**NOTES**

**🔌 What is Networking?**

**Networking** refers to the process of **connecting two or more computing devices** (computers, servers, routers, switches, etc.) to **share resources**, such as files, internet, printers, and data.

These connections can be:

* **Wired** (using Ethernet cables)
* **Wireless** (using Wi-Fi, Bluetooth, etc.)

**🌐 Types of Networks**

**1. LAN (Local Area Network)**

* **Covers**: Small geographical area like a home, school, office, or building.
* **Speed**: High (100 Mbps to 10 Gbps)
* **Example**: Computers in your computer lab connected to a common switch/router.

🔹 **Private ownership**, limited range (~100 meters).

**2. WAN (Wide Area Network)**

* **Covers**: Large geographical areas like countries or continents.
* **Uses**: Routers and public communication links (fiber optics, satellite, etc.)
* **Example**: The **Internet** is the largest WAN.

🔹 **Expensive**, slower than LAN, often managed by ISPs (Internet Service Providers).

**3. MAN (Metropolitan Area Network)**

* **Covers**: A city or a large campus.
* **Size**: Bigger than LAN, smaller than WAN.
* **Example**: A university campus network across different buildings in a city.

🔹 Typically used by government or large companies.

**4. PAN (Personal Area Network)**

* **Covers**: Very small area (around a person).
* **Devices**: Smartphones, laptops, tablets, smartwatches.
* **Example**: Connecting your phone to Bluetooth earphones.

🔹 **Range**: About 10 meters.

**🖥️ Network Architectures**

**1. Client-Server Model**

* One **central server** provides resources/services.
* Other devices (**clients**) request those services.

**🔹 Features:**

* Centralized control.
* Better security and data management.
* Example: Web server and browser; File server in an office.

**2. Peer-to-Peer (P2P) Model**

* Every device (peer) acts as **both client and server**.
* Shares resources **directly** without a central server.

**🔹 Features:**

* Decentralized.
* Easier and cheaper to set up.
* Example: File sharing apps like BitTorrent.

**📡 Network Devices Explained**

**1. Hub**

* 📌 **Function**: Basic device that connects multiple computers in a LAN.
* 📤 **Data Transmission**: Broadcasts data to **all connected devices**, regardless of destination.
* ❌ **No intelligence**: Doesn’t filter data or check destinations.

🔸 Outdated and **less secure** than switches.

**2. Switch**

* 📌 **Function**: Connects multiple devices in a LAN **and intelligently forwards data** only to the destination device.
* 🧠 **More intelligent** than a hub – uses **MAC addresses**.
* ✅ **Efficient, faster**, and more secure.

🔸 Common in modern offices and homes.

**3. Router**

* 📌 **Function**: Connects **different networks** (e.g., your home network to the internet).
* 🌍 Routes data **between networks**, using **IP addresses**.
* 📶 Often includes Wi-Fi and DHCP features.

🔸 Every home internet setup has a router.

**4. Bridge**

* 📌 **Function**: Connects **two similar networks** (e.g., LAN to LAN).
* 🧠 Filters traffic based on **MAC addresses**.
* 🧱 Used to reduce network traffic and divide large networks into segments.

🔸 Example: Connect two buildings’ LANs.

**5. Access Point (AP)**

* 📌 **Function**: Allows **wireless devices** to connect to a **wired network**.
* 📶 Acts like a wireless **hub** or **switch**.
* 📡 Used in Wi-Fi systems (home, office, cafés).

🔸 Works with routers but doesn’t assign IPs on its own.

**6. Modem (Modulator-Demodulator)**

* 📌 **Function**: Converts **digital signals** from your computer into **analog** for transmission over telephone or fiber lines — and **vice versa**.
* 🌐 Enables **Internet access** from an ISP.

🔸 Required for DSL, cable, or fiber internet.

**7. Gateway**

* 📌 **Function**: Acts as a **translator** between **different protocols or networks** (e.g., LAN ↔ Internet, IPv4 ↔ IPv6).
* 🌉 A gateway can be a router **with added protocol conversion** features.

🔸 Used when two networks speak different “languages.”

**8. Repeater**

* 📌 **Function**: **Boosts and regenerates** weak signals to extend network range.
* 🔁 Especially used in **wireless networks** or long-distance wired connections.

🔸 Example: Wi-Fi range extender at home.

Here’s a concise and easy-to-understand explanation of **Cabling and Media** in networking:

**🧵 Cabling and Transmission Media in Networking**

Network communication depends on **transmission media** — the physical pathways for data. These can be **wired** (like Ethernet or fiber) or **wireless** (like Wi-Fi).

**1. Ethernet**

* 📌 A widely used **wired network technology** (IEEE 802.3).
* 🧰 Uses twisted pair cables or fiber optics.
* ⚡ Common speeds:
  + **Fast Ethernet**: 100 Mbps
  + **Gigabit Ethernet**: 1 Gbps
  + **10 Gigabit Ethernet**: 10 Gbps+
* 🔌 Connects computers in a **LAN**.

**2. Fiber Optic Cable**

* 📌 Uses **light signals** to transmit data through glass/plastic fibers.
* 🚀 Very high speed (up to **100 Gbps** or more).
* 🌍 Long-distance communication (up to several kilometers).
* ⚡ **Immune to EMI** (Electromagnetic Interference).

🔸 Used in backbones of ISPs, data centers, and between buildings.

**3. Wireless Standards**

* 📶 Based on **IEEE 802.11** standards (Wi-Fi).

| **Standard** | **Frequency** | **Speed** | **Range (approx.)** | **Notes** |
| --- | --- | --- | --- | --- |
| 802.11b | 2.4 GHz | 11 Mbps | 100–150 ft | Older, more interference |
| 802.11g | 2.4 GHz | 54 Mbps | 100–150 ft | Better than b, still common |
| 802.11n | 2.4/5 GHz | 600 Mbps | 230 ft | MIMO support |
| 802.11ac | 5 GHz | Up to 1.3 Gbps | 115 ft | Fast, used in modern routers |
| 802.11ax (Wi-Fi 6) | 2.4/5 GHz | Up to 10 Gbps | 120 ft | High performance, dense networks |

🔸 Higher frequency = higher speed, but lower range.

**4. UTP vs STP Cables**

| **Feature** | **UTP (Unshielded Twisted Pair)** | **STP (Shielded Twisted Pair)** |
| --- | --- | --- |
| Shielding | ❌ No shielding | ✅ Has metallic shielding |
| Cost | 💸 Cheaper | 💰 More expensive |
| EMI protection | ❌ Low resistance | ✅ High resistance |
| Flexibility | ✅ More flexible | ❌ Less flexible |
| Use-case | Homes, schools | Factories, high-interference areas |
| Example | Cat5, Cat6 cables | FTP, STP Cat6a cables |

Here’s a clear breakdown of the **OSI** and **TCP/IP models** with their layers and key **protocols** used at each layer:

**📦 1. OSI Model (Open Systems Interconnection)**

🔹 Conceptual model with **7 layers**  
🔹 Helps understand how data moves across a network

**🧱 OSI 7 Layers (Top to Bottom):**

| **Layer #** | **Layer Name** | **Function** | **Example Protocols** |
| --- | --- | --- | --- |
| 7 | **Application** | User interface, network services | HTTP, FTP, SMTP, DNS |
| 6 | **Presentation** | Data format (encryption, compression) | SSL/TLS, JPEG, ASCII, MPEG |
| 5 | **Session** | Establish, manage sessions | NetBIOS, PPTP, RPC |
| 4 | **Transport** | End-to-end communication, reliability | TCP, UDP |
| 3 | **Network** | Logical addressing, routing | IP, ICMP, IGMP, IPSec |
| 2 | **Data Link** | MAC addressing, error detection | Ethernet, ARP, PPP, Switch |
| 1 | **Physical** | Raw data transmission (cables, bits) | RJ45, Fiber, Hubs, Repeaters |

**🌐 2. TCP/IP Model (Internet Model)**

🔹 Practical 4-layer model used in real networks  
🔹 Basis of the **Internet**

**📘 TCP/IP 4 Layers:**

| **Layer #** | **Layer Name** | **Corresponds to OSI** | **Example Protocols** |
| --- | --- | --- | --- |
| 4 | **Application** | OSI Layers 5, 6, 7 | HTTP, FTP, SMTP, DNS, DHCP |
| 3 | **Transport** | OSI Layer 4 | TCP, UDP |
| 2 | **Internet** | OSI Layer 3 | IP, ICMP, ARP, IGMP, IPSec |
| 1 | **Network Access** | OSI Layers 1 & 2 | Ethernet, Wi-Fi, Frame Relay |

**🔁 OSI vs TCP/IP: Layer Mapping**

| **OSI Model** | **TCP/IP Model** | **Description** |
| --- | --- | --- |
| 7. Application | 4. Application | Interface for apps to access network |
| 6. Presentation | 4. Application | Data format, encryption |
| 5. Session | 4. Application | Session control |
| 4. Transport | 3. Transport | Reliable delivery, segmentation |
| 3. Network | 2. Internet | Routing, IP addressing |
| 2. Data Link | 1. Network Access | MAC address, framing |
| 1. Physical | 1. Network Access | Cables, signals, bits |

**🧠 Protocols by Layer (Quick Reference)**

**🔸 OSI Layers and Common Protocols**

| **Layer** | **Protocols/Devices** |
| --- | --- |
| Application | HTTP, HTTPS, FTP, SMTP, DNS |
| Presentation | SSL/TLS, JPEG, MPEG, ASCII |
| Session | NetBIOS, RPC, PPTP |
| Transport | TCP, UDP |
| Network | IP, ICMP, IGMP, IPSec |
| Data Link | Ethernet, ARP, PPP, MAC, Switch |
| Physical | Cables, Hubs, NIC, Wi-Fi, Bluetooth |

**🎯 Key Differences: OSI vs TCP/IP**

| **Feature** | **OSI Model** | **TCP/IP Model** |
| --- | --- | --- |
| Layers | 7 | 4 |
| Developed by | ISO | DARPA (U.S. Dept.) |
| Concept vs Use | Theoretical | Practical (used) |
| Protocol Binding | Independent | Protocol-specific |

Here’s a **Networking Cheat Sheet** on **IP Addressing**, covering:  
📌 IPv4 vs IPv6  
📌 Public vs Private IP  
📌 Subnetting  
📌 CIDR

**🌐 IP Addressing Cheat Sheet**

**🧮 1. IPv4 (Internet Protocol v4)**

* 🔢 Format: 32-bit → written as 4 octets (e.g., 192.168.1.1)
* 🌍 Total Addresses: ~4.3 billion
* 📦 Example: 192.168.10.5

**🗂️ IPv4 Classes**

| **Class** | **Range** | **Default Subnet Mask** | **Use** |
| --- | --- | --- | --- |
| A | 1.0.0.0 – 126.255.255.255 | 255.0.0.0 ( /8 ) | Large networks |
| B | 128.0.0.0 – 191.255.255.255 | 255.255.0.0 ( /16 ) | Medium networks |
| C | 192.0.0.0 – 223.255.255.255 | 255.255.255.0 ( /24 ) | Small networks |
| D | 224.0.0.0 – 239.255.255.255 | - | Multicast |
| E | 240.0.0.0 – 255.255.255.255 | - | Experimental/research |

**🌍 2. IPv6 (Internet Protocol v6)**

* 🔢 Format: 128-bit → 8 groups of hexadecimal (e.g., 2001:0db8:85a3::8a2e:0370:7334)
* 🌐 Total Addresses: 340 undecillion+
* 💡 Solves IPv4 exhaustion
* ✅ Built-in security (IPSec), no NAT needed

**🏠 3. Public vs Private IP**

| **Type** | **Description** | **IPv4 Ranges** |
| --- | --- | --- |
| Private | Used inside local networks (not routable on Internet) | - 10.0.0.0 – 10.255.255.255 - 172.16.0.0 – 172.31.255.255 - 192.168.0.0 – 192.168.255.255 |
| Public | Routable on the Internet (unique) | All others not reserved as private |

🌐 **Private IP** → Router assigns  
🌍 **Public IP** → Provided by ISP

**📏 4. Subnetting**

* 📌 Divides a large network into smaller sub-networks.
* ⚡ Improves security, performance, and management.
* 🧠 Uses **Subnet Mask** to define network vs host parts.

**🧮 Example:**

* IP: 192.168.1.10
* Subnet Mask: 255.255.255.0  
  → Network = 192.168.1.0  
  → Host range = 192.168.1.1 – 192.168.1.254
* Here's a **short, clear explanation of subnetting** with just enough detail to understand how it works, including examples.
* **📚 What is Subnetting?**
* **Subnetting** is the process of dividing a **large network** into **smaller, manageable sub-networks** (called **subnets**).
* 🔹 **Why Subnet?**
* Efficient use of IP addresses
* Improves security and performance
* Reduces broadcast traffic
* Easier network management
* **🔧 How It Works**
* Each IP address has two parts:
* **Network part**: Identifies the network
* **Host part**: Identifies the device within that network
* We use a **Subnet Mask** to determine how many bits are for the network and how many for hosts.
* **🧮 Subnet Mask & CIDR Notation**

| * **CIDR** | * **Subnet Mask** | * **# Subnets** | * **Hosts/Subnet (usable)** |
| --- | --- | --- | --- |
| * /24 | * 255.255.255.0 | * 1 | * 254 |
| * /25 | * 255.255.255.128 | * 2 | * 126 |
| * /26 | * 255.255.255.192 | * 4 | * 62 |
| * /27 | * 255.255.255.224 | * 8 | * 30 |
| * /28 | * 255.255.255.240 | * 16 | * 14 |
| * /29 | * 255.255.255.248 | * 32 | * 6 |
| * /30 | * 255.255.255.252 | * 64 | * 2 |

* 📝 **Formula to calculate usable hosts**:  
  2^(32 - subnet bits) - 2
* **🧾 Example:**
* Suppose IP: 192.168.1.0/26
* Subnet mask: 255.255.255.192
* Subnet bits: 26 (→ 6 bits for hosts)
* Hosts per subnet: 2^6 = 64 → usable: 64 - 2 = 62
* Subnets: 4 (because /26 creates 4 subnets in a /24 block)
* **Subnets created:**

| * **Subnet #** | * **Network Address** | * **First Host** | * **Last Host** | * **Broadcast Address** |
| --- | --- | --- | --- | --- |
| * 1 | * 192.168.1.0 | * .1 | * .62 | * .63 |
| * 2 | * 192.168.1.64 | * .65 | * .126 | * .127 |
| * 3 | * 192.168.1.128 | * .129 | * .190 | * .191 |
| * 4 | * 192.168.1.192 | * .193 | * .254 | * .255 |

**🧮 5. CIDR (Classless Inter-Domain Routing)**

* 📌 Replaces class-based system with flexible subnetting.
* 📎 Format: IP/prefix
  + e.g., 192.168.1.0/24 means:
    - 24 bits for network
    - 8 bits for hosts → 256 addresses (254 usable)

**🧾 CIDR Table (Common Prefixes)**

| **CIDR** | **Subnet Mask** | **# Hosts (usable)** |
| --- | --- | --- |
| /8 | 255.0.0.0 | 16 million+ |
| /16 | 255.255.0.0 | 65,534 |
| /24 | 255.255.255.0 | 254 |
| /30 | 255.255.255.252 | 2 |
| /32 | 255.255.255.255 | 1 (single IP) |

Here’s a clear and concise explanation of **MAC vs IP Address** and the **ARP Protocol**, perfect for quick learning or revision:

**🆔 MAC Address vs IP Address**

| **Feature** | **MAC Address** | **IP Address** |
| --- | --- | --- |
| 📌 Full Form | Media Access Control Address | Internet Protocol Address |
| 🧠 Type | **Physical / Hardware Address** | **Logical / Software Address** |
| 🔗 Layer | Data Link Layer (Layer 2, OSI) | Network Layer (Layer 3, OSI) |
| 🎯 Purpose | Identifies **device** on local network | Identifies **device location** on network |
| 📃 Format | Hexadecimal (e.g., 00:1A:2B:3C:4D:5E) | Decimal (IPv4: 192.168.1.10) |
| 🔄 Changes? | Fixed in NIC (usually permanent) | Can change (dynamic/static) |
| 🛠 Assigned By | Manufacturer (via NIC) | Admin or ISP (via DHCP or static config) |

🔸 **MAC** = Like your device's fingerprint  
🔸 **IP** = Like your device's address in a city

**🔄 ARP (Address Resolution Protocol)**

**Purpose**:  
👉 ARP is used to **map an IP address to its corresponding MAC address** in a local network.

**🔁 How ARP Works:**

1. Device A wants to send data to IP 192.168.1.5
2. It checks ARP cache — if no MAC found, sends ARP request:

“Who has IP 192.168.1.5? Tell me your MAC.”

1. Device B replies:

“I have that IP. My MAC is 00:1A:2B:3C:4D:5E.”

1. Device A stores this in the **ARP table** and sends data directly to MAC.

**🧠 ARP Table (or Cache)**

* Temporary memory where the device stores IP-to-MAC mappings.
* Avoids repeating ARP requests unnecessarily.

**🔒 ARP Spoofing (Security Note)**

* Attackers can **fake ARP replies** to redirect traffic.
* Can lead to **Man-in-the-Middle (MITM)** attacks.

Here’s a **cheat sheet** on 🔌 **Common Network Protocols and Port Numbers** —

**🌐 Common Protocols & Ports Cheat Sheet**

| **🔢 Port** | **🌍 Protocol** | **🧠 Purpose** | **📶 Transport Layer Protocol** |
| --- | --- | --- | --- |
| 20, 21 | **FTP** | File Transfer Protocol (data/control) | TCP |
| 22 | **SSH** | Secure Shell – remote access & file transfer | TCP |
| 23 | **Telnet** | Remote terminal access (not secure) | TCP |
| 25 | **SMTP** | Simple Mail Transfer Protocol (sending email) | TCP |
| 53 | **DNS** | Domain Name System (resolves domain to IP) | TCP/UDP |
| 67, 68 | **DHCP** | Dynamic Host Configuration Protocol | UDP |
| 69 | **TFTP** | Trivial FTP – simple file transfer | UDP |
| 80 | **HTTP** | HyperText Transfer Protocol (web access) | TCP |
| 110 | **POP3** | Post Office Protocol v3 (receive email) | TCP |
| 123 | **NTP** | Network Time Protocol (clock sync) | UDP |
| 143 | **IMAP** | Internet Message Access Protocol (email sync) | TCP |
| 161/162 | **SNMP** | Simple Network Management Protocol | UDP |
| 443 | **HTTPS** | Secure HTTP (encrypted websites) | TCP |
| 3389 | **RDP** | Remote Desktop Protocol (remote access) | TCP/UDP |

**💡 Transport Layer Reference**

| **Protocol** | **Layer** | **Uses TCP or UDP?** |
| --- | --- | --- |
| HTTP/HTTPS | Application (Layer 7) | TCP |
| DNS | Application (Layer 7) | TCP (zone transfer) / UDP (query) |
| DHCP | Application (Layer 7) | UDP (67: Server, 68: Client) |
| SSH/Telnet | Application (Layer 7) | TCP |
| FTP | Application (Layer 7) | TCP |

**🔐 Secure vs Non-Secure Protocols**

| **Non-Secure** | **Secure Alternative** |
| --- | --- |
| HTTP (80) | HTTPS (443) |
| Telnet (23) | SSH (22) |
| FTP (21) | SFTP or FTPS (22/990) |

**Network Topologies**

**1. Star Topology**

* All devices connect to a central device like a switch or hub.
* If the central device fails, the whole network goes down.
* Easy to manage and troubleshoot.

**2. Bus Topology**

* All devices share one single communication line (backbone).
* Data sent by one device is available to all others.
* If the main cable breaks, the entire network stops working.
* Less common today, mostly in small or legacy networks.

**3. Ring Topology**

* Devices are connected in a circular fashion, each connected to two neighbors.
* Data travels in one direction (or both in some cases).
* Failure of one device can disrupt the whole network unless it’s a dual ring.
* Used in some fiber networks.

**4. Mesh Topology**

* Every device connects directly to every other device.
* Very reliable and redundant but expensive and complex to set up.
* Used in critical systems or large WANs.

Here’s a simple explanation of **basic network commands** you’ll often use for troubleshooting and information gathering:

**🖥️ Basic Network Commands**

**1. ping**

* **Purpose**: Tests connectivity between your computer and another device (server, router, website).
* **How it works**: Sends ICMP “echo request” packets and waits for “echo reply.”
* **Use case**: Check if a device is reachable and measure response time.

**Example:**

ping google.com

**2. tracert (Windows) / traceroute (Linux/macOS)**

* **Purpose**: Traces the path packets take to reach a destination.
* **How it works**: Shows each hop (router) between your PC and the destination.
* **Use case**: Identify where delays or failures occur on the path.

**Example:**

tracert google.com

**3. ipconfig (Windows) / ifconfig or ip (Linux)**

* **Purpose**: Displays your computer’s IP configuration details.
* **Details shown**: IP address, subnet mask, default gateway, DNS servers.
* **Use case**: Check network settings or troubleshoot IP problems.

**Example:**

ipconfig

**4. nslookup**

* **Purpose**: Queries DNS to find the IP address associated with a domain name, or vice versa.
* **Use case**: Troubleshoot DNS issues or verify domain name resolution.

**Example:**

nslookup google.com

**Switching Concepts**

**1. VLAN (Virtual LAN)**

* A VLAN is a logical segmentation of a physical LAN into different broadcast domains.
* Devices in the same VLAN can communicate directly, even if physically connected to different switches.
* VLANs improve security and reduce broadcast traffic by isolating groups of devices.
* Example: VLAN 10 for Sales, VLAN 20 for HR.

**2. Trunking**

* Trunking allows multiple VLANs to be carried over a single physical link between switches.
* Uses tagging protocols like **IEEE 802.1Q** to identify which VLAN a frame belongs to.
* Trunk ports carry traffic for multiple VLANs simultaneously.
* Essential for connecting switches in a VLAN-aware network.

**3. Inter-VLAN Routing**

* VLANs separate broadcast domains, so devices in different VLANs can't communicate directly.
* **Inter-VLAN routing** allows communication between different VLANs.
* Achieved using a router or a Layer 3 switch.
* Router-on-a-stick: A router with one physical interface connected via trunk to the switch, configured with subinterfaces for each VLAN.

**4. STP (Spanning Tree Protocol)**

* A protocol that prevents loops in a Layer 2 switched network.
* Switches send BPDU (Bridge Protocol Data Units) to elect a root bridge and block redundant paths.
* Ensures a loop-free topology by blocking some links while keeping others active.
* Variants include RSTP (Rapid STP) for faster convergence.

**5. VTP (VLAN Trunking Protocol)**

* Cisco proprietary protocol to manage VLAN configuration across multiple switches.
* Allows centralized VLAN management by propagating VLAN information from a VTP server switch to clients.
* Modes: Server (can add/delete VLANs), Client (receive VLAN info), Transparent (ignore VTP updates, but forward them).
* Helps maintain VLAN consistency but can be risky if misconfigured.

**Routing Types**

Routing is the process of selecting paths in a network along which to send data packets. There are several types of routing based on how routes are determined and maintained.

**1. Static Routing**

* **Definition:** Routes manually configured by a network administrator.
* **How it works:** You specify exact routes in the routing table.
* **Characteristics:**
  + Simple to configure in small or stable networks.
  + Does not adapt to topology changes (no automatic updates).
  + Uses minimal resources since no routing protocol overhead.
  + Good for default routes or backup routes.
* **Pros:**
  + Predictable routing behavior.
  + No routing protocol overhead.
  + More secure (no routing updates to intercept).
* **Cons:**
  + Not scalable for large or frequently changing networks.
  + Manual intervention needed on network changes.
* **Use cases:** Small networks, stub networks, security-sensitive segments.
* **Example command (Cisco IOS):**
* ip route [destination\_network] [subnet\_mask] [next\_hop\_ip or exit\_interface]

**2. Dynamic Routing**

* **Definition:** Routers automatically discover and maintain routing information by exchanging routing updates using routing protocols.
* **How it works:** Routers send and receive routing updates to learn network topology dynamically.
* **Characteristics:**
  + Automatically adapts to network topology changes.
  + Supports large, complex networks.
  + Uses more CPU, memory, and bandwidth than static routing.
* **Main Dynamic Routing Protocol Types:**

**a. Distance Vector Routing Protocols**

* Routers share routing tables with their neighbors periodically.
* Use metrics like hop count to choose best path.
* Count-to-infinity problem can cause slow convergence.
* Examples:
  + **RIP (Routing Information Protocol)**
* **Characteristics:**
  + Simple, easy to configure.
  + Uses periodic updates.
  + Limited scalability (RIP max 15 hops).
  + Slower convergence (especially RIP).

**b. Link-State Routing Protocols**

* Routers build a complete map of the network topology by exchanging Link State Advertisements (LSAs).
* Use algorithms like Dijkstra’s shortest path first to compute best paths.
* Faster convergence and scalable.
* Examples:
  + **OSPF (Open Shortest Path First)**
  + **IS-IS (Intermediate System to Intermediate System)**
* **Characteristics:**
  + More complex but efficient.
  + Requires more CPU and memory.
  + Supports hierarchical design with areas.
  + Sends incremental updates when topology changes.

**c. Hybrid Routing Protocols**

* Combine features of distance vector and link-state.
* Example:
  + **EIGRP** (uses distance vector for routing updates, but maintains topology table and uses DUAL algorithm for loop-free paths).
* **Characteristics:**
  + Fast convergence.
  + Supports unequal-cost load balancing.
  + Cisco proprietary.

**3. Default Routing**

* A special static route used to send packets when no specific route exists in the routing table.
* Often used in small networks or as a route to the internet.
* Configuration example:
* ip route 0.0.0.0 0.0.0.0 [next-hop IP or interface]

**4. Summary of Routing Protocols**

| **Routing Type** | **Examples** | **Metric** | **Update Method** | **Convergence Speed** | **Scalability** | **Classful/Classless** |
| --- | --- | --- | --- | --- | --- | --- |
| Static Routing | Manual config | N/A | N/A | N/A | Low | Classless |
| Distance Vector | RIP, EIGRP | Hop count (RIP), bandwidth/delay (EIGRP) | Periodic routing updates | Slow (RIP), Fast (EIGRP) | Small (RIP), Medium (EIGRP) | RIP: Classful (v1), Classless (v2); EIGRP: Classless |
| Link-State | OSPF, IS-IS | Cost (bandwidth) | Event-driven updates | Fast | Large | Classless |
| Hybrid | EIGRP | Composite metric | Event-driven | Fast | Medium to Large | Classless |

**Routing Concepts**

**1. Static Routing**

* Manually configured routes by network administrator.
* Simple and predictable but doesn’t adapt to network changes automatically.
* Used in small or stable networks or for default routes.
* Example command (Cisco IOS):
* ip route 192.168.2.0 255.255.255.0 10.0.0.2

**2. RIP (Routing Information Protocol)**

* One of the oldest distance-vector routing protocols.
* Uses hop count as metric (max 15 hops).
* Broadcasts routing updates every 30 seconds.
* Simple but not scalable for large networks.
* Versions: RIPv1 (classful), RIPv2 (classless, supports subnet masks).
* Slower convergence compared to newer protocols.

**3. OSPF (Open Shortest Path First)**

* Link-state routing protocol widely used in enterprise networks.
* Uses cost metric based on bandwidth.
* Faster convergence than RIP.
* Organizes networks into **areas** to optimize routing and reduce overhead.
* Sends Link State Advertisements (LSAs) to maintain a database of network topology.
* Supports VLSM and CIDR.

**4. EIGRP (Enhanced Interior Gateway Routing Protocol)**

* Cisco proprietary advanced distance-vector routing protocol (sometimes called hybrid).
* Uses metrics based on bandwidth, delay, load, reliability.
* Fast convergence using Diffusing Update Algorithm (DUAL).
* Supports VLSM, summarization, and unequal-cost load balancing.
* More efficient and scalable than RIP.

**5. Summarization (Route Aggregation)**

* Combines multiple IP routes into a single summary route.
* Reduces size of routing tables and improves routing efficiency.
* Example: Summarize 192.168.1.0/24 and 192.168.2.0/24 into 192.168.0.0/22.
* Supported by dynamic routing protocols like OSPF and EIGRP.
* Helps reduce routing updates and speeds convergence.

**Access Control Lists (ACLs)**

ACLs are used on routers and switches to filter traffic based on rules. They control which packets are allowed or denied through an interface based on criteria like source/destination IP, protocol, ports, etc.

**1. Standard ACLs**

* **Filter traffic based only on the source IP address.**
* Simple and less granular.
* Can only permit or deny traffic by source IP.
* Applied **close to the destination** to avoid blocking legitimate traffic.
* Range: 1-99 and 1300-1999 (expanded range).
* **Example:**
* access-list 10 permit 192.168.1.0 0.0.0.255

This allows all packets with source IP in 192.168.1.0/24.

**2. Extended ACLs**

* **Filter traffic based on multiple criteria:**
  + Source IP
  + Destination IP
  + Protocol (TCP, UDP, ICMP, etc.)
  + Source and destination ports (e.g., HTTP, FTP)
* More granular and flexible.
* Applied **close to the source** to block unwanted traffic early.
* Range: 100-199 and 2000-2699.
* **Example:**
* access-list 110 permit tcp 192.168.1.0 0.0.0.255 10.0.0.0 0.0.0.255 eq 80

This permits TCP traffic from 192.168.1.0/24 to 10.0.0.0/24 destined to port 80 (HTTP).

**3. Wildcard Masks**

* Wildcard mask is used with ACLs to specify which bits of an IP address to **ignore** in matching.
* It is the inverse of a subnet mask.
* **Wildcard bit = 0** means “match this bit exactly.”
* **Wildcard bit = 1** means “ignore this bit.”
* Format example: 0.0.0.255 means match first 24 bits exactly, ignore last 8 bits.
* **How to calculate wildcard mask:**
  + Wildcard mask = 255.255.255.255 - subnet mask
  + Example: Subnet mask 255.255.255.0 → Wildcard mask 0.0.0.255
* **Example use:**
* access-list 10 permit 192.168.1.0 0.0.0.255

Matches all IPs from 192.168.1.0 to 192.168.1.255.

**Summary Table**

| **ACL Type** | **Filter Criteria** | **Range** | **Applied Close To** | **Use Case** |
| --- | --- | --- | --- | --- |
| Standard ACL | Source IP only | 1-99,1300-1999 | Destination | Simple IP-based filtering |
| Extended ACL | Source IP, Destination IP, Protocol, Ports | 100-199,2000-2699 | Source | Granular traffic control |

**NAT (Network Address Translation)**

* **Purpose:** Translates private (internal) IP addresses to public (external) IP addresses so devices inside a private network can communicate with the internet.
* **Why needed:** Private IP addresses (e.g., 192.168.x.x, 10.x.x.x) are not routable on the internet.
* **Types of NAT:**
  1. **Static NAT:**
     + One-to-one mapping between a private IP and a public IP.
     + Used when a device inside needs to be reachable from outside (e.g., a web server).
  2. **Dynamic NAT:**
     + Many private IPs mapped to a pool of public IPs.
     + One-to-one mapping but from a pool; once pool exhausted, no new translations.
  3. **NAT Overload (PAT):**
     + Many private IPs share a **single public IP** by differentiating connections with port numbers.
     + Also called **Port Address Translation (PAT)** or **NAT overload**.
     + Most commonly used form of NAT on home and office routers.

**PAT (Port Address Translation)**

* A subtype of NAT that maps multiple private IP addresses to a single public IP address by using **different source port numbers**.
* Allows thousands of devices to share a single public IP.
* Translates source IP **and** source port.
* Example:
  + Private IP 192.168.1.10:3456 → Public IP 203.0.113.5:1025
  + Private IP 192.168.1.11:4000 → Public IP 203.0.113.5:1026
* Enables many simultaneous connections with one public IP.

**Summary Table**

| **Feature** | **NAT (General)** | **PAT (NAT Overload)** |
| --- | --- | --- |
| IP Address Mapping | Private IP → Public IP | Multiple private IPs → Single public IP + port |
| Number of Public IPs Needed | One or more | Usually one |
| Port Translation | No (except PAT) | Yes |
| Use Case | Servers needing fixed public IPs | Most home/office internet access |

**Example Cisco Configuration for PAT:**

ip nat inside source list 1 interface GigabitEthernet0/0 overload

access-list 1 permit 192.168.1.0 0.0.0.255

* This config allows all devices in 192.168.1.0/24 to share the public IP of GigabitEthernet0/0 interface using port translation.

**Wireless Networking Concepts**

**1. Encryption in Wireless Networks**

Wireless networks are vulnerable to unauthorized access and eavesdropping because they broadcast data over the air. Encryption protects the data by scrambling it so only authorized devices can read it.

* **WEP (Wired Equivalent Privacy)**
  + Oldest and weakest encryption.
  + Uses a static key; vulnerable to attacks.
  + Not recommended for modern networks.
* **WPA (Wi-Fi Protected Access)**
  + Improvement over WEP.
  + Uses TKIP (Temporal Key Integrity Protocol).
  + More secure but now considered outdated.
* **WPA2**
  + Uses AES (Advanced Encryption Standard), stronger and more secure.
  + Most common today.
  + Supports Personal (PSK) and Enterprise (RADIUS) modes.
* **WPA3**
  + Latest standard.
  + Improved security even on open networks.
  + More resistant to brute force attacks.

**2. Wireless LAN Controller (WLC)**

* A centralized device that manages multiple Access Points (APs) in a wireless network.
* Key functions:
  + Centralized configuration and management of APs.
  + Load balancing clients across APs.
  + Enforces security policies.
  + Manages roaming between APs seamlessly.
  + Firmware updates and troubleshooting.
* Common in enterprise environments with many APs to simplify administration.

**3. Access Points (APs)**

* Devices that provide wireless connectivity to clients (laptops, phones, etc.).
* Connect wireless clients to the wired network.
* Types:
  + **Standalone APs:** Configured and managed individually.
  + **Lightweight APs:** Managed by a WLC, rely on the controller for configuration.
* Functions:
  + Transmit and receive wireless signals.
  + Authenticate and encrypt wireless clients.
  + Support multiple wireless standards (802.11a/b/g/n/ac/ax).

**Summary Table**

| **Component** | **Role** | **Key Points** |
| --- | --- | --- |
| Encryption | Secures wireless data | WEP (weak), WPA, WPA2 (AES), WPA3 (best) |
| Wireless LAN Controller (WLC) | Central management of APs | Central config, security, roaming, load balancing |
| Access Points (APs) | Provide wireless access to clients | Connect clients to network, standalone or lightweight |

**WAN Technologies**

**1. PPP (Point-to-Point Protocol)**

* A data link layer protocol used to establish a direct connection between two nodes.
* Provides authentication, encryption, and compression.
* Commonly used for dial-up connections, leased lines, and some broadband.
* Supports multiple network layer protocols like IP.
* Features include:
  + Link establishment and termination.
  + Authentication protocols: PAP, CHAP.
  + Encapsulation of network layer packets.

**2. MPLS (Multiprotocol Label Switching)**

* A high-performance WAN technology that directs data from one node to the next based on short path labels rather than long network addresses.
* Combines the speed of switching with the scalability of routing.
* Supports multiple protocols (IP, ATM, Ethernet).
* Enables efficient traffic engineering and QoS (Quality of Service).
* Widely used by ISPs to create VPNs, manage bandwidth, and improve reliability.

**3. DSL (Digital Subscriber Line)**

* A technology for high-speed internet access over traditional copper telephone lines.
* Provides broadband connectivity without interfering with voice calls.
* Types include:
  + ADSL (Asymmetric DSL) – higher download speeds than upload.
  + SDSL (Symmetric DSL) – equal download and upload speeds.
* Commonly used for home and small business internet connections.

**4. VPN (Virtual Private Network)**

* A technology that creates a secure, encrypted tunnel over a public or shared network (usually the internet).
* Used to securely connect remote users or sites to a private network.
* Types:
  + Site-to-site VPN: Connects entire networks.
  + Remote-access VPN: Connects individual users.
* Protocols include:
  + IPsec (Internet Protocol Security)
  + SSL/TLS (Secure Sockets Layer / Transport Layer Security)
  + MPLS VPNs for enterprise WANs.

**Summary Table**

| **Technology** | **Layer/Type** | **Purpose** | **Typical Use Case** |
| --- | --- | --- | --- |
| PPP | Data Link Protocol | Point-to-point link setup, authentication | Dial-up, leased lines |
| MPLS | Layer 2.5 (Label Switching) | Efficient, scalable WAN transport | ISP backbones, VPNs, traffic engineering |
| DSL | Physical & Data Link | Broadband internet over copper lines | Home/business internet access |
| VPN | Network/Transport | Secure communication over public networks | Secure remote access, site-to-site connectivity |

**Network Security Concepts**

**1. Firewalls**

* **What is a Firewall?**  
  A security device (hardware or software) that monitors and controls incoming and outgoing network traffic based on predetermined security rules.
* **Purpose:**  
  To protect a network by allowing legitimate traffic and blocking unauthorized or malicious traffic.
* **Types of Firewalls:**
  + **Packet Filtering Firewall:**  
    Filters packets based on source/destination IP, ports, and protocol. Works at Layer 3 and 4.
  + **Stateful Inspection Firewall:**  
    Tracks the state of active connections and makes decisions based on connection state and packet content.
  + **Proxy Firewall:**  
    Acts as an intermediary between users and the internet, filtering requests and responses.
  + **Next-Generation Firewall (NGFW):**  
    Combines traditional firewall features with deep packet inspection, intrusion prevention, and application-level filtering.
* **Common Features:**
  + Access control (permit/deny rules)
  + NAT (Network Address Translation)
  + VPN support
  + Logging and alerting

**2. DHCP Snooping**

* **What is DHCP Snooping?**  
  A security feature on switches that monitors DHCP traffic and filters untrusted DHCP messages to prevent attacks like rogue DHCP servers.
* **Why is it important?**  
  Prevents unauthorized DHCP servers from giving out incorrect IP addresses, which could lead to man-in-the-middle attacks or denial of service.
* **How it works:**
  + Switch ports are classified as **trusted** or **untrusted**.
  + Trusted ports (usually uplinks to DHCP servers) can send DHCP server messages.
  + Untrusted ports (end-user devices) cannot send DHCP server messages.
  + The switch builds a DHCP snooping binding table mapping MAC addresses, IP addresses, lease times, and ports.
  + Packets that violate DHCP rules are dropped.
* **Benefits:**
  + Prevents IP address spoofing.
  + Protects against rogue DHCP servers.
  + Improves network security and stability.

**Summary Table**

| **Security Feature** | **Purpose** | **Key Functionality** | **Typical Use Case** |
| --- | --- | --- | --- |
| Firewall | Control network traffic | Packet filtering, stateful inspection, proxy | Protect network perimeter |
| DHCP Snooping | Prevent rogue DHCP servers | Filter DHCP messages, build binding table | Secure LAN against DHCP attacks |

**Troubleshooting Commands**

**1. SHOW Commands**

These commands display current device status, configuration, and statistics — very useful for diagnosing issues.

| **Command** | **Purpose** |
| --- | --- |
| show ip interface brief | Quick overview of interfaces and their status (up/down, IP assigned) |
| show interfaces | Detailed info on interfaces, errors, traffic stats |
| show running-config | Displays current active configuration |
| show startup-config | Shows configuration saved in NVRAM (used at boot) |
| show ip route | Displays the routing table |
| show arp | Shows ARP table (IP to MAC address mappings) |
| show mac address-table | Shows MAC addresses learned on switch ports |
| show vlan | Displays VLAN info and port assignments |
| show spanning-tree | Shows STP status and root bridge info |
| show logging | Displays system logs and error messages |
| show ip protocols | Shows routing protocols enabled and their status |
| show users | Displays active user sessions on the device |
| show cdp neighbors | Shows Cisco Discovery Protocol neighbors |
| show version | Shows device hardware, software version, uptime |
| show processes cpu | CPU utilization and processes |
| show ip dhcp binding | DHCP bindings learned on the device |

**2. DEBUG Commands**

Debug commands show real-time, detailed output about system events, protocols, or packet flow. Use with caution — on production devices, they can overload CPU.

| **Command** | **Purpose** |
| --- | --- |
| debug ip packet | Displays packets matching certain criteria (e.g., IP packets) |
| debug ip routing | Shows routing updates received and sent |
| debug arp | Shows ARP requests and replies |
| debug interface | Displays interface-level events |
| debug spanning-tree events | Shows STP state changes |
| debug dhcp detail | Displays DHCP traffic details |
| debug ppp authentication | Shows PPP authentication process |
| debug crypto ipsec | Shows IPSec encryption/decryption events |
| debug vlan | VLAN-related debug info |

**3. Useful Tips**

* Always **use debug commands carefully** on production systems because of the high CPU usage.
* Use undebug all or no debug all to stop all debug outputs.
* Use **show** commands first to gather information before enabling debug.
* Combine debug with **terminal monitor** to view debug output on your console or SSH session.

**Advanced Routing Concepts**

**1. OSPFv3 (Open Shortest Path First version 3)**

* The updated version of OSPF designed to support **IPv6**.
* Similar to OSPFv2 (for IPv4) but with enhancements for IPv6:
  + Uses **link-local addresses** for neighbor communication.
  + Supports multiple instances per link.
  + Authentication is handled by IPv6’s built-in IPsec rather than OSPF itself.
  + Advertises IPv6 prefixes instead of IPv4.
* Uses the same link-state routing principles, Dijkstra’s SPF algorithm, and areas for scalability.
* Key difference: Protocol packet format and addressing updated for IPv6.

**2. BGP (Border Gateway Protocol)**

* The primary routing protocol used on the **Internet** to exchange routing info between autonomous systems (AS).
* A **path-vector protocol** (not distance-vector or link-state).
* Uses **TCP port 179** for reliable communication.
* Supports **policy-based routing** — decisions based on attributes like AS path, origin, local preference.
* Can handle huge routing tables and complex policies.
* Two types of BGP:
  + **External BGP (eBGP):** Between different ASes.
  + **Internal BGP (iBGP):** Within the same AS.
* BGP is slow to converge but very scalable and flexible.

**3. Route Redistribution**

* Process of sharing routes learned from one routing protocol into another.
* Useful in multi-protocol environments, e.g., OSPF and EIGRP running on different parts of a network.
* Helps maintain connectivity across different routing domains.
* Care must be taken to avoid routing loops and routing table bloat.
* Redistribution can be done with filtering and metrics adjustments to control route propagation.
* **Example use case:** Redistribute OSPF routes into EIGRP and vice versa.
* **Cisco example command:**
* router ospf 1
* redistribute eigrp 10 metric 100 subnets
* router eigrp 10
* redistribute ospf 1 metric 10000 100 255 1 1500
* Key points:
  + Use metrics since protocols have different metric systems.
  + Use route-maps or distribute-lists for filtering routes.

**Summary Table**

| **Protocol/Concept** | **Purpose/Use** | **Key Features** |
| --- | --- | --- |
| OSPFv3 | IPv6 routing within an AS | Link-state, SPF, IPv6 support, IPsec auth |
| BGP | Inter-AS routing on the Internet | Path-vector, policy-based, scalable, TCP-based |
| Redistribution | Exchange routes between different routing protocols | Controlled sharing of routes, metrics & filtering needed |

**Redundancy in Networking**

Redundancy ensures **high availability** and **fault tolerance** in networks by providing backup paths or devices if the primary fails.

**1. HSRP (Hot Standby Router Protocol)**

* A Cisco proprietary protocol for **default gateway redundancy**.
* Provides a **virtual IP and MAC address** shared by a group of routers.
* One router is **active** (forwarding traffic), another is **standby** (backup).
* If the active router fails, the standby router takes over seamlessly.
* Helps prevent single point of failure for gateway.

**Key Features:**

* Virtual IP used by hosts as their default gateway.
* Active router handles all traffic.
* Preemption allows higher priority router to become active.

**2. VRRP (Virtual Router Redundancy Protocol)**

* An **open standard** alternative to HSRP.
* Similar operation: group of routers share a **virtual IP**.
* One router is **master**, others are **backup**.
* If master fails, backup takes over.
* Used in multi-vendor environments.

**Key Features:**

* Faster failover compared to HSRP.
* Supports multiple backups.
* Routers exchange advertisements to monitor status.

**3. Load Balancing**

* Distributes network traffic across multiple devices or links to optimize resource use, maximize throughput, and avoid overload.
* Types:
  + **Static Load Balancing:** Manual configuration of traffic distribution.
  + **Dynamic Load Balancing:** Automatically balances traffic based on metrics or algorithms.
* Common Load Balancing methods in routing/switching:
  + **Equal-Cost Multi-Path (ECMP):** Uses multiple paths with the same cost.
  + **Per-packet or per-flow balancing:** Distributes packets or flows across links.
* Benefits:
  + Improved bandwidth utilization.
  + Increased redundancy and fault tolerance.
  + Better performance and reliability.

**Summary Table**

| **Protocol/Concept** | **Type** | **Purpose** | **Key Points** |
| --- | --- | --- | --- |
| HSRP | Cisco proprietary | Default gateway redundancy | Active/standby routers, virtual IP |
| VRRP | Open standard | Default gateway redundancy | Master/backup routers, virtual IP |
| Load Balancing | Traffic distribution | Optimize resource use | ECMP, per-packet/flow balancing |

**Quality of Service (QoS)**

QoS refers to techniques that manage network resources by prioritizing certain types of traffic to ensure reliable and predictable performance.

**1. DSCP (Differentiated Services Code Point)**

* A field in the IP header used for **classifying and managing network traffic**.
* Part of the **Differentiated Services (DiffServ)** architecture.
* Uses 6 bits in the IP header to mark packets with a value (0–63) indicating priority level.
* Routers and switches read DSCP values to apply QoS policies like prioritization or bandwidth guarantees.
* Common DSCP classes: EF (Expedited Forwarding) for voice, AF (Assured Forwarding) for video/data, default (best effort).

**2. Policing**

* A QoS mechanism that **enforces a maximum traffic rate** by **dropping or remarking packets** exceeding a configured rate.
* Works by monitoring traffic and discarding or lowering priority of excess traffic immediately.
* Suitable for **strict rate limiting** (e.g., enforcing ISP contracts).
* Does not buffer or delay traffic; excess packets are lost or re-marked.
* Can cause traffic bursts to be dropped abruptly.

**3. Shaping**

* A QoS technique that **buffers excess traffic and sends it out at a configured rate**, smoothing out bursts.
* Helps to prevent packet loss by **delaying packets** instead of dropping.
* Useful for **controlling traffic flow** to stay within bandwidth limits gracefully.
* Ideal for traffic that can tolerate some delay but should avoid loss (e.g., video streaming).
* Shaping requires buffering and queuing.

**Summary Table**

| **QoS Feature** | **Function** | **Behavior on Excess Traffic** | **Use Case** |
| --- | --- | --- | --- |
| DSCP | Traffic classification & marking | Marks packets for priority | Prioritizing voice, video, data |
| Policing | Enforce max bandwidth by dropping/remarking | Drops or remarks packets immediately | Strict rate limiting |
| Shaping | Smooth traffic by buffering and delaying | Buffers excess traffic, delays sending | Control bursty traffic flow |

**Virtualization**

**1. VRF (Virtual Routing and Forwarding)**

* Creates multiple isolated routing tables on the same router.
* Allows multiple instances of a routing table to coexist without overlapping.
* Used to segregate traffic for different customers or departments over the same physical infrastructure.
* Enables overlapping IP addresses in different VRFs without conflict.

**2. VXLAN (Virtual Extensible LAN)**

* A network virtualization technology to extend Layer 2 networks over Layer 3 infrastructure.
* Encapsulates Ethernet frames inside UDP packets.
* Supports large-scale multi-tenant environments by providing 24-bit segment IDs (up to 16 million VXLAN segments).
* Used in data centers to create virtual networks across physical infrastructure.

**3. GRE (Generic Routing Encapsulation)**

* A tunneling protocol to encapsulate a wide variety of network layer protocols inside virtual point-to-point links.
* Creates a private “tunnel” between routers over an IP network.
* Often used to carry multicast or non-IP traffic over an IP network.
* Simple and widely supported, but no built-in encryption (can be combined with IPsec).

**Automation**

**1. Python**

* Popular scripting language used for network automation.
* Enables device configuration, data collection, and automation workflows using libraries like Netmiko, NAPALM.

**2. Cisco DNA (Digital Network Architecture)**

* Cisco’s intent-based networking platform.
* Provides automation, assurance, and security across enterprise networks.
* Uses APIs, machine learning, and analytics to simplify network management.

**3. APIs (Application Programming Interfaces)**

* Allow programmatic interaction with network devices and software.
* Common APIs in networking include RESTful APIs used to automate configurations and monitoring.

**4. Ansible**

* Open-source automation tool for configuration management and orchestration.
* Uses simple YAML playbooks to define automation tasks.
* Supports network automation via modules for Cisco, Juniper, Arista, and others.

**Monitoring**

**1. SNMP (Simple Network Management Protocol)**

* Protocol to monitor and manage network devices.
* Devices expose data as MIBs (Management Information Bases).
* Network Management Systems (NMS) query devices to collect stats like CPU, interface status.

**2. Syslog**

* Protocol for sending event messages from devices to a central logging server.
* Useful for troubleshooting and auditing.
* Logs can include errors, warnings, info messages from devices.

**3. NetFlow**

* Cisco technology to collect IP traffic flow information.
* Provides detailed visibility of traffic patterns, source/destination IPs, ports, and protocols.
* Used for bandwidth monitoring, anomaly detection, and capacity planning.

**Network Design**

**1. Hierarchical Network Design**

* A modular approach dividing the network into layers for scalability, performance, and ease of management.
* **Layers:**
  + **Core Layer:** High-speed backbone for fast, reliable transport between distribution layers.
  + **Distribution Layer:** Aggregates access layer switches, implements policies (routing, filtering, QoS).
  + **Access Layer:** Connects end devices like PCs, printers; controls user access.
* Benefits:
  + Scalability, redundancy, simplified troubleshooting.
  + Clear separation of functions.

**2. Campus Network Design**

* A network design focused on a localized area (like a university or corporate campus).
* Usually follows hierarchical design principles.
* Includes:
  + **Access Layer:** Connects user devices, supports VLANs.
  + **Distribution Layer:** Policy enforcement, routing between VLANs.
  + **Core Layer:** Fast backbone connecting multiple buildings or distribution switches.
* Often integrates **Power over Ethernet (PoE)**, **Wireless Access Points**, and **Security** features.

**3. WAN Design**

* Connects geographically dispersed sites.
* Can use technologies like MPLS, VPNs, leased lines, DSL.
* Design goals:
  + Reliable and secure site-to-site connectivity.
  + Efficient bandwidth usage.
  + Redundancy and failover mechanisms.
* May include WAN optimization techniques.

**Wireless Enterprise Deployment**

* Designed for large-scale wireless access with many users and high reliability.
* Key components:
  + **Access Points (APs):** Strategically placed for coverage and capacity.
  + **Wireless LAN Controllers (WLCs):** Central management and control of APs.
  + **Security:** WPA2/WPA3, 802.1X authentication, rogue AP detection.
  + **RF Planning:** Site surveys to optimize AP placement, minimize interference.
  + **QoS:** Prioritize voice and video over wireless.
  + **Seamless Roaming:** Users can move across APs without losing connection.
  + **Guest Access:** Segregated networks for visitors.

**ROLES**

**1. Network Security Specialist**

* **Focus:** Protecting network infrastructure from threats, monitoring, and responding to attacks.
* **Key Technologies:**
  + **Firewalls:** Control and filter traffic, enforce security policies (packet filtering, stateful inspection, NGFW).
  + **VPNs (Virtual Private Networks):** Secure remote access and site-to-site connectivity using protocols like IPsec and SSL/TLS.
  + **IDS/IPS (Intrusion Detection/Prevention Systems):** Monitor network traffic for malicious activity and actively block threats.
  + **Penetration Testing (Pen Testing):** Ethical hacking to find and fix vulnerabilities.
  + **Security Tools:** Wireshark, Nmap, Metasploit, Snort, OSSEC, SIEM solutions (Splunk, QRadar).

**2. Cloud Network Engineer**

* **Focus:** Designing, implementing, and managing cloud-based network architectures.
* **Key Technologies:**
  + **AWS VPC (Virtual Private Cloud):** Create isolated cloud networks, subnets, routing, security groups, and VPN gateways.
  + **Azure VNets (Virtual Networks):** Similar to AWS VPC but in Microsoft Azure ecosystem.
  + **Hybrid Networking:** Connecting on-premises networks with cloud environments securely using VPN, Direct Connect, or ExpressRoute.
  + **Cloud Security:** Identity and access management, firewall rules, encryption.
  + **Automation & Monitoring:** Cloud-native tools like CloudWatch, Azure Monitor.

**3. Network Automation / DevNet Engineer**

* **Focus:** Automating network configuration, monitoring, and management to improve efficiency and reduce errors.
* **Key Skills & Tools:**
  + **Python:** Primary scripting language for automation.
  + **Netmiko:** Python library simplifying SSH connections to network devices.
  + **RESTCONF / NETCONF:** Network management protocols for programmatic device control.
  + **Ansible:** Automation platform using YAML playbooks to automate network tasks.
  + **APIs:** Using REST APIs to interact with network devices and controllers.
  + **Cisco DevNet:** Cisco’s developer program offering APIs, sandboxes, and learning resources.