UNIT I

NETWORKING FUNDAMENTALS AND PROTOCOL MODELS

Basics of storage, Need for modern storage, Overview of data growth and digital transformation, Features of DAS, NAS, SAN, CAS, Introduction to RAID: types, levels, and configurations, Disk subsystems, Performance and reliability considerations, Traditional vs. software-defined storage.

◈ Basics of Storage

Computer storage is a core function of a computer that involves the **retention of digital data** on various physical components and media. It's how a computer remembers information, whether it's for a split second or for many years. Think of it as the computer's memory bank, allowing it to store, retrieve, and process information. □

The fundamental building block of all digital data is the **bit** (binary digit), which has a value of either a 0 or a 1. All information, from text and images to videos and programs, is converted into a long string of these bits. Bits are then grouped into **bytes** (8 bits), which serve as the basic unit for measuring storage capacity (e.g., kilobytes, megabytes, gigabytes, and terabytes).

Types of Storage 🗀

Computer storage is organized into a hierarchy based on its speed, cost, and capacity. The faster the storage, the more expensive it is and the less data it can typically hold.

Primary Storage (Volatile Memory)

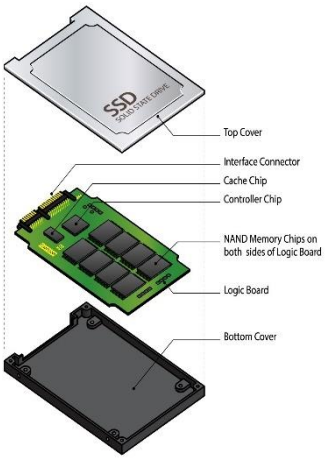
This is the computer's main memory and is directly accessible by the CPU. It's used for storing data and instructions that are actively being used. It's volatile, meaning it loses its contents when the power is turned off.

- Random Access Memory (RAM): The most common type of primary storage. It's a temporary workspace for the CPU, allowing for quick reading and writing of data. When you open a program or a file, it's loaded into RAM for fast access.
- Cache Memory: A very small, extremely fast memory that stores frequently accessed data and instructions. The CPU can access cache memory much faster than RAM, further speeding up performance.

Secondary Storage (Non-Volatile Storage)

This is a computer's long-term memory. It's non-volatile, meaning it retains data even when the power is off. It's used to permanently store the operating system, applications, and user files.

- Hard Disk Drive (HDD): A traditional storage device that uses spinning magnetic platters to store data. They are relatively inexpensive and offer large capacities, but are slower and more prone to mechanical failure than SSDs.
- Solid State Drive (SSD): A newer technology that uses interconnected flash memory chips to store data. SSDs have no moving parts, making them much faster, more durable, and more energy-efficient than HDDs. They are, however, more expensive per gigabyte.
- Optical Discs: Storage media like CDs, DVDs, and Blu-ray discs that use lasers to read and write data. They are largely considered obsolete for data storage today, but are still used for entertainment media.
- Flash Storage: Non-volatile memory used in devices like USB drives, memory cards (e.g., SD cards), and SSDs. It's compact, durable, and offers fast data transfer speeds.



HDD

How It Works 💻

SDD

Different storage technologies work in different ways to represent the binary bits (0s and 1s).

- Magnetic Storage (HDDs): Tiny areas on a spinning platter are magnetized in a specific direction. One direction represents a "1" and the other a "0." A read/write head moves over the platters to detect and change the magnetic fields.
- Solid-State Storage (SSDs): Data is stored as electrical charges within a grid of flash memory cells. The presence of a charge represents a "1" and the absence represents a "0."
- Optical Storage (CDs/DVDs): A laser burns tiny pits into the disc's surface. A laser is then used to read the disc, and it reflects differently off the "lands" (the flat areas) and the "pits," which is interpreted as a stream of 0s and 1s.

Cloud Storage

Cloud storage is a service where a third-party provider stores and manages your data on remote servers (data centers) and makes it accessible via the internet. It's a "pay-as-you-go" or subscription-based model. Think of it as renting space in a massive, professionally managed digital warehouse.

How it works: When you use a service like Google Drive or Dropbox, you upload your files to the provider's servers. The provider is responsible for the hardware, security, maintenance, and redundancy (often replicating your data across multiple servers and locations to prevent loss). You access your files through a web browser, a desktop application, or a mobile app.

- Examples: Google Drive, Dropbox, OneDrive, AWS S3.

2. Key Characteristics of Storage

- Speed – How fast data can be read/written.
- Capacity – How much data it can hold.
- Volatility – Does it keep data after power is off?
- Cost – Price per GB of storage.

Need for Modern Storage

With the explosion of data, cloud services, IoT, and real-time applications, traditional storage (like basic hard drives or tapes) is not enough. Organizations and users now require modern storage solutions.

✅ Reasons Why Modern Storage is Needed

1. Massive Data Growth (Big Data)
    - Social media, IoT, AI, and businesses generate huge amounts of data daily.
    - Modern storage can handle petabytes or more efficiently.
2. High Speed & Low Latency
    - Real-time apps (video calls, online gaming, AI, analytics) need instant access.
    - Modern solutions like SSDs, NVMe, and in-memory storage provide this speed.
3. Reliability & Availability
    - Data loss or downtime is unacceptable for banking, healthcare, or enterprises.
    - Modern storage uses RAID, replication, and cloud backups for continuous availability.
4. Scalability
    - Data is growing rapidly. Modern storage (like cloud and SAN systems) can scale up or down as needed.
5. Accessibility Anywhere
    - Users want to access files across devices and locations.
    - Cloud and distributed storage make this possible.
6. Security & Compliance
    - Sensitive data must be protected.
    - Modern storage includes encryption, authentication, and legal compliance tools.

Overview of Data Growth

Data growth refers to the continuous and exponential increase in the amount of digital information generated, collected, and stored globally. This phenomenon is driven by the proliferation of devices, the rise of the Internet of Things (IoT), and the widespread adoption of digital services. The volume of data has exploded from petabytes and exabytes to zettabytes and yottabytes, transforming every sector from business and science to daily life.

Key Drivers 🚀

Several factors are fueling this explosive growth:

- The Internet of Things (IoT): Billions of interconnected devices, from smartwatches to industrial sensors, are constantly generating data on everything from personal health to factory performance.

- Social Media: Platforms like Facebook, Twitter, and TikTok produce massive amounts of user-generated content, including photos, videos, messages, and interaction data.
- Video and Streaming Services: High-definition video content from platforms like Netflix and YouTube, along with video conferencing and surveillance, accounts for a significant portion of global data traffic.
- Cloud Computing: The ease of storing and accessing data via cloud services encourages both individuals and businesses to save more information than ever before.
- Big Data Analytics: Companies are collecting vast amounts of data to gain insights into customer behavior, optimize operations, and drive innovation. This need for analysis fuels further data collection.

Impact of Data Growth 💡

The exponential growth of data has a profound impact across various fields:

- Business: Companies can leverage big data to personalize customer experiences, predict market trends, and improve operational efficiency.
- Science and Research: Scientists can analyze massive datasets to accelerate discoveries in medicine, climate science, and astronomy.
- Infrastructure: The need to store and process this data requires a massive expansion of data centers, servers, and network capacity, leading to challenges in energy consumption and infrastructure management.
- Privacy and Security: With more data being collected, the challenges of protecting sensitive information from breaches and misuse become increasingly critical.

Trends in Data Growth

- Exponential Increase: Global data is doubling roughly every 2–3 years.
- Video Dominance: Over 80% of internet traffic is video.
- Edge Computing Growth: Local processing reduces cloud bandwidth but increases distributed data.
- Data-Intensive Applications: AI, ML, and analytics drive storage and bandwidth demands.

Challenges Due to Data Growth

1. Storage: Need for larger, faster, and more reliable storage solutions.
2. Bandwidth & Network Load: Networks must handle increased traffic efficiently.
3. Data Management: Organizing, retrieving, and securing massive datasets.
4. Security & Privacy: More data means higher risk of breaches.
5. Energy Consumption: Data centers and networks require more energy.

Solutions and Adaptations

- Scalable Cloud Storage (AWS, Azure, GCP)
- High-Speed Networks (5G, fiber optics)
- Data Compression & Efficient Protocols (QUIC, HTTP/3)
- Edge & Fog Computing to process data closer to sources
- AI-Driven Data Management for automated processing and insights

Overview of Digital Transformation 🚀

Digital transformation is the process of using digital technologies to create new—or modify existing—business processes, culture, and customer experiences to meet changing market and business requirements. It's not just about technology; it's a fundamental change in how a business operates and delivers value to its customers.

Digital transformation (DX) is the integration of digital technologies into all areas of a business or organization, fundamentally changing how it operates, delivers value, and interacts with customers. It is not just about technology—it also involves culture, processes, and business models.

Key Components 💡

Digital transformation typically involves several interconnected areas:

1. Technology Integration: This involves adopting and integrating modern technologies like cloud computing, artificial intelligence (AI), machine learning (ML), big data analytics, and the Internet of Things (IoT) to improve efficiency and create new capabilities.
2. Process Re-engineering: Businesses redesign their core processes to be more agile, data-driven, and automated. This could mean automating supply chains, digitizing customer service, or using data to personalize marketing.
3. Cultural Shift: This is arguably the most challenging part. Digital transformation requires a change in mindset, encouraging employees to be more innovative, adaptable, and comfortable with change. It fosters a culture of collaboration and continuous learning.
4. Customer-Centricity: The goal is to provide a seamless and personalized experience for the customer. Companies use digital tools to better understand customer needs, provide real-time support, and offer customized products and services.

Why It's Important ☑

The push for digital transformation is driven by several factors:

- Competitive Pressure: Businesses that fail to adapt risk being outpaced by more agile, tech-savvy competitors.
- Changing Customer Expectations: Today's customers expect convenient, personalized, and instant services.
- Operational Efficiency: Digital tools can automate repetitive tasks, reduce errors, and provide valuable insights, leading to significant cost savings and improved productivity.

2. Key Drivers of Digital Transformation

1. Customer Expectations
    - Demand for faster, personalized, and seamless digital experiences.
2. Technological Advancements
    - Cloud computing, AI/ML, IoT, big data, mobile technologies, and 5G.
3. Competitive Pressure
    - Businesses must innovate to stay ahead of agile digital-first competitors.
4. Operational Efficiency
    - Automation, data analytics, and integrated systems improve productivity and reduce costs.
5. Global Connectivity
    - Remote work, global markets, and online platforms accelerate digital adoption.

4. Benefits of Digital Transformation

- Improved Customer Experience: Personalization and faster service.
- Operational Efficiency: Automation reduces errors and saves time.
- Data-Driven Decisions: Real-time analytics enables smarter strategies.

- Innovation and Agility: Ability to quickly adapt to market changes.
- Competitive Advantage: Digital-first organizations outperform traditional ones.

5. Challenges in Digital Transformation

1. Resistance to Change: Employees may resist new processes and tools.
2. Legacy Systems: Older infrastructure can slow down adoption.
3. Cybersecurity Risks: Increased reliance on digital systems brings new threats.
4. High Costs: Initial investment in technology and training can be significant.
5. Skill Gaps: Need for employees with expertise in AI, cloud, and analytics.

6. Examples of Digital Transformation

- Retail: E-commerce platforms, personalized recommendations, and mobile apps.
- Banking: Mobile banking, digital wallets, and AI-driven fraud detection.
- Healthcare: Telemedicine, electronic health records, and AI diagnostics.
- Manufacturing: Smart factories, IoT-enabled equipment, and predictive maintenance.

Features of DAS, NAS, SAN AND CAS

Direct-Attached Storage (DAS)

Definition

Direct-Attached Storage (DAS) is a digital storage system that is directly connected to a single computer or server, without a network in between. It provides local storage for that specific device.

Key Point: Unlike NAS (Network-Attached Storage) or SAN (Storage Area Network), DAS is not shared over a network.

It is the simplest and oldest form of data storage, commonly used in personal computers and small-scale server setups.

How it Works ⚙

DAS is a straightforward storage solution where the storage device is physically connected to the host machine. The connection is typically made through an interface like SATA, SCSI, USB, or Thunderbolt. The operating system of the host computer manages the storage, treating it as a local drive.

Think of it as the hard drive inside your laptop or an external hard drive you plug into your computer's USB port. The data is accessed directly by the host machine's bus, making it a very fast and low-latency storage solution for that specific computer.

Key Features and Characteristics ✨

- Simplicity: DAS is easy to set up and manage, as it's a simple plug-and-play connection. It doesn't require complex network configuration or management.
- Performance: It offers high performance because there is no network bottleneck. The speed is limited only by the drive's read/write speed and the bus interface.
- Cost-Effective: For a single user or a small business with limited storage needs, DAS is often the most affordable solution.
- Limited Scalability and Sharing: The major drawback of DAS is that it is limited to the capacity of the connected server. It cannot be easily shared among multiple computers or users without using a file-sharing service or a dedicated server.

- No Redundancy: If the host computer fails, the DAS storage becomes inaccessible. There is no built-in redundancy or failover mechanism.

2. Components of DAS

1. Storage Devices
    - Hard Disk Drives (HDDs)
    - Solid-State Drives (SSDs)
    - NVMe drives
2. Host Connection
    - Connected directly via interfaces like SATA, SAS, or USB.
3. Controller
    - Manages the storage devices and ensures data access by the host.

3. Types of DAS

1. Internal DAS
    - Storage drives installed inside the server or computer.
2. External DAS
    - Storage devices connected externally via USB, Thunderbolt, or eSATA.

4. Advantages of DAS

- High Performance: Direct connection ensures low latency and high data transfer speeds.
- Simplicity: Easy to install and manage since it doesn't require network setup.
- Cost-Effective: Less expensive than SAN or complex NAS setups.
- Suitable for Single-Server Use: Ideal for local storage needs.

5. Limitations of DAS

- Limited Scalability: Expanding storage requires physical addition of drives to each server.
- No Centralized Management: Each server manages its own storage independently.
- Poor for Shared Access: Cannot easily provide storage to multiple users or servers.
- Backup Complexity: Requires separate backup solutions for each system.

Network Attached Storage(NAS)

Network Attached Storage (NAS) is a specialized storage device that connects to a network, allowing multiple users and devices to store and access data centrally. Unlike a traditional external hard drive, which connects to a single computer, a NAS is accessible over a local network (LAN) or even remotely via the internet.

How it Works ⚙

A typical NAS device consists of:

1. Hardware: One or more hard drives or solid-state drives (SSDs) to store the data.
2. Processor and RAM: A CPU and memory to handle network requests and file operations.
3. Operating System: A simplified operating system optimized for file storage and sharing, often a version of Linux.

When a user on the network wants to access a file, their computer sends a request to the NAS device's IP address. The NAS processes the request and sends the file back to the user's computer. This process is seamless and appears to the user as if they are accessing a folder on their own computer.

Key Features of NAS:

1. Network Connectivity:
    - NAS devices have their own IP address and connect directly to your network through Ethernet or Wi-Fi.
    - Any device on the same network can access its files, depending on permissions.
2. Centralized Storage:
    - Acts as a central hub for storing files, making it easier to share data among multiple users or devices.
3. File Sharing:
    - Supports common protocols like SMB/CIFS (Windows), NFS (Linux), and AFP (Mac), allowing seamless file sharing across platforms.
4. Data Protection:
    - Many NAS devices support RAID configurations, backups, and snapshots to prevent data loss.
5. Remote Access:
    - Some NAS systems allow users to access files over the internet securely.
6. Expandable and Scalable:
    - You can add more storage by inserting additional hard drives or upgrading existing ones.
7. Additional Features:
    - Some NAS devices include apps for media streaming, surveillance storage, virtualization support, and cloud syncing.

Storage area network

A Storage Area Network (SAN) is a high-speed, specialized network that provides block-level storage to servers. Unlike NAS, which provides file-level access, SAN provides storage at the “raw disk” level, which servers can format and manage as if they were local drives. SANs are commonly used in enterprise environments where high performance, reliability, and scalability are required.

How It Works ⚙

A SAN consists of three main components:

1. Cabling and Interconnects: This includes high-speed networking hardware like Fibre Channel (the traditional choice for SANs) or Ethernet (in the case of iSCSI SANs).
2. Host Bus Adapters (HBAs): These are special adapters on the servers that connect them to the SAN.
3. Storage Devices: A pool of storage devices, such as disk arrays, that are shared among all the connected servers.

When a server needs to access data, it sends a request over the SAN. The SAN routes this request to the appropriate storage device, which then returns the requested data block. This process is highly efficient and transparent to the server's operating system.

Key Features and Benefits ✨

- Block-Level Access: Data is accessed in chunks called blocks, which is ideal for databases and virtual machines that require high-speed, direct access to storage.
- High Performance: Because a SAN is a dedicated network, it eliminates the network congestion that can occur with a NAS, providing much higher throughput and lower latency.
- Scalability: SANs are highly scalable. You can add more storage capacity or more servers to the network without disrupting operations.
- Data Consolidation: SANs allow you to centralize storage management, making it easier to back up, protect, and restore data.

SAN vs. NAS 📊

The main difference between a SAN and a NAS is the level of data access and the underlying technology.

- SAN: Provides block-level storage to servers via a dedicated network. It's best for high-performance applications.
- NAS: Provides file-level storage to users over a standard Ethernet network. It's best for simple file sharing and collaboration.

Content-Addressed Storage

Content-Addressed Storage (CAS) is a storage system where data is retrieved based on its content (usually a unique identifier or hash) rather than its location or file name. CAS is widely used for archival storage, compliance, and immutable data storage, where it’s critical to ensure that data is tamper-proof and cannot be modified once written.

How It Works 🛠️

When you store data in a CAS system, it follows a simple, yet powerful, two-step process:

1. Hashing: The system uses a cryptographic hash function (like SHA-256) to process the entire content of the data. This algorithm produces a fixed-size, unique string of characters called a hash. This hash is the content's unique address.
2. Storage: The data is stored in a simple, flat structure, and the system records which hash corresponds to which piece of data. There's no hierarchical folder structure; it's a simple key-value relationship where the hash is the key and the data is the value.

To retrieve the data, you simply provide the hash, and the system uses it as a lookup key to find and deliver the content.

Key Features and Benefits ✨

- Immutability: Once a piece of data is stored, its content is fixed and cannot be changed. Any change to the content, no matter how small, will produce a completely different hash, resulting in a new, separate stored object. This guarantees the integrity and authenticity of the data.
- Data Deduplication: Because the same content will always produce the same hash, a CAS system automatically avoids storing duplicate copies of the same data. If you try to store an identical file that already exists, the system simply creates a new reference to the existing data, saving significant storage space.
- Location Independence: The data's physical location is irrelevant. You only need the hash to retrieve it. This makes CAS ideal for distributed and decentralized systems, where data can be stored across multiple servers.

- Data Integrity: The hash acts as a checksum. The system can continuously verify that the stored data has not been corrupted by re-hashing the content and comparing the new hash to the original one

Here's a clear comparison of DAS, NAS, SAN, and CAS with their main features:

Feature / Storage Type	DAS (Direct Attached Storage)	NAS (Network Attached Storage)	SAN (Storage Area Network)	CAS (Content-Addressed Storage)
Access Type	Direct (local to server/computer)	File-level over network	Block-level over network	Content-based (using hash/identifier)
Connection	SATA, SCSI, USB, or PCIe	Ethernet / LAN	Fibre Channel, iSCSI, InfiniBand	Network or storage system (software-defined)
Use Case	Single server storage, simple setups	File sharing, home/small business storage	Enterprise applications, databases, virtualization	Archival, compliance, immutable storage
Scalability	Limited (local expansion only)	Moderate (add drives or expand NAS)	High (add multiple storage arrays/servers)	High (object-based storage, easy to scale)
Performance	High for single server	Moderate, depends on network	Very high, low latency	Moderate to high, optimized for retrieval by content
Data Management	Basic, depends on host OS	Centralized file management, backups	Advanced storage management, RAID, replication	Data integrity, deduplication, immutable storage
Access by Multiple Users	Usually single server	Multiple users over network	Multiple servers with shared block storage	Multiple users/systems using content identifiers
Cost	Low to moderate	Moderate	High (enterprise-level)	Moderate to high (depending on implementation)
Data Integrity / Security	Basic, depends on local management	Basic to moderate (RAID, backups)	High (RAID, replication, failover)	Very high (immutable, hash-based integrity)

RAID(Redundant Array of Independent Disks)

RAID is a data storage technology that combines multiple physical hard drives into a single logical unit to improve performance, reliability, and/or storage capacity. It is widely used in servers, data centers, and enterprise storage systems.

The term RAID stands for Redundant Array of Independent (or Inexpensive) Disks. The main idea is to use multiple disks together to achieve benefits that a single disk cannot provide.

Key Objectives of RAID

1. Redundancy / Fault Tolerance

- Protects data from disk failures. Some RAID levels can survive one or more disk failures without data loss.

2. Improved Performance
   - Certain RAID configurations can read/write data faster than a single disk because data is split across multiple disks.
3. Increased Storage Capacity
   - Multiple disks can be combined into a single logical drive, increasing usable storage.

How RAID works

RAID works by placing data on multiple drives and allowing input/output (I/O) operations to overlap in a balanced way, improving performance.

Because using multiple drives increases the mean time between failures, storing data redundantly also increases fault tolerance.

RAID arrays appear to the operating system (OS) as a single logical drive. RAID employs the techniques of disk mirroring or disk striping. Mirroring copies identical data onto more than one drive.

Striping partitions help spread data over multiple drives. Each drive's storage capacity is divided into units ranging from a sector of 512 bytes up to several megabytes. The stripes of all the drives are interleaved and addressed in order.

Disk mirroring and disk striping can also be combined in a RAID array.

In a single-user storage system where large records are stored, the stripes are typically set up to be small -- 512 bytes, for example -- so that a single record spans all the drives and can be accessed quickly by reading all the drives at the same time. In a multiuser system, better performance requires a stripe wide enough to hold the typical or maximum size record, enabling overlapped disk I/O across drives.

RAID levels

RAID devices use different versions called *levels*. The original paper that coined the term and developed the RAID setup concept defined six levels of RAID -- 0 through 5. This numbered system differentiated the capabilities of the RAID versions. The number of levels has since expanded and has been broken into three categories: standard, nested and nonstandard RAID levels.

1. RAID 0 – Striping

RAID 0, also known as striping, is a storage technology that distributes data across multiple disk drives in a single, striped set. It's used to enhance performance for both read and write operations.

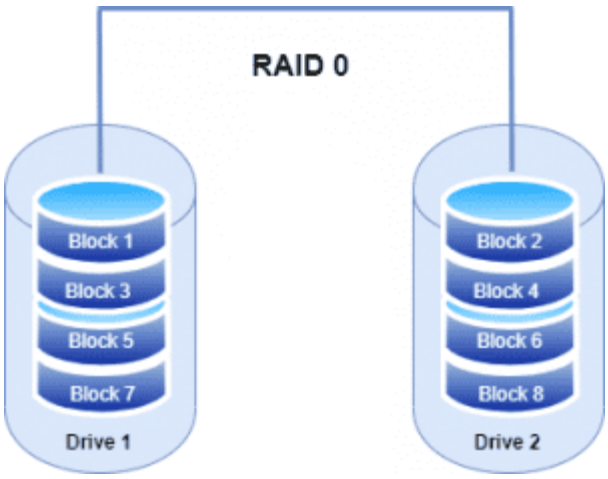
How It Works ⚙

RAID 0 doesn't provide any data redundancy. Instead, it breaks data into smaller blocks, called "stripes," and writes them simultaneously across all the drives in the array. This parallel writing

process significantly reduces the time it takes to save data. When reading data, the process is reversed, with the system reading from all drives at once, which drastically improves performance.

Key Characteristics and Use Cases 🚀

- Performance: This is the primary benefit of RAID 0. By utilizing multiple drives, it dramatically increases both the input/output (I/O) speed and throughput.
- No Redundancy: This is the major drawback. If even one of the drives in the array fails, all the data stored on the entire array is lost. The failure of a single drive is catastrophic.
- Capacity: The total capacity of the RAID 0 array is the sum of the capacities of all the drives in the set.
- Use Cases: RAID 0 is best used in non-critical environments where speed is the most important factor and data loss is acceptable, such as for temporary files, video editing, or gaming installations.
- **Advantages:**
  - High read and write performance because multiple disks work in parallel.
  - Full storage capacity of all disks is utilized.
- **Disadvantages:**
  - No redundancy; if one disk fails, all data is lost.
- **Use Cases:**
  - High-performance computing, gaming systems, temporary data storage where speed is critical and data loss is acceptable.



2. RAID 1 – Mirroring

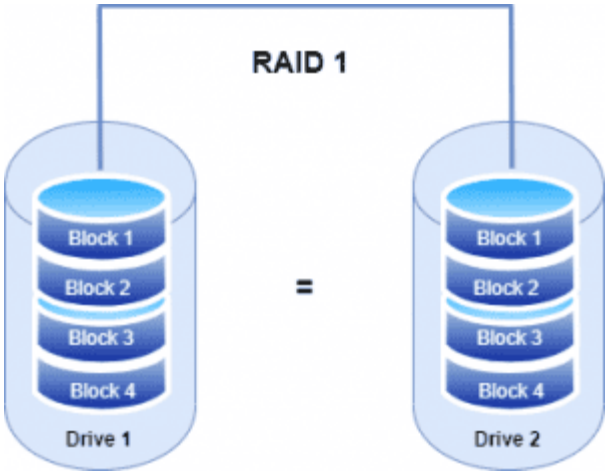
RAID 1, also known as mirroring, is a storage technology that creates an exact, real-time duplicate of data on a second disk drive. Its primary goal is to provide data redundancy and fault tolerance.

How It Works ⚙

When data is written to a RAID 1 array, it's simultaneously written to two or more disk drives. Each drive contains an identical copy of the data. If one disk fails, the other disk (the "mirror") continues to operate, and the system can seamlessly switch to using the working disk without any data loss or interruption.

Key Characteristics and Use Cases

- Redundancy: RAID 1 offers excellent redundancy. It can withstand the failure of a single drive, making it a very reliable option for storing critical data.
- Performance: Read performance can be slightly improved because the system can read from either disk. However, write performance is limited by the speed of the slowest drive, as data must be written to both drives.
- Capacity: The usable storage capacity is only half the total capacity of the drives in the array. For example, two 1 TB drives in a RAID 1 configuration will provide only 1 TB of usable space, with the other 1 TB used for the mirror.
- Use Cases: RAID 1 is ideal for storing crucial data where reliability is more important than capacity or raw speed. This includes operating system drives, financial data, and other critical information.
- **Advantages:**
    - High fault tolerance; can survive a single disk failure.
    - Simple to implement and recover data.
- **Disadvantages:**
    - Storage efficiency is only 50%, as half of the total disk space is used for the mirror.
- **Use Cases:**
    - Critical systems, operating system drives, financial databases where data protection is crucial.



3. RAID 5 – Striping with Parity

RAID 5 is a storage technology that combines data striping with distributed parity to provide both good performance and data redundancy. It is one of the most common RAID levels used today.

How It Works

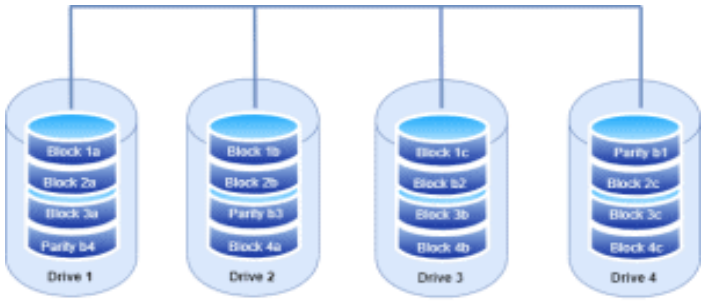
RAID 5 requires a minimum of three disk drives. It stripes data across all the drives in the array, similar to RAID 0. However, it also calculates a parity block for each stripe and distributes this parity across all the disks.

- Striping: Data is broken into chunks and written to different disks simultaneously for faster performance.
- Parity: The parity block is a checksum of the data blocks in a stripe. It is used to reconstruct the data on a failed disk.

For example, if you have three data blocks (A, B, C) and a parity block (P), the parity is calculated such that A + B + C = P. If one of the data blocks fails, the missing block can be calculated from the remaining data blocks and the parity block.
If a single disk fails, the remaining disks can use the parity information to rebuild the lost data. The array can continue to operate in a degraded state until the failed drive is replaced and the data is rebuilt.

Key Characteristics and Use Cases

- Performance: RAID 5 offers good read performance, as data is read from multiple disks. Write performance is slightly slower than RAID 0 because it has to calculate and write the parity information.
- Redundancy: It can withstand the failure of one disk. If a second disk fails before the first one is replaced, all data is lost.
- Capacity: The usable storage capacity is the sum of all the disks' capacities minus one disk's capacity (which is used for parity). For example, three 1 TB drives will provide 2 TB of usable storage.
- Cost: It is a cost-effective solution for providing redundancy without the 50% capacity overhead of RAID 1.
- Use Cases: RAID 5 is an excellent choice for general-purpose servers, file servers, and applications that require a balance of performance, capacity, and fault tolerance.
- **Advantages:**
  - Good balance of performance, fault tolerance, and storage efficiency.
  - Can survive a single disk failure without losing data.
- **Disadvantages:**
  - Write operations are slower due to parity calculations.
  - Minimum of 3 disks required.
- **Use Cases:**
  - File servers, application servers, small to medium business storage.



RAID 5
stripping with parity across drives

4. **RAID 6 – Striping with Double Parity**

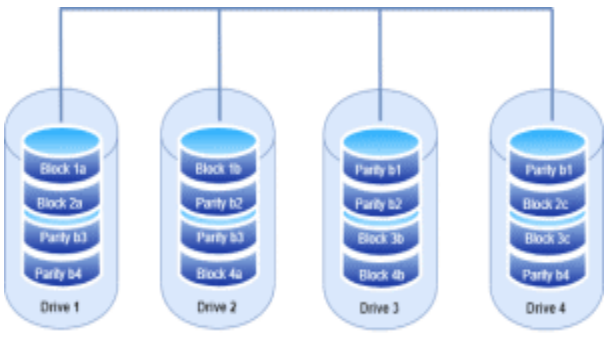
- RAID 6 is a storage technology that uses data striping with two independent parity blocks to provide a higher level of data redundancy than RAID 5. It's designed to protect against the simultaneous failure of two disks.

How It Works 🛡️

- RAID 6 requires a minimum of four disk drives. It stripes data across all the drives in the array and, for each stripe, it calculates two separate parity blocks. These parity blocks are then distributed across the drives.
- Parity Calculation: Two different algorithms are used to calculate the parity blocks (often a simple XOR and a more complex polynomial function). This creates two independent sets of checksums.
- Data Reconstruction: If a single disk fails, the remaining drives and the parity information can be used to reconstruct the data, just like in RAID 5. However, if a second disk fails before the first is replaced and rebuilt, the two sets of parity data are still sufficient to reconstruct the lost data from both drives.

Key Characteristics and Use Cases 🗄️

- Redundancy: The main advantage of RAID 6 is its high fault tolerance. It can withstand the failure of two disks simultaneously, which is crucial for very large arrays where the risk of a second disk failing during a lengthy rebuild process is significant.
- Performance: Write performance is slower than RAID 5 because the system must calculate and write two parity blocks for every stripe. Read performance is generally good.
- Capacity: The usable storage capacity is the sum of all disks' capacities minus two disks' worth of capacity (used for the two parity blocks).
- Cost: RAID 6 is more expensive than RAID 5 in terms of storage efficiency due to the extra parity drive.
- Use Cases: It's an excellent choice for large-scale storage systems, data archives, and environments where data integrity is paramount and a large amount of data needs to be protected, such as in enterprise-level data centers and video surveillance systems.
- **Advantages:**
  - Higher fault tolerance than RAID 5.
  - Safe for critical data storage in larger disk arrays.
- **Disadvantages:**
  - More complex implementation.
  - Write performance slightly slower than RAID 5.
- **Use Cases:**
  - Enterprise storage systems, mission-critical databases, large-scale archival storage.



RAID 6
striping with dual parity across drives

5. **RAID 10 (1+0) – Mirroring + Striping**

RAID 10, also known as RAID 1+0, is a nested RAID level that combines mirroring and striping. It provides the high performance of a RAID 0 array and the excellent data redundancy of a RAID 1 array. It's often used for mission-critical applications that require both speed and fault tolerance.

How It Works ⚙

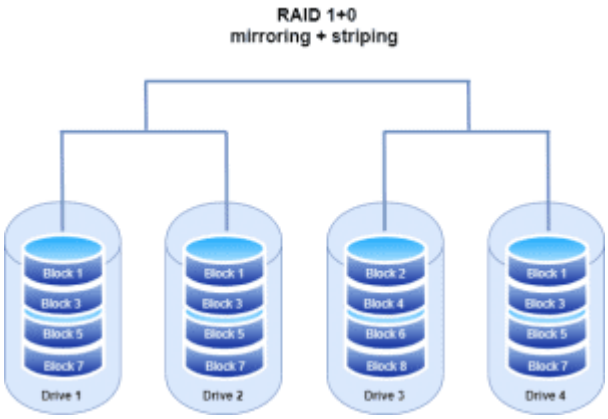
RAID 10 requires a minimum of four disk drives and is configured in two steps:

1. Mirroring (RAID 1): First, data is mirrored across pairs of drives. This creates a redundant copy of the data on each pair.
2. Striping (RAID 0): The mirrored pairs are then striped together. This distributes the data across all the mirrored sets, significantly improving read and write performance.

If a single drive fails within a mirrored pair, its data can be recovered from the other drive in the same pair. If both drives in a pair fail, the entire array will fail. Therefore, the array can withstand the failure of one drive per mirrored pair.

Key Characteristics and Use Cases 🚀

- Performance: RAID 10 offers excellent performance for both read and write operations. The striping provides high speed, while the mirroring ensures quick data recovery in case of a drive failure.
- Redundancy: It provides high fault tolerance and can withstand the loss of a drive in each mirrored pair.
- Capacity: The usable storage capacity is half of the total capacity of all the drives, similar to RAID 1. For example, four 1 TB drives will provide 2 TB of usable space.
- Cost: Due to the mirroring, it is one of the most expensive RAID levels in terms of storage efficiency.
- Use Cases: RAID 10 is the preferred choice for applications where both high performance and data protection are critical, such as database servers, email servers, and other high-transaction environments.
- **Advantages:**
    - High performance and high fault tolerance.
    - Can survive multiple disk failures as long as they are not in the same mirrored pair.
- **Disadvantages:**
    - Expensive; requires at least 4 disks.
    - Storage efficiency is 50%.
- **Use Cases:**
    - High-performance databases, enterprise applications, critical workloads where both speed and redundancy are needed.



RAID Configuration Summary Table

RAID Level	Minimum Disks	Fault Tolerance	Performance	Storage Efficiency
RAID 0	2	None	High	100%
RAID 1	2	Yes	Moderate	50%
RAID 5	3	Yes (1 disk)	Good	N-1 / N
RAID 6	4	Yes (2 disks)	Moderate	N-2 / N
RAID 10	4	Yes	High	50%

Disk Subsystem

What is a Disk Subsystem?

A disk subsystem is the combination of hard disks, controllers, interfaces, and software that work together to store and manage data for a computer system. Disk subsystems are responsible for data storage, retrieval, and protection in servers, workstations, and enterprise storage systems.
In essence, it's the hardware and software infrastructure that supports disk storage.

Components of a Disk Subsystem

1. Disk Drives
   - Physical storage media such as HDDs (Hard Disk Drives) or SSDs (Solid State Drives).
   - Store user data, operating systems, applications, and backups.
2. Disk Controllers
   - Hardware that manages communication between the computer and the disk drives.
   - Examples: RAID controllers, SCSI controllers, SATA controllers.
3. Disk Interfaces / Buses
   - Connect disks to the controller and CPU.
   - Common interfaces: SATA, SAS, Fibre Channel, NVMe.
4. RAID / Storage Management Software
   - Handles data distribution, redundancy, and fault tolerance.
   - Allows creation of RAID arrays, snapshots, and backups.
5. Cache / Buffer Memory
   - Temporary storage used to improve read/write performance by storing frequently accessed data.

Functions of a Disk Subsystem

1. Data Storage and Retrieval
    - Stores persistent data and retrieves it efficiently when requested.
2. Fault Tolerance / Data Protection
    - Implements RAID or other redundancy mechanisms to prevent data loss.
3. Performance Optimization
    - Uses striping, caching, and high-speed interfaces to improve read/write speed.
4. Scalability
    - Can add more disks or expand storage capacity as needed.

Types of Disk Subsystems

1. Direct-Attached Storage (DAS)
    - Disks are directly connected to a single server or workstation.
2. Network-Attached Storage (NAS)
    - Disks are part of a networked storage system accessed via file-level protocols.
3. Storage Area Network (SAN)
    - High-speed network connecting servers to block-level storage arrays.
4. Content-Addressed Storage (CAS)
    - Specialized storage where data is stored and retrieved based on content identifiers.

Importance of Disk Subsystems

- Central to data integrity, availability, and performance in computing systems.
- Critical for enterprise environments, databases, virtualization, and high-performance applications.
- Helps balance cost, performance, and reliability in storage architectures.

---

1. Performance Considerations

Performance refers to how fast a storage system can read and write data. Key factors that affect performance include:

a) Disk Type

- HDD (Hard Disk Drive): Mechanical, slower, higher latency.
- SSD (Solid State Drive): Faster access times, higher IOPS (Input/Output Operations Per Second).
- NVMe SSD: Even faster than SATA/SAS SSDs due to direct PCIe connection.

b) RAID Level

- RAID 0: High performance due to striping, no redundancy.
- RAID 1: Moderate performance; mirrors data.
- RAID 5/6: Balanced performance; write speed may be slower due to parity calculations.
- RAID 10: High performance and redundancy; requires more disks.

c) Disk Interface

- SATA: Moderate speed, cost-effective.
- SAS: Faster, more reliable, suitable for enterprise.
- Fibre Channel / NVMe: High-speed interfaces for SAN and high-performance systems.

d) Caching / Buffering

- Using cache memory reduces access time by storing frequently used data temporarily.
- Improves both read and write performance.

e) Disk Subsystem Configuration

- Number of disks in RAID, striping size, and controller efficiency affect throughput and latency.

---

2. Reliability Considerations

Reliability refers to the ability of the storage system to protect data and function without failure. Key factors include:

a) Redundancy

- Implemented via RAID levels to protect against disk failures.
- Example: RAID 1, 5, 6, or 10 provide redundancy, while RAID 0 does not.

b) Fault Tolerance

- The system can continue operating even if one or more disks fail (depending on RAID level).
- Hot-swappable drives allow replacing failed disks without downtime.

c) Data Integrity

- Checksums, parity, and error-correcting codes (ECC) help detect and correct errors.

d) Backup and Recovery

- Regular backups ensure data can be restored in case of catastrophic failure.
- Snapshots and replication improve reliability in enterprise environments.

e) Disk Quality and MTBF

- MTBF (Mean Time Between Failures): Higher MTBF indicates more reliable drives.
- Enterprise-grade disks are built for 24/7 operation, unlike consumer drives.

---

1. Traditional Storage

Definition:

Traditional storage refers to storage systems where hardware and software are tightly integrated. Storage functionality is mostly provided by proprietary hardware from vendors, like SAN or NAS appliances.

Key Features:

1. Hardware-Dependent: Storage features rely on specific vendor hardware.
2. Proprietary Management: Uses vendor-specific tools for configuration and monitoring.
3. Fixed Architecture: Scaling often requires purchasing new hardware from the same vendor.
4. Performance Optimized: Often highly optimized for specific workloads.
5. Cost: Generally expensive due to proprietary hardware and licensing.

Advantages:

- High performance and reliability out of the box.
- Mature and proven in enterprise environments.
- Vendor support available for troubleshooting.

Disadvantages:

- Expensive to scale and maintain.
- Less flexible; difficult to integrate with heterogeneous hardware.
- Upgrades often require replacing hardware.

2. Software-Defined Storage (SDS)

Definition:

SDS separates storage software from underlying hardware, allowing storage management to be done via software on commodity servers or storage devices.

Key Features:

1. Hardware-Agnostic: Can run on standard servers and disks.
2. Centralized Management: Software controls storage policies, provisioning, replication, and snapshots.
3. Scalable: Easily scale-out by adding more commodity hardware.
4. Flexibility: Supports hybrid setups (HDD + SSD, different vendors).
5. Automation & Integration: Can integrate with cloud and virtualization platforms.

Advantages:

- Cost-effective; uses commodity hardware.
- Highly scalable and flexible.
- Easy to integrate with virtualization and cloud environments.
- Supports advanced features like deduplication, compression, and automated tiering.

Disadvantages:

- Performance depends on software and underlying hardware.
- Requires expertise to configure and manage.
- Can be less predictable than proprietary systems in some high-performance scenarios.

Comparison Table

Feature	**Traditional Storage**	**Software-Defined Storage (SDS)**
Hardware Dependency	Vendor-specific	Hardware-agnostic (commodity servers)
Management	Proprietary tools	Centralized software-based
Scalability	Limited; costly	Easy, scale-out architecture
Cost	High	Lower, flexible hardware choices
Flexibility	Low	High, can integrate heterogeneous devices
Automation & Features	Basic to moderate	Advanced (tiering, deduplication, snapshots)
Performance	Highly optimized	Depends on hardware and software layer
Use Cases	Enterprise SAN/NAS	Virtualization, cloud, big data, hybrid storage

UNIT II

DATA LINK AND NETWORK LAYER PROTOCOLS

Data link layer services, design issues, error detection and correction, Medium Access Control (MAC): Ethernet, CSMA/CD, CSMA/CA, Network layer functions and IP routing, store and forward packet switching, connection less and connection, oriented networks, Routing algorithms: Distance vector, Link state, OSPF, BGP, ARP, ICMP, and DHCP operations

1. Introduction to Data Link Layer

The Data Link Layer is the second layer in the OSI (Open Systems Interconnection) model. It sits above the Physical Layer and below the Network Layer. Its main role is to provide reliable data transfer across a single physical link between two directly connected nodes.

Data Link Layer Services

The Data Link Layer, which is the second layer of the OSI model, provides essential services to ensure reliable and efficient data transfer between two directly connected nodes. Its main services include:

1. Framing
   - Converts a raw bitstream from the Physical Layer into frames, which are structured units of data.
   - Each frame contains a header, payload, and trailer.
   - Makes it easier to detect errors and manage data transfer.

2. Error Detection and Correction
   - Ensures that data transmitted across a link is received accurately.
   - Common techniques:
     - Parity Check – Adds a parity bit for basic error detection.
     - Checksum – Computes a value from the data to verify integrity.
     - Cyclic Redundancy Check (CRC) – Strong error-detection method.
   - Some protocols also allow error correction through retransmission.

3. Flow Control
   - Regulates the rate of data transmission so the sender does not overwhelm the receiver.
   - Common methods:
     - Stop-and-Wait – Sender waits for acknowledgment before sending the next frame.
     - Sliding Window – Allows multiple frames in transit for higher efficiency.

4. Physical Addressing
   - Adds MAC (Media Access Control) addresses to frames so that devices on the same network can be uniquely identified.
   - Ensures that frames are delivered to the correct destination node on a LAN.

5. Media Access Control (MAC)
   - Determines how devices share the communication medium, especially in networks where multiple devices use the same channel.
   - Common MAC methods:
     - CSMA/CD (Ethernet) – Detects collisions and retransmits.
     - CSMA/CA (Wi-Fi) – Avoids collisions using a contention mechanism.

6. Reliable Delivery
   - Provides mechanisms to acknowledge received frames and retransmit lost or corrupted frames.
   - Ensures data integrity and correctness over unreliable physical links.

Data Link Layer Design Issues

1. Framing
- Issue: How to divide a continuous stream of bits from the Physical Layer into discrete frames.
- Considerations:
  - Detecting start and end of a frame.
  - Handling variable-length frames.
- Solutions:
  - Character-oriented framing – Uses special characters as frame delimiters.
  - Bit-oriented framing – Uses specific bit patterns to indicate frame boundaries (e.g., HDLC).
  - Physical layer coding violation – Detecting frame boundaries using illegal physical codes.

2. Error Control
- Issue: How to detect and correct errors introduced by the Physical Layer.
- Considerations:
  - Single-bit or burst errors.
  - How to recover lost or corrupted data.
- Solutions:
  - Error detection codes: Parity, checksum, CRC.
  - Error correction techniques: Automatic Repeat Request (ARQ), Forward Error Correction (FEC).

3. Flow Control
- Issue: How to prevent the sender from overwhelming the receiver.
- Considerations:
  - Receiver buffer size.
  - Variable transmission rates.
- Solutions:
  - Stop-and-Wait – Sender waits for acknowledgment before sending the next frame.
  - Sliding Window – Allows multiple frames to be sent before receiving an acknowledgment.

4. Addressing
- Issue: How to deliver frames to the correct device on a shared link.
- Considerations:
  - Unique identification of devices in a LAN.
- Solution:
  - Use MAC (Media Access Control) addresses in frame headers.

5. Medium Access Control
- Issue: How to manage access to a shared communication medium and avoid collisions.

- Considerations:
  - Multiple devices trying to transmit simultaneously.
- Solutions:
  - Controlled access protocols: Polling, token passing.
  - Random access protocols: ALOHA, CSMA/CD (Ethernet), CSMA/CA (Wi-Fi).

6. Reliability

- Issue: How to ensure that frames are delivered accurately and in order.
- Considerations:
  - Lost, delayed, or duplicated frames.
- Solutions:
  - Frame numbering and sequencing.
  - Acknowledgments and retransmissions.

ERROR DETECTION AND CORRECTION

Error Detection

In the Data Link Layer (DLL) of the OSI model, error detection is the process of identifying errors that may have occurred during data transmission. These errors are often caused by electrical interference, signal degradation, or hardware faults. While error detection can tell you that an error has occurred, it typically does not correct the error. Instead, it signals the need to discard the corrupted data and request a retransmission from the sender.

Common Error Detection Methods

The Data Link Layer employs several techniques to detect errors. These methods work by adding a small amount of redundant information to the data frame before it is transmitted. The receiver then uses this redundant information to check the integrity of the received frame.

1. Simple Parity Check
2. Cyclic Redundancy Check (CRC)
3. Checksum

Common Techniques:

1. Parity Check

- How it works: A single bit, called a parity bit, is added to a block of data. The value of this bit is chosen so that the total number of 1s in the data block is either even (even parity) or odd (odd parity).
- Detection: The receiver counts the 1s in the received data. If the count does not match the parity scheme, an error is detected.
- Limitations: It can only detect an odd number of bit errors. If two bits are flipped, the parity check will still pass, and the error will go undetected.
  - Example: For 1011 (odd parity), parity bit = 0 → 10110
  - Pros: Simple and fast.
  - Cons: Can detect only single-bit errors; fails with multiple-bit errors.

Types of Parity Check

a) Even Parity

- The parity bit is set so that the total number of 1s in the data including the parity bit is even.

- Example:
  - Data: 1011 (3 ones → odd)
  - Parity bit: 1 (to make total 4 → even)
  - Transmitted: 10111

b) Odd Parity

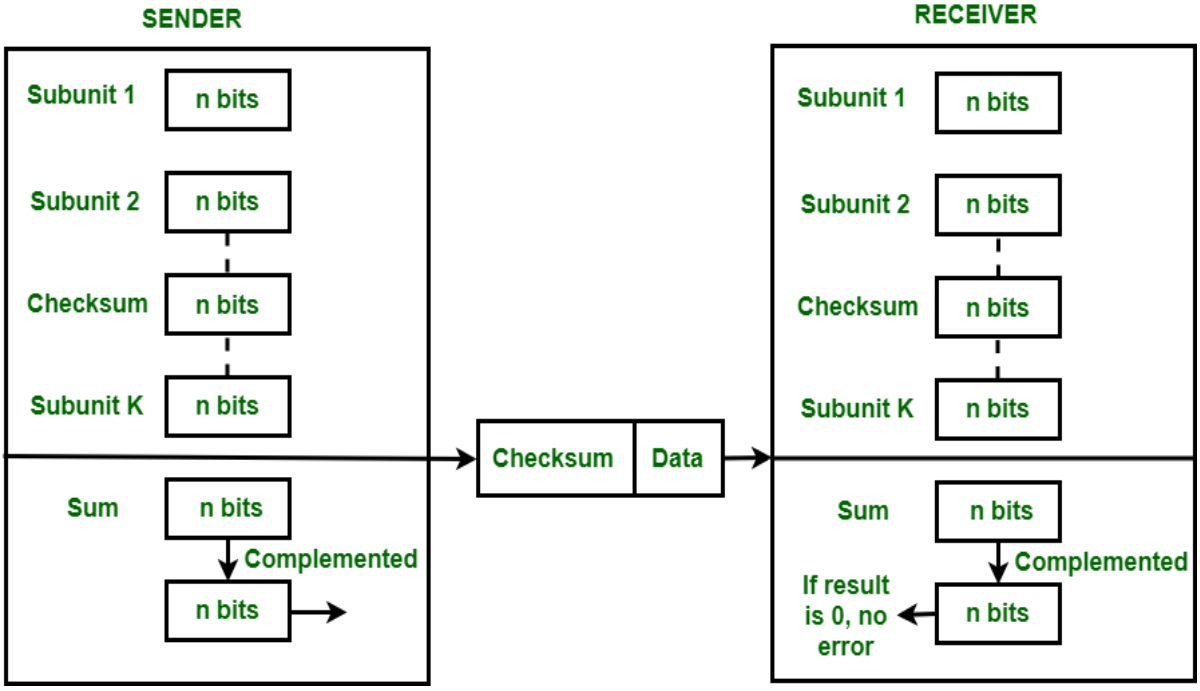
- The parity bit is set so that the total number of 1s in the data including the parity bit is odd.
- Example:
  - Data: 1010 (2 ones → even)
  - Parity bit: 1 (to make total 3 → odd)
  - Transmitted: 10101

Data Bits	Parity Type	Parity Bit	Transmitted Data	Error Detected?
1011	Even	1	10111	Yes, if 1-bit error occurs
1010	Odd	1	10101	Yes, if 1-bit error occurs

2. Checksum

- How it works: The sender divides the data into equal-sized segments and adds them up. The sum is then complemented (all 0s become 1s and vice-versa) to get the checksum, which is appended to the data.
- Detection: The receiver adds up all the received segments, including the checksum. The final sum should be zero. If it isn't, an error is detected.
- Use: While used in the Internet Layer (e.g., in the IP header), its use in the Data Link Layer is less common than CRC, as it's less robust.

  - Computes a sum of all data bytes in the frame and sends it along with the data.
  - Receiver recalculates the sum and compares it with the transmitted checksum.
  - Pros: Simple and works for multiple-bit errors.
  - Cons: Not as robust as CRC for detecting all error patterns.



How Checksum Works

Sender Side:

1. Data is divided into equal-sized units (e.g., 8-bit bytes or 16-bit words).
2. All units are added together using binary addition.
3. The sum may be complemented or truncated to fit a specific size.
4. The resulting value is sent as the checksum along with the data.

Receiver Side:

1. Receives the data and checksum.
2. Adds all data units together in the same manner.
3. Compares the computed sum with the received checksum:
    - Match: No error detected
    - Mismatch: Error detected

Example

Suppose a sender wants to transmit three 8-bit data bytes:

Data Byte	Binary
1	10110011
2	11001100
3	11110000

Step 1 – Compute Sum:

- Add all bytes together:
  10110011 + 11001100 + 11110000 = 1 11111111 (overflow bit ignored or wrapped around)

Step 2 – Compute Checksum:

- Take complement of sum (optional, depending on protocol).

Step 3 – Transmission:

- Send data bytes along with checksum.

Step 4 – Receiver Verification:

- Receiver adds all data bytes and the checksum.
- If result = 0 (or expected value), no error; otherwise, an error is detected.

3. Cyclic Redundancy Check (CRC)

- How it works: The sender treats the data block as a large binary number and divides it by a pre-agreed-upon divisor (generator polynomial). The remainder of this division is the CRC checksum, which is appended to the data frame.
- Detection: The receiver performs the same division on the received data. If the new remainder is not zero, an error is detected.
- Detection capability: CRC is very effective at detecting a wide range of common errors, including burst errors (multiple consecutive bits flipped). It is the most common and robust error detection method used in the DLL, found in Ethernet and Wi-Fi.
    - Treats data as a polynomial, divides it by a generator polynomial, and appends the remainder to the data.
    - Receiver performs the same division; if the remainder is zero, the data is considered correct.

- Pros: Very powerful; can detect burst errors.
- Cons: Slightly more complex to implement than parity or checksum.

Working Steps of CRC:

1. Message Polynomial:
   Represent the binary data (message) as a polynomial M(x).
2. Generator Polynomial (G(x)):
   A predefined divisor polynomial known to both sender and receiver. Example: $x^3 + x + 1$ (binary 1011).
3. Append Zeros:
   Add r zeros to the end of the message, where r = degree of G(x).
4. Binary Division:
   Divide the new message (M(x) × x^r) by G(x) using modulo-2 division (XOR instead of subtraction).
5. CRC Remainder:
   The remainder from this division is the CRC bits.
6. Transmit:
   Append CRC bits to the original message and transmit it.
7. Receiver Side:
   Receiver divides the received bits by the same G(x).
   - If remainder = 0, no error.
   - If remainder ≠ 0, error detected.

Example of CRC:

- Message: 1101
- Generator (G(x)): 1011 (degree = 3 → append 3 zeros)
- Message after appending zeros: 1101000
- Perform modulo-2 division by 1011.
- Remainder (CRC): 011
- Transmitted Data: 1101 011

Receiver checks again using the same process.

Example CRC Problem

- Message bits (M): 100100
- Generator polynomial (G): 1101 (degree = 3)

Step 1: Append zeros

- Since generator is degree 3 → append 3 zeros to message:
- Dividend = 100100000

Step 2: Perform Modulo-2 Division
We divide 100100000 by 1101.
(Remember: XOR replaces subtraction.)
Long Division (binary):

```
1101 ) 100100000
       1101        ← XOR with first 4 bits
       -------
        1010
        1101       ← XOR again
        -------
        1110
         1101
         -------
          0110
```

Now bring down remaining zeros step by step:
0110 → bring down 0 → 1100
1100 XOR 1101 = 0001
Bring down 0 → 010
Not enough bits, bring down 0 → 100
100 XOR 1101 = 001 (remainder)
Step 3: Final remainder (CRC)
The remainder = 001

Step 4: Transmitted Frame
Attach remainder to original message:
Message = 100100
CRC = 001
Transmitted Frame = 100100001
At the receiver side, if you divide 100100001 by 1101,
the remainder will be 0 (means no error).

Error Correction

Error correction allows the receiver not only to detect errors but also to correct them without asking for retransmission.

Common Techniques:

1. Automatic Repeat Request (ARQ)
    - The receiver detects errors and requests retransmission of corrupted frames.
    - Types of ARQ:
        - Stop-and-Wait ARQ – Sender waits for acknowledgment before sending the next frame.
        - Go-Back-N ARQ – Sender can transmit multiple frames, but if an error occurs, all subsequent frames are resent.
        - Selective Repeat ARQ – Only the corrupted frames are retransmitted.
    - Pros: Reliable; ensures correct data delivery.
    - Cons: May increase transmission delay due to retransmissions.
2. Forward Error Correction (FEC)
    - Adds redundant bits to the data so that the receiver can correct certain errors without retransmission.
    - Examples: Hamming code, Reed-Solomon code.

- Pros: Useful for links with high latency where retransmission is costly (e.g., satellite links).
- Cons: Extra overhead for redundancy; cannot correct all errors if they exceed correction capacity.

Hamming code

Definition

Hamming Code is an error-correcting code invented by Richard Hamming. It can:

- Detect up to 2-bit errors
- Correct single-bit errors

It is widely used in computer memory (ECC RAM), telecommunication, and data transmission.

The Hamming Code is a single-error-correcting, double-error-detecting linear block code used in digital communication to detect and correct errors. It adds redundant parity bits to a data message to achieve this.

Here's how it works:

1. Calculating Parity Bits: The Hamming Code uses a formula to determine how many parity bits are needed for a given number of data bits. The formula is $2p \geq m+p+1$, where p is the number of parity bits and m is the number of data bits. The parity bits are placed at positions that are powers of 2 (1, 2, 4, 8, etc.). The other positions are filled with the data bits.
2. Parity Bit Calculation: Each parity bit is calculated based on a subset of the data bits. The parity bit at position 2k is an even parity check for all positions that have the kth bit set in their binary representation. For example:
   - P1 (position 1): Checks all bits at odd positions (1, 3, 5, 7, 9, 11...).
   - P2 (position 2): Checks all bits in groups of 2 (2, 3, 6, 7, 10, 11...).
   - P4 (position 4): Checks all bits in groups of 4 (4, 5, 6, 7, 12, 13...). This systematic arrangement of checks is the key to its error-correcting capability.
3. Error Detection and Correction: When the receiver gets the message, it re-calculates the parity bits. If all parity checks pass, there is no error. If one or more parity checks fail, the position of the error can be determined by the binary value of the failed checks. For example, if P1 and P4 checks fail, the error is at position 5 (1 + 4 = 5). The receiver can then flip the bit at that position to correct the error. The code can also detect, but not correct, two-bit errors.

Example: Encode 4-bit data 1011

Step 1: Place parity bits

Data = 1 0 1 1 (4 bits)

We need r = 3 (since $23=8 \geq 4+3+12^3 = 8 \geq 4+3+123=8 \geq 4+3+1$)

So total = 7 bits (4 data + 3 parity).

Positions:

1 2 3 4 5 6 7

p1 p2 d1 p4 d2 d3 d4

Now insert data (1011) into positions:

_ _ 1 _ 0 1 1

Step 2: Calculate parity bits (even parity assumed)

- p1 covers bits (1,3,5,7): → (p1, d1=1, d2=0, d4=1) → 1+0+1=2 (even) → p1=0

- p2 covers bits (2,3,6,7): → (p2, d1=1, d3=1, d4=1) → 1+1+1=3 (odd) → p2=1
- p4 covers bits (4,5,6,7): → (p4, d2=0, d3=1, d4=1) → 0+1+1=2 (even) → p4=0

Final encoded word:
0 1 1 0 0 1 1

Medium Access Control (MAC)

Definition

Medium Access Control (MAC) is a sublayer of the Data Link Layer in the OSI model. It controls how devices share and access the transmission medium (like cables or wireless channels) to avoid collisions and ensure efficient communication.

MAC works just above the Physical Layer and just below the Logical Link Control (LLC) sublayer.

Functions of MAC

1. Channel Allocation – Decide which device can use the medium at a given time.
2. Addressing – Assigns MAC addresses (48-bit unique IDs) to network devices.
3. Framing – Defines how data is packaged for transmission.
4. Error Detection – Uses CRC or checksum to detect errors in frames.
5. Collision Handling – Prevent or recover from collisions in shared media.
6. Flow Control – Ensure sender does not overwhelm the receiver.

MAC Techniques (Access Methods)

1. Random Access Protocols (contention-based)

Devices compete for the channel.

- ALOHA
  - *Pure ALOHA*: Send anytime, retransmit if collision occurs.
  - *Slotted ALOHA*: Time is divided into slots; reduces collisions.
- CSMA (Carrier Sense Multiple Access)
  - Listen before sending (sense the carrier).
  - Variants:
    - 1-persistent CSMA (send immediately if idle)
    - Non-persistent CSMA (wait random time if busy)
    - p-persistent CSMA (send with probability p in slotted systems).
- CSMA/CD (Collision Detection)
  - Used in Ethernet.
  - Detects collision during transmission and retries after random delay.
- CSMA/CA (Collision Avoidance)
  - Used in Wi-Fi.
  - Avoid collisions using RTS/CTS (Request to Send / Clear to Send).

2. Controlled Access Protocols (no contention)

The channel is allocated in an organized way.

- Polling – A central controller asks each station if it has data.
- Reservation – Stations reserve slots in advance.

- Token Passing – A “token” (special frame) is passed around; only the device with the token can send (used in Token Ring, FDDI).

3. Channelization Protocols (sharing by splitting resources)

The channel is divided among users.

- FDMA (Frequency Division Multiple Access) – Each user gets a separate frequency band.
- TDMA (Time Division Multiple Access) – Each user gets a time slot.
- CDMA (Code Division Multiple Access) – All users transmit simultaneously but with unique codes.
- OFDMA (Orthogonal FDMA) – Used in 4G/5G, divides channel into subcarriers.

MAC Address

- Unique 48-bit identifier for each device’s Network Interface Card (NIC).
- Written in Hexadecimal, e.g., 08:00:27:AF:32:19.
- First 24 bits = manufacturer (OUI), last 24 bits = device ID.

Medium Access Control (MAC) in Ethernet

Definition

In Ethernet networks, the MAC sublayer of the Data Link Layer manages how devices access and share the common transmission medium (cable or wireless). It ensures efficient frame transmission, addressing, and collision handling.

Key Roles of MAC in Ethernet

1. Framing – Encapsulates data from the upper layers into Ethernet frames (header + data + CRC).
2. Addressing – Uses 48-bit MAC addresses to identify source and destination devices.
    - Example: 00:1A:2B:3C:4D:5E
    - Ensures frames are delivered to the correct NIC.
3. Error Detection – Uses CRC (Cyclic Redundancy Check) in the Frame Check Sequence (FCS) field.
4. Access Control – Prevents multiple devices from transmitting at the same time.

Ethernet in MAC

Where it fits

Ethernet works at the Data Link Layer of the OSI model, which has two sublayers:

1. Logical Link Control (LLC) – Provides services to the Network Layer.
2. Medium Access Control (MAC) – Manages how frames are sent and received over the shared medium.

👉 So, Ethernet MAC = the set of rules and protocols that decide *when* and *how* devices transmit Ethernet frames.

Functions of MAC in Ethernet

1. Framing
    - Adds headers and trailers to create Ethernet frames.

- Includes source MAC address, destination MAC address, and CRC.

2. Addressing
   - Uses 48-bit MAC addresses (unique to each device's NIC).
   - Example: 00-14-22-3B-59-4A.
3. Error Detection
   - Uses CRC (Cyclic Redundancy Check) in the Frame Check Sequence (FCS) to detect errors.
4. Access Control (Medium Sharing)
   - Old Ethernet (bus topology / hubs): Used CSMA/CD (Carrier Sense Multiple Access with Collision Detection).
     - Devices listen before sending.
     - If collision → stop + retransmit after random backoff.
   - Modern Ethernet (switches): Uses Full-duplex mode.
     - Each device has a dedicated link.
     - No collisions, so CSMA/CD is no longer needed.

1. CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

- Used in: Ethernet (wired LANs, legacy hubs, coaxial bus).
- Working:
  1. Device listens to channel before sending.
  2. If channel is idle → transmit.
  3. If another device transmits at the same time → collision occurs.
  4. Both detect collision → send jam signal → stop transmission.
  5. Devices wait a random backoff time (binary exponential backoff) → retry.
- Why possible?
  - In wired networks, stations can transmit and listen simultaneously, so they can detect collisions.

2. CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)

- Used in: Wireless LANs (Wi-Fi, IEEE 802.11).
- Working:
  1. Device listens to channel before sending.
  2. If channel idle → wait for DIFS (Distributed Inter-Frame Space).
  3. If busy → wait + set a random backoff timer.
  4. Optional: Use RTS/CTS (Request to Send / Clear to Send) handshake to avoid hidden node collisions.
  5. Transmit data → receiver sends ACK if received correctly.
  6. If no ACK → sender assumes collision → retries after backoff.
- Why needed?
  - In wireless, devices cannot detect collisions (because they cannot listen and transmit at the same time).
  - Hidden & exposed node problems → collision *avoidance* is better.

Network layer functions and IP routing

Network Layer Functions

The Network Layer (Layer 3 in OSI) is responsible for end-to-end delivery of packets across multiple networks.

Main Functions

1. Logical Addressing (IP Addressing)
   - Provides unique logical addresses (IP addresses) to identify devices across networks.
   - Example: IPv4 (32-bit), IPv6 (128-bit).
2. Routing
   - Finds the best path from source to destination across routers.
   - Uses routing algorithms & tables.
3. Packet Forwarding
   - Once the route is known, packets are forwarded hop by hop toward the destination.
4. Fragmentation & Reassembly
   - If a packet is too large for a network's MTU (Maximum Transmission Unit), it is fragmented.
   - Destination reassembles fragments.
5. Error Handling & Diagnostics
   - Provides error messages (e.g., ICMP – Internet Control Message Protocol).
   - Example: ping uses ICMP to test connectivity.

IP Routing

What is IP Routing?

IP routing is the process of selecting a path for data packets to travel across one or more networks. A router is a specialized device that performs this function. When a packet arrives at a router, the router examines its destination IP address and forwards the packet to the next network hop on its journey to the final destination.

How It Works

1. Packet Arrival: A data packet, containing its destination IP address, arrives at a router's interface.
2. Routing Table Lookup: The router checks its routing table for a matching network address. The routing table is a database that stores information about the network's topology and the best paths to different destinations.
3. Path Selection: Based on the routing table, the router determines the next best "hop"—the next router or destination host—to forward the packet to. This decision is based on various factors, including the shortest path, network congestion, and link cost.
4. Packet Forwarding: The router forwards the packet to the appropriate output interface, sending it closer to its final destination. This process repeats at every router along the path until the packet reaches its final destination.

Types of Routing

- Static Routing: An administrator manually configures all routes in the routing table. This is simple for small networks but becomes unmanageable and inflexible for large, dynamic networks.

- Dynamic Routing: Routers use routing protocols to automatically share and update routing information with each other. This allows the network to adapt to changes in topology, such as when a link goes down.
    - Distance Vector Protocols (e.g., RIP): Routers share their routing table with their immediate neighbors. They learn the best path by counting the number of hops (router jumps) to a destination.
    - Link-State Protocols (e.g., OSPF): Routers share the state of their links with all other routers in the network. This gives each router a complete map of the network, allowing it to calculate the most efficient path.

The Routing Table 📋

A routing table is the heart of the routing process. It typically contains the following information for each route:

- Destination Network: The network address of the destination.
- Next Hop: The address of the next router to send the packet to.
- Interface: The outgoing port on the router to use.
- Metric: A value used to determine the "cost" of the route, which helps in selecting the best path.

IP Routing is the process of selecting a path for IP packets to travel from source to destination across interconnected networks.

🔑 **Steps in Routing**

1. Routing Table Lookup
    - Routers maintain a routing table with destination networks and next-hop information.
    - Each incoming packet is matched to the longest prefix match in the table.
2. Best Path Selection
    - Based on metrics like hop count, bandwidth, delay, cost.
3. Packet Forwarding
    - Router forwards the packet to the next-hop router or final destination.

🛠 **Types of Routing**

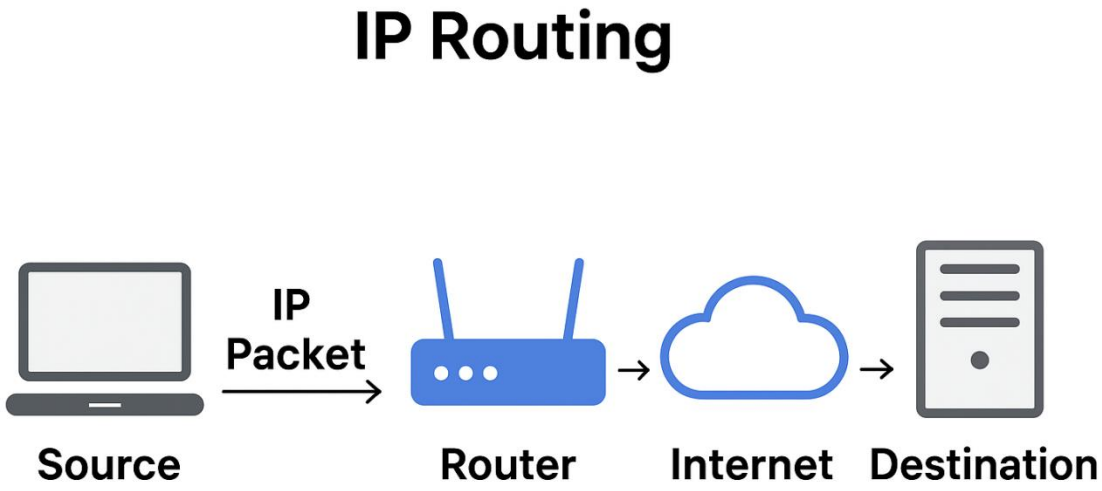
1. **Static Routing**
    - Routes are manually configured by admin.
    - Simple, but not scalable.
    - Example: ip route add ... in Cisco/Unix.
2. **Dynamic Routing**
    - Routers exchange routing information automatically.
    - Protocols:
        - Distance Vector – RIP (Routing Information Protocol)
        - Link State – OSPF (Open Shortest Path First)
        - Hybrid – EIGRP (Enhanced Interior Gateway Routing Protocol)
3. **Default Routing**
    - A single default route is used when no specific route is found.
    - Example: Internet traffic often goes through a default gateway.

🗂 Routing Protocol Categories

- Interior Gateway Protocols (IGP): Used inside an organization (e.g., RIP, OSPF, EIGRP).
- Exterior Gateway Protocols (EGP): Used between organizations (e.g., BGP – Border Gateway Protocol, backbone of the Internet).

Example of Routing

- Source: 192.168.1.10 (PC in LAN)
- Destination: 8.8.8.8 (Google DNS)
- Process:
    1. PC sends packet to default gateway (router).
    2. Router checks routing table → finds best next hop.
    3. Packet is forwarded router by router until destination.
    4. Response comes back via reverse route.



Store-and-Forward Packet Switching

Definition

Store-and-forward is a packet switching technique in which every intermediate node (like a router or switch) receives the entire packet, stores it temporarily in memory (buffer), checks it for errors, and then forwards it to the next node.

This ensures that only error-free packets are transmitted further.

🔑 Key Characteristics

1. Complete Packet Reception:
    - The entire packet must be received before forwarding begins.
2. Error Checking:
    - Nodes check packets using error detection techniques (e.g., CRC).
    - Corrupted packets are discarded.
3. Buffering:
    - Temporary storage is used at each hop.
    - Helps handle congestion and traffic bursts.

4. Reliable Transmission:
    - Ensures only valid packets are forwarded.
    - But adds processing delay.

🛠 Advantages

- High reliability (corrupted packets are not propagated).
- Better error control.
- Supports variable-length packets.

⚠ Disadvantages

- Higher latency (must receive the whole packet before forwarding).
- Requires more memory (buffers at each hop).
- Slower than cut-through switching.

📌 Example in Use

- Wide Area Networks (WANs): Used in packet-switched networks like X.25 and Frame Relay.
- Modern Routers: Often use store-and-forward when error detection is needed.

Example of Store-and-Forward Packet Switching

Scenario:

- Source: Computer A (in New York)
- Destination: Computer B (in London)
- Intermediate Routers: R1 (New York), R2 (Paris), R3 (London)

Step-by-Step Process:

1. Computer A → R1
    - A sends a packet of 1500 bytes to R1.
    - R1 waits until it receives the entire packet, stores it in memory.
    - R1 checks for errors (using CRC).
    - If valid → forwards to R2. If not → drops packet.
2. R1 → R2
    - Once R1 has the full packet, it sends it across the Atlantic link to R2.
    - R2 again stores the full packet, checks for errors, then forwards it.
3. R2 → R3
    - R2 sends the error-free packet to R3 (London).
    - R3 stores it, verifies it, then passes it to Computer B.
4. Computer B Receives
    - B finally receives the complete error-checked packet.

🔍 Example in Action:

- Suppose packet size = 1500 bytes,
- Link speed = 1 Mbps (1,000,000 bits per second).
- Time to transmit packet =

1500×81,000,000=0.012 seconds (12 ms)\frac{1500 \times 8}{1,000,000} = 0.012 \text{ seconds (12 ms)}1,000,0001500×8=0.012 seconds (12 ms)

- At each router, an extra 12 ms delay occurs (because packet must be fully received first).

So, across 3 routers:

- Total delay ≈ 12 ms × 3 = 36 ms + propagation delay.

This shows why store-and-forward is slower than cut-through switching.

Connection Less And Connection, Oriented Networks

Connection-Oriented Networks 🤝

These networks are similar to a telephone call: you must first dial and get a response before you can begin a conversation. This setup process, known as a three-way handshake, ensures that both the sender and receiver are ready for data transfer.

- Reliability: High. Packets are guaranteed to arrive in the correct order, and the protocol includes mechanisms for error detection, acknowledgments, and retransmission if a packet is lost.
- Overhead: High. The initial setup, ongoing session management, and final teardown of the connection add overhead.
- Examples: The Transmission Control Protocol (TCP) is the most common example, used for applications where reliability is critical, such as email, file transfers, and web browsing.

Connectionless Networks 📭

Think of this as sending a letter in the mail. You simply address the packet and send it off, without any prior notification or guarantee that the recipient is ready. Each data packet is treated as an independent unit and may take a different path to its destination.

- Reliability: Low. Packets may arrive out of order, be lost, or duplicated. There are no built-in mechanisms for error handling or retransmission.
- Overhead: Low. The absence of a connection setup and management makes these networks very efficient.
- Examples: The User Datagram Protocol (UDP) is a prime example, used for applications where speed and low latency are more important than guaranteed delivery, such as video streaming, online gaming, and VoIP.

Feature	**Connection-Oriented**	**Connectionless**
Setup	Requires a connection to be established (e.g., TCP handshake)	No setup phase, data sent directly
Reliability	Guaranteed delivery, error handling, and retransmission	No guarantee, data may be lost
Order of Data	Maintains the order of data packets	No guarantee of order
Protocols	TCP, SCTP, etc.	UDP, IP, etc.
Overhead	Higher (due to connection management)	Lower (no connection management)
Use Cases	Web browsing, file transfers, emails	Streaming, real-time gaming, DNS

Routing Algorithms

Definition

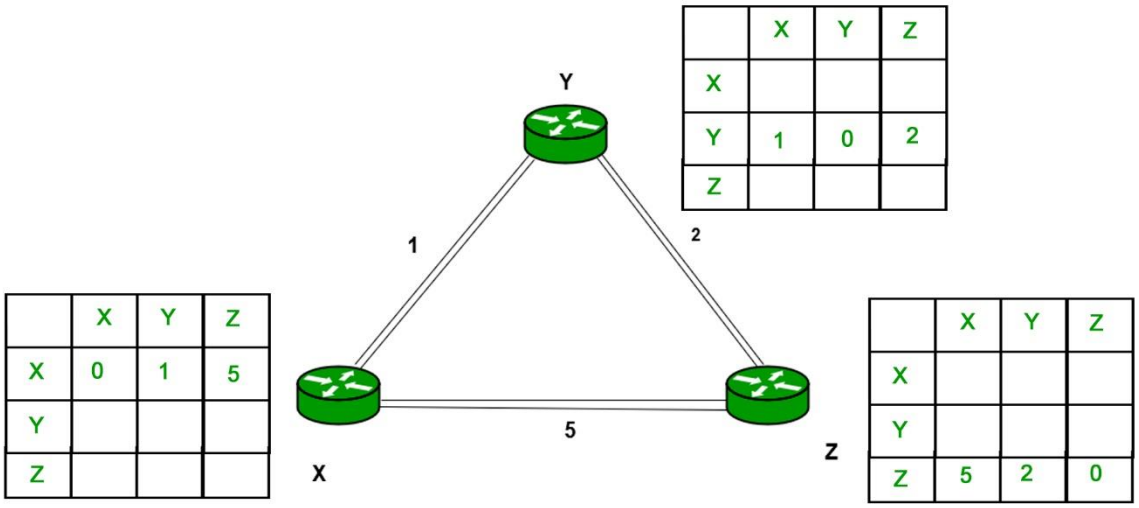
A routing algorithm is the method used by routers to decide the best path for forwarding packets from source to destination across a network.

Routers use these algorithms to build and update their routing tables.

Distance Vector Routing (DVR) Protocol is a method used by routers to find the best path for data to travel across a network. Each router keeps a table that shows the shortest distance to every other router, based on the number of hops (or steps) needed to reach them. Routers share this information with their neighbors, allowing them to update their tables and find the most efficient routes. This protocol helps ensure that data moves quickly and smoothly through the network.

How Distance Vector Algorithm works?

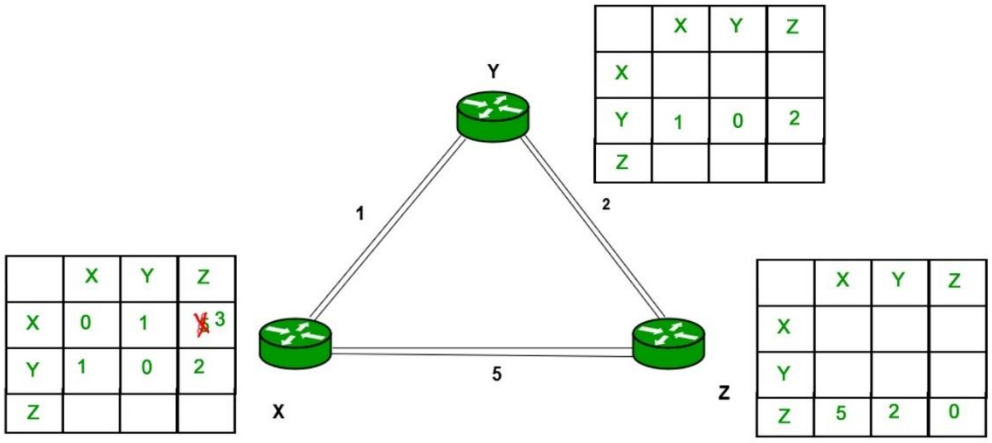
- A[router]transmits its distance vector to each of its neighbors in a routing packet.
- Each router receives and saves the most recently received distance vector from each of its neighbors.
- A router recalculates its distance vector when:
    - It receives a distance vector from a neighbor containing different information than before.
    - It discovers that a link to a neighbor has gone down.
- Example :
- Consider 3-routers X, Y and Z as shown in figure. Each router have their routing table. Every routing table will contain distance to the destination nodes.



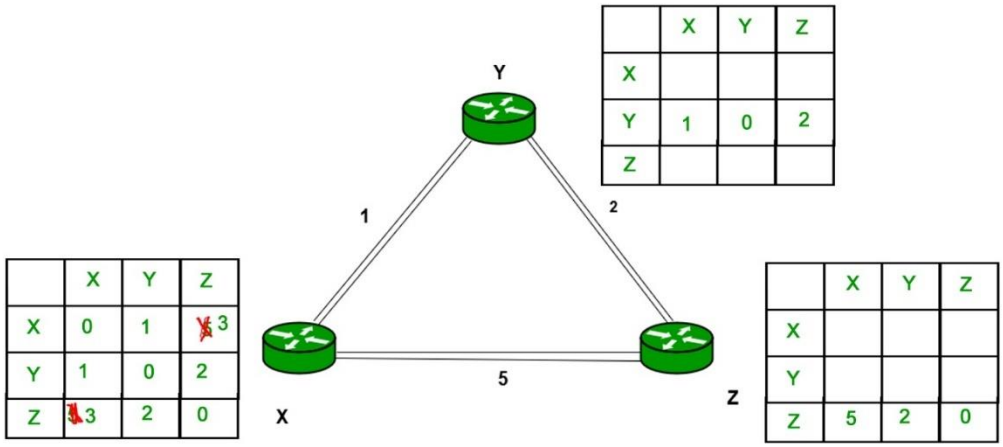
Consider router X , X will share it routing table to neighbors and neighbors will share it routing table to it to X and distance from node X to destination will be calculated using bellmen- ford equation.

Dx(y) = min { C(x,v) + Dv(y)} for each node y ? N

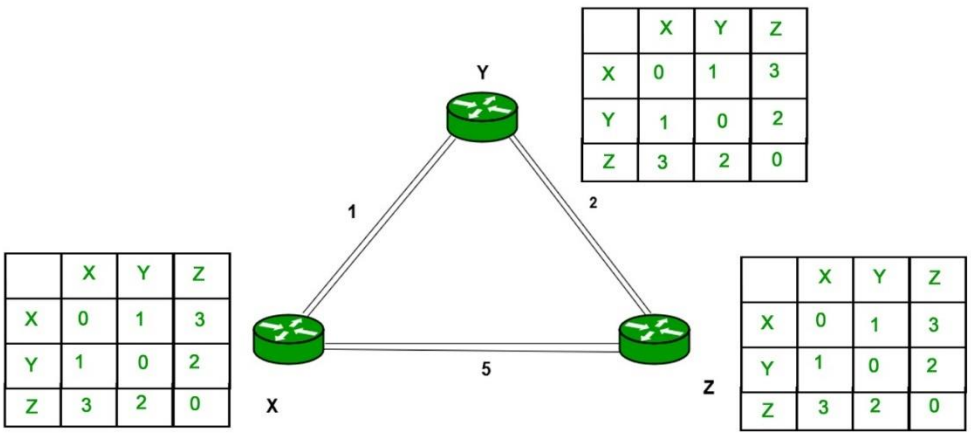
As we can see that distance will be less going from X to Z when Y is intermediate node(hop) so it will be update in routing table X.



Similarly for Z also –



Finally the routing table for all –



Applications of Distance Vector Routing Algorithm

The Distance Vector Routing Algorithm has several uses:

- Computer Networking : It helps route data packets in networks.
- Telephone Systems : It's used in some telephone switching systems.
- Military Applications : It has been used to route missiles.

Advantages of Distance Vector routing

- Shortest Path : Distance Vector Routing finds the shortest path for data to travel in a network.
- Usage : It is used in local, metropolitan, and wide-area networks.
- Easy Implementation : The method is simple to set up and doesn't require many resources.

Disadvantages of Distance Vector Routing Algorithm

- It is slower to converge than link state.
- It is at risk from the count-to-infinity problem.
- It creates more traffic than link state since a hop count change must be propagated to all routers and processed on each router. Hop count updates take place on a periodic basis, even if there are no changes in thenetwork topology , so bandwidth -wasting broadcasts still occur.
- For larger networks, distance vector routing results in larger routing tables than link state since each router must know about all other routers. This can also lead to congestion onWAN links.

---

3. Link State Routing

- Idea: Each router has a map of the entire network topology.
- Routers flood the network with Link State Advertisements (LSAs).
- Algorithm: Dijkstra's Shortest Path First (SPF).
- Metric: Cost (based on bandwidth, delay).
- Example: OSPF, IS-IS.
- ✅ Faster convergence, more scalable.

Key Steps in Link State Routing

1. Neighbor Discovery:
   Each router identifies its directly connected neighbors and the cost (metric) to reach them.
2. Link State Advertisement (LSA):
   Each router creates a packet containing its link information (neighbors + costs).
3. Flooding:
   The LSAs are flooded across the entire network. Every router receives LSAs from all others.
4. Topology Database:
   Each router builds a complete map of the network topology from received LSAs.
5. Shortest Path Calculation (Dijkstra's Algorithm):
   Using the map, each router independently computes the shortest path tree and derives its routing table.

✅ **Features of Link State Routing**

- Uses Dijkstra's Algorithm for shortest path calculation.
- Routers know the entire network topology, not just next-hop info.

- Updates are sent only when a change occurs (not periodically).
- Faster convergence compared to distance vector routing.
- More scalable and reliable for large networks.

⚖️ Advantages

- Fast convergence → less chance of routing loops.
- Accurate routing due to complete topology knowledge.
- Efficient updates → only changes are flooded.

❌ Disadvantages

- Complex implementation (needs more CPU and memory).
- Flooding overhead in very large networks.

🛠️ Example

Consider a network of 4 routers (A–B–C–D).

- Router A knows links: (A–B: 2), (A–C: 5).
- Router B knows links: (B–A: 2), (B–C: 3), (B–D: 4).
- Router C knows links: (C–A: 5), (C–B: 3), (C–D: 2).
- Router D knows links: (D–B: 4), (D–C: 2).

After LSAs are exchanged and flooded, every router knows the whole network. Each then runs Dijkstra's Algorithm to compute shortest paths.

📌 Real-World Protocols Using Link State Routing

- OSPF (Open Shortest Path First)
- IS-IS (Intermediate System to Intermediate System)

🔷 OSPF (Open Shortest Path First)

Definition:

OSPF is a Link State Routing Protocol used in IP networks. It's designed for large, complex networks and is widely used inside organizations (Interior Gateway Protocol – IGP).

🔑 Key Characteristics of OSPF

1. Link State Protocol → uses Link State Advertisements (LSAs).
2. Algorithm Used → Dijkstra's Shortest Path First (SPF).
3. Administrative Distance → 110 (used when comparing with other protocols).
4. Supports Hierarchical Routing with Areas (e.g., backbone Area 0).
5. Convergence is fast compared to Distance Vector protocols.
6. Metric → Cost (based on link bandwidth, usually: Cost = 100 / Bandwidth in Mbps).

🔄 OSPF Working Process

1. Neighbor Discovery
   Routers exchange Hello packets to identify neighbors on the same link.
2. Database Exchange
   Routers exchange LSAs describing their links and costs.
3. Flooding LSAs
   Each router floods LSAs to all other routers in the area.
4. Topology Database
   Each router builds a Link State Database (LSDB) with full network topology.
5. Shortest Path Calculation
   Each router independently runs Dijkstra's SPF algorithm to compute the best paths.

6. Routing Table Update
   The shortest paths are added to the routing table.

✅ Advantages of OSPF
- Scalable → supports large hierarchical networks with Areas.
- Efficient → only changes are flooded, not entire routing tables.
- Fast Convergence → quickly adapts to network changes.
- Supports VLSM & CIDR → classless addressing.
- Authentication → provides security for routing updates.

❌ Disadvantages of OSPF
- Complex configuration compared to RIP.
- More resource-intensive (CPU, memory).
- Flooding overhead in very large networks if not structured into areas.

🔑 OSPF Network Types
- Point-to-Point (direct link between routers)
- Broadcast (Multi-access) (e.g., Ethernet – DR/BDR election happens)
- Non-Broadcast Multi-Access (NBMA) (e.g., Frame Relay, ATM)

🛠 Example Topology (Simple)
- Routers: R1 – R2 – R3 – R4

For example:
If R1–R2 cost = 10, R2–R3 cost = 5, R1–R4 cost = 20,
then shortest path from R1 to R3 = R1 → R2 → R3 (cost 15), instead of R1 → R4 → R3 (cost 25).

📌 Real-World Use
- Used inside organizations (ISP backbones, enterprise LAN/WAN).
- Usually combined with BGP for external routing.

🔷 BGP (Border Gateway Protocol)

Definition:
BGP is a path vector routing protocol used to exchange routing information between Autonomous Systems (ASes) on the Internet. It is the protocol that literally makes the Internet work by connecting ISPs, enterprises, and data centers.
- Type: Exterior Gateway Protocol (EGP)
- Algorithm: Path Vector Protocol
- Administrative Distance: 20 (External BGP), 200 (Internal BGP)
- Port: TCP 179

🗝 Key Characteristics of BGP
1. Policy-based Routing → decisions are made not only by shortest path but also by policies (business agreements, security, etc.).
2. Path Vector → advertises the full AS-path (list of ASes a route has passed).
3. Scalable → designed for large networks like the Internet.
4. Loop Prevention → uses the AS-PATH attribute to avoid loops.
5. Reliable Transport → runs on TCP (not directly on IP like OSPF/RIP).

🔄 Types of BGP
1. eBGP (External BGP)
   - Used between routers in different ASes (ISP to ISP, ISP to enterprise).

2. iBGP (Internal BGP)
    - Used between routers in the same AS.
    - Ensures all routers in the AS know external routes.

🔁 BGP Path Selection Process

When multiple routes exist, BGP selects the best path using attributes in order of preference:

1. Highest Local Preference (policy preference within AS).
2. Shortest AS-PATH (fewer AS hops preferred).
3. Lowest Origin Type (IGP < EGP < Incomplete).
4. Lowest MED (Multi-Exit Discriminator).
5. Prefer eBGP over iBGP.
6. Lowest IGP metric to next-hop.
7. Oldest route.
8. Lowest Router ID (as a final tie-breaker).

✅ Advantages of BGP

- Scales globally (used in the Internet backbone).
- Policy-based control over routing.
- Loop-free via AS-PATH attribute.
- Supports route aggregation (CIDR).

❌ Disadvantages of BGP

- Complex configuration & management.
- Slower convergence compared to OSPF/IS-IS.
- Resource-intensive → requires strong CPU/memory.
- Not ideal for small networks.

🌍 Example

- AS 65001 (ISP 1) and AS 65002 (ISP 2) exchange routes via eBGP.
- Inside AS 65001, routers use iBGP to share external routes learned from AS 65002.
- If a packet from Enterprise A (AS 65001) wants to reach Enterprise B (AS 65002), BGP determines the AS-path and applies policies to select the best route.

📌 Real-World Use

- ISPs use BGP to connect customers and peers.
- Enterprises with multiple ISPs use BGP for redundancy and load balancing.
- Data centers& cloud providers use BGP for large-scale routing.

🔷 ARP (Address Resolution Protocol)

Definition:

ARP is a protocol used to map a 32-bit IP address (logical address) to a 48-bit MAC address (physical address) in a local area network (LAN).

It works at the Data Link Layer (Layer 2) and is crucial for communication inside a LAN.

🔄 How ARP Works (Step by Step)

1. A device wants to send data to another device on the same network.
    - It knows the IP address of the destination but not the MAC address.
2. The sender broadcasts an ARP Request frame:
    - "Who has IP address 192.168.1.5? Tell me your MAC."

3. The device with that IP replies with an ARP Reply:
    - "I am 192.168.1.5, and my MAC is AA:BB:CC:DD:EE:FF."
4. The sender stores the mapping in its ARP cache (temporary memory) to use in future communication.

📌 **Types of ARP**

1. Normal ARP Request/Reply → used for IP-to-MAC resolution.
2. Reverse ARP (RARP) → obsolete, used to get IP when only MAC was known.
3. Proxy ARP → one device answers ARP on behalf of another (used in complex networks).
4. Gratuitous ARP → a device announces its own IP–MAC mapping (often used for redundancy or updating caches).

✅ **Advantages of ARP**

- Automatic address resolution (no manual mapping needed).
- Works dynamically in real time.

❌ **Disadvantages of ARP**

- Broadcast traffic may increase in large networks.
- Vulnerable to ARP spoofing/poisoning attacks (attackers can trick devices with fake MAC addresses).

🌍 **Example**

- PC1 (192.168.1.10) wants to send data to PC2 (192.168.1.20).
- PC1 checks its ARP cache → no entry found.
- PC1 sends ARP request (broadcast).
- PC2 replies with its MAC → entry stored in cache.
- PC1 can now send data directly using PC2's MAC.

🔷 ICMP (Internet Control Message Protocol)

Definition:

ICMP is a network layer protocol (part of the IP suite) used for error reporting and diagnostics in an IP network.

- It is not used for sending data but for sending control messages about network conditions.
- Defined in RFC 792.

🔄 **How ICMP Works**

- If something goes wrong while delivering an IP packet (e.g., host unreachable, TTL expired), the router or host generates an ICMP message back to the source.
- ICMP messages are carried inside IP packets (Protocol number: 1).

📌 **Common ICMP Message Types**

1. Error Reporting Messages
    - Destination Unreachable (Type 3): Host, network, port, or protocol not reachable.
    - Time Exceeded (Type 11): TTL expired (used by traceroute).
    - Source Quench (deprecated): Used to control congestion (no longer used).
    - Redirect (Type 5): Tells host to use a better gateway.
2. Query Messages
    - Echo Request (Type 8) and Echo Reply (Type 0): Used by ping to test reachability.
    - Timestamp Request/Reply: For clock synchronization.

- Address Mask Request/Reply: For subnet mask discovery (rarely used now).

✅ **Uses of ICMP**

- Troubleshooting tools:
  - Ping: Uses Echo Request/Reply to check if a host is alive.
  - Traceroute: Uses Time Exceeded messages to trace the path to a destination.
- Error reporting: Alerts sender about unreachable hosts/networks.
- Network diagnostics: Helps administrators detect failures.

❌ **Limitations / Issues**

- ICMP messages themselves may be lost.
- Not secure → vulnerable to abuse (e.g., ICMP flood attacks, ping of death).
- Many firewalls block ICMP for security reasons.

🌍 **Example**

1. PC1 sends data to PC2, but the router cannot reach PC2.
2. The router sends back an ICMP Destination Unreachable message to PC1.
3. PC1 knows the packet was not delivered.

If you run:
ping 8.8.8.8
Your PC sends ICMP Echo Requests → Google DNS replies with ICMP Echo Replies.

🔷 DHCP (Dynamic Host Configuration Protocol)

Definition:
DHCP is a network protocol that automatically assigns IP addresses and other network configuration parameters (subnet mask, default gateway, DNS, etc.) to devices on a network.

- Works at the Application Layer (but messages are transported using UDP).
- Uses UDP port 67 (server) and UDP port 68 (client).

🔄 **DHCP Operations (DORA Process)**

The DHCP process has 4 main steps, often remembered as DORA:

1. Discover
   - When a device (client) connects to the network, it does not know its IP.
   - It sends a DHCP Discover message (broadcast) to locate DHCP servers.
2. Offer
   - A DHCP server receives the Discover and replies with a DHCP Offer (unicast or broadcast).
   - The Offer contains an available IP address and configuration details (lease time, gateway, DNS).
3. Request
   - The client receives one or more Offers and chooses one.
   - It sends a DHCP Request (broadcast) to inform all servers about the selected offer.
4. Acknowledge
   - The chosen DHCP server sends a DHCP Acknowledgment (ACK) to finalize the lease.
   - The client can now use the assigned IP and other settings.

📌 **Example Flow (DORA)**

1. Client: DHCP Discover → “I need an IP!”

2. Server: DHCP Offer → "You can use 192.168.1.10."
3. Client: DHCP Request → "I'd like 192.168.1.10, please."
4. Server: DHCP ACK → "Confirmed! You have 192.168.1.10 for 24 hours."

🔑 Additional DHCP Operations

- Lease Renewal:
  Before lease expires, client sends a DHCP Request (unicast) → server replies with ACK to renew.
- Lease Rebinding:
  If the original server doesn't respond, the client tries other DHCP servers to extend the lease.
- Release:
  Client can send a DHCP Release to free its IP.

✅ Advantages of DHCP

- Automatic and fast IP assignment.
- Reduces human error (e.g., duplicate IPs).
- Supports mobile devices (IP changes automatically when moving networks).
- Centralized management of IPs.

❌ Disadvantages of DHCP

- Single point of failure (if DHCP server goes down).
- May cause delays during IP acquisition.
- Vulnerable to attacks (e.g., DHCP spoofing).

UNIT III

Transport and Application Layer Protocols

Transport service, elements of transport protocol, Simple Transport Protocol, Internet transport layer protocols: UDP and TCP, Domain name system, electronic mail, World Wide Web: architectural overview, dynamic web document and http. Simple Network Management Protocol, File Transfer Protocol, Simple Mail Transfer Protocol.

Definition:

The Transport Layer provides end-to-end communication between applications running on different hosts. It sits between the Network Layer (IP) and the Application Layer (HTTP, FTP, DNS, etc.).
Its job is to ensure reliable, efficient, and correct data delivery to the right application.
The Transport Layer (Layer 4 of OSI) provides end-to-end communication between applications on different hosts.

✅ Services Provided by Transport Layer

1. Process-to-Process Delivery → Uses port numbers to deliver data to the correct application (e.g., HTTP = 80, SMTP = 25).
2. Error Control → Ensures reliable delivery (acknowledgements, retransmissions).
3. Flow Control → Prevents sender from overwhelming receiver.
4. Segmentation & Reassembly → Breaks large messages into smaller segments and reassembles them at the destination.
5. Connection Control → Supports connection-oriented (TCP) and connectionless (UDP) communication.
6. Multiplexing/Demultiplexing → Multiple apps share the same network channel using port numbers.

🔗 Transport Layer Protocols

1. TCP (Transmission Control Protocol):
    - Connection-oriented
    - Reliable (acknowledgments, retransmissions)
    - Ordered delivery
    - Example: Web (HTTP/HTTPS), Email (SMTP, IMAP), File transfer (FTP).
2. UDP (User Datagram Protocol):
    - Connectionless
    - Unreliable (no ACKs, no retransmission)
    - Faster, less overhead
    - Example: DNS, VoIP, Video streaming, Online gaming.
3. Other Transport Protocols:
    - SCTP (Stream Control Transmission Protocol) – used in telecom.
    - DCCP (Datagram Congestion Control Protocol) – used for streaming.
    - Simple Transport Protocol (STP) – academic/learning use.

Elements of a Transport Protocol

1. Addressing (Port Numbers): Identifies sending and receiving processes.
2. Connection Establishment/Release: Rules for starting/ending communication (e.g., TCP's 3-way handshake).

3. Error Control: Error detection & retransmission of lost/corrupted segments.
4. Flow Control: Matching sender's speed to receiver's capacity.
5. Sequencing: Numbering segments to ensure correct order.
6. Acknowledgement: Positive or negative feedback from receiver.
7. Multiplexing/Demultiplexing: Multiple applications share transport service.
8. Transmission Services:
    - Reliable (TCP) vs Unreliable (UDP) delivery.
    - Connection-oriented (TCP) vs Connectionless (UDP).

◈ 3. Simple Transport Protocol (STP)

Definition:
A basic model of a transport protocol used in academic study and small systems to illustrate transport-layer concepts. It is much simpler than TCP/UDP but covers core functions.

✨ Features of STP

- Provides end-to-end delivery between processes.
- Implements port numbers for addressing.
- Provides error detection using checksums.
- Provides flow control (stop-and-wait or sliding window).
- Supports acknowledgements and retransmissions for reliability.

📌 Working of STP (Typical)

1. Sender adds STP header (with port numbers, sequence number, checksum).
2. Receiver checks header:
    - If error → discard and wait for retransmission.
    - If correct → send ACK.
3. Sender retransmits if no ACK received within a timeout.

✅ Example Use

- Used in teaching concepts of TCP, without its complexity.
- Can be seen in lightweight embedded systems or simulations.

Internet transport layer protocols: UDP and TCP

◈ 1. UDP (User Datagram Protocol)

Definition:
UDP is a connectionless, lightweight transport protocol. It does not guarantee delivery, order, or reliability.

✅ Features of UDP

- Connectionless → no handshake before sending.
- Unreliable → no acknowledgment, no retransmission.
- Fast, low overhead → header is only 8 bytes.
- Message-oriented → sends data as independent datagrams.
- Multiplexing via port numbers.

📌 Use Cases

- DNS queries
- Online gaming
- Video/audio streaming (VoIP, YouTube Live, Zoom)

- IoT data transfer

◈ 2. TCP (Transmission Control Protocol)

Definition:
TCP is a connection-oriented, reliable transport protocol. It ensures ordered, error-free delivery of data.

✅ Features of TCP

- Connection-oriented → uses 3-way handshake.
- Reliable → acknowledgments & retransmissions.
- Ordered delivery → sequence numbers ensure correct order.
- Flow Control → sliding window adjusts speed.
- Error Control → checksum, retransmission.
- Full duplex → data can flow both ways simultaneously.
- Larger header (20–60 bytes).

📌 Use Cases

- Web browsing (HTTP/HTTPS)
- Email (SMTP, IMAP, POP3)
- File transfer (FTP, SFTP)
- Remote login (SSH, Telnet)

TCP vs UDP (Comparison Table)

Feature	TCP	UDP
Type	Connection-oriented	Connectionless
Reliability	Reliable (ACKs, retransmissions)	Unreliable
Ordering	Ensures packets arrive in sequence	No ordering guarantee
Error Handling	Yes (checksums + retransmit)	Yes (checksum only, no recovery)
Speed	Slower (more overhead)	Faster (less overhead)
Header Size	20–60 bytes	8 bytes
Use Cases	Web, Email, File Transfer	Gaming, Streaming, VoIP, DNS

◈ Domain Name System (DNS)

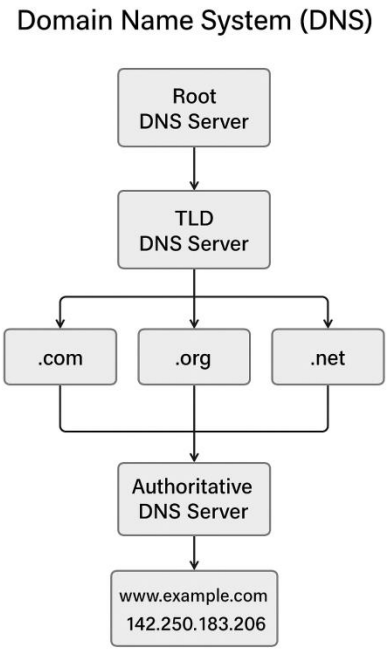
Definition:
DNS is a hierarchical, distributed naming system that translates human-readable domain names (like www.google.com) into IP addresses (like 142.250.183.206) so computers can communicate over the Internet.
It works like the phonebook of the Internet.

🔄 How DNS Works (Name Resolution Process)

When you type a URL in your browser:

1. DNS Query Initiation
    - You type www.example.com. The browser asks the DNS Resolver (usually your ISP or a public DNS like Google 8.8.8.8).

2. Recursive Query (Resolver → DNS Hierarchy)
    - If not cached, the resolver queries step by step:
        1. Root DNS Server → directs to Top-Level Domain (TLD) server.
        2. TLD Server (.com, .org, .net, etc.) → directs to the Authoritative DNS Server.
        3. Authoritative Server → gives the actual IP address of www.example.com.
3. Response
    - The resolver sends the IP address back to your browser.
4. Connection Established
    - Browser contacts the web server at that IP using TCP/UDP depending on the service.



📌 Types of DNS Servers

1. DNS Resolver (Recursive Resolver): First point of contact (usually at ISP or public DNS like Google 8.8.8.8, Cloudflare 1.1.1.1).
2. Root DNS Servers: The top of DNS hierarchy (only 13 root server clusters worldwide).
3. TLD Servers: Manage domains like .com, .org, .in, etc.
4. Authoritative DNS Servers: Hold actual records of a domain (managed by the domain owner or hosting provider).

📌 Types of DNS Records

- A Record: Maps domain → IPv4 address.
- AAAA Record: Maps domain → IPv6 address.
- CNAME Record: Alias name mapping (e.g., mail.example.com → gmail.com).
- MX Record: Mail exchange (for email delivery).
- NS Record: Specifies authoritative DNS servers.
- PTR Record: Reverse DNS (IP → domain name).
- TXT Record: Stores arbitrary text (used for verification, SPF, DKIM in email).

✅ Advantages of DNS

- Easy to remember names instead of IPs.
- Distributed & scalable system.

- Provides load balancing via multiple IPs.
- Caching improves speed.

❌ **Disadvantages / Challenges**

- DNS attacks (DNS spoofing, cache poisoning).
- Single failure point if misconfigured.
- Latency if not cached.

🌍 **Example**

- You type: www.wikipedia.org.
- DNS resolves to IP: 208.80.154.224.
- Browser connects to Wikipedia's server using that IP.

Electronic Mail

Definition:

Electronic Mail (E-Mail) is a method of exchanging digital messages between people using computer networks, primarily the Internet.

It follows a client–server model where mail servers handle sending, storing, and delivering emails, and mail clients are used to read/write them.

🔑 **Components of E-Mail System**

1. User Agent (UA):
   - The application used by users to compose, read, and manage emails.
   - Examples: Outlook, Gmail, Thunderbird.
2. Message Transfer Agent (MTA):
   - The mail server responsible for transferring messages between sender and receiver servers.
   - Uses protocols like SMTP.
3. Mailbox:
   - Storage location for a user's emails (on the server).
4. Message Formats:
   - Standardized by MIME (Multipurpose Internet Mail Extensions) to support text, images, audio, video.

🔄 **Working of E-Mail**

1. Composing:
   User writes an email using a client (e.g., Gmail).

2. Sending (SMTP):
   The client sends the email to the sender's mail server using SMTP (Simple Mail Transfer Protocol).
3. Relaying:
   The sending mail server forwards the email across the Internet to the recipient's mail server (again via SMTP).
4. Receiving (POP3/IMAP):
   - The recipient's client retrieves the message using either:
     - POP3 (Post Office Protocol v3): Downloads email locally and optionally deletes from server.
     - IMAP (Internet Message Access Protocol): Keeps emails on server, allows multiple device sync.
5. Reading:
   Recipient reads the email via their mail client.

How Email Works

The process of sending and receiving an email involves several key components and protocols:

1. Mail User Agent (MUA): This is the email client software on your computer or phone (e.g., Outlook, Gmail app). You use it to compose, send, and read messages.
2. Mail Transfer Agent (MTA): This is the server software that handles sending and routing email messages. When you click "send," your MUA forwards the message to your local MTA.
3. Mail Delivery Agent (MDA): This server is responsible for receiving the message from the MTA and placing it into the recipient's mailbox.

Protocols Used in E-Mail

 SMTP (Simple Mail Transfer Protocol): Used for sending email from a client to a server and between servers. It is a push protocol.

 POP3 (Post Office Protocol 3): A pull protocol used to retrieve email from a server. It is a one-way protocol, with messages typically downloaded and then deleted from the server.

 IMAP (Internet Message Access Protocol): A more advanced pull protocol used to access and manage email on a server. It allows users to view headers, search messages, and organize mail on the server without downloading it. It is ideal for users who access their email from multiple devices.

MIME: Supports multimedia attachments (images, audio, video, etc.).

Advantages

- Fast communication across the globe.
- Can include text, files, multimedia.
- Accessible from anywhere.
- Supports group communication.

Disadvantages

- Spam and phishing attacks.
- Privacy concerns (can be intercepted if not encrypted).
- Requires Internet connection.

🌍 Example Flow

- Alice (alice@gmail.com) sends a mail to Bob (bob@yahoo.com).
- Gmail server uses SMTP to send the mail to Yahoo's mail server.
- Bob's Yahoo client retrieves the mail using IMAP/POP3.
- Bob reads the message.

🌐 World Wide Web (WWW): Architectural Overview

🔷 What is the WWW?

The World Wide Web (WWW) is a distributed system of interlinked hypertext documents and resources, accessed through the Internet using web browsers.
It uses client–server architecture and follows the request–response model.

🔑 Key Components of WWW Architecture

1. Clients (Web Browsers)
    - Applications like Chrome, Firefox, Safari.
    - Send HTTP/HTTPS requests to web servers.
    - Render HTML pages with multimedia (text, images, video, audio).
2. Web Servers
    - Store and provide web pages/resources.
    - Respond to client requests using HTTP/HTTPS.
    - Examples: Apache, Nginx, Microsoft IIS.
3. Protocols
    - HTTP/HTTPS (Hypertext Transfer Protocol): Communication protocol between client and server.
    - DNS (Domain Name System): Converts domain names into IP addresses.
    - TCP/IP: Transport and network layer protocols for reliable delivery.
4. Resources
    - Identified using URLs (Uniform Resource Locators).
    - Formats: HTML, CSS, JavaScript, images, video, etc.
5. Middleware / Application Layer Services
    - Search engines, web APIs, caching servers, CDNs, proxies.
    - Enhance performance, scalability, and functionality.

How It Works ⚙️

1. Request: When you type a URL into your browser, the browser sends an HTTP request to the web server that hosts the website.
2. DNS Lookup: Before sending the request, the browser uses the Domain Name System (DNS) to translate the human-readable domain name (e.g., www.example.com) into a machine-readable IP address.
3. Transmission: The request is sent over the Internet, passing through various routers and networks, until it reaches the correct web server.
4. Response: The web server receives the request, finds the requested resource, and sends an HTTP response back to the browser.

5. Rendering: The browser receives the response, which usually contains HTML, CSS, and JavaScript, and renders the web page on your screen.

🔄 Working of WWW (Flow)

1. User enters a URL (e.g., https://www.example.com).
2. DNS translates domain name into an IP address.
3. Browser establishes a TCP connection (port 80 for HTTP, 443 for HTTPS).
4. Browser sends an HTTP request (e.g., GET /index.html).
5. Web server processes request and responds with an HTTP response (e.g., an HTML file).
6. Browser renders the web page (fetching additional CSS, JS, images as needed).
7. User interacts (hyperlinks, forms, scripts), generating more requests/responses.

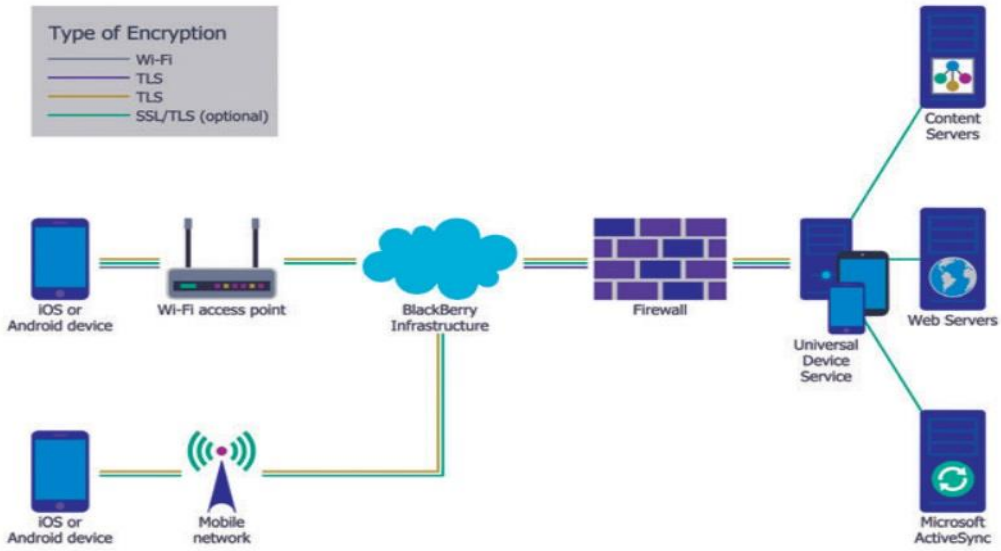


Figure 15.8 World Wide Web Architecture

📌 Features of WWW Architecture

- Distributed: Resources are spread across many servers worldwide.
- Scalable: Billions of pages accessible.
- Platform-independent: Works across devices & OS.
- Multimedia support: Text, images, audio, video, interactive content.
- Dynamic & Interactive: Enabled via JavaScript, APIs, and databases.

✅ Example

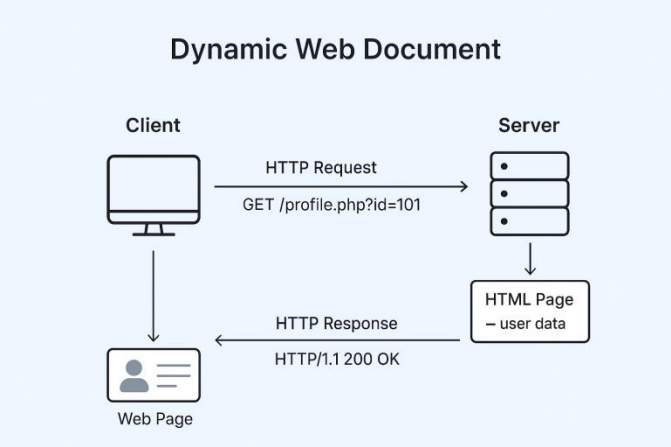
- You search "news" in Google Chrome.
- Browser contacts Google's DNS → finds IP.
- Sends HTTP request to Google's web server.
- Google responds with an HTML search results page.
- Browser displays it interactively with JavaScript & CSS.

🌍 Dynamic Web Document

🔷 Definition

A dynamic web document is a web page whose content is generated at runtime by the server or client, instead of being fixed like a static HTML page.

- Static Web Document: Pre-written, same for every user. (e.g., a plain HTML file)
- Dynamic Web Document: Content changes based on user input, database queries, time, session data, etc



🔑 Characteristics of Dynamic Web Documents

- Generated by server-side scripts (PHP, Python, Node.js, Java, ASP.NET) or client-side scripts (JavaScript).
- Can pull data from databases (MySQL, MongoDB, etc.).
- Personalized for each user.
- Frequently updated (e.g., social media feeds, e-commerce pages, dashboards).

📌 Example Flow

1. User requests profile.php?id=101.
2. Web server forwards request to an application server.
3. Application server runs code (e.g., PHP/Java/Python), queries the database.
4. Server generates an HTML page dynamically with user data.
5. Browser displays the page.

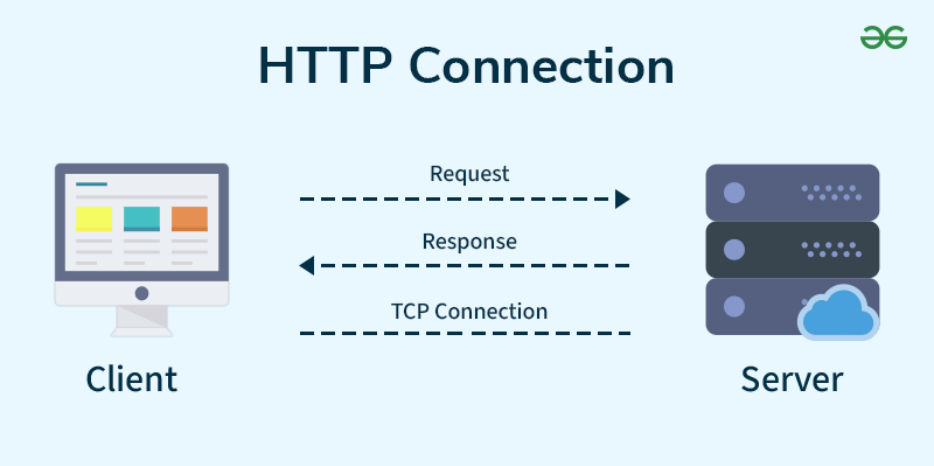
✔Example: Facebook profile page, Amazon product page.

🌐 HTTP (Hypertext Transfer Protocol)

🔷 Definition

HTTP is the application layer protocol used for communication between web clients (browsers) and web servers.

- Connectionless: Each request is independent.
- Stateless: Server does not remember past requests (cookies/sessions are used to maintain state).
- Runs over TCP (port 80) or HTTPS over port 443 (with SSL/TLS encryption).



🔑 HTTP Operations (Methods)

- GET: Request data from server.
- POST: Send data to server (e.g., form submission).
- PUT: Update existing data.
- DELETE: Remove data.
- HEAD: Get headers only.
- OPTIONS: Ask server about supported methods.

📌 Structure of HTTP Communication

1. HTTP Request (from client to server):
2. GET /index.html HTTP/1.1
3. Host: www.example.com
4. User-Agent: Chrome/120.0
5. Accept: text/html
6. HTTP Response (from server to client):
7. HTTP/1.1 200 OK
8. Content-Type: text/html
9. Content-Length: 1024
10. <html> ... web page content ... </html>

✅Key Features of HTTP

- Simple & human-readable.
- Stateless (but cookies/sessions fix this).
- Flexible (supports multimedia, JSON, APIs).
- Extensible (works with modern web apps, REST APIs).

🔄 Connection Between Dynamic Web Documents and HTTP

- HTTP acts as the transport protocol to request and deliver web documents.
- When a dynamic page is requested, HTTP sends the request to the server, which then generates the document dynamically and sends it back via an HTTP response.

Simple Network Management Protocol (SNMP)

Definition

SNMP is an application layer protocol used to manage and monitor network devices (routers, switches, servers, printers, IoT devices) over IP networks.
It allows administrators to collect information, configure devices, and detect faults in a standardized way.

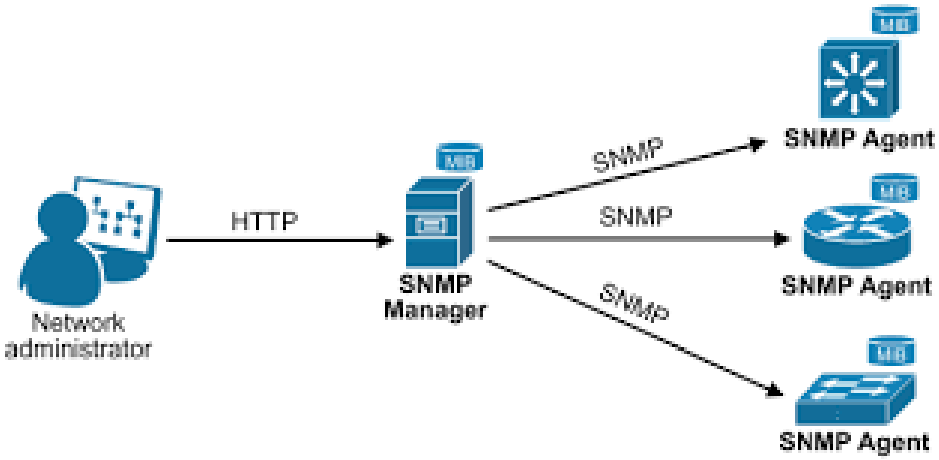
Key SNMP Operations

The SNMP manager communicates with the agent using a few key message types:

- GET: The manager sends a GET request to retrieve a specific value from the MIB of an agent.
- SET: The manager sends a SET request to change a value on an agent, which can be used to configure a device.
- TRAP: An agent can send an unsolicited TRAP message to the manager to alert it of a significant event, such as a device failure or a critical error. This is a crucial feature for real-time monitoring.

Components of SNMP

1. SNMP Manager (Network Management System – NMS):
    - Software running on a management workstation.
    - Sends requests and receives responses from agents.
    - Collects device statistics and generates reports.
2. SNMP Agent:
    - A software module running on a managed device (router, switch, server).
    - Collects data from the device (CPU usage, memory, interface stats).
    - Responds to the SNMP manager.
3. Management Information Base (MIB):
    - A database of network objects that can be managed via SNMP.
    - Objects are identified by Object Identifiers (OIDs) in a tree structure.
4. Managed Devices:
    - Network devices that run an SNMP agent and store performance/usage data.



SNMP Communication Model

1. Manager → Agent (Request): The manager queries the agent for information (e.g., device uptime).
2. Agent → Manager (Response): Agent responds with requested data.

3. Agent → Manager (Trap): Agent sends an unsolicited alert (e.g., link failure, high CPU load).

📌 **SNMP Operations**

- GET: Retrieve the value of a variable from an agent.
- SET: Modify the value of a variable on an agent.
- GET-NEXT: Retrieve the next variable in the MIB tree.
- GET-BULK: Retrieve large blocks of data efficiently.
- TRAP: Unsolicited message from agent to manager (alert/notification).
- INFORM: Similar to trap but requires acknowledgment.

🔑 **Versions of SNMP**

1. SNMPv1: First version, simple, minimal security.
2. SNMPv2: Enhanced performance and data types, introduced GET-BULK.
3. SNMPv3: Most secure version, adds authentication & encryption.

✅**Advantages**

- Centralized network monitoring.
- Lightweight and widely supported.
- Can automate fault detection and alerts.

❌**Disadvantages**

- Earlier versions (v1, v2) lacked security (clear-text community strings).
- Scalability issues in very large networks.

🌍 **Example**

- A network admin wants to monitor router uptime.
- SNMP manager sends a GET request to the router's SNMP agent.
- The agent looks up the MIB and replies with the uptime value.

File Transfer Protocol

FTP – File Transfer Protocol (in Computer Networks)File Transfer Protocol (FTP) is a standard network protocol used to transfer files between a client and a server on a computer network. FTP is built on a client-server model and uses separate control and data connections between the client and the server. It's a foundational protocol that's been used for decades to upload and download files.

- FTP (File Transfer Protocol) is a standard application layer protocol in computer networks.
- Used to transfer files between a client and a remote server.
- Works on the Client-Server model.
- Based on TCP/IP for reliable communication

How FTP Works □

FTP operates with two distinct channels for communication:

1. Control Connection (Port 21): This channel is used to send commands from the client to the server and receive replies. It handles administrative tasks like authentication (username and password) and directory navigation. This connection remains open throughout the FTP session.
2. Data Connection (Ports 20 or other): This channel is used for the actual transfer of data, such as a file. The data connection is opened and closed for each file transfer.

2. Key Features

- File upload (client → server) and download (server → client).
- Supports user authentication (username/password) or anonymous access.
- Allows directory operations (create, delete, list, navigate).
- Transfer in ASCII mode (text files) or Binary mode (images, videos, executables).
- Can resume interrupted transfers.

3. Ports Used

- Port 21 → Control connection (commands, authentication).
- Port 20 → Data connection (actual file transfer).

4. Modes of Operation

1. Active Mode
    - Client opens a port and waits.
    - Server connects back to the client for data transfer.
    - Problematic with firewalls.
2. Passive Mode
    - Server opens a port and waits.
    - Client connects to that port for data transfer.
    - Firewall/NAT friendly → widely used.

5. FTP Commands (Examples)

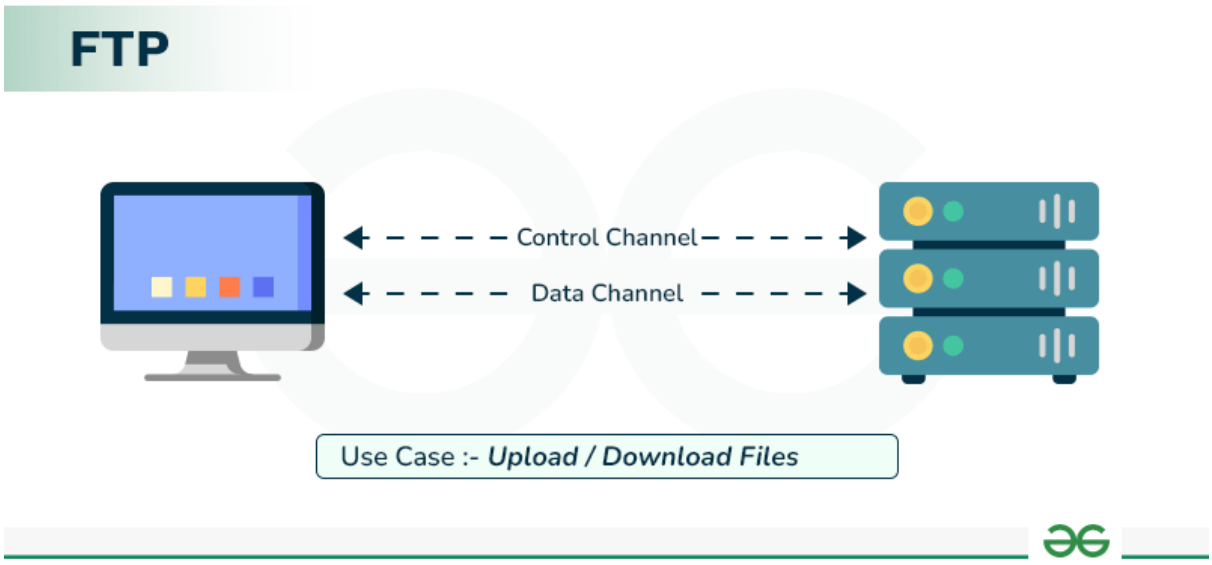
- USER – send username.
- PASS – send password.
- LIST – list directory contents.
- RETR – download file.
- STOR – upload file.
- QUIT – close connection.

6. Advantages

- Simple, easy-to-use protocol.
- Allows transfer of large files.
- Widely supported on all platforms.

7. Disadvantages

- Insecure – credentials and data sent in plain text.
- Requires two connections (complex through firewalls).
- Replaced by secure protocols:
    - SFTP (SSH File Transfer Protocol).
    - FTPS (FTP Secure – over SSL/TLS).



SMTP – Simple Mail Transfer Protocol

1. Definition

- SMTP = Simple Mail Transfer Protocol.
- Standard application layer protocol used for sending emails over the Internet.
- Based on TCP/IP → provides reliable communication.
- Works on store-and-forward model between mail servers.

2. Ports Used

- Port 25 – Default SMTP port (sending between servers).
- Port 587 – Secure mail submission from client to server.
- Port 465 – SMTP over SSL (deprecated, but still used).

3. Features

- Used to send, relay, or forward emails.
- Supports client-server and server-server communication.
- Only handles sending emails, not receiving.
    - For receiving → POP3 or IMAP is used.

4. Working Process

1. User Agent (Email Client) (e.g., Outlook, Gmail app) composes email.
2. SMTP Client connects to SMTP Server on port 25/587.
3. Message is sent to recipient's mail server using SMTP.
4. Recipient retrieves email using POP3/IMAP.

5. SMTP Commands

- HELO – identify the client to server.
- MAIL FROM: – sender's address.
- RCPT TO: – recipient's address.

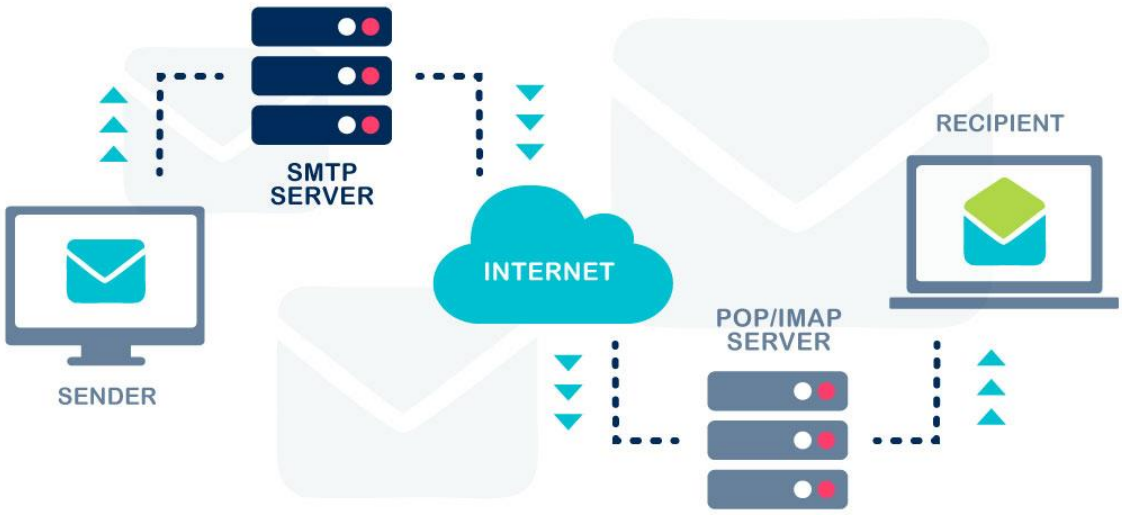
- DATA – start sending email body.
- QUIT – terminate connection.

6. Advantages

- Reliable and widely supported.
- Enables communication between different email systems.
- Efficient for bulk email transfer.

7. Disadvantages

- No encryption by default → insecure.
- Can be exploited for spamming.
- Needs extra protocols (POP3/IMAP) for email retrieval.



UNIT IV
Wireless and Mobile Communication Systems

Cellular networks and mobile IP,Wireless LANs and IEEE 802.11, Bluetooth, ZigBee, and NFC basics, Mobile communication challenges: handoff, roaming, latency, MANETs and VANETs

Introduction

Wireless and mobile communication systems allow the exchange of information (voice, data, video, etc.) without the need for physical connections like cables.

- Wireless Communication → Any communication over air using electromagnetic waves.
- Mobile Communication → Wireless communication with the added ability for the user/device to move while staying connected.

Evolution of Mobile Communication

- 1G (Analog, 1980s): Voice only, low capacity, e.g., AMPS.
- 2G (Digital, 1990s): Digital voice, SMS, GSM, CDMA.
- 3G (2000s): Internet, video calls, multimedia, WCDMA.
- 4G (2010s): High-speed internet, HD streaming, LTE.
- 5G (Present): Ultra-high speed, low latency, IoT, AI integration.

Key Components

1. Mobile Devices (UE – User Equipment): Phones, tablets, IoT devices.
2. Base Station (BTS / eNodeB / gNodeB): Provides wireless coverage.
3. Switching Centers (MSC): Manages call setup and mobility.
4. Backhaul Network: Connects base stations to the core network.
5. Core Network: Handles routing, authentication, billing, and internet access.

Types of Wireless Communication Systems

- Cellular Networks (GSM, LTE, 5G)
- Satellite Communication
- Wi-Fi (Wireless LANs)
- Bluetooth, NFC (Short-range communication)
- WiMAX (Broadband wireless)

Features of Mobile Communication

- Mobility – Seamless connectivity while moving.
- Handoff/Handover – Switching connection from one cell to another.
- Roaming – Service access outside home network.
- Security – Encryption, authentication for safe communication.
- Scalability – Support for millions of users

Applications

- Voice & Data Communication – Calls, SMS, internet.
- Business & Banking – Mobile payments, online transactions.
- IoT & Smart Devices – Smart homes, vehicles, healthcare.
- Emergency Services – Disaster management, tracking.
- Entertainment – Video streaming, gaming, social media.

Challenges

- Limited spectrum availability.
- Interference between signals.
- Security & privacy concerns.
- Battery consumption of mobile devices.
- Infrastructure cost for 5G and beyond.

Cellular Networks

Definition

A cellular network is a wireless communication system where the service area is divided into small regions called cells.
Each cell is served by a base station (BS), and all cells are interconnected to form a large network, enabling mobile communication.

Key Concepts

- Cell: A geographic area covered by one base station.
- Frequency Reuse: Same frequencies can be used in different non-adjacent cells to maximize spectrum efficiency.
- Handoff (Handover): Seamless transfer of an active call or data session from one cell to another when the user moves.
- Roaming: Ability to use network services outside the home area (national or international).

Cellular Network Architecture

1. Mobile Station (MS): User device (mobile phone, tablet, IoT).
2. Base Station (BS/BTS): Connects mobiles to the network via radio signals.
3. Base Station Controller (BSC): Manages multiple base stations and handles handoffs.
4. Mobile Switching Centre (MSC): Main switching node, responsible for routing calls, SMS, and data.
5. Public Switched Telephone Network (PSTN) / Internet: Provides connectivity to external networks.

Features of Cellular Networks

- High Capacity: Supports millions of users.
- Mobility: Users can move freely while staying connected.
- Wide Coverage: Networks can cover cities, countries, and globally.
- Scalability: Easy to expand by adding new cells.
- Security: Encryption and authentication protect user data.

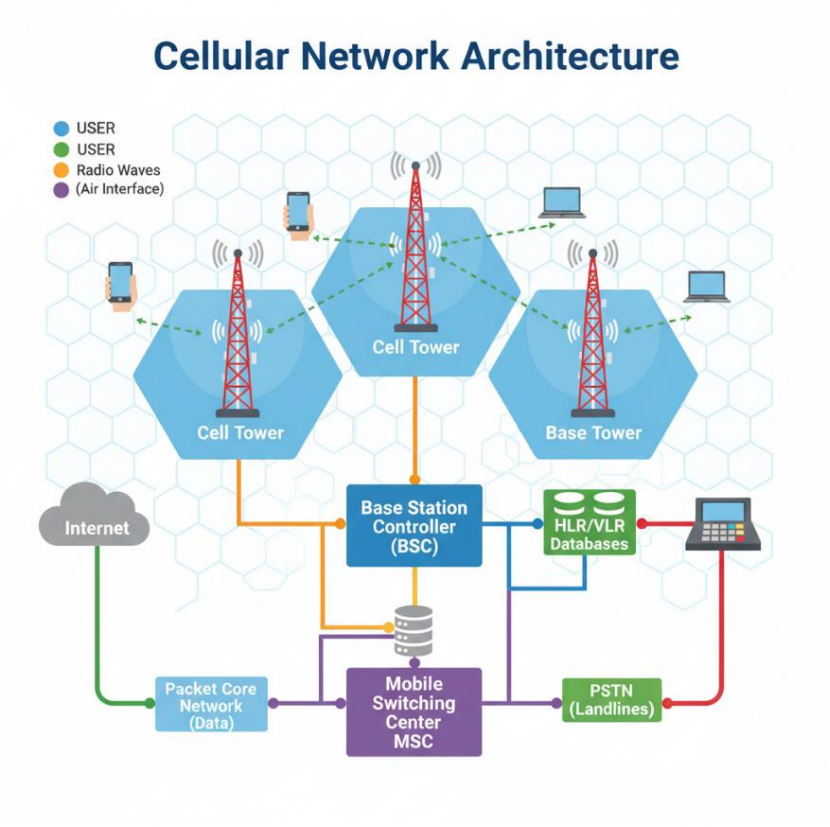
How it Works

The fundamental principle behind cellular networks is the division of a large service area into smaller, manageable cells. Each cell has its own base station (cell tower) that communicates with mobile devices within its boundaries using radio waves.

1. Mobile Device (e.g., your phone): When you make a call or use data, your device sends and receives radio signals.
2. Base Station (Cell Tower): The closest base station picks up your device's signal. It's equipped with transceivers that handle communication within its cell.
3. Base Station Controller (BSC): Multiple base stations are connected to a BSC. The BSC manages radio resources, controls handovers between base stations, and forwards calls.
4. Mobile Switching Center (MSC): This is the core of the 2G/3G network (and part of the core for 4G/5G). The MSC acts like a telephone exchange, routing calls between mobile users, to landline networks (PSTN), or to the internet. It also handles subscriber authentication and billing.

5. Databases (HLR/VLR): The Home Location Register (HLR) stores permanent subscriber information, while the Visitor Location Register (VLR) temporarily stores data about roaming subscribers.
6. Packet Core Network (for Data): For data services (like internet browsing), the network connects through a packet core, which handles IP packet routing to and from the internet.

When a user moves from one cell to another during a call, the network performs a handoff (or handover), seamlessly transferring the connection to the new cell's base station without interruption.



Generations of Cellular Networks

- 1G (1980s): Analog voice (AMPS).
- 2G (1990s): Digital voice, SMS, GSM, CDMA.
- 3G (2000s): Multimedia, internet, video calling (UMTS, WCDMA).
- 4G (2010s): IP-based, high-speed internet, LTE, VoLTE.
- 5G (Present): Ultra-fast speeds, IoT, AI, low latency.
- 6G (Future): AI-driven, holographic communication, terahertz spectrum.

Advantages

- Efficient spectrum usage.
- Seamless mobility support.
- Supports large populations.
- Flexibility in network expansion.

Challenges

- Limited spectrum availability.

- High infrastructure costs.
- Interference between neighboring cells.
- Security and privacy issues.

Mobile IP

Mobile IP is an IETF standard communications protocol that allows a mobile device (called a Mobile Node) to maintain a consistent, permanent IP address while moving between different networks. It ensures that data sessions remain active and uninterrupted, even as the device changes its physical location and network connection. Think of it as a virtual forwarding service for your device's IP address.

How Mobile IP Works 📱💻

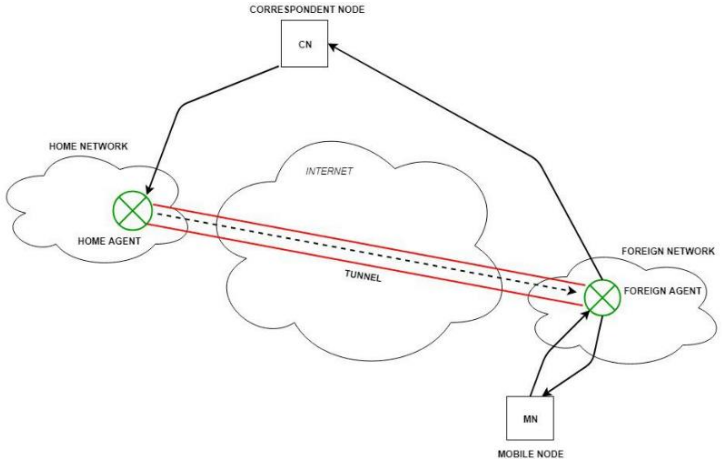
Mobile IP uses a "home address" and a "care-of address" to manage the device's location.

1. Home Network: When a Mobile Node (MN) is on its home network, it behaves like a regular host. All communication uses its permanent Home Address (HoA).
2. Moving to a Foreign Network: When the MN moves to a new, or Foreign Network, it needs a temporary address to receive data. This new address is called the Care-of Address (CoA).
3. Registration: The MN must inform its Home Agent (HA) of its new location. It does this by sending a registration request, which includes its new CoA. The HA is a router on the home network that knows the MN's permanent address.
4. Packet Forwarding (Tunneling): A Correspondent Node (CN), which is any device communicating with the MN, sends packets to the MN's permanent Home Address. The HA intercepts these packets, encapsulates them (wraps them in a new IP header with the CoA as the destination), and sends them through a tunnel to the foreign network.
5. Packet Delivery: The Foreign Agent (FA) on the foreign network receives the tunneled packets, de-encapsulates them, and forwards them to the MN at its Care-of Address.

This process ensures that data packets always find their way to the MN, regardless of its location, without the sender needing to know the MN's temporary address.

Key Components 🗺️

- Mobile Node (MN): The mobile device itself (e.g., a smartphone, laptop).
- Home Agent (HA): A router on the MN's home network that acts as an anchor point. It intercepts and tunnels packets destined for the MN when it's away.
- Foreign Agent (FA): A router on the foreign network that assists the MN. It receives tunneled packets from the HA and delivers them to the MN. (Note: in some implementations, the MN can acquire its own "co-located" care-of address and act as its own FA.)
- Correspondent Node (CN): Any other device on the internet that is communicating with the Mobile Node.
- Home Address (HoA): The permanent IP address of the Mobile Node.
- Care-of Address (CoA): The temporary IP address used by the Mobile Node while it's on a foreign network.



Wireless LAN

A Wireless Local Area Network (WLAN) is a computer network that uses wireless communication to connect devices within a limited area, like a home or office. Instead of physical cables, a WLAN uses radio waves, following the IEEE 802.11 standards, which are widely known as Wi-Fi.

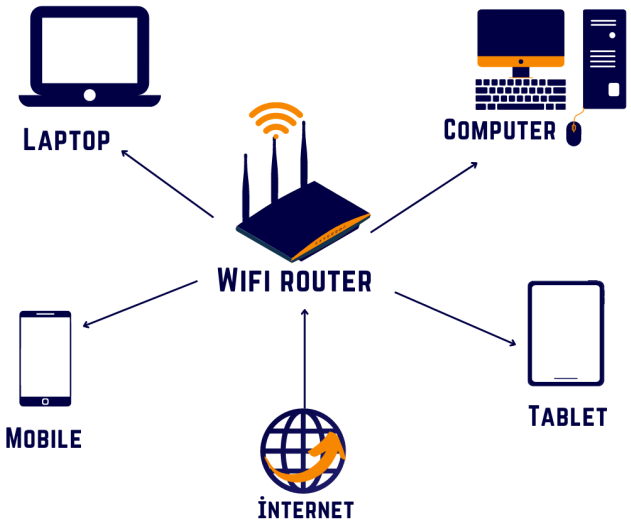
Key Components Explained

1. Internet: The ultimate source of information and services. The WLAN provides a bridge for wireless devices to access the internet.
2. Modem: This device connects your home or office network to your Internet Service Provider (ISP). It translates the digital signals from the network into a format that can travel over the physical connection (e.g., fiber, cable).
3. Wireless Router / Access Point (AP): This is the central hub of the WLAN.
    - It's connected to the modem via an Ethernet cable, giving it access to the internet.
    - It broadcasts a radio signal, creating a wireless coverage area. This is where wireless devices connect.
    - It's responsible for managing all the wireless traffic, assigning IP addresses to devices, and routing data to and from the internet.
    - The wireless router typically combines the functions of a router, a switch, and an access point into one device.
4. Wireless Devices: These are the devices that connect to the WLAN wirelessly, such as:
    - Laptops, Desktops, Tablets: Computers with built-in Wi-Fi adapters.
    - Smartphones: Your mobile phone uses Wi-Fi to connect to the internet at home or in public places.
    - Printers, Smart TVs, Gaming Consoles: Many modern devices have integrated Wi-Fi for seamless network connectivity.
5. Basic Service Set (BSS): This is the area of coverage provided by a single Access Point. All the devices within this area can communicate with each other through the AP.
6. Wired Network: The wireless network often exists as an extension of a wired network. The wireless router connects to wired devices (like a desktop computer) and acts as a gateway for the wireless devices.

How it All Works Together

When a device wants to connect to the WLAN, its wireless adapter scans for the router's signal (its SSID or network name). Once the user enters the correct password, the router authenticates the

device and grants it access. All data sent from the device is converted into radio waves, transmitted to the AP, and then converted back into a wired signal to be sent out to the internet. This process is reversed for incoming data.



Advantages of WLAN:

- Mobility: Devices can connect from anywhere within the coverage area of the access point, offering flexibility and convenience.
- Easy Setup: Installing a WLAN is relatively simple and doesn't require laying cables, which can be expensive and cumbersome.
- Scalability: Additional devices can be added without the need for extra wiring.

Disadvantages of WLAN:

- Interference: Wireless networks are subject to interference from physical obstructions (walls, furniture) and other devices (microwaves, other routers).
- Security Risks: Wireless networks can be more vulnerable to unauthorized access if not properly secured with encryption (like WPA2 or WPA3).
- Range Limitations: The signal strength of WLANs decreases as the distance from the router increases.

IEEE 802.11 Architecture

The IEEE 802.11 standard, commonly known as Wi-Fi, outlines the architecture and defines the MAC and physical layer specifications for wireless LANs (WLANs). Wi-Fi uses high-frequency radio waves instead of cables for connecting the devices in LAN. Given the mobility of WLAN nodes, they can move unrestricted within the network coverage zone. The 802.11 structure is designed to accommodate mobile stations that participate actively in network decisions. Furthermore, it can seamlessly integrate with 2G, 3G, and 4G networks.

The Wi-Fi standard represents a set of wireless LAN standards developed by the Working Group of IEEE LAN/MAN standards committee (IEEE 802). The term 802.11x is also used to denote the set of standards. Various specifications and amendments include 802.11a, 802.11b, 802.11e, 802.11g, 802.11n etc.

Important Terminologies of IEEE 802.11 Architecture

Station: Stations (STA) comprise all devices and equipment that are connected to the wireless LAN. It can be of two types:

- Wireless Access Point (WAP): WAPs or simply access points (AP) are wireless routers that bridge connections for base stations.
- Client: Examples include computers, laptops, printers, and smartphones.

Access Point: It is a device that can be classified as a station because of its functionalities and acts as a connection between wireless medium and distributed systems.

Distribution System: A system used to interconnect a set of BSSs and integrated LANs to create an ESS.

Frame: It is a MAC protocol data unit.

SSID (Service Set Identifier): It's the network name for a particular WLAN. All-access points and devices on a specific WLAN must use the same SSID to communicate.

SDU: It is a data unit that acts as an input to each layer. These can be fragmented or aggregated to form a PDU.

PDU: It is a data unit projected as an output to communicate with the corresponding layer at the other end. They contain a header specific to the layer.

Network Interface Controller: It is also known as network interface card. It is a hardware component that connects devices to the network.

Portal: Serves as a gateway to other networks

IEEE 802.11 Standards

The IEEE 802.11 family of standards defines the technical specifications for WLANs, including the radio frequencies, data rates, and security protocols. The continuous development of these standards has led to the different generations of Wi-Fi.

- 802.11b, 802.11a, 802.11g: These early standards (Wi-Fi 1, 2, and 3) laid the foundation for modern wireless networking with speeds up to 54 Mbps.
- 802.11n (Wi-Fi 4): Introduced MIMO (Multiple-Input Multiple-Output) technology, using multiple antennas to boost speeds and range significantly, up to 600 Mbps.
- 802.11ac (Wi-Fi 5): Operates exclusively in the 5 GHz band, offering multi-gigabit speeds and support for more users at once with MU-MIMO (Multi-User MIMO).
- 802.11ax (Wi-Fi 6): The current standard, designed for high-density environments. It uses OFDMA and other technologies to improve network efficiency, allowing more devices to communicate simultaneously with faster speeds and lower latency.

IEEE 802.11 Architecture and Services

In the year 1990, IEEE 802.11 Committee formed a new working group, the IEEE 802.11 standard which defines protocols for Wireless Local Area Networks (WLANs). Just like how Ethernet provides services for wired media, IEEE 802.11 architecture is designed to provide features for wireless networks.

An AP supports both wired and wireless connections. The 802.11 standard calls the upstream wired network the distribution system (DS). The AP bridges the wireless and wired L2 Ethernet frames, allowing traffic to flow from the wired to the wireless network and vice versa. Each wireless network has a unique SSID.

The 802.11 architecture provides some basic services for WLANs whose implementation is supported by MAC layer:

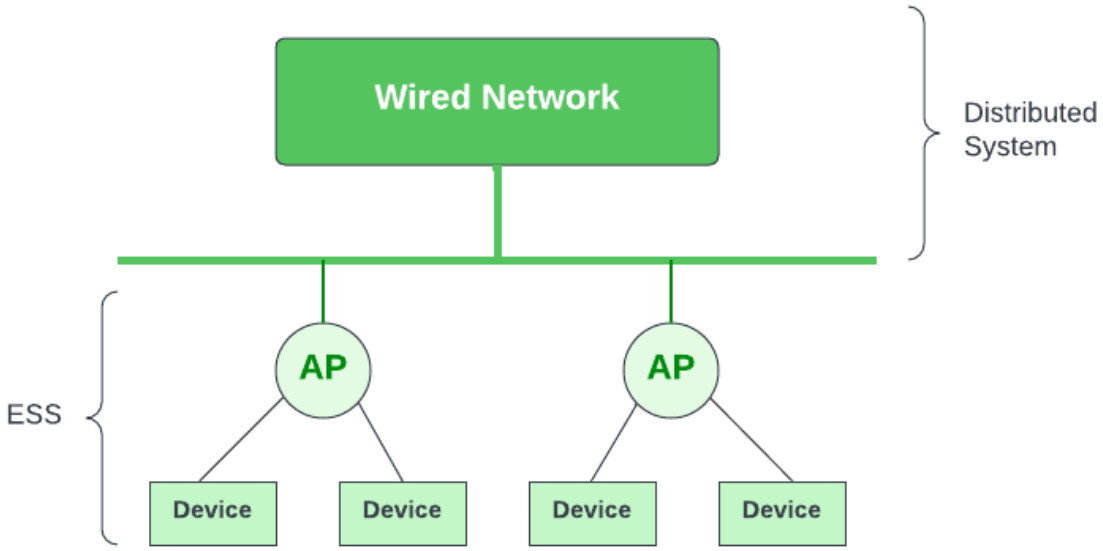
Basic Service Set

The Basic Service Set configuration consists of a group of stations and relies on an Access Point (AP), which serves as a logical hub. Stations from different BSSs interact through the AP, which functions as a bridge, linking multiple WLAN cells or channels.

Operating Modes

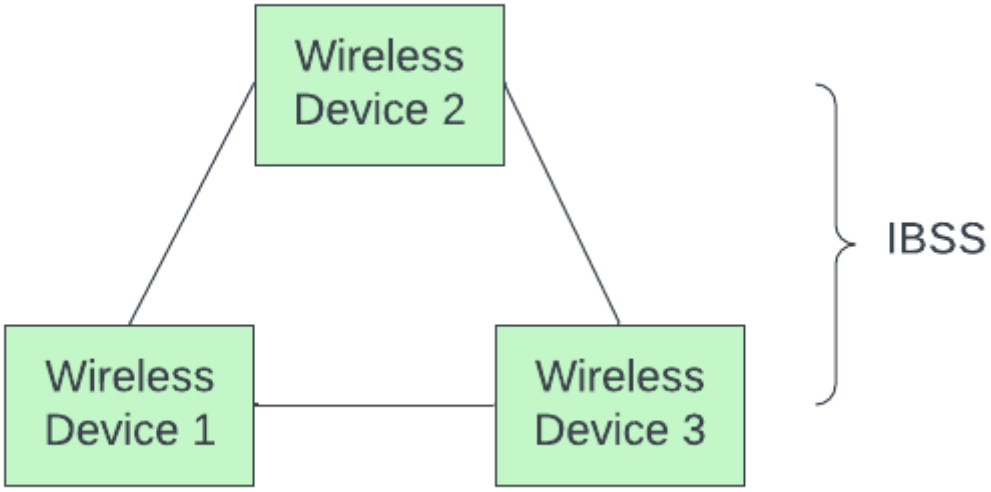
Depending upon the mode of operation, BSS can be categorized into the following types:

- Infrastructure BSS: Communication between stations takes place through access points. The AP and its associated wireless clients define the coverage area and form the BSS.



Infrastructure BSS

- Independent BSS - Supports mutual communication between wireless clients. An ad-hoc network is spontaneously created and does not support access to wired networks.



Independent BSS

Independent Basic Service Set

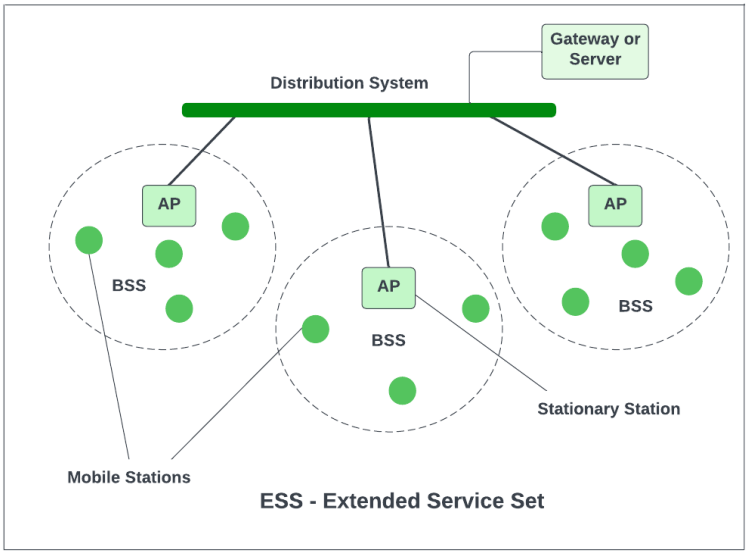
In the IBSS configuration, also referred to as independent configuration or ad-hoc network, no single node is required to act as a server. The stations communicate directly with one another in a peer-to-peer basis. Generally, IBSS covers a limited area instead of a large network. Typically covering a specific area, IBSS is used for specific, short-term purposes with a limited number of nodes.

Extended Service Set

ESS connects multiple BSSs and consists of several BSS cells, which can be interlinked through wired or wireless backbones known as a distributed system. Multiple cells use the same channel to boost aggregate throughput to network. The equipment outside of the ESS, the ESS and all of its mobile stations comprise a single MAC layer network where all stations are virtually stationary. Thus, all stations within the ESS appear stationary from an outsider's perspective.

Other components include:

- Distribution System (DS): Links APs within the ESS.
- Portal: Serves as a gateway to other networks.



Architecture for IEEE 802.11 Configuration

- Roaming: In an environment with multiple access points (like a large office building or campus), a device can move from the range of one AP to another and still maintain its connection. This is possible due to the underlying architecture of the IEEE 802.11 standard which allows for roaming between APs.
- Authentication and Association: Before a station can send or receive data frames on a WLAN, it needs to establish its identity with an AP. This process is called authentication. After authentication, the station then establishes a data link-layer connection with the AP through a process called association.

Advantages of IEEE 802.11 Architecture

- Fault Tolerance: The centralized architecture minimizes the bottlenecks and introduces resilience in the WLAN equipment.
- Flexible Architecture: Supports both temporary smaller networks and larger, more permanent ones.

- Prolonged Battery Life: Efficient power-saving protocols extend mobile device battery life without compromising network connections.

Disadvantages of IEEE 802.11 Architecture

- Noisy Channels: Due to reliance on radio waves, signals may experience interference from nearby devices.
- Greater Bandwidth and Complexity: Due to necessary data encryption and susceptibility to errors, WLANs need more bandwidth than their wired counterparts.
- Speed: Generally, WLANs offer slower speeds compared to wired LANs.

Applications of IEEE 802.11 Architecture

- Home Networking: Connecting devices, laptops, smart TVs, speakers, gaming consoles etc.
- Wi-Fi Hotspots: Free or paid internet access to visitors in coffee shops, hotels, airports, malls and restaurants.
- Connectivity in Campus: Provide internet access in university, colleges, schools or corporate campuses.

Bluetooth

Bluetooth is a short-range wireless technology for exchanging data between devices. It operates on the 2.4 GHz frequency band and is a key technology for creating Personal Area Networks (PANs), connecting peripherals like headphones, keyboards, and mice to a phone or computer.

How Bluetooth Works 🛠️

Bluetooth uses a master-slave model to establish connections. One device, the master, controls the communication and can connect with up to seven other devices, which are called slaves. Slaves can only communicate with their master, not directly with other slaves.

A group of devices connected in this way forms a piconet. All devices within a piconet share the same frequency-hopping sequence, which is determined by the master device. This process, called frequency hopping spread spectrum (FHSS), involves devices rapidly changing between 79 different frequencies to minimize interference from other wireless technologies on the same band (like Wi-Fi) and to enhance security.

For more complex connections, multiple piconets can be interconnected to form a scatternet. This is achieved when a device acts as a bridge, serving as a slave in one piconet and a master in another.

Key Features & Concepts 💡

- Piconet: A basic Bluetooth network consisting of one master device and up to seven active slave devices.
- Scatternet: A larger network formed by multiple interconnected piconets, where a device acts as a bridge to link them.
- Pairing: The initial process of connecting two Bluetooth devices. During pairing, the devices exchange a shared secret key and other security information to establish a trusted relationship for future connections.
- Bluetooth Classic vs. Bluetooth Low Energy (BLE): Bluetooth has evolved into two main versions.

- Bluetooth Classic is designed for continuous, high-data-rate connections, like streaming music to headphones. It consumes more power.
- Bluetooth Low Energy (BLE) is optimized for applications that require short, burst-like data transfers, such as fitness trackers, smartwatches, and IoT sensors. It's designed to conserve battery life and is a cornerstone of the Internet of Things.

There are two types of Bluetooth networks −

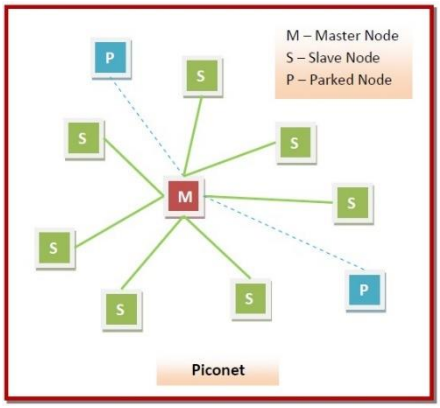
- Piconets
- Scatternets

Piconets

Piconets are small Bluetooth networks, formed by at most 8 stations, one of which is the master node and the rest slave nodes (maximum of 7 slaves). Master node is the primary station that manages the small network. The slave stations are secondary stations that are synchronized with the primary station.

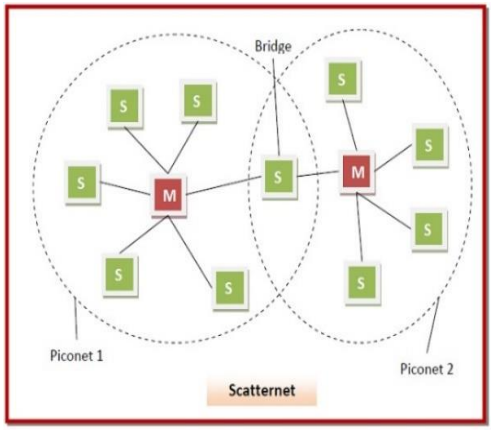
Communication can take place between a master node and a slave node in either one-to-one or one-to-many manner. However, no direct communication takes place between slaves. Each station, whether master or slave, is associated with a 48-bit fixed device address.

Besides the seven active slaves, there can be up to 255 numbers of parked nodes. These are in a low power state for energy conservation. The only work that they can do is respond to a beacon frame for activation from the master node.



Scatternodes

A scatternet is an interconnected collection of two or more piconets. They are formed when a node in a piconet, whether a master or a slave, acts as a slave in another piconet. This node is called the bridge between the two piconets, which connects the individual piconets to form the scatternet.



ZigBee

ZigBee is a wireless technology standard built on the IEEE 802.15.4 specification, designed for low-power, low-data-rate, and secure wireless communication.
It's especially popular for Internet of Things (IoT) applications, particularly in smart home automation, industrial control, and sensor networks, where battery life and network reliability are critical.

Features of ZigBee

- Range: 10–100 meters (can be extended using mesh networking).
- Data Rate: 20 kbps (868 MHz), 40 kbps (915 MHz), 250 kbps (2.4 GHz).
- Power Consumption: Very low (suitable for battery-powered devices).
- Topology: Supports star, tree, and mesh networks.
- Devices: Can connect up to 65,000 devices in one network.
- Security: Uses 128-bit AES encryption.

How ZigBee Works 🏠

ZigBee networks typically operate in the 2.4 GHz ISM band (though other frequencies like 868 MHz and 915 MHz are also used in certain regions).
Its main strength lies in its mesh networking capability, which allows devices to relay data for each other, extending the network's range and improving its resilience.

A ZigBee network consists of three types of devices:

1. **ZigBee Coordinator (ZC):**
    - The "brain" of the network. There can only be one coordinator per network.
    - Initializes and forms the network.
    - Stores security keys and manages the network's configuration.
    - Can act as a bridge to other networks (e.g., Wi-Fi, Ethernet) to connect to the internet.
    - Functions as a router.
2. **ZigBee Router (ZR):**
    - Relays data between other devices, extending the network's range and robustness.
    - Can also run application functions (e.g., a smart light switch that also relays signals).
    - Must be always-on, as they forward data.
3. **ZigBee End Device (ZED):**
    - Simple, low-power devices (e.g., sensors, light bulbs).
    - Can only communicate with a parent router or coordinator.
    - Cannot relay data for other devices.
    - Can enter a deep "sleep" mode for extended periods, making them ideal for battery-powered applications that need to last months or years.

ZigBee Network Topologies

1. Star Topology – All devices connect to a central Coordinator.
2. Tree Topology – Hierarchical structure; routers extend range.
3. Mesh Topology – Devices interconnect, providing redundancy and reliability.

ZigBee Network Diagram

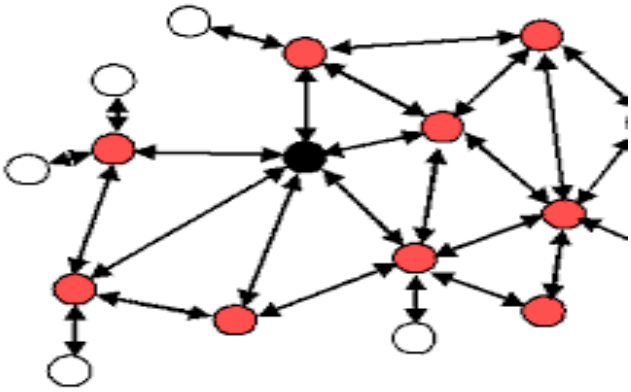
Here's a simple representation:

```
        (Coordinator)
             ZC
           /   |   \
         ZR  ZR  ZED
         |       |
       ZED     ZED
```

- ZC = Coordinator
- ZR = Router
- ZED = End Device

Key Features ⚡

- Low Power Consumption: Designed for devices that run on batteries for very long durations.
- Low Data Rate: Ideal for small data packets (e.g., sensor readings, on/off commands), not for video streaming.
- Mesh Networking: Creates a self-healing, robust network where data can find multiple paths to its destination. This increases reliability and coverage.
- Scalability: Supports a large number of nodes (up to 65,000 devices theoretically).
- Security: Employs 128-bit AES encryption for secure data transmission.



Near Field Communication

NFC, or Near Field Communication, is a short-range wireless technology that enables two electronic devices to communicate when they're very close to each other, typically within 4 cm (about 1.5 inches).

It's a subset of Radio-Frequency Identification (RFID) technology and is most commonly used for contactless payments, data sharing, and quick device pairing.

Key Features of NFC

1. Short Range – Secure communication (needs close proximity).
2. Low Power – Consumes very little energy, suitable for small devices.
3. Ease of Use – No manual pairing; communication starts when devices are close.
4. Secure Communication – Harder to intercept compared to Bluetooth/Wi-Fi.
5. Two-way communication – Unlike RFID, NFC allows both devices to send and receive data.

How it Works 🤝

NFC operates on the 13.56 MHz frequency and relies on inductive coupling between two antennas. It doesn't require a power source for one of the devices, which is a key advantage.

- Active Device (Reader): This device, like a smartphone or a payment terminal, actively generates a radio frequency (RF) field to power and communicate with a passive device.
- Passive Device (Tag): This is a simple, unpowered tag (like an NFC sticker or a chip in a credit card) that draws its power from the active device's RF field. It modulates the field to send data back to the active device.

This setup is what allows you to "tap to pay" with your phone or use a transit card without a battery.

Communication Modes 💬

NFC supports three main modes of operation:

- Reader/Writer Mode: An active device (like a smartphone) reads or writes data to a passive NFC tag. This is how you can tap your phone on a smart poster to open a website or on a key card to access a building.
- Peer-to-Peer Mode: Two active devices (e.g., two smartphones) exchange data. This is used for sharing photos, contact information, or other files with a simple tap. Both devices can send and receive information.
- Card Emulation Mode: An active device (e.g., a smartphone) acts as a passive, unpowered card. This allows your phone to function like a contactless credit card or a transit pass, interacting with a payment terminal or a card reader.

Security 🔐

The very nature of NFC makes it highly secure:

- Short Range: The extremely short communication distance of a few centimeters makes it difficult for a third party to intercept the signal. A hacker would need to be uncomfortably close to a transaction.
- Tokenization: For payments, NFC systems often use tokenization. Instead of transmitting your actual credit card number, a unique, one-time-use digital token is sent. Even if this token is intercepted, it's useless for future transactions.
- Encryption: Many NFC transactions, especially those involving sensitive data, are encrypted, providing an additional layer of protection against unauthorized access.

Common Applications

- Contactless Payments (Google Pay, Apple Pay, Samsung Pay).
- Public Transport Tickets (metro/bus cards).
- Access Control (door locks, ID cards).
- Smart Posters (tap phone to get website link).
- Pairing Devices (Bluetooth speaker pairing via NFC).

◈ Advantages

- Quick and convenient (just tap).
- Secure (short range reduces hacking risk).
- No internet needed for basic operations.

◈ Limitations

- Very short range (must be within a few cm).

- Limited data transfer speed (slower than Wi-Fi/Bluetooth).
- Requires compatible hardware on both devices

Mobile communication faces several key challenges related to mobility, network management, and emerging technologies.

Handoff

A handoff, or handover, is the process of transferring an ongoing call or data session from one base station to another without interruption. It's a critical process in cellular networks to ensure seamless communication as a user moves. The main challenges include:

- Signaling Overhead: The handoff process requires significant communication between the mobile device, base stations, and the core network, which can consume network resources.
- Handoff Latency: The time it takes to complete a handoff. If this is too long, it can cause the call to drop or the data session to be interrupted.
- Decision-making: The network must accurately decide when and where to hand off a device based on signal strength, network load, and other factors to prevent unnecessary or failed handoffs.

Roaming

Roaming allows a mobile device to use a different network when it leaves its home network's service area. This is a major challenge due to:

- Billing and Authentication: The visited network must be able to authenticate and authorize the roaming user and correctly bill their home network for the services used. This requires complex agreements between network operators.
- Security: Roaming introduces security risks, as the user's data is handled by a network they are not a subscriber of, requiring robust security protocols.
- Latency: Data from a roaming user often has to be routed back to their home network, which can introduce significant latency and a degraded user experience.

Latency

Latency is the time delay in data transmission. It's a critical challenge in mobile communication, particularly for real-time applications like video calls, online gaming, and IoT. Factors contributing to latency include:

- Network Congestion: Too many users on the network can slow down data transmission.
- Processing Delays: The time it takes for routers, switches, and servers to process and forward data packets.
- Wireless medium: The nature of wireless signals, which are susceptible to interference and require retransmissions, can add to delays. 5G technology is specifically designed to address latency.

MANETs and VANETs

MANETs (Mobile Ad Hoc Networks)

- Definition: A decentralized wireless network formed by mobile nodes (like laptops, smartphones, IoT devices) without fixed infrastructure (no routers or base stations).
- Mobility: Random and relatively slow movement of nodes.
- Topology: Changes dynamically but less frequently compared to VANETs.

- Communication: Multi-hop wireless communication among mobile devices.
- Applications: Disaster recovery, military operations, temporary communication setups.
- Challenges:
    - Limited battery and processing power.
    - Frequent link breaks due to mobility.
    - Scalability issues with large node counts.

 VANETs (Vehicular Ad Hoc Networks)

- Definition: A specialized form of MANET where vehicles act as mobile nodes to establish communication without centralized infrastructure.
- Mobility: Very high-speed node movement (vehicles).
- Topology: Changes rapidly and frequently due to high mobility.
- Communication:
    - V2V (Vehicle-to-Vehicle)
    - V2I (Vehicle-to-Infrastructure)
- Applications:
    - Intelligent Transport Systems (ITS).
    - Road safety (collision warnings, traffic management).
    - Infotainment (real-time traffic updates, media sharing).
- Challenges:
    - Extremely dynamic topology.
    - Short connection time between vehicles.
    - Reliability and latency are critical for safety applications.

Feature	**MANETs**	**VANETs**
Mobility	Random, low/medium speed	High-speed (vehicles)
Topology change	Moderate	Very frequent & rapid
Energy constraints	Limited battery	Vehicles have ample power
Communication	Node-to-node	V2V, V2I
Applications	Military, disaster recovery, conferences	Traffic safety, ITS, infotainment
Challenges	Power saving, scalability	Low latency, fast handoff, reliability

---

UNIT V

Modern Network Trends, IoT, Cloud & Secure Networking

Cloud networking and virtualization, IoT architecture and protocols (MQTT, CoAP), Edge and fog computing, Network security basics: firewalls, encryption, VPNs, Network challenges in modern computing (5G, SDN, NFV)

Cloud Networking ☁

Cloud networking is a type of networking that resides in a public or private cloud environment, allowing users to access and share network resources like virtual routers, firewalls, and subnets on demand. Instead of relying on physical networking hardware, cloud networking uses software-defined networking (SDN) and virtualization to manage and automate network services.

Key features and benefits:

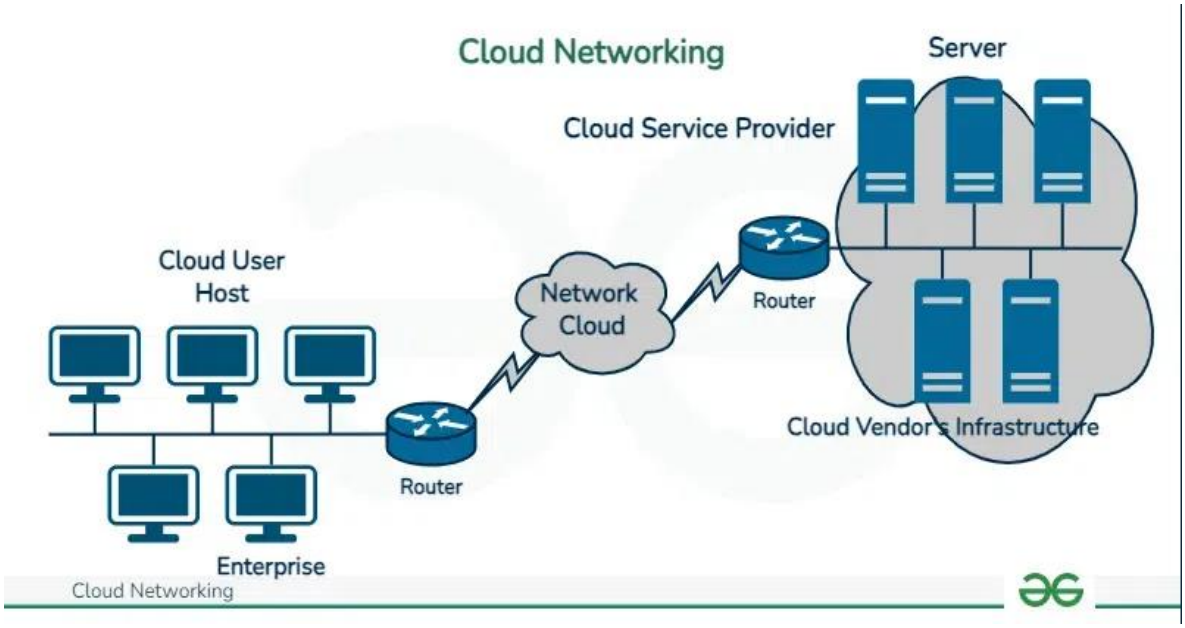
- Scalability: Network resources can be provisioned and de-provisioned quickly to meet changing demands.
- Flexibility: Users can customize their network topology and security policies without buying or installing new hardware.
- Cost-effectiveness: It operates on a pay-as-you-go model, eliminating the need for large upfront investments in physical infrastructure.
- Global Reach: Cloud providers' data centers are distributed globally, allowing businesses to create a worldwide network presence.

 Types:

1. Public Cloud Networking – Networking resources provided by public cloud vendors (AWS, Azure, GCP).
2. Private Cloud Networking – Cloud-based networking inside an organization’s private datacenter.
3. Hybrid Cloud Networking – Combination of public + private networks.

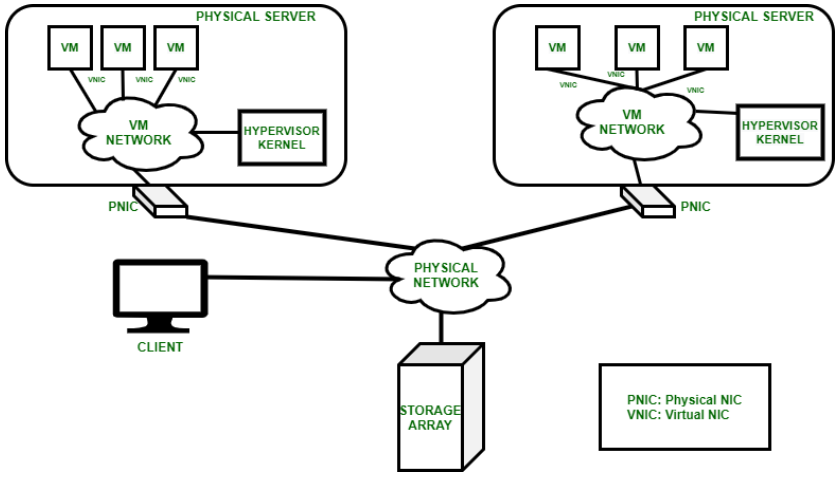
 Applications:

- Connecting branch offices via cloud VPNs.
- Hosting virtual routers, firewalls, load balancers.
- Supporting remote work and global collaboration.



Virtualization

Virtualization is the technology that creates a virtual, software-based version of a physical resource, such as a server, storage device, or network. It allows a single piece of physical hardware to run multiple isolated instances of operating systems or applications.



Types of Virtualization:

- Server Virtualization: This is the most common type. It involves using software called a hypervisor to create and manage multiple virtual machines (VMs) on a single physical server. Each VM has its own operating system and applications, completely isolated from others.
- Network Virtualization: This technology abstracts network resources and services, such as switches, routers, and firewalls, from the underlying physical hardware. It is the foundation of cloud networking.
- Storage Virtualization: This pools physical storage from multiple devices into a single, virtual storage resource that can be managed centrally.
- Desktop Virtualization – Running virtual desktops (VDI) instead of physical PCs.

 Benefits:

- Better utilization of hardware.
- Isolation between virtual machines.
- Easy scalability and migration.
- Cost reduction in IT infrastructure.

 Applications:

- Cloud computing backbone (IaaS, PaaS, SaaS).
- Running multiple OS environments on one machine.
- Disaster recovery and load balancing.
- Software testing and development environments.

Cloud Networking vs Virtualization

Aspect	**Cloud Networking**	**Virtualization**
Focus	Networking services in the cloud	Creating virtual versions of physical resources
Scope	Connects systems, apps, users across the	Abstracts compute, storage, and networks

Aspect	Cloud Networking	Virtualization
	cloud	
Examples	Cloud VPNs, SD-WAN, cloud firewalls	Virtual machines, virtual networks, virtual storage
Dependency	Often built on top of virtualization	Foundation of cloud computing (IaaS, SDN, etc.)
Use Case	Remote access, secure networking, hybrid cloud	Efficient resource utilization, multi-OS environments

---

IoT Architecture

The architecture of the Internet of Things (IoT) is typically organized into a layered structure that handles data from the physical world and delivers it to applications. While different models exist, a common and comprehensive one consists of four layers:

1. Sensing Layer (Physical Layer): This is the foundation of IoT. It includes all the physical devices that sense, gather, and act on data from the environment. Examples are sensors (for temperature, motion, light), actuators (to open doors, control motors), and the actual "things" like smart lights or connected vehicles.
2. Network Layer (Data Transmission Layer): This layer is responsible for securely and reliably transmitting the data collected by the sensing layer to the processing center. It uses various communication technologies such as Wi-Fi, Bluetooth, Zigbee, cellular networks (LTE, 5G), and dedicated low-power WAN protocols.
3. Data Processing Layer (Analytics Layer): At this layer, the raw data received from the network layer is processed, analyzed, and prepared for use. This includes data cleansing, aggregation, and filtering. Edge computing, where processing happens closer to the data source, is an important part of this layer for reducing latency.
4. Application Layer: This is the top layer where the end-user interacts with the IoT system. It includes the user interface (e.g., a smartphone app, a web dashboard) and the business logic that uses the processed data to perform tasks. Examples are smart home automation, traffic management, and industrial monitoring applications.

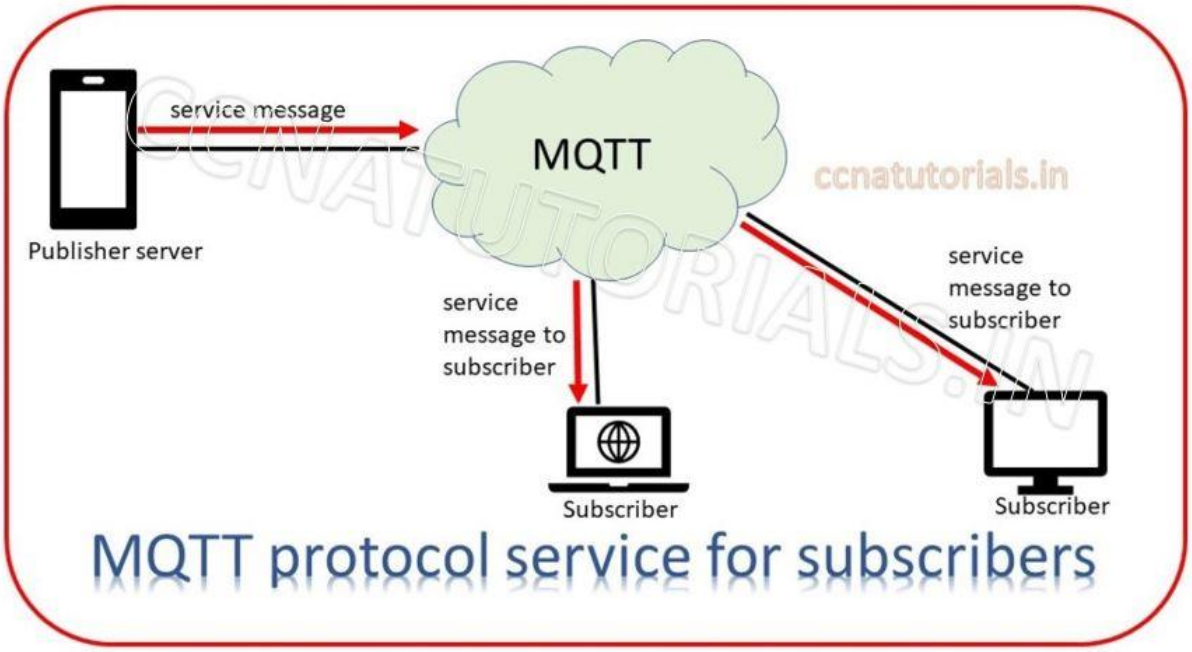
IoT Protocols

IoT protocols are the communication languages that allow devices to talk to each other and to the cloud. They are optimized for the specific constraints of IoT, such as low bandwidth, limited power, and high latency.

MQTT (Message Queuing Telemetry Transport)

MQTT is a lightweight, publish-subscribe messaging protocol. It's designed for resource-constrained devices and low-bandwidth, high-latency networks.

- How it works: Devices don't communicate directly with each other. Instead, they connect to a central broker. A device that wants to send data "publishes" a message to a specific topic on the broker. Devices interested in that data "subscribe" to the topic and receive the message. This decouples the sender from the receiver.

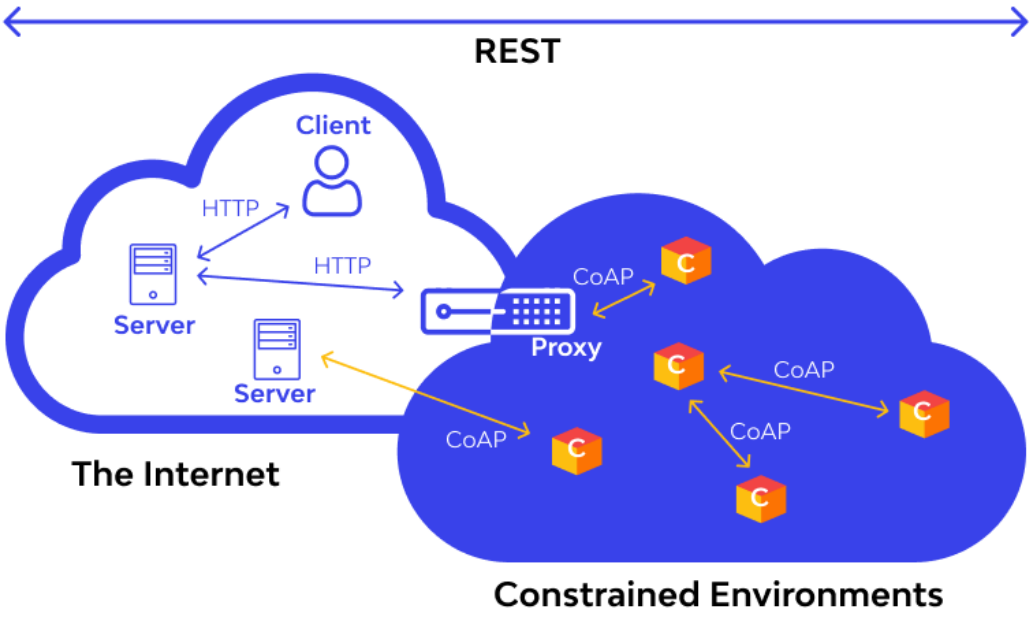


- **Key features:**
  - Lightweight: Minimal overhead, making it efficient for small data packets.
  - Publish/Subscribe: Provides flexible, one-to-many communication.
  - Stateful Connection: Uses persistent TCP connections, which can be useful for reliability.
  - Quality of Service (QoS): Offers different levels of delivery assurance (at most once, at least once, exactly once).

CoAP (Constrained Application Protocol)

CoAP is a specialized web transfer protocol for constrained devices and networks. It's designed to be similar to HTTP but optimized for the IoT environment.

- How it works: CoAP is a request/response protocol, where a client sends a request to a server, and the server sends a response back. It uses the UDP (User Datagram Protocol) transport layer, which is more efficient but less reliable than TCP.



- **Key features:**
    - Low Overhead: Smaller header and message sizes than HTTP.
    - Request/Response Model: Simple and familiar for developers.
    - Stateless: Each request is independent of previous ones.
    - Multicast Support: Allows a single message to be sent to multiple devices.

MQTT vs. CoAP

Feature	**MQTT**	**CoAP**
Communication Model	Publish/Subscribe	Request/Response (REST-like)
Transport Protocol	TCP	UDP
Message Overhead	Low	Very low
Reliability	High (with QoS levels)	Lower (best-effort, retransmit)
Best for	Continuous data streams (sensors)	Simple queries, device control

---

Edge and fog computing

Edge and fog computing are two related architectures that extend cloud computing by processing data closer to its source, rather than sending it all to a centralized data center. This approach reduces latency, conserves bandwidth, and enhances real-time applications. While they share a similar goal, they operate at different levels of a network.

Edge Computing □

Edge computing refers to processing data at or very near the source of the data generation, such as sensors, IoT devices, or local servers. It's the most decentralized of the three models (cloud, fog, edge).

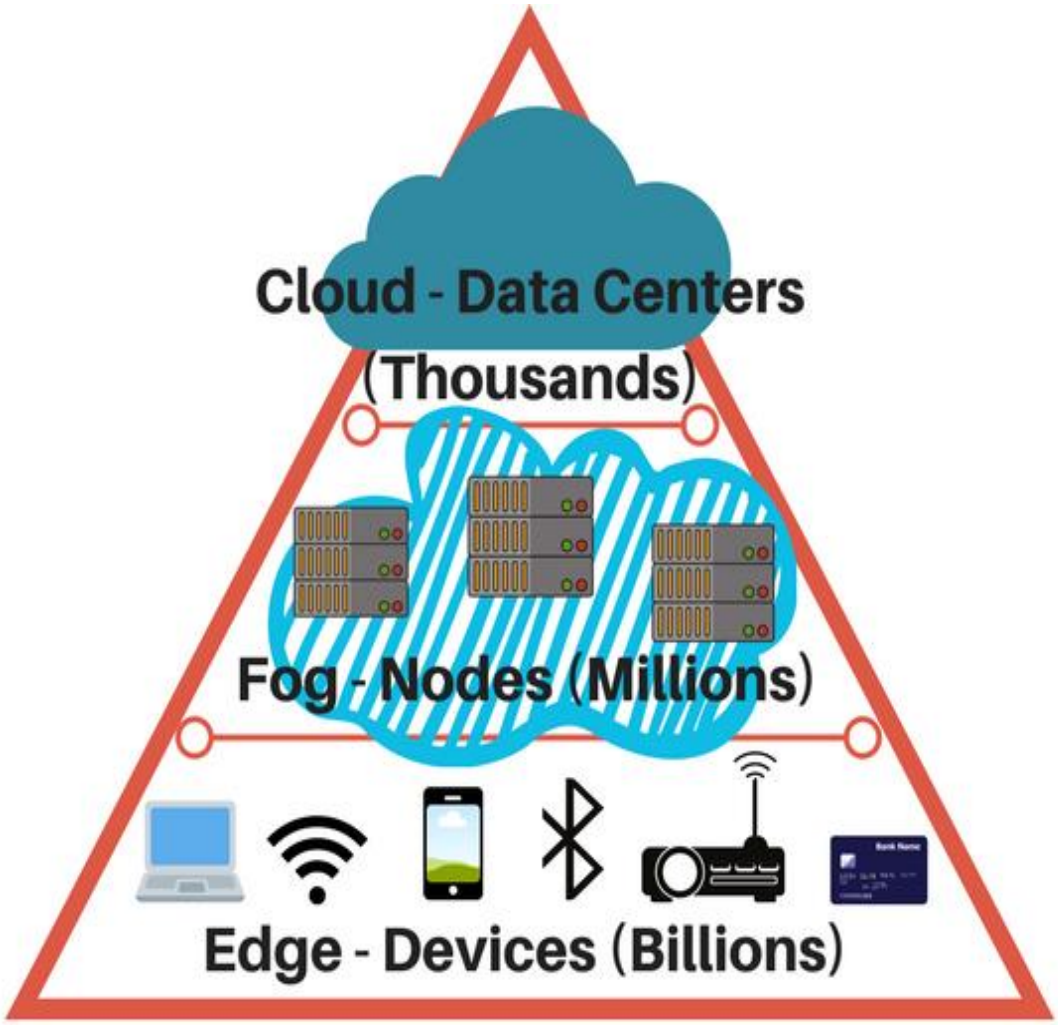
- How it works: Data is processed directly on the device itself or on a nearby gateway. For example, a smart camera might analyze a video stream for motion and only send an alert to the cloud, rather than sending the entire video.
- Purpose: Primarily focused on reducing latency for time-sensitive applications.
- Location: At the very "edge" of the network, right where the data is created.
- Example: Self-driving cars that process sensor data in real time to make immediate decisions, or smart factory machines that analyze their own performance to predict maintenance needs.

Fog Computing □

Fog computing acts as a middle layer between the edge and the cloud. It extends the cloud's functionality to the local network, serving as a distributed network of nodes that can process and store data.

- How it works: Data from multiple edge devices is sent to a local fog node (e.g., a server, router, or gateway). This node aggregates and processes the data before sending a summarized version to the cloud. It's a stepping stone between the edge and the cloud.
- Purpose: Provides a more robust layer for local data processing and storage, handling tasks that are too complex for a single edge device but don't require the full power of the cloud.

- Location: Between the edge devices and the cloud, often in a local area network.
- Example: A city-wide traffic management system where traffic signals and cameras send data to a local server (the fog node), which then coordinates traffic flow in a specific neighborhood before sending aggregated data to a central cloud for long-term analysis.



Key Differences: Edge vs Fog

Aspect	**Edge Computing**	**Fog Computing**
Location	On/very close to IoT devices	Between edge and cloud (gateways, local servers)
Latency	Ultra-low, real-time	Low (slightly higher than edge)
Data Handling	Processes raw data locally	Aggregates/filter data from multiple edge devices
Scalability	Device-level	Network/regional-level
Example	Self-driving car decisions	Smart traffic management across a city

---

Firewalls

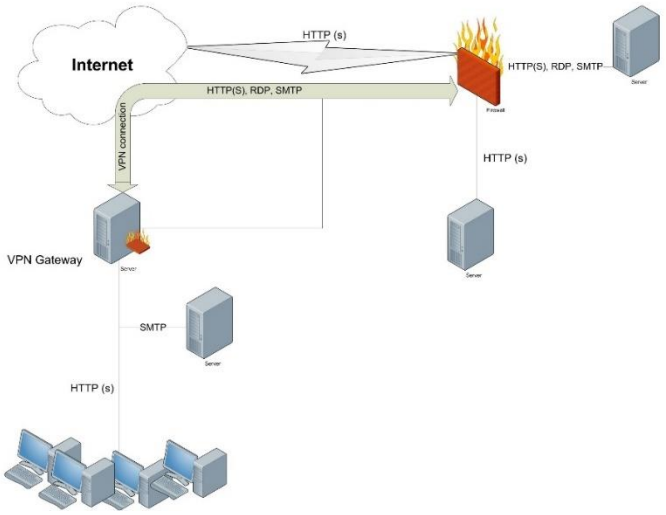
A firewall is a network security device that monitors and controls incoming and outgoing network traffic based on a set of predetermined security rules. It acts as a barrier, or fence, between a trusted internal network (like a company's network) and an untrusted external network (like the internet).
Firewalls operate by inspecting data packets and can be implemented as either hardware or software.

Types of Firewalls

- Packet-Filtering Firewall: This is the most basic type. It inspects the header of a data packet, looking at information like the source and destination IP addresses and port numbers. If the packet's information matches a rule in the firewall's access control list (ACL), it is either allowed to pass or is dropped.
- Stateful Inspection Firewall: This is a more advanced and common type. It not only inspects the packet header but also keeps track of the "state" of the connection. For example, once a connection is established, the firewall remembers it and allows return traffic without re-inspecting every packet. This makes it more efficient and secure than simple packet filtering.
- Application-Level Gateway (Proxy Firewall): This type operates at the application layer, providing a deeper level of inspection. It acts as an intermediary, or proxy, for all traffic to and from the network, making decisions based on the application protocol being used (e.g., HTTP, FTP). This gives it the ability to filter content and block specific commands within an application.
- Next-Generation Firewall (NGFW) : Combines traditional firewall + advanced features (intrusion detection, app control, malware prevention).

Functions of a Firewall

1. Traffic Filtering – Allows or blocks data packets based on IP addresses, ports, or protocols.
2. Access Control – Ensures only authorized devices/users access the network.
3. Protection Against Attacks – Prevents threats like malware, denial-of-service (DoS), and unauthorized access.
4. Monitoring & Logging – Keeps track of traffic for security analysis.



Key Functions

- Traffic Control: Firewalls filter traffic to prevent unauthorized access to a network.
- Threat Blocking: They can block common threats like malware, viruses, and hacking attempts.
- Logging: Firewalls keep logs of network activity, which are crucial for security audits and detecting suspicious behavior.

Example in Real Life

- In a company, the firewall might block employees from accessing malicious websites or unauthorized external servers, while allowing business apps like email and cloud storage.
- At home, your Wi-Fi router has a basic firewall that prevents random internet devices from connecting to your laptop.

Encyption

Encryption is the process of converting data into a coded, unreadable form called ciphertext. This is done to secure information and ensure that only authorized parties who possess the correct key can decrypt and access the original data, or plaintext. It's a fundamental pillar of network security.

Why Encryption is Important

- Ensures confidentiality – only intended recipients can read data.
- Protects against eavesdropping on networks (like Wi-Fi or the internet).
- Provides data integrity – prevents alteration of data in transit.
- Supports authentication – verifies the identity of sender/receiver.

How it Works □

The encryption process involves three main components:

1. Plaintext: The original, readable data that needs to be secured.
2. Encryption Algorithm (Cipher): A mathematical formula used to scramble the plaintext.
3. Key: A secret string of characters that works with the algorithm to encrypt and decrypt the data. The security of the encryption largely depends on the strength of this key.

The basic process is: Plaintext + Key + Algorithm → Ciphertext.

Types of Encryption

There are two primary types of encryption based on how they use keys:

1. Symmetric Encryption

This method uses a single, shared secret key for both encryption and decryption. Both the sender and the receiver must have this same key.

- Pros: It's very fast and efficient, making it ideal for encrypting large amounts of data.
- Cons: The biggest challenge is securely sharing the key with the intended recipient. If the key is compromised during transmission, the security of all data encrypted with it is at risk.
- Example: AES (Advanced Encryption Standard) is the most widely used symmetric encryption algorithm today, employed in Wi-Fi security and data storage.

2. Asymmetric Encryption

Also known as public-key cryptography, this method uses a pair of keys: a public key for encryption and a private key for decryption.

- Public Key: Can be shared with anyone.
- Private Key: Must be kept secret by the owner.

- How it works: A sender uses the recipient's public key to encrypt a message. Only the recipient's corresponding private key can decrypt it.
- Pros: It solves the key distribution problem of symmetric encryption.
- Cons: It's much slower and computationally intensive, so it's not practical for encrypting large files.
- Example: RSA is a classic example. Asymmetric encryption is primarily used for securing communication channels and for digital signatures, where it is often used to securely exchange the shared key for symmetric encryption.

Real-Life Examples of Encryption

- Online Banking: When you log in, your username and password are encrypted to prevent hackers from stealing them.
- Messaging Apps (WhatsApp, Signal): Use end-to-end encryption, meaning only sender and receiver can read messages.
- HTTPS Websites: Use SSL/TLS encryption so your credit card info is safe during online shopping.

Virtual Private Network (VPN)

A Virtual Private Network (VPN) extends a private network across a public network, allowing users to send and receive data as if their computing devices were directly connected to the private network. It achieves this by creating a secure, encrypted "tunnel" through the public network, such as the internet.

How a VPN Works □

1. VPN Client: You install a VPN application, or client, on your device (e.g., a laptop or smartphone).
2. Encryption: When you turn on the VPN, the client encrypts all your internet traffic. This makes your data unreadable to anyone who might intercept it.
3. Secure Tunnel: The encrypted data is then sent through a secure tunnel to a VPN server.
4. IP Address Masking: The VPN server acts as an intermediary. It decrypts your traffic and sends it on to its final destination on the internet. Your internet traffic appears to originate from the VPN server's IP address, not your own.
5. Data Return: When data returns, it first goes to the VPN server, which encrypts it and sends it back through the secure tunnel to your device.

This process essentially masks your real IP address and location, making your browsing activity private and secure.

Key Benefits □

- Data Security: By encrypting your data, a VPN protects you from hackers, ISPs, and government surveillance, especially on public Wi-Fi networks.
- Privacy: Your IP address is hidden, making it difficult for websites, advertisers, and third parties to track your online activity.
- Access to Geo-restricted Content: By connecting to a server in another country, you can access content (like streaming services or websites) that is only available in that region.

- Secure Remote Access: Businesses use VPNs to allow employees to securely access company networks and resources from home or while traveling.

Key Functions of a VPN

1. Encryption of Traffic – Protects your internet data so hackers, ISPs, or governments cannot read it.
2. Privacy Protection – Hides your IP address and location, making your online activity anonymous.
3. Secure Remote Access – Allows employees to safely connect to company servers while working from home.
4. Bypass Restrictions – Lets users access blocked websites or services (e.g., streaming sites, region-specific apps).

Real-Life Examples of VPN Usage

- Corporate Security: Employees working remotely use a VPN to access internal company systems safely.
- Public Wi-Fi Safety: Using a VPN at airports or cafes to prevent hackers from stealing data.
- Streaming & Gaming: Accessing region-restricted services like Netflix or bypassing censorship.

Network Challenges in Modern Computing

Modern networks face challenges due to rapid growth in data demand, devices, cloud services, and IoT. Three major technologies – 5G, Software Defined Networking (SDN), and Network Function Virtualization (NFV) – address these challenges, but they also introduce new complexities.

1. 5G (Fifth Generation Networks)

Opportunities:

- Ultra-fast speeds (up to 10 Gbps).
- Low latency (<1 ms) → essential for autonomous cars, AR/VR, and remote surgery.
- Supports massive IoT (billions of devices).

Challenges:

- High Infrastructure Cost – Need dense deployment of small cells and fiber backhaul.
- Spectrum Management – Limited frequency bands, interference issues.
- Energy Consumption – More base stations increase power usage.
- Network Densification: To achieve high speeds, 5G requires a much denser network of small cells, which are smaller base stations placed closer together. This makes network planning, deployment, and maintenance more complex and costly than with previous generations.
- Security: The increased number of connected devices (IoT) and the reliance on a distributed network architecture create a larger attack surface, making the network more vulnerable to cyber threats. Securing billions of new endpoints is a major challenge.
- Energy Consumption: The dense network of small cells and the increased data traffic can lead to higher power consumption, which is a significant operational and environmental concern.
- Interoperability: Ensuring seamless handoffs and interoperability between 5G and older networks (4G, 3G) is crucial for a smooth user experience.

---

2. SDN (Software Defined Networking)

What it is:

- Separates the control plane (decision-making) from the data plane (traffic forwarding).
- Enables centralized network management via software controllers.

Opportunities:

- Flexible and programmable networks.
- Simplifies management of large-scale data centers and cloud networks.
- Enables automation and dynamic traffic routing.

Challenges:

- Controller as a Single Point of Failure – If it's compromised, the whole network is at risk.
- Scalability – Handling large-scale traffic with centralized controllers.
- Interoperability – Integrating SDN with legacy hardware.
- Security Concerns – Attackers may target SDN controllers or APIs.

---

3. NFV (Network Function Virtualization)

What it is:

- Replaces dedicated hardware appliances (like firewalls, load balancers, routers) with software-based virtual functions running on commodity servers.

Opportunities:

- Reduces cost of specialized hardware.
- Improves scalability and flexibility – network functions can be deployed on demand.
- Speeds up innovation by allowing quick rollout of new services.

Challenges:

- Performance Overheads – Virtualized functions may be slower than dedicated hardware.
- Resource Management – Requires efficient orchestration of virtual machines/containers.
- Reliability Issues – Virtual environments can introduce instability.
- Security Risks – Virtual functions are more exposed to cyberattacks.

Technology	Opportunities	Challenges
5G	High speed, low latency, IoT support	High cost, spectrum issues, energy use, security risks
SDN	Programmability, automation, centralized control	Controller failure, scalability, interoperability, security
NFV	Cost savings, flexibility, fast deployment	Performance, orchestration complexity, reliability, security

---