 10
 encr aes
 hash sha256
 authentication pre-share
```

```
 group 5
 lifetime 86400

crypto isakmp key VPNKEY address 12.0.0.2
```

On **R2**:

```
crypto isakmp policy 10
 encr aes
 hash sha256
 authentication pre-share
 group 5
 lifetime 86400

crypto isakmp key VPNKEY address 12.0.0.1
```

**4. IKE Phase 2 – IPSec Transform Set**

On **both routers**:

```
crypto ipsec transform-set MYSET esp-aes esp-sha-hmac
```

**5. Crypto Map (Bind Everything Together)**

On **R1**:

```
crypto map MYMAP 10 ipsec-isakmp
 set peer 12.0.0.2
 set transform-set MYSET
 match address 101
```

On **R2**:

```
crypto map MYMAP 10 ipsec-isakmp
 set peer 12.0.0.1
 set transform-set MYSET
```

match address 101

**6. Apply Crypto Map to WAN Interfaces**

On **R1**:

interface fa0/1

crypto map MYMAP

On **R2**:

interface fa0/1

crypto map MYMAP

**7. Verification**

- From **PC1 (192.168.1.2)** → ping **PC2 (192.168.2.2)**
- On routers, check VPN status:

  show crypto isakmp sa ← Phase 1

  show crypto ipsec sa ← Phase 2

You should see **QM_IDLE** (Phase 1 up) and encrypted packets in IPSec SA.

🔷 **How It Works**

- When PC1 sends traffic to PC2, the router sees it matches ACL 101.
- IPSec VPN encrypts that traffic.
- The encrypted packet is sent via 12.0.0.x.
- R2 decrypts and forwards it to PC2.

⚠️ **Important Note:**

- Cisco **Packet Tracer has limited IPSec support** (crypto commands exist, but full VPN behavior may not work).
- For **full IPSec VPN testing** (with encryption counters), use **GNS3, EVE-NG, or real Cisco IOS**.

## 🔷 Encryption in IPSec

In IPSec, encryption is provided mainly by **ESP (Encapsulating Security Payload)**.
There are two big categories:

### 1. Symmetric Encryption

- Uses the **same key** for encryption and decryption.
- Fast, efficient → ideal for VPN tunnels.
- Keys are exchanged securely via **IKE (Diffie-Hellman)**.

**Common Symmetric Algorithms in IPSec:**

1. **DES (Data Encryption Standard)**
    - 56-bit key.
    - Very weak today (obsolete, easily broken).
    - ⚠️ Do not use in production.
2. **3DES (Triple DES)**
    - Applies DES three times with different keys.
    - Stronger than DES but **slow**.
    - Legacy, being phased out.
3. **AES (Advanced Encryption Standard)**
    - Industry standard today.
    - Key sizes: **128-bit, 192-bit, 256-bit**.
    - Very secure + efficient.
    - Most recommended for IPSec VPNs.
4. **Blowfish / CAST**
    - Supported in some implementations.
    - Not as common as AES.

**2. Asymmetric Encryption**

- Uses a **public key** to encrypt and a **private key** to decrypt.
- Too slow for bulk data encryption.
- Used only in **IKE Phase 1** (to exchange keys securely, via Diffie-Hellman).
- Example algorithms:
    - RSA
    - Diffie-Hellman (for key exchange, not actual encryption)

**3. Hashing (Data Integrity – not encryption, but related)**

While encryption scrambles data, IPSec also uses hashing for **integrity**:

- **MD5 (128-bit hash)** → Weak today.
- **SHA-1 (160-bit hash)** → Better, but also considered weak.
- **SHA-2 (256, 384, 512-bit)** → Modern standard, secure.

🔷 **Summary Table**

| Purpose | Algorithms Used in IPSec |
|---|---|
| **Encryption (ESP)** | DES, 3DES, AES (128/192/256), Blowfish |
| **Key Exchange** | Diffie-Hellman, RSA |
| **Integrity (HMAC)** | MD5, SHA-1, SHA-2 |

🔷 **Real-World Best Practice**

For a secure IPSec VPN today, you'd normally configure:

- **AES-256** for encryption
- **SHA-256** for integrity
- **DH Group 14/19/20** (strong Diffie-Hellman groups)
- **Pre-shared key or digital certificates** for authentication

✅ So in short:
**IPSec VPN uses symmetric encryption (AES/3DES) for the actual data, asymmetric encryption (DH/RSA) only for key exchange, and hashing (SHA/MD5) for integrity.**

🔷 **IPSec VPN Syntax & Explanation**

**1. Define Interesting Traffic (Access-List)**

This defines **which traffic should be encrypted**.

access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255

✅ Meaning: Any traffic between LAN 192.168.1.0/24 and LAN 192.168.2.0/24 will be protected by IPSec.

---

**2. IKE Phase 1 – ISAKMP Policy**

This is where peers agree on how to build the **secure channel**.

```
crypto isakmp policy 10
 encr aes         ! Encryption algorithm (AES, 3DES, DES…)
 hash sha256      ! Hashing for integrity (SHA, MD5…)
 authentication pre-share   ! Authentication method
 group 5          ! Diffie-Hellman group (key exchange strength)
 lifetime 86400    ! Time (seconds) before re-keying (default 86400 = 1 day)
```

Then set the **pre-shared key**:

```
crypto isakmp key VPNKEY address 12.0.0.2
```

✅ Meaning: Use the key VPNKEY when talking to peer at 12.0.0.2.

---

**3. IKE Phase 2 – IPSec Transform Set**

Defines **how data will be encrypted** inside the tunnel.

```
crypto ipsec transform-set MYSET esp-aes esp-sha-hmac
```

✅ Meaning: Use AES encryption + SHA for integrity inside IPSec.

---

**4. Crypto Map (Tie It All Together)**

Binds peer, transform-set, and ACL into a **VPN policy**.

```
crypto map MYMAP 10 ipsec-isakmp
 set peer 12.0.0.2        ! Remote peer address
 set transform-set MYSET   ! Use this IPSec transform-set
 match address 101         ! Match traffic defined in ACL 101
```

**5. Apply Crypto Map to Interface**

Apply the crypto policy to the **WAN interface** (the one facing the peer).

```
interface fa0/1
 crypto map MYMAP
```

✅ Now, any traffic that matches ACL 101 and goes through this interface will be encrypted.

---

🔷 **Verification Commands**

- **Check IKE Phase 1 (ISAKMP SA)**

  ```
  show crypto isakmp sa
  ```

  QM_IDLE = success (tunnel is up).

- **Check IKE Phase 2 (IPSec SA)**

  ```
  show crypto ipsec sa
  ```

Shows encaps/decaps counters (encrypted/decrypted packets).

🔷 **Full Example (R1 side)**

```
! Define interesting traffic
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255

! Phase 1 - ISAKMP policy
crypto isakmp policy 10
 encr aes
 hash sha256
 authentication pre-share
 group 5
 lifetime 86400

! Pre-shared key with R2
crypto isakmp key VPNKEY address 12.0.0.2

! Phase 2 - IPSec transform set
crypto ipsec transform-set MYSET esp-aes esp-sha-hmac

! Crypto map (bind everything together)
crypto map MYMAP 10 ipsec-isakmp
 set peer 12.0.0.2
 set transform-set MYSET
 match address 101

! Apply crypto map to WAN interface
interface fa0/1
 crypto map MYMAP
```

🔷 R2 Config (Mirror of R1)

```
! Define interesting traffic
access-list 101 permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255

! Phase 1 - ISAKMP policy
crypto isakmp policy 10
 encr aes
 hash sha256
 authentication pre-share
 group 5
 lifetime 86400

! Pre-shared key with R1
crypto isakmp key VPNKEY address 12.0.0.1

! Phase 2 - IPSec transform set
crypto ipsec transform-set MYSET esp-aes esp-sha-hmac

! Crypto map (bind everything together)
crypto map MYMAP 10 ipsec-isakmp
 set peer 12.0.0.1
 set transform-set MYSET
 match address 101

! Apply crypto map to WAN interface
interface fa0/1
 crypto map MYMAP
```

🔷 **Verification**

From **PC1 (192.168.1.2)**, ping **PC2 (192.168.2.2)**.

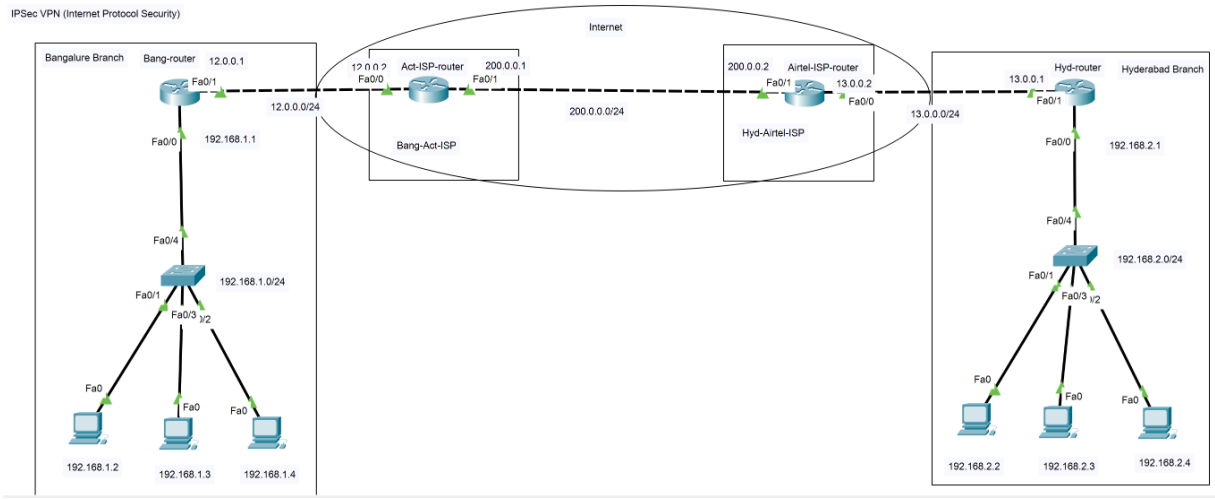
Then on routers:

show crypto isakmp sa ! Check Phase 1 (should show QM_IDLE if up)

show crypto ipsec sa ! Check Phase 2 (should show packet counters increasing)

✅ Now your Site-to-Site IPSec VPN tunnel is complete.

IPSec VPN (Internet Protocol Security)




```
! On Bang-router
enable
configure terminal
host Bang-router
!
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 12.0.0.1 255.255.255.0
no shutdown
exit
!
router ospf 1
router-id 1.1.1.1
network 192.168.1.0 0.0.0.255 area 0
network 12.0.0.0 0.0.0.255 area 0
exit
!
!Configuring acl
access-list 100 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
!
! Configuring ipsec phase 1 isakmp policy
crypto isakmp policy 10
encryption aes
hash sha
group 5
lifetime 86400
exit
!
! setting pre-shared key
crypto isakmp key prasanth.key 13.0.0.1
!
! Configuring phase 2 ipsec transform set
! defines how data will encrypted inside the tunnel
crypto ipsec transform-set MYSET esp-aes esp-sha-hmac
!
! crypto map
! binds peer, transform-set, acl into a vpn policy
crypto map MYMAP 10 ipsec-isakmp
set peer 13.0.0.1          ! remote peer address
set transform-set MYSET        ! use the ipsec transform-set
match address 100      ! match traffic defined acl 100
exit
!
! Apply crypto map to interface
Interface Fa0/1
crypto map MYMAP
exit
!
```

```
! On Act-ISP-router
enable
configure terminal
hostname Act-ISP-router
!
interface Fa0/0
ip address 12.0.0.2 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 200.0.0.1 255.255.255.0
exit
!
router ospf 1
router-id 2.2.2.2
network 12.0.0.0 0.0.0.255 area 0
network 200.0.0.0 0.0.0.255 area 0
exit
!
```

```
! On Airtel-ISP-router
enable
configure terminal
hostname Airtel-ISP-router
!
interface Fa0/1
ip address 200.0.0.2 255.255.255.0
no shutdown
exit
!
interface Fa0/0
ip address 13.0.0.2 255.255.255.0
no shutdown
exit
!
router ospf 1
router-id 3.3.3.3
network 200.0.0.0 0.0.0.255 area 0
network 13.0.0.0 0.0.0.255 area 0
exit
!
```

```
! On Hyd-router
enable
configure terminal
hostname Hyd-router
!
interface Fa0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
!
interface Fa0/1
ip address 13.0.0.1 255.255.255.0
no shutdown
exit
!
router ospf 1
router-id 4.4.4.4
network 192.168.2.0 0.0.0.255 area 0
network 13.0.0.0 0.0.0.255 area 0
exit
!
! configuring acl
access-list 100 permit 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
!
! configuring ipsec phase 1: isakmp policy
crypto isakmp policy 100
encryption aes
hash sha
group 5
lifetime 86400
exit
!
! setting pre-shares key
crypto isakmp key prasanth.key 12.0.0.1
!
! configuring phase 2: ipsec transform set
! defines how data will encrypted inside the tunnel
crypto ipsec transform-set MYSET esp-aes esp-sha-hmac
!
! crypto map
! binds peer, transform-set, acl into a vpn policy
crypto map MYMAP 10 ipsec-isakmp
set peer 12.0.0.1
set transform-set MYSET
match address 100
exit
!
!Apply crypto map to interface
interface fa0/1
crypto map MYMAP
exit
!
```

# VTP (VLAN Trunking Protocol)

**VTP (VLAN Trunking Protocol)** is a Cisco proprietary Layer 2 protocol that helps manage VLANs across a network.
It allows switches to **share and synchronize VLAN configurations** so you don't need to configure VLANs manually on every switch.

---

## 🔑 Key Points of VTP

1. **Purpose** – Simplifies VLAN management by propagating VLAN information (like VLAN ID, name, and state) across switches.
2. **Works on** – Only across trunk links (802.1Q or ISL).
3. **Domain** – All switches must belong to the same **VTP domain name** to share VLAN info.
4. **Advertisement** – VTP uses **multicast frames** to send VLAN updates.

---

## 📌 VTP Modes

1. **Server Mode** (default on Cisco switches)
    - Can create, modify, and delete VLANs.
    - Advertises VLAN info to other switches in the domain.
    - Stores VLANs in **NVRAM**.
2. **Client Mode**
    - Cannot create, delete, or modify VLANs.
    - Only receives VLAN updates from VTP servers.
    - Stores VLANs in **RAM** (not NVRAM → lost after reboot).
3. **Transparent Mode**
    - Does not participate in VTP advertisements.
    - Forwards VTP advertisements through trunk links.
    - VLAN changes made locally stay on that switch only.
    - Stores VLANs in **NVRAM**.

4. **Off Mode (Optional in IOS 15+)**
    - Completely disables VTP on the switch.
    - Neither sends nor forwards VTP advertisements.

---

### 🔐 VTP Security

- Can be configured with a **VTP password** to prevent unauthorized VLAN changes.
- VTP uses **revision numbers** → the highest revision number wins.
  ⚠️ This can be dangerous: if a new switch with a higher revision number and empty VLAN database is connected, it may wipe out VLANs across the network.

---

### ⚙️ VTP Versions

- **VTPv1** – Original version, supports normal VLANs (1–1005).
- **VTPv2** – Adds support for Token Ring VLANs, more stable.
- **VTPv3** – Supports extended VLANs (1006–4094), better authentication, and role-based operation.

---

### 💻 Basic VTP Configuration

Example on a Cisco switch:

```
Switch(config)# vtp domain MyDomain
Switch(config)# vtp mode server
Switch(config)# vtp password Cisco123
Switch(config)# vtp version 2
```

Check status:

```
Switch# show vtp status
```

✅ In short:
VTP is used to centrally manage VLANs in a Cisco switched network. It reduces admin overhead but can also be risky if misconfigured.

🔷 **If a switch is in Server or Client mode:**

- ✅ **Must be in the same VTP domain name** (and same password, if configured)
- Otherwise, they will **not exchange VLAN information**.

✅ **Best practice**: Keep all switches in the **same VTP domain name** for consistency, even if some are Transparent.

---

🔷 **Scenario: Small Company Network with VTP**

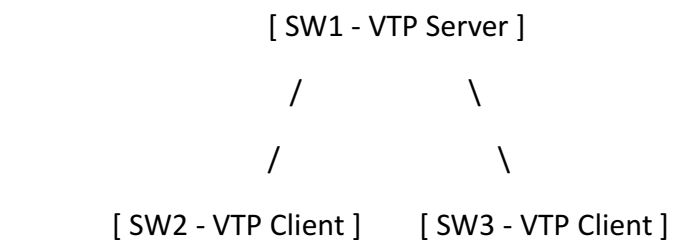
Imagine a company with **3 switches** in one building:

- **SW1 (Core Switch)** → will be the **VTP Server**
- **SW2 (Access Switch 1)** → will be a **VTP Client**
- **SW3 (Access Switch 2)** → will be a **VTP Client**

Goal:

- Configure VLANs **10 (HR)** and **20 (Finance)** only on the **Server (SW1)**.
- Let VLAN info automatically propagate to SW2 and SW3 through VTP.

🖥️ **Topology**




All connections between switches are **trunk links**.

⚙️ **Configuration Steps**

**On SW1 (VTP Server)**

```
SW1> enable
SW1# configure terminal
```

```
SW1(config)# vtp domain COMPANY
SW1(config)# vtp mode server
SW1(config)# vtp password Cisco123
SW1(config)# vtp version 2

! Create VLANs only on Server
SW1(config)# vlan 10
SW1(config-vlan)# name HR
SW1(config-vlan)# exit
SW1(config)# vlan 20
SW1(config-vlan)# name FINANCE
SW1(config-vlan)# exit

! Make trunk to other switches
SW1(config)# interface fa0/1
SW1(config-if)# switchport mode trunk
SW1(config-if)# no shutdown
SW1(config)# interface fa0/2
SW1(config-if)# switchport mode trunk
SW1(config-if)# no shutdown
```

On SW2 (VTP Client):

```
SW2> enable
SW2# configure terminal
SW2(config)# vtp domain COMPANY
SW2(config)# vtp mode client
SW2(config)# vtp password Cisco123
```

```
SW2(config)# vtp version 2


! Trunk towards SW1
SW2(config)# interface fa0/1
SW2(config-if)# switchport mode trunk
SW2(config-if)# no shutdown
```

On SW3 (VTP Client):

```
SW3> enable
SW3# configure terminal
SW3(config)# vtp domain COMPANY
SW3(config)# vtp mode client
SW3(config)# vtp password Cisco123
SW3(config)# vtp version 2


! Trunk towards SW1
SW3(config)# interface fa0/1
SW3(config-if)# switchport mode trunk
SW3(config-if)# no shutdown
```

✅ **Verification**

On SW2 or SW3:

```
SW2# show vlan brief
```

👉 You should see VLAN 10 (HR) and VLAN 20 (FINANCE) even though you **never created them locally**.

On any switch:

```
SW1# show vtp status
```

- Domain Name = COMPANY

- Mode = Server/Client
- Revision Number increases when VLANs are changed

🔍 **Key Learning**

- Only **create VLANs on the Server**.
- Clients **learn automatically**.
- If you add a new VLAN on SW1, it spreads to all clients instantly.

🔷 **Scenario: VTP Transparent in Action**

We'll use **3 switches** again:

- **SW1 → VTP Server** (creates and advertises VLANs)
- **SW2 → VTP Transparent** (does not learn VLANs but forwards advertisements)
- **SW3 → VTP Client** (receives VLANs from SW1 through SW2)

💻 **Topology**

[ SW1 - Server ] --- [ SW2 - Transparent ] --- [ SW3 - Client ]

⚙️ **Configuration Steps**

**On SW1 (VTP Server)**

```
SW1> enable
SW1# configure terminal
SW1(config)# vtp domain COMPANY
SW1(config)# vtp mode server
SW1(config)# vtp password Cisco123
SW1(config)# vtp version 2

! Create VLANs only on Server
```

```
SW1(config)# vlan 10
SW1(config-vlan)# name HR
SW1(config-vlan)# exit
SW1(config)# vlan 20
SW1(config-vlan)# name FINANCE
SW1(config-vlan)# exit

! Trunk link to SW2
SW1(config)# interface fa0/1
SW1(config-if)# switchport mode trunk
SW1(config-if)# no shutdown
```

On SW2 (VTP Transparent):

```
SW2> enable
SW2# configure terminal
SW2(config)# vtp domain COMPANY
SW2(config)# vtp mode transparent
SW2(config)# vtp password Cisco123
SW2(config)# vtp version 2

! Trunk ports
SW2(config)# interface fa0/1
SW2(config-if)# switchport mode trunk
SW2(config-if)# no shutdown
SW2(config)# interface fa0/2
SW2(config-if)# switchport mode trunk
SW2(config-if)# no shutdown
```

On SW3 (VTP Client):

```
SW3> enable
SW3# configure terminal
SW3(config)# vtp domain COMPANY
SW3(config)# vtp mode client
SW3(config)# vtp password Cisco123
SW3(config)# vtp version 2

! Trunk towards SW2
SW3(config)# interface fa0/1
SW3(config-if)# switchport mode trunk
SW3(config-if)# no shutdown
```

✅ **Verification**

On SW3:

```
SW3# show vlan brief
```

👉 You should see **VLAN 10 (HR)** and **VLAN 20 (FINANCE)**, even though they came from **SW1 through SW2 (transparent)**.

On SW2:

```
SW2# show vlan brief
```

👉 You will NOT see VLAN 10/20 (since transparent mode doesn't apply them locally).

On SW2:

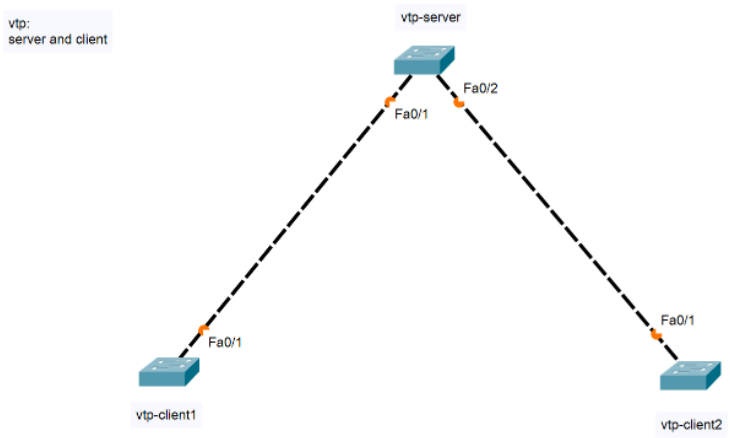
```
SW2# show vtp status
```

👉 Mode = Transparent, Revision number won't change.

## 🔍 Key Learning

- **Transparent switch does not learn VLANs**.
- It **forwards advertisements** to other switches.
- Useful in large networks where you want some switches to act as “pass-through” without modifying VLANs locally.

VTP (VLAN Trunking Protocol)

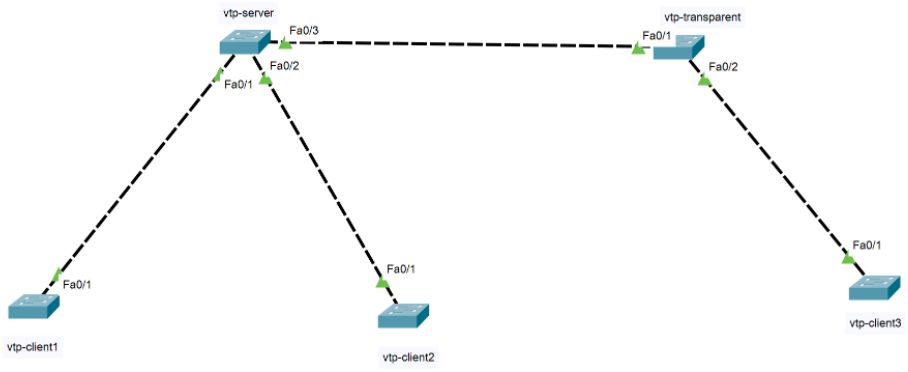



```
! On vtp-server
enbale
configure terminal
host vtp-server
!
! configuring vtp
vtp mode server
vtp domain prasanth
vtp password prasanth@123
!
! configuring vlans
vlan 2
name HR
exit
!
vlan 3
name IT
exit
!
vlan 4
name Finance
exit
!
! Configuring trunk
interface Fa0/1
switchport mode trunk
exit
!
interface Fa0/2
switchport mode trunk
exit
!
```

```
! On vtp-client1
enable
configure terminal
hostname vtp-client1
!
vtp mode client
vtp domain prasanth
vtp password prasanth@123
!
interface Fa0/1
switchport mode trunk
exit
!
```

```
! On vtp-client2
enable
configure terminal
hostname vtp-client2
!
vtp mode client
vtp domain prasanth
vtp password prasanth@123
!
interface Fa0/1
switchport mode trunk
exit
!
```

vtp:
server, client and transperant




```
! On vtp-server
enbale
configure terminal
host vtp-server
!
! configuring vtp
vtp mode server
vtp domain prasanth
vtp password prasanth@123
!
! configuring vlans
vlan 2
name HR
exit
!
vlan 3
name IT
exit
!
vlan 4
name Finance
exit
!
! Configuring trunk
interface Fa0/1
switchport mode trunk
exit
!
interface Fa0/2
switchport mode trunk
exit
!
```

```
! On vtp-client1
enable
configure terminal
hostname vtp-client1
!
vtp mode client
vtp domain prasanth
vtp password prasanth@123
!
interface Fa0/1
switchport mode trunk
exit
!
```

```
! On vtp-client2
enable
configure terminal
hostname vtp-client2
!
vtp mode client
vtp domain prasanth
vtp password prasanth@123
!
interface Fa0/1
switchport mode trunk
exit
!
```

```
! On vtp-client3
enable
configure terminal
hostname vtp-client3
!
vtp mode client
vtp domain prasanth
vtp password prasanth@123
!
interface Fa0/1
switchport mode trunk
exit
!
```

```
! On vtp-transparent
enable
configure terminal
hostname vtp-transparent
!
vtp mode client
vtp domain prasanth
vtp password prasanth@123
!
interface Fa0/1
switchport mode trunk
exit
!
interface Fa0/2
switchport mode trunk
exit
!
```

In Cisco routers (and switches), you can set **different types of passwords** for security and access control.

Here's a step-by-step guide:

---

**1. Console Password (for local access via console cable)**

```
Router> enable
Router# configure terminal
Router(config)# line console 0
Router(config-line)# password cisco123
Router(config-line)# login
Router(config-line)# exit
```

✅ Now, whenever someone connects through the console, they'll be asked for the password.

---

**2. VTY (Telnet/SSH) Password (for remote access)**

```
Router(config)# line vty 0 4
Router(config-line)# password remote123
Router(config-line)# login
Router(config-line)# exit
```

✅ Sets password for remote connections (Telnet/SSH).

---

**3. Enable Password (for privileged EXEC mode)**

```
Router(config)# enable password cisco123
```

⚠️ But enable password is stored in plain text (not secure).

---

**4. Enable Secret (recommended – encrypted)**

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

✅ This overrides enable password and is encrypted in the config file.

---

**5. Encrypt All Passwords**

```
Router(config)# service password-encryption
```

✅ Encrypts all plain-text passwords (console, vty, etc.) in the configuration.

---

🔒 **Best Practice**

- Always use enable secret instead of enable password.
- Use service password-encryption.
- Prefer SSH over Telnet for security.

Let's do it step by step for **SSH login with username + password** (recommended in real networks):

---

🔷 **Step 1: Set a Domain Name (needed for SSH key generation)**

```
Router(config)# ip domain-name mynetwork.com
```

🔷 **Step 2: Create a Local User (with password)**

```
Router(config)# username admin privilege 15 secret MyStrongPass
```

- privilege 15 → gives full admin rights.
- secret → stores password in encrypted form.

---

🔷 **Step 3: Generate RSA Keys (for SSH)**

```
Router(config)# crypto key generate rsa
How many bits in the modulus [512]: 1024
```

- Recommend 1024 or 2048 for security.

---

🔷 **Step 4: Enable SSH on VTY Lines**

```
Router(config)# line vty 0 4
Router(config-line)# transport input ssh
Router(config-line)# login local
Router(config-line)# exit
```

- login local → uses the local username/password instead of just line password.

---

🔷 **Step 5: (Optional) Restrict Telnet**

```
Router(config)# line vty 0 4
Router(config-line)# transport input ssh   ! only SSH allowed
```

✅ Now you can connect via SSH from your PC:

```
ssh admin@<router-ip>
```

It will ask for the password you set.

Nice 👍 let's verify SSH on your router after configuration.

---

🔎 **Step 1: Check SSH Version**

```
Router# show ip ssh
```

You should see something like:

```
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
```

🔎 Step 2: Check VTY Lines

```
Router# show running-config | section vty
```

You should see:

```
line vty 0 4
 login local
```

```
transport input ssh
```

🔎 Step 3: Test SSH from Router Itself (loopback test)

```
Router# ssh -l admin 192.168.1.1
```

- Replace 192.168.1.1 with the router's interface IP.
- -l admin means login with username admin.
- It will ask for the password you set.

---

🔎 **Step 4: Test from a PC**

From your PC (Command Prompt / Terminal):

```
ssh admin@192.168.1.1
```

✅ If everything is correct, you'll be logged in securely using SSH.

🔷 **Where IOS Files Are Stored**

- **Flash memory** → Stores the IOS image file (.bin).
- **NVRAM** → Stores startup-config.
- **RAM** → Stores running-config and decompresses IOS to run.
- **ROM** → Stores bootstrap and a mini-IOS (ROMMON mode).

🔷 Common Commands for IOS Files

```
Router# show version              ! Shows IOS version
Router# show flash:               ! Lists IOS files in flash memory
Router# show startup-config       ! View startup config
Router# show running-config       ! View running config
Router# copy tftp: flash:         ! Copy IOS from TFTP to flash
Router# boot system flash:filename  ! Set boot IOS image
```

**Cisco Router Booting Sequence**

When you power on a Cisco router/switch, it goes through **5 main steps**:

---

**1. POST (Power-On Self-Test)**

- Hardware check (CPU, memory, interfaces, etc.)
- Runs from **ROM**.
- If hardware is faulty, error messages are displayed.

---

**2. Bootstrap Program**

- Small program stored in **ROM**.
- Its job: **locate and load the IOS image** into RAM.
- Think of it like a BIOS in a PC.

---

**3. Locate and Load IOS**

- Bootstrap checks the **configuration register** (default value = 0x2102).
- It looks for IOS in this order (default):
    1. Flash memory (most common, e.g., flash:c1900-universalk9-mz.SPA.bin)
    2. TFTP server (if configured for network boot)
    3. ROMMON (mini-IOS if everything else fails)
- Once found, the IOS image is **decompressed** and loaded into **RAM**.

---

**4. Load Configuration File**

- Router looks for **startup-config** in **NVRAM**.
- If found → loaded into RAM as **running-config**.
- If NOT found → router enters **setup mode** (initial configuration dialog).

---

**5. Router is Ready**

- IOS is running in RAM.
- Interfaces initialized.
- Router is ready to accept CLI commands.

---

🔷 **Special Cases (ROMMON & Config Register)**

- If the IOS image is missing/corrupted → Router enters **ROMMON mode** (ROM Monitor). Example prompt:

  ```
  rommon 1>
  ```

- Config register values matter:
  - 0x2102 → Normal boot (load IOS from flash, config from NVRAM).
  - 0x2142 → Ignore startup-config (used for password recovery).
  - 0x2101 → Boot into ROMMON mini-IOS.

---

🔷 **Quick Summary**

1. **POST** → Hardware check
2. **Bootstrap** → Run from ROM
3. **IOS** → Load from Flash into RAM
4. **Startup-config** → Load from NVRAM into RAM
5. **Router ready** 🚦

## 🔑 Cisco Router Password Recovery Steps

1. **Connect to the router**
    - Use a console cable + terminal software (PuTTY, Tera Term, HyperTerminal).
2. **Reboot the router**
    - Power cycle it, and during startup press Ctrl + Break (or Ctrl + Pause/Break) to enter **ROMMON mode** (rom monitor).
3. **Change configuration register**
    - At the ROMMON prompt, type:

        rommon 1 > confreg 0x2142

        This tells the router to ignore the startup-config (where the password is saved).
4. **Reboot**
    - Enter:

        rommon 2 > reset

        Router will restart and ignore the saved password.
5. **Enter privileged mode**

- After boot, you'll be in initial setup mode. Type no to skip setup.
- Go to enable mode:

    Router> enable

6. Load the startup configuration into running config

    Router# copy startup-config running-config

Now you have the old configuration (interfaces, routing, etc.) **without the password restriction**.

7. **Change the password**

    Example:

        Router(config)# enable secret NEWPASSWORD

8. **Restore configuration register**

   Set it back to normal:

   ```
   Router(config)# config-register 0x2102
   ```

9. Save the configuration

   ```
   Router# write memory
   ```

   Or

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

10. **Reload**

- Now the router will boot with your new password.

---

✅ After this, you can log in with the new password you configured.

⚠️ Note: This procedure works if you **physically own or have authorized access** to the router. Unauthorized access is illegal.

**1. If You Have Console Access (Local, via Console Cable)**

- Follow the **ROMMON password recovery** steps I gave earlier.
- You can always reset the password even if you forgot it.

---

**2. If You Still Have Privileged Access (via SSH/Telnet)**

- You don't need ROMMON. Just change it directly:

```
Router> enable

Router# configure terminal

Router(config)# enable secret NEWPASSWORD

Router(config)# line vty 0 4

Router(config-line)# password NEWVTYPASS

Router(config-line)# login

Router(config-line)# exit
```

```
Router(config)# line console 0
Router(config-line)# password NEWCONPASS
Router(config-line)# login
Router(config-line)# exit
Router(config)# end
Router# write memory
```

✅ This changes:

- **Enable password** (for enable mode).
- **VTY password** (for Telnet/SSH).
- **Console password** (for local console login).

---

**3. If You Don’t Have Any Access**

- Only solution is **password recovery via ROMMON** (needs console cable + physical access).

Backing up a Cisco router means saving the **running configuration** or **startup configuration** to an external server (TFTP/FTP) or locally.

Here are the main methods:

---

### 🔷 1. Backup to TFTP Server

Most common in real labs/production.

1. Make sure you have a **TFTP server** running on your PC (for example, using tftpd32).
2. Verify connectivity (ping from router to your PC's IP).
3. Use the command:

   Router# copy running-config tftp

   Or

   Router# copy startup-config tftp

4. The router will ask:
   Address or name of remote host []? 192.168.1.100 ← your PC/TFTP server
   Destination filename [running-config]? backup.cfg
5. After confirmation, the file will be copied to your PC.

### 🔷 2. Backup to FTP Server

Router# copy running-config ftp

Then provide:

- FTP server IP
- Username & password
- Destination filename

### 🔷 3. Backup Locally in Router's Flash

You can save the config as a file in flash memory:

Router# copy running-config flash:backup.cfg

To view files:

Router# dir flash:

🔷 4. Show & Copy Manually

You can simply display and copy-paste:

Router# show running-config

Router# show startup-config

Then paste the output into a text editor and save as .cfg.

✅ Best Practice: Always back up **startup-config**, since that's what loads when the router reboots.

Router# copy startup-config tftp

the **restore process** (how to bring back your backup into the router).

---

### 🔄 Restoring a Backup Configuration

**1. From TFTP Server**

If you already saved to TFTP:

```
Router# copy tftp running-config
```

Then enter:

```
Address or name of remote host []? 192.168.1.100   ← TFTP server IP
Source filename []? backup.cfg
Destination filename [running-config]?
```

✅ This merges the backup into the current **running-config** (active memory).
If you want it to load after reboot:

```
Router# copy tftp startup-config
```

2. **From FTP Server**

```
Router# copy ftp running-config
```

You'll provide:

- FTP server IP
- Username/password
- Filename

3. **From Flash (local backup file)**

If you had saved config in flash earlier:

```
Router# copy flash:backup.cfg running-config
```

4. **4. Manual Restore**

If you have the backup in a text file:

- Enter **global config mode**:

```
Router# configure terminal
```

Copy-paste the config lines from your text file.

⚠ **Important Notes**

- copy … running-config → applies immediately but not saved permanently.
- copy … startup-config → replaces saved config (loads on reboot).
- Best practice:

  Router# copy tftp running-config

  Router# copy running-config startup-config

🔷 Cisco Router Backup & Restore Workflow