🔥 Disaster Recovery Planning (DR)

Rebuilding the operations or infrastructures **after** the disaster has passed.
Focus on **availability (24/7)**. Must be **documented**.

📁 Types of DR

- **Virtualized**
- **Network**
- **Cloud**
- **Data**

🎯 Goals of Disaster Recovery

- **Risk Reduction**
- **Resume Operations:**
    - Identify **critical operations**
    - **Prioritize** essential services
    - Ensure **preservation** of vital assets
    - Plan for **replacement** of damaged/lost resources
- Address **investor** and **industry** concerns

🔄 Business Continuity Plan (BCP)

Keep the organization **running** during periods of **displacement** or **interruption** of normal operations.

🛠️ Methods to Achieve Business Continuity

- **Cold Site**
    - Any physical location **without infrastructure**
- **Hot Site**
    - Has both **infrastructure** and **physical location**
- **Mirrored Site**
    - Has **all facilities**, **backups**
    - Considered the **best option**
    - Also the most **expensive**

✅ Advantages of BCP

- **Reduced Risk**
- **Process Improvements**
- Improved **Organizational Maturity**
- Increased **Reliability** and **Availability**
- **Marketplace Advantages**

IP Addressing & Network Concepts

🔗 Types of Addresses

Type	Layer	Details
Physical Address	Layer 2: Data Link	- Stored in NIC by manufacturer - Permanent, 48 bits
Logical Address	Layer 3: Network	- Consists of NetID + HostID - Depends on protocol - IPv4: 32 bits - IPv6: 128 bits (≈ 2^{128} addresses)
Port Address	Layer 2 (Application Service Port)	- Identifies specific processes on a device
Socket	—	- Combination of IP address and Port number (e.g., 192.168.1.5:80)

🔦 Layer Devices

Layer	Devices
Layer 2	Bridge, Switch
Layer 3	Router

🧩 IP Address Classes

Class	Range	Bit Pattern	Use
A	0 – 127	N H H H	Large networks
B	128 – 191	N N H H	Medium networks
C	192 – 223	N N N H	Small networks
D	224 – 239	—	Multicasting (host or multicast group)
E	240 – 255	—	Reserved for experimental use

📡 Broadcast Addresses

- **Direct Broadcast (2 LANs)**
    - Netbit: specific
    - Hostbit: all **1s**
- **Limited Broadcast (same LAN)**
    - Address: **255.255.255.255**
    - Stays within the network, not routed externally

🌍 IP Address Types

Type	Description
Public	- Must be purchased - Globally routable and visible
Private	- Free to use within an organization - Not publicly routed or announced

📌 **Private IP Ranges:**

Class	Range	No. of Networks
A	10.0.0.0	1
B	172.16.0.0 – 172.31.0.0	16
C	192.168.0.0 – 192.168.255.0	256

🧮 **Address Components**

Component	Net Bits	Host Bits
Network Address	Written as-is	All **0s**
Broadcast Address	Written as-is	All **1s**
Subnet Mask	All **1s**	All **0s**
Usable IP Range	From Network + 1 to Broadcast - 1	
Usable Hosts per Subnet	$2^n - 2$ (n = number of host bits)	
Actual Networks	2^n (n = number of network bits)	
Prefix	Number of host bits (e.g., /24)	

🧠 VLSM – Variable Length Subnet Masking

Purpose: Allocate IPs more efficiently by subnetting based on device count.

✅ Steps:

1. Rearrange required device counts in **descending** order
2. Apply **subnetting** accordingly
3. Fill the table:

netID	1st Host	Last Host	Broadcast	Subnet Mask

Routing Concepts

🚦 Basic Concepts

Concept	Description
Routing Goals	High bandwidth, low traffic, shortest path, loop-free routes
Default Gateway	Nearest router in a network
Routing Table	Stores only the **best path** to each destination

🔁 Types of Routing

Type	Static Routing	Dynamic Routing
Configuration	Manually entered by administrator	Learned via routing protocols (RIP, OSPF, EIGRP)
Use Case	Small networks	Large/complex networks
Update	Manual	Automatic
Downtime	Higher	Lower

📩 Routing Behavior

Concept	Description
Trigger Update	Only updated routes are sent, not the full table
Direct Delivery	Devices are in the **same network(**source = destination**)**; switching is used,
Indirect Delivery	Devices are in **different networks**; routers and routing tables are needed

🔄 Adaptive Routing

Aspect	Details
Definition	Routes dynamically adjust based on network changes
Advantages	Low latency, minimal traffic, best routes selected frequently
Disadvantages	High memory usage in routers

🧭 Routing Methods (in Adaptive Routing)

Method	Details
Next Hop Routing	Stores only the next hop—not the entire route
Host-Specific Routing	One entry per host ❌ Requires many records, harder to update
Network-Specific Routing	One entry per network ✅ Easier to manage and update

📋 Routing Table Update Methods

Method	Description
Connected	Automatically added if a network is directly connected
Static	Manually configured by administrator
Dynamic	Updated via routing protocols (RIP, OSPF, EIGRP, etc.)

⚙️ Static Routing (Expanded View)

Aspect	Details
✅ **Advantages**	Minimal CPU usage Easy to configure and understand
❌ **Disadvantages**	Manual updates, error-prone, not scalable, high effort required

🛣️ Default Routing

Aspect	Details
Purpose	Route used when no other entries match in the routing table
Use Case	Common in **stub networks** (single path in/out)
Advantage	Reduces routing table size

🔄 Dynamic Routing

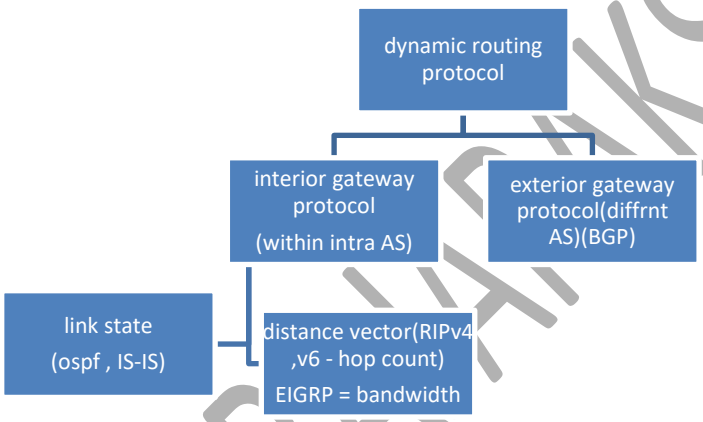
Aspect	Details
Uses Protocols	RIP, OSPF, EIGRP
Table Build Order	1. Connected routes 2. Static routes 3. Dynamically learned routes(via routing protocols)
✅ **Advantages**	Automatically adjusts to changes, less effort, scalable,errors -
❌ **Disadvantages**	Consumes CPU/memory, complex to set up and troubleshoot, need more knowldge

📡 Routing Protocols Overview

Concept	Description
Routing Protocol	A set of rules for routers to share and select best paths (**communicate routing information**)
Key Features	Auto-updates routes, selects best among multiple paths, supports load balancing

- IS-IS (Intermediate System to Intermediate System)

Routing protocol



🏢 **Autonomous System (AS):** A group of routers **under the control of a single organization or administrator**, exchanging routing info using an IGP or EGP

🔄 **Distance Vector vs 🔗 Link State**

Feature	**Distance Vector** (hop count)	**Link State**
Algorithm	Bellman-Ford	Dijkstra (Short path 1st)
Metric	Hop count, speed, delay, throughput	Qualitative link state info (e.g., bandwidth, state of link)
Network Info Shared	Periodically shares **entire routing table** with neighbors	Shares **link-state information** only on changes
Routing Updates	Periodic (every 30s in RIP)	Triggered by topology changes
Best For	Small networks	Large and complex networks
Convergence	Slower	Faster

📡 RIP (Routing Information Protocol)

Feature	RIP Details
Protocol Type	Distance Vector
Algorithm	Bellman-Ford
Metric	Hop count (Max: 15; 16 = Unreachable)
Direction Info	Next-hop (exit interface)
Bandwidth Consideration	❌ Not considered
Routing Table Updates	Every **30 seconds**
Administrative Distance (AD)	120 (Lower AD = more preferred)
Timer Values	Update = 30s Invalid = 180s Hold Down = 180s Flush = 240s
Supported Addressing	RIPv4: Classful (no subnet mask) RIPv6: Classless (supports subnet mask)
Multicast IP	RIPv4: 224.0.0.9 RIPv6: FF02::9
Scalability	Limited to **15 routers max** in a path

Switches (Layer 2 Devices)

Topic	Details
Switch Layer	Layer 2 (Data Link Layer) — works with MAC addresses
Ports	Multiple ports, **no serial ports**
Remote Management	Uses **IP address** for remote connection and management

Switch Form Factors

Type	Description
Fixed Configuration	- Cannot physically modify - Low port density - Cannot add/remove slots - Cheaper
Modular Configuration	- More expensive but scalable - Can increase port density - Flexible with slots
Stackable Configuration	- Connect up to 9 switches with daisy wheel cabling - Huge port density - Expensive - Power supplied to only 1 switch

Factors to Consider When Selecting a Switch

Factor	Explanation
Cost	Depends on port speed, supported features, number of ports
Port Density	Number of devices connected to the network
Power	- POE (Power Over Ethernet): single cable for power + data - Redundant power supply for reliability
Reliability	Should provide **24/7 continuous access**
Port Speed	Ethernet speeds: 10 Mbps (Ethernet), 100 Mbps (Fast Ethernet), 1 Gbps (Gigabit Ethernet)
Scalability	Ability to support future network growth

Switch Functions

Function	Description
Address Learning	- Learns MAC addresses from incoming frames - Known destination = **Unicast** - Unknown destination = **Broadcast** - MAC table initially empty; inactive timer = 0 - Table stores: MAC, VLAN, incoming interface
Forwarding & Filtering	- Forwards frames only to the destination port based on MAC address
Loop Avoidance	- Multiple connections create loops - **STP (Spanning Tree Protocol)** prevents loops while allowing redundancy

Switch Internal Processing Methods

Method	Description
Store and Forward	Receives entire frame, checks error with FCS (Frame Check Sequence), forwards if error-free
Cut Through	Forwards frame immediately **without** checking errors (faster, but risks errors)
Fragment Free	Checks first 64 bits (TCP/IP headers), forwards if no error detected

Switch Security

MAC Address Table (CAM Table)

Topic	Details
- Dynamic MAC Address	Learned dynamically from incoming frames; volatile (lost on reboot)
- Sticky MAC Address	Learned dynamically but stored permanently when saved to startup config; retained after reboot
- Permanent MAC Address	Manually configured; directly connected; timer ~5 minutes

Port Security : Limits the number of valid MAC addresses allowed to transmit on a switch port

Default Allowed MACs : 1 MAC address per port by default

Violation Mode	Description
Protect	Drops unknown MAC address frames; no security notification sent (no SNMP alert).
Restrict	Drops unknown MAC address frames; increments violation count; sends security notification (SNMP alert).
Shutdown	Puts interface into error-disabled state; port LED off; increments violation count; sends security notification; port is disabled.

Violation Situations:

- MAC address not in the MAC address table
- MAC address used on two secure interfaces within the same VLAN

Error-disable Recovery Steps (if Shutdown mode triggered):

1. Disconnect the attacker's cable
2. Issue shutdown command on the interface
3. Issue no shutdown command to bring the port back up

🔹 **VLAN**

🔹 **What is a VLAN?**

- **A VLAN (Virtual LAN) is a logical partition of a switch's network.**
- **Each VLAN acts like a separate broadcast domain.**
- **VLANs are separated by the switch, not physically but logically.**
- **VLANs cannot communicate directly without routing.**
- **Each VLAN maintains its own MAC address table.**

🔹 **Benefits of Using VLANs**

1. ✅ **Improved Security – Limits broadcast traffic and unauthorized access.**
2. 💰 **Reduced Cost – No need for extra hardware like switches/routers.**
3. 🚀 **Better Performance – Less broadcast traffic per VLAN.**
4. 📉 **Smaller Broadcast Domains – Efficient data flow.**
5. 🛠️ **IT Efficiency – Easier management of users and resources.**
6. 👷 **Management Efficiency – Simplified administration.**
7. 📂 **Simpler Project & App Management – VLANs for different teams/projects.**

🔹 **Types of VLANs**

- **Data VLAN: Carries user-generated data (e.g., web traffic).**
- **Default VLAN: All ports are initially part of this VLAN (usually VLAN 1).**
- **Management VLAN: Used for remote switch management (via SSH/Telnet).**

- **VLAN Trunk**
  - **A trunk link is a point-to-point connection (usually between switches or switch–router).**
  - **It carries traffic for multiple VLANs.**
  - **Cisco switches use IEEE 802.1Q protocol for trunking.**

- **VLAN Tagging (Trunking)**
  - **Adds a VLAN ID to Ethernet frames to identify VLAN traffic.**
  - **Two tagging protocols:**
    - **ISL (Inter-Switch Link): Cisco proprietary, encapsulates the whole frame.**
    - **IEEE 802.1Q: Industry standard, inserts a 12-bit VLAN ID in the frame.**

- **VLAN Troubleshooting Commands**

Command	Purpose
interface f0/5	Enters interface configuration mode.
switchport mode trunk	Sets port to trunk mode.
switchport trunk allowed vlan	Specifies which VLANs are allowed on the trunk.
show interfaces trunk	Displays current trunk interfaces.
show vlan brief <id .. | name .. | summary>	Shows VLAN configuration summary.

🌐 Router-on-a-Stick (Inter-VLAN Routing)

🔸 **What is it?**

- A **single router interface** is used to route traffic between VLANs.
- This interface is divided into **subinterfaces**, one for each VLAN.
- Each subinterface is configured with:
    - A **VLAN ID** (via 802.1Q tagging).
    - An **IP address** (acts as the default gateway for that VLAN).

TCP (Transmission Control Protocol)

Topic	Details
Layer	Layer 4 (Transport Layer)
Type	Connection-oriented, reliable communication protocol
Main Services	- Reliable data transfer - Error recovery - Data sequencing - Flow control
Port Numbers	Identify application layer protocols and provide session multiplexing
Port Ranges (IANA)	- 0-1023: Well-known ports - 1024-49151: Registered ports - 49152-65535: Dynamic/Ephemeral/ private
Session	Data exchange between two or more devices; separate sessions based on source/destination ports
Connection Establishment	Uses **Three-Way Handshake**: SYN, SYN-ACK, ACK
Connection Termination	Uses **Four-Way Handshake**: FIN, ACK, FIN, ACK
TCP Header Fields	Source port, Destination port, Sequence number, Acknowledgement number, Flags, Window size
Flag Bits	CWR, ECE, URG, ACK, PSH, RST, SYN, FIN
Sequencing & Acknowledgement	Sequence number tracks data order; ACK confirms received data
Flow Control	Window size indicates how much data receiver can handle; sliding window adjusts dynamically
Retransmission	Lost segments are resent if no ACK received

Three-Way Handshake (Connection Establishment)

Step	Client Sends	Server Responds
1	SYN seq=x	
2		SYN-ACK seq=y, ack=x+1
3	ACK seq=x+1, ack=y+1	

- Three-way handshake

  C SYN → S
  C ← SYN /ACK S
  C ACK → S

- TERMINATION CONNECTION : 4 way handshake

  C FIN → S

  C ← ACK S

  C ← FIN S

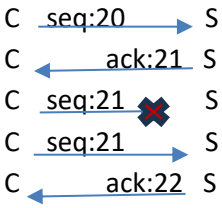
  C ACK → S

- Sequencing / acknowledgement
  - Use :sequence , acknowledgement
  - Host sends random initial sequence number
  - Forward ACK is used to indicate the sequence number of next segment the host expets to receive

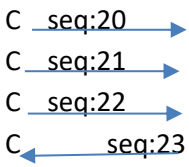
  C seq:10 (3way- SYN) → S
  C ← (3way- SYN /ACK) seq :50 ack:11 S
  C seq :11 ack:51 (3way- ACK) → S
  C ← seq :51 ack:12 S
  C seq :12 ack: 52 → S
  C ← seq :52 ack:13 S

  - Ack = previous sequence received + 1
  - Sequence = prevouse sent ack

- TCP retransmission



- Window seize
  - Acknowledging every single segment , no matter what size is insufficient
  - The TCP header's window field allows more data to be sent before an acknowledgment is required



UDP (User Datagram Protocol)

Feature	**UDP Characteristics**
Connection Type	Connectionless (no connection established)
Reliability	Unreliable, no acknowledgments, no retransmission
Sequencing	No sequencing or ordering of packets
Flow Control	No flow control mechanism
Header Fields	Source port, Destination port, Length, Checksum

TCP Segment Structure (Header)

Field	Size	Description
Source Port	16 bits	Port number of the sender
Destination Port	16 bits	Port number of the receiver
Sequence Number	32 bits	Position of this segment in the byte stream
Acknowledgement Number	32 bits	Next byte expected from the other side
Header Length (HLEN)	4 bits	Length of TCP header in 32-bit words (HLEN × 4 bytes)
Flags	6 bits	URG, ACK, PSH, RST, SYN, FIN
Window Size	16 bits	Amount of data receiver can accept (flow control)
Checksum	16 bits	Error checking for header and data
Urgent Pointer	16 bits	Used only if URG flag is set

- Minimum TCP header size: 20 bytes
- Maximum TCP header size: 60 bytes (with options)

TCP Flags

Flag	Meaning
URG	Data is urgent, prioritize processing immediately
ACK	Acknowledgement field is significant
PSH	Push data immediately to receiving application, no buffering
RST	Reset connection due to errors or invalid requests
SYN	Initiate connection (synchronize sequence numbers)
FIN	Finish/terminate connection

TCP Timers

Timer	Purpose
Retransmission	Retransmit segment if ACK not received
Persistence	Handles zero-window situations (prevents deadlock in flow control)
Keep-alive	Checks if peer is alive after being idle
Time-Wait	Waits to prevent delayed duplicate segments from affecting new connections

TCP Error Control

- **Checksum:** Detects corrupted segments
- **Resending:** Retransmits segments if ACK not received within timeout
- **No NACKs:** Only positive acknowledgments sent (ACKs)
- **Buffers:**
    - Sender buffer holds sent segments until ACK received
    - Receiver buffer holds segments during error checking

TCP Efficiency Example

- Data: 16 bytes
- Headers: Ethernet (20) + IP (20) + TCP (20) + Trailer (4) = 64 bytes
- Total: 80 bytes
- Efficiency = 16 / 80 = 0.2 (20%)

TCP vs UDP Comparison

Feature	TCP	UDP
Reliability	Reliable (ACK, sequencing)	Unreliable (no ACK, no sequencing)
Connection Type	Connection-oriented	Connectionless
Header Size	Minimum 20 bytes	8 bytes
Control	Flow, error, congestion control	No control mechanisms
Common Usage	File transfer, web, email	Streaming, DNS, gaming

TCP Connection States

State	Description
CLOSED	No connection exists
LISTEN	Waiting for incoming connection requests (passive open)
SYN-SENT	SYN sent, waiting for SYN-ACK (active open)
SYN-RCVD	SYN-ACK sent, waiting for ACK (passive open responding)
ESTABLISHED	Connection established, data transfer ongoing
FIN-WAIT-1	FIN sent, waiting for ACK
FIN-WAIT-2	ACK received for FIN, waiting for FIN from peer
CLOSE-WAIT	FIN received, waiting for app to close connection
TIME-WAIT	Waiting for delayed packets to expire (2×MSL)
LAST-ACK	FIN sent, waiting for ACK
CLOSING	Both sides closing simultaneously, waiting for ACK

Some Common UDP Ports

Port Number	Application
69	TFTP (Trivial File Transfer)
53	DNS
161	SNMP
520	RIP

Additional Notes

- URG flag must be 0 if Urgent Pointer is 0 (otherwise inconsistent)
- FIN and ACK flags are usually not sent together initially (except in specific handshake sequences)
- Header Length (HLEN) calculation:
    - Header length in bytes = HLEN × 4
    - Example: If header size = 36 bytes, HLEN = 36 / 4 = 9 (binary: 1001)
    - Standard TCP header size: 20 bytes; Options add extra bytes (up to 40)

ACL (Access Control List) — Summary

What is an ACL?

- **Definition:**
  ACLs are lists of conditions used to filter network traffic passing through router interfaces.
- **Purpose:**
  Control and secure traffic by telling the router which packets to accept or deny.

How ACLs Work

- Packets are checked **in order** against ACL conditions.
- Router takes action (permit or deny) based on the **first matching rule**.
- If no rules match, **implicit deny** (default deny) applies at the end.

Filtering Conditions Can Include:

- Source IP address
- Destination IP address
- Protocols (TCP, UDP, ICMP, etc.)
- Port numbers (e.g., HTTP = 80, HTTPS = 443)

What ACLs Can Do:

- Prevent unwanted or unauthorized traffic
- Block hackers and unauthorized system usage
- Filter routing updates
- Match packets for prioritization (QoS)
- Match packets for VPN tunneling

Wildcard Masking

- Wildcard masks specify which bits in an IP address to check or ignore.
- **0** = check corresponding bit
- **1** = ignore corresponding bit

Types of ACLs

ACL Type	Filters Based On	Number Range
Standard ACL	Source IP only	1–99, 1300–1999
Extended ACL	Source, Destination, Protocol, Port	100–199, 2000–2699

ACL Configuration Guidelines

- One ACL per interface, protocol, and direction (inbound/outbound).
- ACL statements tested in sequence — **order matters!**
- Most restrictive rules should be **first**.
- ACLs have an implicit **deny any** at the end.
- Always include **permit** statements for desired traffic.
- Create ACL before applying to interfaces.
- ACLs filter traffic **passing through** the router, not originating from it.

Applying ACLs to Interfaces

Direction	Description	When Applied
Inbound	Applied before routing decision (more efficient)	Packets filtered immediately upon arrival
Outbound	Applied after routing decision	Packets filtered just before sending out

Named Standard IPv4 ACLs

- Use ip access-list standard <name> to create named ACLs.
- Easier to read and manage than numbered ACLs.
- Use permit and deny inside the list.
- Apply with ip access-group <name> in|out on interface.

Advantages of Named ACLs

- Descriptive, easier to understand.
- Can delete or insert individual entries (sequence numbers).
- Easier to edit than numbered ACLs.

Other ACL Types

- **Dynamic ACLs:** Temporary access after authentication.
- **Time-based ACLs:** Access controlled by date/time.
- **Reflexive ACLs:** Session-aware, allow return traffic only if outbound session exists.
- **Turbo ACLs:** Optimized for performance with large lists.

Best Practices for ACL Placement

- Place ACLs **as close as possible** to the source of traffic to block unwanted traffic early.
- **Extended ACLs:** Near source (filter by source & destination).
- **Standard ACLs:** Near destination (filter only source IP).

Lecture 8 – DHCP and IPv6 Address Assignment

DHCPv4 (Dynamic Host Configuration Protocol for IPv4)

What is DHCPv4?

- DHCPv4 dynamically **leases** IPv4 addresses to clients from a pool for a limited time.
- Lease time is set by the administrator (e.g., 24 hours to a week).
- When lease expires, client requests a new lease (usually gets the same IP).

DHCPv4 Operation (4-Step Process)

1. **DHCP Discover (DHCPDISCOVER)**
   Client broadcasts to find DHCP servers.
2. **DHCP Offer (DHCPOFFER)**
   Server unicasts offer with IP and lease info.
3. **DHCP Request (DHCPREQUEST)**
   Client broadcasts request to accept offer.
4. **DHCP Acknowledgment (DHCPACK)**
   Server unicasts acknowledgment confirming lease.

DHCPv4 Message Format

- Uses **UDP** transport:
  Client → Server: source port 68, destination port 67
  Server → Client: source port 67, destination port 68
- Important fields:
  - **Operation (OP) Code:** 1 = Request, 2 = Reply
  - **Hardware Type:** 1 = Ethernet, 15 = Frame Relay, 20 = Serial
  - **Hardware Address Length:** e.g., 6 for MAC
  - **Hops:** For relay agents (client sets 0)
  - **Transaction ID:** Matches requests & replies
  - **Flags:** 1 means server should broadcast reply
  - **Client IP Address:** Used during renewal

- **Your IP Address:** Assigned IP to client
- **Server IP Address:** DHCP server's IP
- **Gateway IP Address:** For relay agents
- **Client Hardware Address:** Client MAC
- **Server Name & Boot Filename:** Optional
- **DHCP Options:** Config params (subnet mask, DNS, router, etc.)

DHCP Relay

- Used when client & server are on **different subnets**.
- Client broadcasts request → router forwards it to DHCP server.
- Relay agent fills the **gateway IP address** field with its own address.

IPv6 Address Assignment Methods

1. SLAAC (Stateless Address Autoconfiguration)

- Default method on Cisco routers.
- No DHCPv6 server required.
- Uses **ICMPv6 Router Solicitation (RS)** and **Router Advertisement (RA)** messages.
- Clients create IPv6 address using router prefix + interface ID (EUI-64 or random).
- Stateless: no server tracks assigned addresses.

2. DHCPv6 (Stateful)

- Similar to DHCPv4.
- DHCPv6 server assigns full IPv6 address and other config options.

SLAAC Operation Steps

1. Enable IPv6 routing on router:

```
ipv6 unicast-routing
```

2. Client sends **Router Solicitation (RS)** to multicast address FF02::2 (all routers).
3. Router replies with **Router Advertisement (RA)** containing IPv6 prefix and prefix length.
4. Client generates IPv6 address by combining prefix + EUI-64/random interface ID.
5. Client verifies uniqueness with **Neighbor Solicitation** messages.

NAT (Network Address Translation)

Purpose & Function

- Translates **private IPs** to **public IPs** for Internet communication.
- Enables multiple internal devices to share one public IP.
- Hides internal IPs, enhancing network security.
- Typically enabled on the **edge router**.

Key Terminology

Term	**Meaning**
Inside Address	Internal device IP being translated
Outside Address	External destination IP
Local Address	IP visible inside the private network
Global Address	IP visible outside the private network

How NAT Works (Basic Flow)

1. Private IP → Translated to Public IP for outgoing traffic.
2. Server responds to Public IP.
3. NAT router maps response back to original Private IP.

Types of NAT

Type	Description	Use Case
Static NAT	One-to-one IP mapping	Servers needing external access
Dynamic NAT	Maps private IPs to public IP pool	Limited by pool size
PAT (NAT Overload)	Many-to-one using port numbers	Thousands of devices sharing one public IP

PAT Port Usage:

Uses source port or assigns ports from ranges:

- 0–511
- 512–1023
- 1024–65535

NAT vs PAT Comparison

Feature	NAT	PAT (Port Address Translation)
Mapping	One-to-One	Many-to-One (using ports)
Port Usage	Not used	Essential for session tracking
Protocols	TCP/UDP	TCP/UDP + ICMP

Advantages and disadvantages of NAT

Advantages of NAT	Disadvantages of NAT
Conserves public IPv4 addresses	Can degrade performance due to session tracking overhead
Adds flexibility and consistent internal IP usage	Breaks end-to-end traceability
Makes it easier to change ISPs	Some applications require static IPs and may face issues
Hides internal IP addresses, improving network security	Troubleshooting NAT-related issues can be complex
	Disrupts tunneling protocols and affects TCP performance

🔹 NAT Configuration

⚙️ **Static NAT**

✅ **Configuration Steps:**

1. Configure inside interface:

```
interface fa0/0
ip nat inside
```

2. Configure outside interface:
   ```
   interface fa0/1
   ip nat outside
   ```

3. Map private to public IP:
   ip nat inside source static [local_net] [global_ip]

🔍 **Verification:**

- show ip nat translations
- show ip nat statistics

⚙️ **Dynamic NAT**

✅ **Configuration Steps:**

1. Define NAT pool:

ip nat pool [name] [start_net] [end_net] netmask [netmask]

2. Create an access list to match internal traffic:

3. Link the ACL with the pool:

access-list [number] permit [source_net] [wildcard]

ip nat inside source list [acl number] pool [name]

⚙️ **PAT (Port Address Translation)**

✅ PAT Using Address Pool (Many-to-One):

ip nat inside source list [acl#] pool [pool_name] overload

✅ PAT Using Router Interface IP:

ip nat inside source list 1 interface fa0/1 overload

Port Forwarding (Static PAT)

- Allows external devices to access internal services like web servers or FTP

  ip nat inside source static tcp 192.168.1.10 80 203.0.113.10 80

  //Forwards HTTP (port 80) traffic from public IP to internal web server.

🔸 **4. NAT Troubleshooting**

🛠️ **Useful Commands:**

Command	Purpose
show ip nat translations	View active NAT mappings
show ip nat statistics	View stats on NAT usage
debug ip nat	Real-time NAT operation debugging
clear ip nat statistics	Reset counters

❗ **Common Issues:**

Problem	Cause
No connectivity	No NAT translations present
Interface mismatch	Inside/Outside misconfigured
ACL not working	Wrong IP or wildcard settings
Translations work	Confirmed via show command

🔸 **5. NAT & IPv6 – Is NAT Needed?**

❌ **Not required** for private-to-public address saving like in IPv4
✅ **Used only for IPv6↔IPv4 interoperability**

🔄 **Key IPv6 NAT Techniques:**

Technique	Description
Dual Stack	Runs IPv4 and IPv6 together on devices
Tunneling	Encapsulates IPv6 packets inside IPv4 packets
NAT64	Translates IPv6 ↔ IPv4 traffic

✅ **Summary – Quick Revision**

Topic	Key Points
NAT Purpose	Allows internal (private) devices to access internet
Types	Static (1:1), Dynamic (Pool), PAT (Overload)
Static NAT	One private IP ↔ one public IP
Dynamic NAT	Uses a pool of public IPs with ACL matching
PAT	Multiple private IPs share one public IP via ports
Port Forwarding	For accessing internal servers from external network
IPv6 NAT	Only for compatibility, not address saving
Troubleshooting	Use show, debug, check ACLs and interfaces

🛠️ Cisco CLI Practical Commands Summary

🔹 1. Basic Router Interface Setup & Routing

✅ Interface Configuration

```
Router> enable
Router# configure terminal
Router(config)# interface <interface-id>          # e.g., serial0/1
Router(config-if)# no shutdown                    # Enable interface
Router(config-if)# ip address <IP> <subnet mask>  # e.g., 10.10.10.2 255.255.255.252
Router(config-if)# exit
```

✅ Static Routing

```
Router(config)# ip route <LAN_network> <subnet_mask> <next_hop_IP>
# Example: ip route 192.168.2.0 255.255.255.0 10.10.10.2
```

✅ Default Route

```
Router(config)# ip route 0.0.0.0 0.0.0.0 <next_hop_IP>
```

🔹 2. RIP v2 Routing Protocol

```
Router(config)# router rip
Router(config-router)# version 2
Router(config-router)# no auto-summary
Router(config-router)# network <network_ID>        # e.g., network 10.10.10.0
Router(config-router)# passive-interface <interface>  # Optional  To prevent updates on specific interfaces Router(config-router)# exit
```

🔹 3. Interface Speed, Duplex & Clock Rate

```
Router> enable
Router# configure terminal
Router(config)# interface <interface-id>          # e.g., fa0/0 or s0/0
Router(config-if)# speed <100|1000|auto>
Router(config-if)# duplex <full|half|auto>
Router(config-if)# clock rate 64000               # Only on DCE interface
Router(config-if)# description <optional>
```

🔹 4. Switchport Security & VLAN Configuration

```
Switch> enable
Switch# configure terminal
Switch(config)# interface <interface-id>
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 1
Switch(config-if)# switchport port-security mac-address <MAC>
Switch(config-if)# switchport port-security mac-address sticky
```

🔹 **5. IP Assignment on VLAN Interface (SVI)**

```
Switch(config)# interface vlan 99
Switch(config-if)# ip address <IP> <subnet_mask>
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# ip default-gateway <gateway_IP>
Switch(config)# copy running-config startup-config
```

🔹 6. VLAN Trunk Configuration (Switch)

✅ Configure Trunk

```
Switch(config)# interface fa0/1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan 10,20,30,99
Switch(config-if)# exit
```

🔄 Reset Trunk Configuration

```
Switch(config)# interface <interface-id>
Switch(config-if)# no switchport trunk allowed vlan
Switch(config-if)# no switchport trunk native vlan
Switch(config-if)# exit
```

- 7. Router-on-a-Stick (Inter-VLAN Routing)

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

Router(config)# interface g0/0.10
Router(config-subif)# encapsulation dot1q 10
Router(config-subif)# ip address 172.12.10.1 255.255.255.0
Router(config-subif)# exit

Router(config)# interface g0/0.30
Router(config-subif)# encapsulation dot1q 30
Router(config-subif)# ip address 172.12.30.1 255.255.255.0
Router(config-subif)# exit
```

- 8. Delete VLANs Permanently (Switch)

```
Switch# write erase

Switch# delete flash:

Switch# delete vlan.dat      # Confirm when prompted

Switch# reload

Switch# show vlan
```

- VLAN Troubleshooting Commands

Command	**Purpose**
Switch(config)# interface f0/5	Enter interface config mode
Switch(config-if)# switchport mode trunk	Set port to trunk
Switch(config-if)# switchport trunk allowed vlan	Allow specific VLANs
Switch# show interfaces trunk	Show trunk status
Switch# show vlan brief	Show VLAN config summary

- VLAN Creation & Port Assignment

```
Switch# config terminal
Switch(config)# vlan 20
Switch(config-vlan)# name student
Switch(config)# interface f0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 20
Switch(config)# end
```

- Trunk Port Configuration Example

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

Configuration Scenario: Inter-VLAN Routing

📝 **Assumptions:**

- VLAN 10 = Sales
- VLAN 20 = HR
- One router interface (g0/0)
- Subinterfaces for routing between VLANs

🛠 Step 1: Switch Configuration

```
Switch(config)# vlan 10
Switch(config-vlan)# name Sales
Switch(config)# vlan 20
Switch(config-vlan)# name HR

Switch(config)# interface fa0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10

Switch(config)# interface fa0/2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 20

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

🛠 Step 2: Router Configuration (Router-on-a-Stick)

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

Router(config)# interface g0/0.10
Router(config-subif)# encapsulation dot1q 10
Router(config-subif)# ip address 172.17.10.1 255.255.255.0

Router(config)# interface g0/0.20
Router(config-subif)# encapsulation dot1q 20
Router(config-subif)# ip address 172.17.20.1 255.255.255.0
```

- 9. DHCP Server Configuration on Router

```
Router(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9

Router(config)# ip dhcp excluded-address 192.168.10.254


Router(config)# ip dhcp pool LAN-POOL-1

Router(dhcp-config)# network 192.168.10.0 255.255.255.0

Router(dhcp-config)# default-router 192.168.10.1

Router(dhcp-config)# dns-server 192.168.11.5

Router(dhcp-config)# domain-name example.com

Router(dhcp-config)# exit
```

- **10. Router as DHCP Relay Agent**

```
Router(config)# interface g0/0

Router(config-if)# ip helper-address 192.168.17.6

Router(config-if)# exit
```

- 11. Router as DHCP Clien

```
Router(config)# interface fa0/0

Router(config-if)# ip address dhcp

Router(config-if)# no shutdown

Router(config-if)# exit
```

Standard ACL apply :

✅ Format:

access-list <1–99> <permit | deny> <source IP address > <wildcard mask>

✅ Example (Network-Based):

access-list 1 permit 192.168.10.0 0.0.0.255

✅ Example (Host-Based):

access-list 1 deny host 192.168.10.5

access-list 1 permit any

ℹ️ If you deny something, you must **manually permit the rest** (no implicit "permit all").

✅ To Apply on Interface:

interface fa0/0

ip access-group 1 in ← "in" means inbound traffic<in / out>

🔹 **Extended ACL**

✅ Format:

access-list <100–199> <permit | deny> <protocol TCP/IP > <source net> <source wildcard> <destination net> <destination wildcard> <operator> <port/service(ftp(23) http(80))>

✅ **Example:**

access-list 100 permit tcp 192.168.1.0 0.0.0.255 172.16.5.0 0.0.0.255 eq 80

✅ **Operators:**

- eq = equal to

- lt = less than
- gt = greater than
- neq = not equal
- range = between two port numbers

✅ To Apply on Interface:

interface fa0/0

ip access-group 100 out // in or out

🛑 If first rule is deny, you must add:

access-list 100 permit ip any any

🔹 **Named ACL**

✅ **Purpose:**

Easier to read and modify using a custom name.

✅ **Format:**

ip access-list extended <ACL_NAME>

<permit | deny> <protocol> <source> <destination> [eq <port>]

✅ Example:

ip access-list extended hostC-web

permit tcp host 192.168.1.10 any eq 80

permit ip any any

Permit any any

Ip access- group <hostC-web> out

✅ To Apply:

interface fa0/0

ip access-group hostC-web out

📝 **Notes:**

- access in named ACL implies **protocol**, like TCP/IP.
- eq 80 refers to the **HTTP port**.
- To block **Telnet(23)**, use:

deny tcp any any eq 23