A blue and gold cover

AI-generated content may be incorrect.



2.CLASSIFICATION of Networking

| **Category** | **Type** | **Description / Example** |
| --- | --- | --- |
| **Geographical Range** | **LAN** | **Small area like office or home network.** |
|  | **MAN** | **Covers a city or campus (e.g., university network).** |
|  | **WAN** | **Large area, connects countries (e.g., the Internet).** |
| **Topology (Connection)** | **Bus** | **All devices share a single backbone cable.** |
|  | **Star** | **Devices connected to a central hub or switch.** |
|  | **Ring** | **Each device connects to two others forming a circle.** |
|  | **Mesh** | **Devices interconnected, high reliability.** |
|  |  |  |

1.introduction of Networking

A network connects devices to share information, resources, and services. It supports business needs and modern technologies like IP telephony and video conferencing.

**Basic communication elements**: source, encoder, transmitter, channel, receiver, decoder, destination.

**Networks consist** of **end devices**, **transmission media**, intermediate devices

A screenshot of a computer

AI-generated content may be incorrect.

4.pdu during Encapsulation

| **OSI Layer** | **PDU (Protocol Data Unit)** |
| --- | --- |
| Application / Presentation / Session | Data |
| Transport Layer | Segment (or Datagram) |
| Network Layer | Packet |
| Data Link Layer | Frame |
| Physical Layer | Bits |

3.Reference models

A diagram of a model

AI-generated content may be incorrect.

6.some concepts

**Physical Topology:** The actual layout of cables, devices, and connections in the network (e.g., star, bus, ring).

**Logical Topology:** The way data flows and devices communicate across the physical network (e.g., how signals travel in Ethernet or Token Ring).

**Default Gateway:** The router that forwards traffic from the local network to external networks (e.g., Internet)

The **Subnet Mask** is a 32-bit number used in IP networking to divide an IP address into **network** and **host** portions.

5.Addressing

| **Layer** | **Address Type** | **Example** | **Purpose** | **Extra Notes** |
| --- | --- | --- | --- | --- |
| Data Link (Layer 2) | **MAC Address** | 00:1A:2B:3C:4D:5E | Unique hardware identity for local network communication | Fixed, burned into NIC |
| Network (Layer 3) | **IP Address** | 192.168.1.10 (IPv4), 2001:db8::1 (IPv6) | Logical address to locate devices across networks | **IPv4:** 32-bit, ~4.3B addresses, uses NAT.  **IPv6:** 128-bit, ~3.4×10³⁸ addresses, no NAT needed, better security & mobility. |
| Transport (Layer 4) | **Port Number** | 80 (HTTP), 443 (HTTPS) | Identifies specific application/service on a device | Works with both IPv4 & IPv6 |

A graph on a piece of paper

AI-generated content may be incorrect.

8.Static Routing vs Dynamic

* **Static Routing = Manual, simple, secure, but not adaptive.**
* **Dynamic Routing = Automatic, adaptive, resource-consuming, suitable for large networks.**

7.subneting vs vlsm

| **Feature** | **Subnetting (Fixed)** | **VLSM (Variable)** |
| --- | --- | --- |
| Subnet Mask | Same for all subnets | Different masks for different subnets |
| Flexibility | Low – all subnets must be equal size | High – subnets can be tailored to host needs |
| IP Address Usage | May waste addresses | More efficient, less wasted addresses |
| Example | All subnets /24 | Some /24, some /26, some /30, etc. |

10. **some commends in cisco cmd**

| **Section** | **Command Example** | **Purpose** |
| --- | --- | --- |
| **CLI First Look** | enable configure terminal | Enter privileged EXEC mode and global configuration mode |
| **Basic Configuration** | hostname R1  enable secret cisco123  line console 0 → password cisco → login  interface g0/0 → ip address 192.168.1.1 255.255.255.0 → no shutdown | Set hostname, passwords, console access, interface IP, and warning banner |
| **Viewing, Saving, Erasing** | show running-config  show startup-config  copy running-config  startup-config  erase startup-config | View, save, delete configuration files and reload device |
| **Discovering & Testing** | show ip interface brief ping 192.168.1.2  traceroute 8.8.8.8 | Check interfaces, test connectivity, trace path |
| **TELNET** | line vty 0 4 password telnet123  login | Configure Telnet remote access |

9. **Cisco IOS Booting Steps**

| **Step** | **Process** | **Description** |
| --- | --- | --- |
| 1 | **POST (Power On Self Test)** | Runs hardware diagnostics from ROM (CPU, RAM, NVRAM). |
| 2 | **Bootstrap Loading** | Copied from ROM to RAM; executed by CPU to locate IOS. |
| 3 | **IOS Location** | Usually in Flash; if not found, a minimal IOS from ROM is used. |
| 4 | **IOS Loading** | IOS copied into RAM (current models); older models ran IOS directly from Flash. |
| 5 | **Config File Search** | Bootstrap looks for **startup-config** in NVRAM. |
| 6 | **Config File Loading** | If found, loaded into RAM as **running-config**; if not, user is prompted for setup mode. |

**lab view**

**1. Introduction to the LabVIEW Environment**

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical programming environment based on **dataflow**. Unlike text-based languages (C#, Python), it uses **graphical blocks and wires** to create programs, making it intuitive for engineers and scientists.

**Core Concepts:**

* **Graphical Programming:** Build logic using icons and wires (like a flowchart).
* **Virtual Instruments (VIs):** Every program is a VI, simulating real instruments.

**Key Components of a VI:**

1. **Front Panel:** User interface with controls (inputs) and indicators (outputs).
2. **Block Diagram:** Graphical source code connecting functions and data flow.
3. **Icon & Connector Pane:** Enable reuse of a VI as a SubVI.

**2. Fundamental Programming Structures in LabVIEW**

* **While Loop:** Repeats until a stop condition is met.
* **For Loop:** Runs a set number of times.
* **Case Structure:** Executes different code based on input (like if-else).
* **Sequence Structure:** Forces execution order in a program.
* **Control via LabVIEW (LINX Library):**
  + Upload firmware to Arduino.
  + LabVIEW sends commands (e.g., read sensor, turn LED ON).
  + Arduino executes and returns data to LabVIEW.

**3. Working with Data: Arrays & Clusters**

* **Arrays:** Collections of same-type data (numbers, strings), 1D or multi-D.
* **Clusters:** Group mixed-type data (like a struct). Managed with Bundle/Unbundle.

**4. Data Visualization: Charts & Graphs**

* **Charts:** Show data continuously with history. Types: Strip, Scope, Sweep.
* **Graphs:** Display complete datasets at once, no history buffer.

**5. Arduino UNO: Hardware Gateway**

* **Role:** Open-source microcontroller bridging LabVIEW with the real world.
  + 6 analog inputs (sensors).
  + 14 digital I/O pins (LEDs, motors, relays).
  + 6 PWM pins (control speed, dimming, etc.).

**6. Project Overview**

We use these Express VIs:

* **Acquire Sound:** Record sound from microphone.
* **Filter:** Apply band-pass filter to the sound.
* **Play Waveform:** Output filtered sound through speakers.
* **Tone Measurements:** Measure average frequency and amplitude.