**Networking Devices**

* **Router**
  + Connects different networks and routes data.
  + Common use: Connects home/office networks to the internet.
  + Extra features: Assigns IP addresses, acts as a firewall, and provides security.
* **Switch**
  + Connects multiple devices (computers, printers, etc.) within a LAN.
  + Forwards data to the correct device using MAC addresses.
  + Smarter than a hub (sends only to intended device).
* **Hub** (legacy, rarely used now)
  + Connects multiple Ethernet devices.
  + Broadcasts incoming data to all ports.
  + Not intelligent: cannot filter or identify destination devices.
* **Modem (Modulator–Demodulator)**
  + Converts digital signals to analog (and back).
  + Provides connection between your home/office and the ISP.
  + Types: DSL modem, Cable modem, Fiber modem.
* **Access Point (AP)**
  + Extends a wired network with Wi-Fi capability.
  + Used in WLANs.
  + Many routers already have built-in APs.
* **Network Interface Card (NIC)**
  + Lets a computer or device connect to a network.
  + Two types: Wired (Ethernet NIC) and Wireless (Wi-Fi NIC).
* **Bridge**
  + Connects two LANs that use the same protocol.
  + Divides network into segments to reduce congestion.
  + Smarter than a hub because it filters traffic by MAC addresses.
* **Gateway**
  + Acts as a translator between different protocols or systems.
  + Example: connects an enterprise network to the internet.
* **Repeater**
  + Boosts and regenerates network signals.
  + Extends the coverage area of a network.
  + Useful in large buildings or areas with weak signals.
* **Firewall** (hardware or software)
  + Monitors and controls incoming/outgoing traffic.
  + Protects the network from unauthorized access.
* **Brouter (Bridge + Router)**
  + Hybrid device with features of both a bridge and a router.

**🔹 Network Topologies**

* **Bus Topology**
  + All devices are connected to a single central cable.
  + Data travels in both directions.
  + Pros: Cheap, easy to implement, less cabling.
  + Cons: If the main cable fails, the whole network fails; hard to troubleshoot.
* **Star Topology**
  + All devices connect to a central hub/switch.
  + Pros: Easy to install/manage; failure of one node doesn’t affect others.
  + Cons: Central hub/switch is a single point of failure; uses more cable.
* **Ring Topology**
  + Each device connects to two others, forming a circle.
  + Data flows in one direction (or both in dual rings).
  + Pros: Organized data flow.
  + Cons: Failure of one node/cable can break the loop; hard to troubleshoot.
* **Mesh Topology**
  + Every device connects to every other device (full or partial).
  + Pros: High reliability and redundancy; link failure doesn’t stop communication.
  + Cons: Expensive and complex to set up; requires lots of cabling.
* **Tree Topology (Hierarchical)**
  + Combines multiple star topologies with a main backbone.
  + Pros: Scalable, structured, easy to expand.
  + Cons: If backbone fails, whole network fails; needs more cabling.
* **Hybrid Topology**
  + Mix of two or more topologies (e.g., Star-Bus, Star-Ring).
  + Pros: Flexible and scalable.
  + Cons: Complex design and costly.

**🔹 Basic Network Troubleshooting Commands**

* **ping**
  + Checks connectivity to another device/website.
  + Example: ping google.com or ping 192.168.1.1.
  + Shows reachability, response time, packet loss.
* **ipconfig** (Windows)
  + Displays IP address, subnet mask, default gateway.
  + Useful options:
    - ipconfig /all → detailed info
    - ipconfig /release and ipconfig /renew → refresh IP
    - ipconfig /flushdns → clear DNS cache
* **ifconfig** (Linux/macOS, older command)
  + Displays and configures network interfaces.
  + Modern replacement: ip a.
* **ip** (Linux, modern tool)
  + ip a → show IP addresses
  + ip r → show routing table
  + ip link → show network interfaces
* **tracert (Windows) / traceroute (Linux/macOS)**
  + Shows the path data takes (hops) to reach a destination.
  + Example: tracert google.com
* **nslookup**
  + Queries DNS to find the IP of a domain.
  + Example: nslookup google.com.
* **route**
  + Displays/manages routing tables.
  + Example: route print (Windows), route -n (Linux).
* **telnet**
  + Tests connectivity to a specific host/port.
  + Example: telnet google.com 80.