# ANNAMACHARYA UNIVERSITY

New Boyanapalli, RAJAMPET, 516126, Andhra Pradesh

Department of Artificial Intelligence & Machine Learning


## LECTURE NOTES

### III B TECH – I SEMESTER

### COMPUTER NETWORKS

### 24ACSE52T

### ACADEMIC YEAR 2026 – 2027

Prepared By:

Mr.C.Bhanu Prakash Reddy
Assistant Professor,
Department of AI & ML,
Annamacharya University.

| | | | | |
|---|---|---|---|---|
| Title of the Course: | Computer Networks | | | |
| Category: | PC | | | |
| Year: | III | | | |
| Semester: | I | | | |
| Course Code: | 24ACSE52T | | | |
| Branches: | CSE, AI&ML, CSE(AI), CSE(AIML) & CSE (IoTCSBCT) | | | |
| **Lecture Hours** | **Tutorial Hours** | **Practice Hours** | **Credits** | **Total Hours** |
| 3 | 0 | 0 | 3 | 52 |

**Course Objectives:** This course is designed to

1. Understand the protocol layering and physical level communication.
2. Analyze the performance of a network.
3. Describe the functions of network layer and the various routing protocols.
4. Familiarize the functions and protocols of the Transport layer.
5. Understand the working of various Application Layer Protocols.

**Course Outcomes:**

After completion of the course, students will be able to

1. Explain protocol layering, OSI/TCP-IP models, and physical layer communication.
2. Analyze network performance using metrics like throughput and delay.
3. Apply network layer functions and routing protocols.
4. Explain transport layer functions and differentiate TCP and UDP.
5. Illustrate the working of application layer protocols such as DNS and HTTP.

| **Unit 1** | **Introduction to Physical Layer** | **10** |
|---|---|---|

Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet. Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless transmission.

| **Unit 2** | **Data Link Layer** | **12** |
|---|---|---|

Data link layer: Design issues, framing, Error detection and correction. Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel. Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols. Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols.

| Unit 3 | The Network Layer | 10 |
|---|---|---|
| Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet. | | |

| Unit 4 | The Transport Layer | 10 |
|---|---|---|
| Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols: RTP, RTTP. | | |

| Unit 5 | The Application Layer | 10 |
|---|---|---|
| Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Streaming audio and video. | | |

**Textbook:**

1. Andrew S.Tanenbaum, David j.wetherall, Computer Networks, 7th Edition, PEARSON.

**Reference Books:**


1. James F. Kurose, Keith W. Ross, ―Computer Networking: A Top-Down Approach‖, 6th edition, Pearson, 2019.
2. Forouzan, Datacommunications and Networking, 5th Edition, McGraw Hill Publication.
3. Youlu Zheng, Shakil Akthar, ―Networks for Computer Scientists and Engineers‖, Oxford Publishers, 2016.

# UNIT 1: INTRODUCTION TO PHYSICAL LAYER

## Introduction to Computer Networks

A **computer network** is a collection of computers and other devices connected through communication links to exchange data, share resources, and communicate with one another.

The connected devices may include:

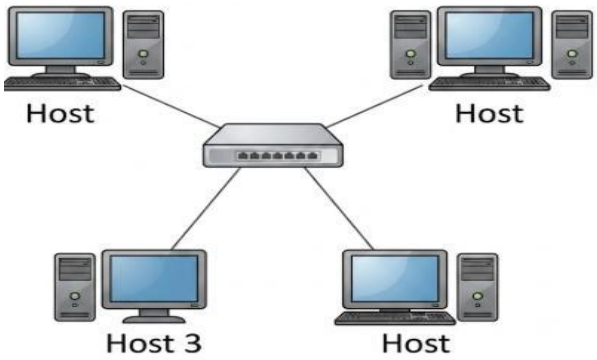
- Desktop computers
- Laptops
- Servers
- Smartphones
- Printers
- Routers
- Switches
- Internet of Things devices

**Example**

In a college computer laboratory, several computers may be connected to:

- A common printer
- A file server
- The internet
- Other computers in the laboratory

This arrangement forms a computer network.




## Objectives of Computer Networks

Computer networks are created for several purposes.

**1. Resource Sharing**

Hardware and software resources can be shared among multiple users.

Examples:

- Sharing a printer
- Sharing an internet connection
- Sharing software applications
- Sharing storage devices

**2. Data Sharing**

Users can exchange files, documents, images, audio, and video through a network.

Example:

A faculty member can upload lecture notes to a college server, and students can access them through the network.

**3. Communication**

Networks support different forms of communication, such as:

- Email
- Video conferencing
- Voice calls
- Instant messaging
- Social networking

**4. Centralized Management**

Network administrators can control users, security, applications, and data from a central location.

**5. Reliability**

Important data can be stored on multiple systems. If one system fails, another system may provide the required data.

**6. Cost Reduction**

Instead of purchasing separate devices for every user, organizations can share resources.

Example:

Twenty employees can use one network printer instead of using twenty individual printers.

## Basic Components of a Computer Network

A computer network contains several basic components.

**1. Sender**

The sender is the device that transmits the data.

Example:

A laptop sending an email.

**2. Receiver**

The receiver is the device that receives the data.

Example:

A smartphone receiving the email.

**3. Message**

The message is the information being transmitted.

It may contain:

- Text
- Images
- Audio
- Video
- Files
- Commands

**4. Transmission Medium**

The transmission medium is the path through which data travels.

Examples:

- Twisted-pair cable
- Coaxial cable
- Optical fibre
- Radio waves

**5. Protocol**

A protocol is a set of rules used by network devices to communicate.

Examples:

- HTTP
- HTTPS
- TCP
- IP
- FTP

# Network Hardware

Network hardware refers to the physical devices and communication equipment used to create, connect, control and operate a computer network. These devices provide the path through which data travels and perform tasks such as signal regeneration, frame forwarding, packet routing, wireless access and protocol conversion. A network cannot function only with software; it requires interfaces, transmission media and interconnecting devices.

## Network hardware includes:


1. **Network Interface Card (NIC):** A NIC connects a computer or smart device to a network. A wired NIC provides an Ethernet port, whereas a wireless NIC uses Wi-Fi. It contains a unique MAC address used for local delivery of frames.
2. **Repeater:** A repeater receives a weak or distorted signal, regenerates the original bit pattern and retransmits it. It extends the useful distance of a communication link but does not understand addresses.
3. **Hub:** A hub is a multiport repeater. Data received on one port is copied to every other port, so all connected devices share the same collision domain. Hubs are simple but inefficient and are largely replaced by switches.
4. **Bridge:** A bridge connects two LAN segments and filters frames using MAC addresses. It learns which devices are present on each side and forwards only necessary traffic.
5. **Switch:** A switch connects devices in a LAN. It learns source MAC addresses and stores them in a forwarding table. A frame is sent only through the port leading to the destination, which reduces collisions and improves performance.
6. **Router:** A router connects different IP networks. It examines destination IP addresses, consults a routing table and forwards packets along a suitable path. A home router connects the private LAN to the Internet and may also provide NAT, DHCP and firewall functions.

7. **Gateway:** A gateway is an entry and exit point that may translate between different protocols, data formats or architectures. Examples include email gateways, VoIP gateways and application gateways.
8. **Modem:** A modem converts digital information into a signal suitable for a service-provider line and converts the received signal back into digital form. Cable, DSL, optical-network terminals and cellular modems provide access to wide-area services.
9. **Wireless Access Point:** An access point connects Wi-Fi devices to a wired LAN. It transmits and receives radio frames and commonly provides authentication and encryption.
10. **Server:** A server provides shared services such as web pages, files, databases, email, authentication or backups to client devices.

## Hub, switch and router comparison

| Feature | Hub | Switch | Router |
|---|---|---|---|
| Main data unit | Bits | Frames | Packets |
| Address used | None | MAC address | IP address |
| Forwarding | To all ports | To selected LAN port | Between different networks |
| Main role | Signal repetition | Efficient LAN connectivity | Path selection and internetworking |
| Typical layer | Physical | Data Link | Network |

## Advantages

- Enables computers and smart devices to communicate and share resources.
- Improves performance by segmenting traffic and selecting efficient paths.
- Supports expansion through additional switches, routers and access points.
- Provides reliability, security filtering and centralized service delivery.

## Limitations

- Initial equipment and installation cost can be high.
- Incorrect configuration may cause loops, address conflicts, congestion or security weaknesses.
- Devices require power, maintenance, firmware updates and skilled administration.
- Failure of a central switch or router can interrupt many users unless redundancy is provided.

## Example and applications

In a college laboratory, every computer contains a NIC and connects to a switch using Ethernet. The switch connects to a router, the router connects to the Internet service provider through a

modem or optical terminal, and wireless access points provide Wi-Fi. Servers host the learning-management system and shared files.

## Conclusion

Network hardware forms the physical foundation of communication. Each device has a specific role: repeaters restore signals, switches forward frames, routers connect networks, gateways translate services and access points provide mobility. Correct selection and configuration of these devices produces a fast, secure and scalable network.

# Types of Networks Based on Coverage Area

**1. Personal Area Network**

A Personal Area Network, or PAN, covers a very small area around an individual.

Examples

- Connecting wireless earphones to a smartphone
- Connecting a smartwatch to a phone
- Sharing files through Bluetooth

The usual coverage is a few metres.

**2. Local Area Network**

A Local Area Network, or LAN, covers a limited geographical area.

Examples

- Computer laboratory
- School building
- Office
- Hospital
- Home network

LANs generally provide high-speed communication.

**3. Metropolitan Area Network**

A Metropolitan Area Network, or MAN, covers a city or a large campus.

Examples

- A network connecting several branches of a bank within one city
- A city-wide cable television network
- A university network connecting multiple campuses

**4. Wide Area Network**

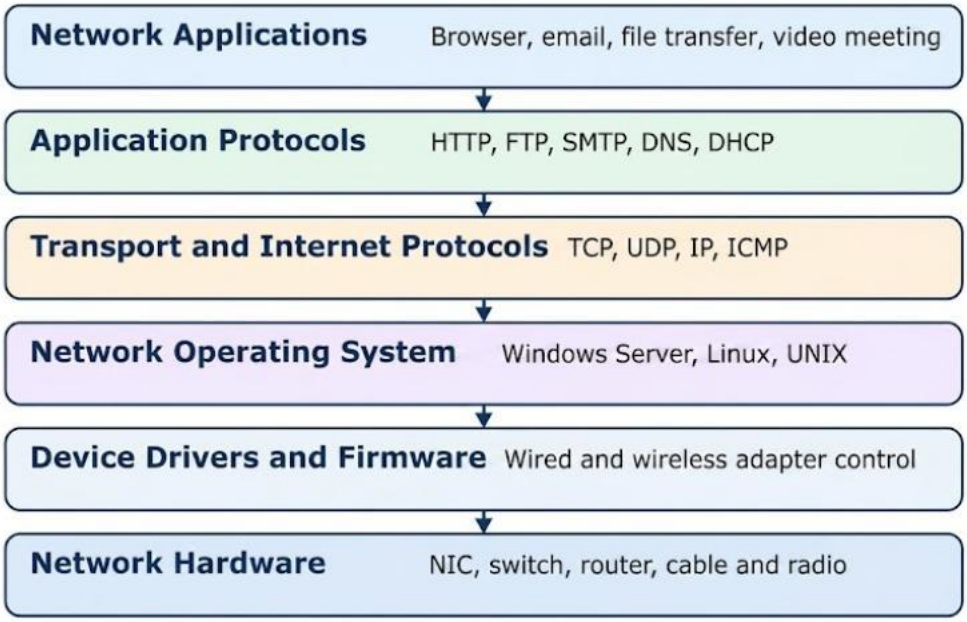
A Wide Area Network, or WAN, covers a large geographical area such as a country, continent, or the entire world.

Examples

- The internet
- A bank network connecting branches across India
- A multinational company network

# Network Software

Network software is the collection of programs, protocols, drivers and services that control network hardware and make communication useful to users and applications. Hardware carries electrical, optical or radio signals, while software determines how data is formatted, addressed, transmitted, received, checked, secured and presented.




## Categories of network software

**Network operating systems:** Windows Server, Linux and UNIX provide user accounts, file and printer sharing, directory services, access control, remote administration and server management.

**Protocols:** Protocols are agreed rules followed by communicating devices. They define message format, meaning, order, timing and error handling.

**Network applications:** Browsers, email clients, file-transfer tools, cloud applications, video-conferencing software and network utilities use protocol services.

**Device drivers and firmware:** A device driver allows the operating system to control a NIC, modem or wireless adapter. Firmware runs inside switches, routers and access points.

**Management software:** Monitoring systems use tools and protocols such as SNMP, logs and performance counters to detect faults, measure traffic and configure devices.

**Security software:** Firewalls, VPN clients, intrusion-detection systems, antivirus products and authentication services protect devices and data.

## Important protocols and their purposes

| Protocol | Purpose |
|---|---|
| HTTP/HTTPS | Transfers web pages; HTTPS adds encryption and server authentication. |
| FTP/SFTP | Transfers files; SFTP provides secure transfer. |
| SMTP, POP3, IMAP | Send and retrieve electronic mail. |
| DNS | Converts human-readable domain names into IP addresses. |
| DHCP | Automatically assigns IP configuration to hosts. |
| TCP | Reliable, connection-oriented and ordered transport. |
| UDP | Fast, connectionless transport with low overhead. |
| IP | Logical addressing and packet delivery across networks. |
| Ethernet / Wi-Fi | Local framing, media access and physical transmission. |

## Layered communication

Networking tasks are divided into layers. Each layer performs a defined function, provides a service to the layer above and uses the service of the layer below. The sender adds control information called headers; the receiver removes it in reverse order. Layering simplifies design, permits standardization, allows one technology to be replaced without redesigning the entire system and makes troubleshooting systematic.

## Connection-oriented and connectionless services

**Connection-oriented service:** A logical connection is established before data is exchanged. Delivery, ordering, acknowledgements and retransmission may be provided. TCP is the main example.

**Connectionless service:** Packets are sent independently without prior connection establishment. It has low overhead but usually provides best-effort delivery. UDP and IP are examples.

## Advantages

- Provides standardized communication between devices from different manufacturers.
- Automates configuration, addressing, routing, security and resource sharing.
- Layering reduces complexity and improves portability and maintainability.
- Management tools support fault detection, performance analysis and centralized control.

## Limitations

- Software errors or misconfiguration can affect many users.
- Protocols add headers and processing overhead.
- Network applications and operating systems require frequent security updates.
- Compatibility problems may occur with proprietary or obsolete protocols.

## Example

When a student opens a website, the browser uses HTTPS, TCP divides the data into segments, IP supplies logical addresses, Wi-Fi or Ethernet forms local frames, and the NIC driver controls the hardware. DNS first resolves the domain name, while security software checks the connection.

## Conclusion

Network software converts raw connectivity into dependable services. Operating systems, protocols, applications, drivers, management tools and security programs work together in layers so that users can communicate without dealing directly with signals and hardware details.

# Reference Models

A reference model divides network communication into layers.

The two major network reference models are:

1. OSI Reference Model
2. TCP/IP Reference Model

# OSI Reference Model

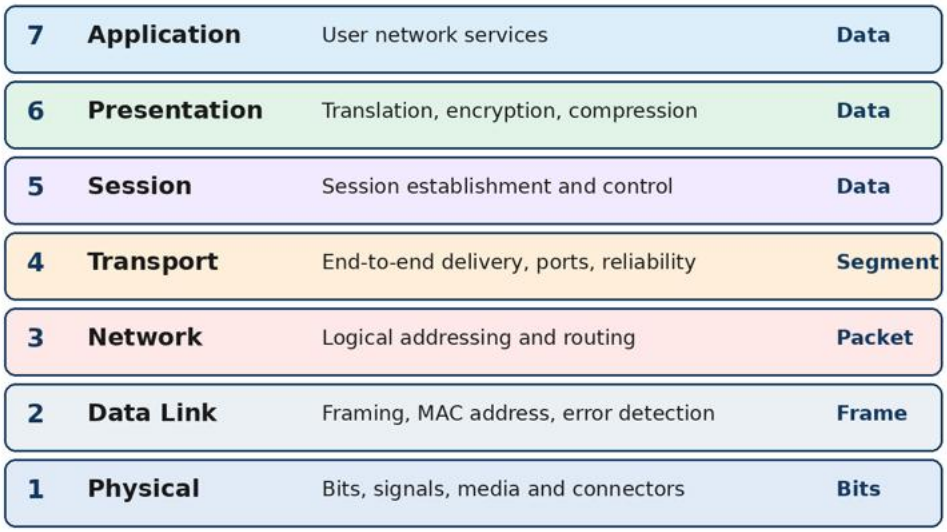
OSI stands for **Open Systems Interconnection**.

The OSI model was developed by the International Organization for Standardization.

It provides a standard framework for understanding how data travels from one device to another through a network.

The OSI model contains seven layers.

Seven Layers of the OSI Model




A common way to remember the layers from Layer 7 to Layer 1 is:

**All People Seem To Need Data Processing**

### 1. Physical Layer

The physical layer is the lowest layer of the OSI model.

It is responsible for transmitting raw bits through a communication medium.

**Functions**

- Transmission of bits
- Definition of cables and connectors
- Representation of binary values as signals

- Data transmission rate
- Physical network topology
- Transmission mode
- Synchronization of bits

**Examples**

- Ethernet cable
- Optical fibre
- Radio signals
- Connectors
- Voltage levels

Data Unit - **Bits**

**2. Data Link Layer**

The data-link layer provides node-to-node data delivery.

It converts raw bits into organized units called frames.

**Functions**

- Framing
- Physical addressing
- Error detection
- Flow control
- Access control

**Example**

A switch mainly operates at the data-link layer.

Data Unit - **Frame**

**3. Network Layer**

The network layer provides logical addressing and routing.

**Functions**

- IP addressing
- Route selection
- Packet forwarding
- Connecting different networks
- Congestion handling

**Example**

A router mainly operates at the network layer.

Data Unit – **Packet**

**4. Transport Layer**

The transport layer provides end-to-end communication between applications.

**Functions**

- Segmentation
- Reassembly
- Error recovery
- Flow control
- Port addressing
- Reliable or unreliable delivery

**Protocols**

- TCP
- UDP

Data Unit - **Segment**

**5. Session Layer**

The session layer establishes, manages, and terminates communication sessions.

**Functions**

- Session establishment
- Session maintenance
- Session termination
- Dialog control
- Synchronization

**Example**

The session layer helps maintain communication between two users during an online meeting.

**6. Presentation Layer**

The presentation layer converts data into a form that the application layer can understand.

**Functions**

- Data translation
- Encryption
- Decryption
- Compression
- Decompression

**Example**

When encrypted data is received, the presentation layer may decrypt it before passing it to the application.

**7. Application Layer**

The application layer is the top layer of the OSI model.

It provides network services directly to user applications.

**Functions**

- Web browsing
- Email communication
- File transfer
- Remote login
- Network resource access

**Protocols**

- HTTP
- HTTPS
- FTP
- SMTP
- DNS

**Example**

A web browser uses application-layer protocols to access websites.

# Data Encapsulation in the OSI Model

When data moves from the application layer to the physical layer, each layer adds control information.

This process is called **encapsulation**.

Encapsulation Process

3. Application data is created.
4. The transport layer adds a transport header.
5. The network layer adds an IP header.
6. The data-link layer adds a frame header and trailer.
7. The physical layer converts the frame into bits and signals.

At the receiving device, the headers are removed layer by layer. This process is called **decapsulation**.

## Protocol data units

| Layer group | PDU |
|---|---|
| Application, Presentation, Session | Data |
| Transport | Segment or datagram |
| Network | Packet |
| Data Link | Frame |
| Physical | Bits |

## Advantages of the OSI model

- Reduces a complex communication process into seven understandable layers.
- Promotes interoperability and standardization.
- Allows protocols and technologies at one layer to change with minimum effect on other layers.
- Provides a common vocabulary for network design, teaching and troubleshooting.
- Helps locate faults by testing one layer at a time.

### Limitations

- It is mainly a reference framework and is not implemented exactly by the Internet.
- Some functions are repeated across layers.
- The session and presentation layers are often combined with applications in practical systems.
- Strict layering can introduce processing overhead and may not match every protocol suite.

## Example: loading a web page

The browser generates an HTTPS request at the application layer. Presentation functions encode and encrypt it; session functions maintain the interaction; TCP at the transport layer provides reliable delivery; IP at the network layer routes the packet; Ethernet or Wi-Fi at the data-link layer forms frames; and the physical layer sends the bits through cable or radio.

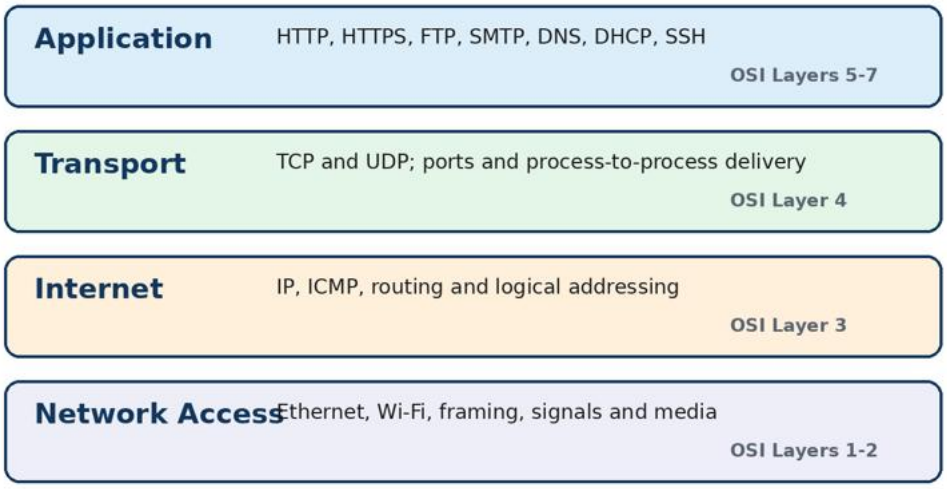
## Conclusion

The OSI model is important because it explains communication in a clear, modular form. Remember the order Application, Presentation, Session, Transport, Network, Data Link and Physical, together with the PDU sequence Data, Segment, Packet, Frame and Bits.

# TCP/IP Reference Model

TCP/IP means Transmission Control Protocol/Internet Protocol. It is the protocol suite and practical reference model used by the Internet. It originated from research funded by the U.S. Department of Defense and was designed to interconnect dissimilar packet-switched networks. The common teaching model contains four layers.

The TCP/IP model commonly contains four layers:




## Layers of the TCP/IP model

**Application layer:** Combines the OSI application, presentation and session functions. It provides services such as web browsing, email, file transfer, name resolution, remote login and configuration. Protocols include HTTP/HTTPS, FTP, SMTP, DNS, DHCP and SSH.

**Transport layer:** Provides communication between application processes. TCP offers connection-oriented, reliable, ordered delivery with acknowledgements and retransmissions. UDP provides connectionless, low-overhead delivery suitable for real-time traffic and simple transactions.

**Internet layer:** Moves packets across interconnected networks. IP provides logical addressing and best-effort delivery, routers forward packets, and ICMP reports errors and diagnostic information.

**Network Access layer:** Covers local frame delivery and physical transmission. It includes MAC addressing, media access, error detection, encoding, connectors and communication media. Ethernet and Wi-Fi are examples.

## How TCP/IP communication works

**1.** An application creates a message using an application protocol.

**2.** TCP or UDP adds source and destination port numbers. TCP may establish a connection and number segments.

**3.** IP adds source and destination IP addresses and creates packets.

**4.** The network-access layer places each packet in a local frame and transmits signals.

**5.** Routers remove the incoming frame, inspect the IP packet and place it in a new frame for the next link.

**6.** The destination removes the headers and delivers the data to the correct application process.

## TCP and UDP comparison

| Feature | TCP | UDP |
|---|---|---|
| Service | Connection-oriented | Connectionless |
| Reliability | Acknowledgement and retransmission | No delivery guarantee |
| Order | Preserves sequence | Packets may arrive out of order |
| Overhead | Higher | Lower |
| Examples | Web, email, file transfer | Streaming, gaming, DNS, voice |

## OSI and TCP/IP comparison

| OSI model | TCP/IP model |
|---|---|
| Seven layers | Four layers |
| Conceptual protocol-independent model | Practical Internet protocol suite |
| Separate session and presentation layers | Functions included in application layer |
| Separate data-link and physical layers | Combined as network access |
| Useful for teaching and troubleshooting | Used in real networks and the Internet |

## Mapping of OSI and TCP/IP Models

| OSI Model | TCP/IP Model |
|---|---|
| Application | Application |
| Presentation | Application |
| Session | Application |
| Transport | Transport |
| Network | Internet |
| Data Link | Network Access |
| Physical | Network Access |

## Advantages

- Open, scalable and independent of a particular hardware platform.
- Supports internetworking across many types of LAN and WAN.
- IP routing can select alternate paths when topology changes.
- Widely implemented and supported by almost every modern operating system.
- Layered design permits new applications and link technologies to be introduced.

## Limitations

- The four-layer model does not clearly separate services, interfaces and protocols.
- The application layer includes many unrelated presentation and session functions.
- Basic IP provides best-effort delivery and requires additional mechanisms for reliability and security.
- Addressing, routing and security configuration can become complex in large networks.

## Conclusion

TCP/IP is the foundation of modern internetworking. Its four layers cooperate to provide application services, end-to-end transport, logical packet delivery and access to physical networks. Knowledge of TCP, UDP and IP is essential for understanding Internet communication.

# EXAMPLE NETWORKS

# ARPANET

ARPANET stands for Advanced Research Projects Agency Network. It was an early wide-area packet-switching network sponsored by the U.S. Advanced Research Projects Agency. Its purpose was to connect research computers, share expensive computing resources and study reliable communication between geographically separated sites.

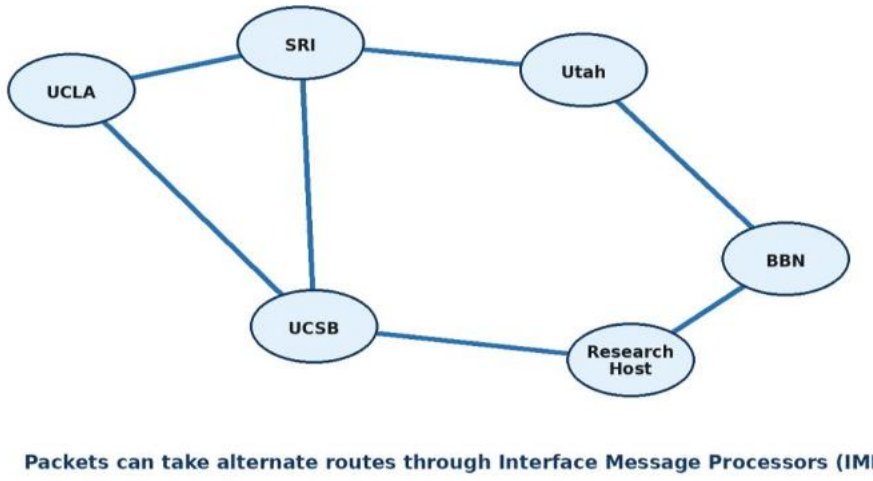



**Figure Simplified ARPANET Packet-Switching Structure**

## Basic architecture

Host computers at universities and research laboratories were connected to specialized packet-switching computers called Interface Message Processors, or IMPs. The IMPs stored and forwarded packets over leased communication links. A host did not need a direct circuit to every other host; packets could pass through several IMPs and use an alternate route when a link was unavailable.

## Packet-switching operation

**1.** The source divided a message into smaller packets.

**2.** Each packet contained addressing and control information.

**3.** An IMP received a packet, temporarily stored it and selected the next link.

**4.** Different packets could follow different paths depending on availability and congestion.

**5.** The destination host reassembled the packets into the original message.

## Important developments

- The first four operational nodes connected UCLA, the Stanford Research Institute, UC Santa Barbara and the University of Utah in 1969.
- The first host-to-host communication attempt occurred between UCLA and SRI.
- ARPANET initially used the Network Control Protocol for host communication.
- Research on connecting independent networks led to TCP/IP.
- ARPANET changed from NCP to TCP/IP on 1 January 1983, an important milestone in Internet history.
- The network was eventually retired after its technologies had spread to newer academic and commercial networks.

## Contributions and advantages

- Demonstrated that packet switching was practical for long-distance computer communication.
- Allowed expensive computing resources and research data to be shared.
- Encouraged the creation of host protocols, email, remote login and file-transfer services.
- Provided experience in routing, congestion, network measurement and fault tolerance.
- Supported the development and testing of TCP/IP and the idea of an internetwork.

## Limitations

- Access was limited mainly to government-funded research institutions.
- Early bandwidth was low and equipment was expensive.
- Initial protocols were not designed for connecting many independent networks.
- Security was not a primary design goal because the community was small and trusted.

- The architecture and addressing methods did not scale indefinitely as the number of networks increased.

## ARPANET and the Internet

| ARPANET | Internet |
|---|---|
| Single pioneering research network | Global network of interconnected networks |
| Limited research community | Public, commercial, academic and government users |
| Early IMP-based packet switching | Routers and diverse wired/wireless technologies |
| Initially NCP, later TCP/IP | TCP/IP is the common protocol suite |
| Historical predecessor | Present worldwide infrastructure |

## Conclusion

ARPANET was not the entire Internet, but it supplied many of the technical ideas, protocols and operational lessons that made the Internet possible. Its most important legacy is the successful use of packet switching and the development of internetworking through TCP/IP.

# Internet

The Internet is a worldwide system of interconnected computer networks that communicate mainly through the TCP/IP protocol suite. It is called a network of networks because millions of private, public, academic, business and government networks are joined through routers and service providers. No single organization owns the entire Internet; standards and resources are coordinated by several technical and administrative bodies.

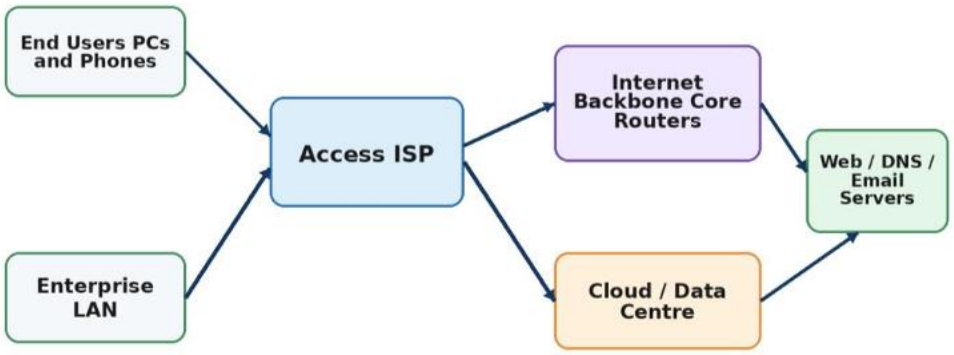



**Figure: Structure of the Internet**

## Main components

- **End systems:** Computers, smartphones, servers, sensors and other hosts that create or consume data.
- **Access networks:** Home Wi-Fi, enterprise LANs, mobile networks and fibre or broadband links that connect users to an Internet service provider.
- **Internet Service Providers:** Organizations that provide connectivity. Local providers connect to regional and global providers through peering or transit arrangements.
- **Routers and backbone links:** High-capacity routers and fibre links carry packets across cities, countries and continents.
- **Data centres and cloud platforms:** Large facilities that host websites, applications, storage, streaming systems and virtual servers.
- **DNS infrastructure:** A distributed naming system that maps domain names to IP addresses.
- **Internet exchange points:** Facilities where networks interconnect and exchange traffic directly.

## Working example: opening a website

1. The user enters a domain name in a browser.
2. DNS resolves the name into an IP address.
3. The device sends packets through the local router and access ISP.
4. Routers examine destination IP addresses and forward packets hop by hop.
5. TCP commonly establishes reliable communication; HTTPS encrypts the request and response.
6. The web server processes the request and returns webpage files.
7. Packets may take different paths and are reassembled at the destination.

## Major Internet services

1. World Wide Web and web applications
2. Electronic mail and messaging
3. File transfer and cloud storage
4. Voice and video communication
5. Streaming media and online gaming
6. Electronic commerce, banking and digital payments
7. Online education, research and collaboration
8. Remote access, Internet of Things and cloud computing

## Advantages

- Provides rapid global communication and access to information.
- Supports education, research, business, entertainment and public services.
- Allows resource sharing through cloud computing and distributed systems.
- Enables remote work, collaboration and low-cost digital publishing.
- Scales by interconnecting independently managed networks using open standards.

## Limitations and risks

- Cyberattacks, malware, phishing, identity theft and privacy loss.
- Misinformation, harmful content and unequal access to reliable connectivity.
- Dependence on service providers, power and infrastructure.
- Congestion and latency can reduce the quality of real-time applications.
- Online services may collect personal data and create legal or ethical concerns.

## Internet, intranet and extranet

| Network | Meaning |
|---|---|
| Internet | Public worldwide interconnection of networks. |
| Intranet | Private network using Internet technologies within an organization. |
| Extranet | Controlled part of an intranet made available to selected external partners. |

## Conclusion

The Internet combines end systems, access networks, service providers, backbone routers, data centres and open protocols into a global communication platform. Its decentralized packet-switched design provides flexibility and scale, but users and administrators must manage security, privacy and reliability risks.

# Physical Layer

- The physical layer is the first and lowest layer of the OSI model.
- It is responsible for transmitting individual bits from one device to another through a physical or wireless medium.
- The physical layer does not examine the meaning of the data. It only transmits binary values in the form of signals.

## Functions of the Physical Layer

### Bit Representation

The physical layer represents binary values using:

- Electrical signals
- Light pulses
- Radio waves

### Data Rate

It defines the number of bits transmitted per second.

The data rate is commonly measured in:

- Bits per second
- Kilobits per second
- Megabits per second
- Gigabits per second

**Bit Synchronization**

The sender and receiver must remain synchronized so that the receiver can identify where each bit begins and ends.

**Line Configuration**

The physical layer defines whether the communication link is:

- Point-to-point
- Multipoint

**Physical Topology**

It defines how devices are physically connected.

Examples:

- Bus
- Star
- Ring
- Mesh

**Transmission Mode**

Communication may occur in:

- Simplex mode
- Half-duplex mode
- Full-duplex mode

**Physical Characteristics**

It defines:

- Cable type
- Connector type
- Signal strength
- Voltage level
- Frequency
- Transmission distance

## Transmission modes

| Mode | Direction | Example |
|---|---|---|
| Simplex | Only one direction | Keyboard to computer; broadcast receiver |

| | | |
|---|---|---|
| Half-duplex | Both directions, but one at a time | Walkie-talkie |
| Full-duplex | Both directions simultaneously | Telephone and switched Ethernet |

## Signal impairments

- **Attenuation:** Loss of signal strength as distance increases. Amplifiers or repeaters may be required.
- **Distortion:** Different frequency components travel differently, changing the signal shape.
- **Noise:** Unwanted electrical or electromagnetic energy is added to the signal. Examples include thermal noise, crosstalk and impulse noise.

## Devices and standards associated with Layer 1

- Repeaters, hubs, transceivers, media converters and the physical part of a NIC.
- Copper cables, optical fibre, antennas, connectors and patch panels.
- Standards that specify voltages, frequencies, modulation, channel width and maximum distance.

## Advantages of standardized physical-layer rules

- Allows devices from different vendors to connect through common media and connectors.
- Provides predictable speed, distance and signal behaviour.
- Supports multiple technologies such as copper, fibre, Wi-Fi and microwave.
- Makes installation, testing and troubleshooting systematic.

## Limitations

- Layer 1 alone cannot identify destinations, detect all errors or guarantee delivery.
- Signals are affected by distance, interference, noise and physical damage.
- Higher speed often requires better cable, shorter distance or more complex modulation.
- Physical installation may be costly in large or difficult locations.

## Example

When a switch transmits an Ethernet frame, its physical interface converts the frame bits into electrical symbols on a twisted-pair cable or light pulses on fibre. The receiving interface recovers timing, identifies the symbols and recreates the bit stream for the data-link layer.

## Conclusion

The physical layer is the foundation on which all higher-layer communication depends. It defines how bits become real signals and how those signals pass through cables or free space. Correct media, connector, speed and transmission-mode choices are therefore essential for reliable networking

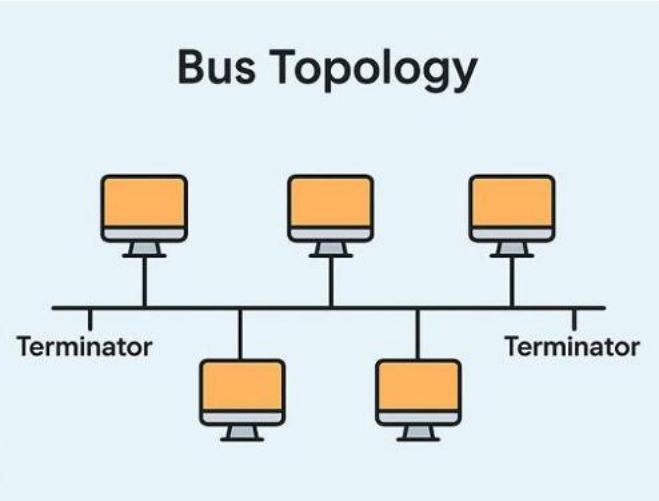
# Physical Network Topologies

A Physical Network Topology refers to the actual physical arrangement of cables, computers, switches, and other networking devices in a network. It shows how devices are physically connected to each other.

The common physical network topologies are:

## 1. Bus Topology

In Bus Topology, all computers are connected to a single communication cable called the backbone cable.




### Working

- Every computer is connected to the main cable.
- When one computer sends data, the signal travels through the entire backbone.
- Only the destination computer accepts the data.
- Terminators are placed at both ends of the cable to prevent signal reflection.

### Advantages

- Easy to install.
- Requires less cable.
- Low installation cost.
- Suitable for small networks.

### Disadvantages

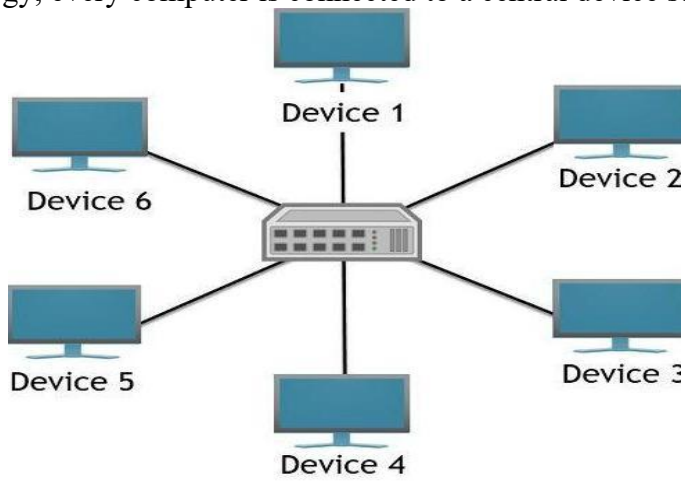
- Backbone failure stops the entire network.
- Difficult to troubleshoot.
- Performance decreases as more devices are added.
- Limited cable length.

### Applications

- Small offices
- Temporary LANs
- Computer laboratories

## 2. Star Topology

In Star Topology, every computer is connected to a central device such as a Hub or Switch.




### Working

- Every device has an individual cable connection to the switch.
- Data always passes through the central device.
- The switch forwards data only to the destination computer.

### Advantages

- Easy to install and manage.
- Failure of one cable affects only one computer.
- Easy to detect faults.
- High network performance.

### Disadvantages

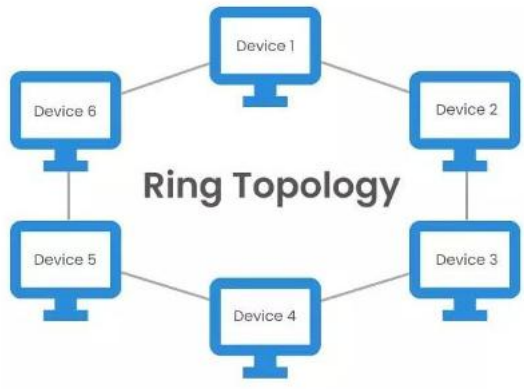
- More cable is required.
- Central switch failure stops the entire network.
- Installation cost is higher.

### Applications

- Schools
- Colleges
- Banks
- Modern office LANs

## 3. Ring Topology

In Ring Topology, each computer is connected to two neighboring computers, forming a closed circular loop.



## Working

- Data travels around the ring.
- Usually, data moves in one direction.
- Each computer forwards the received data to the next computer until it reaches the destination.

## Advantages

- No data collision.
- Equal access for all computers.
- Better performance than Bus topology under heavy traffic.

## Disadvantages

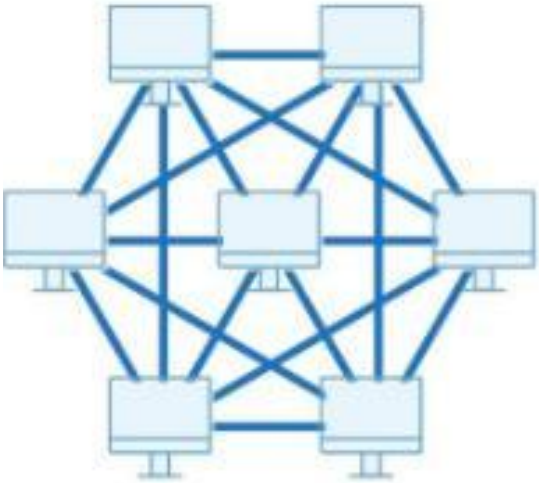
- Failure of one computer may affect the entire network.
- Difficult to add or remove devices.
- Troubleshooting is difficult.

## Applications

- Token Ring networks
- Fiber optic communication systems

# 4. Mesh Topology

In Mesh Topology, every computer is connected directly to every other computer.



## Working

- Multiple communication paths exist.
- If one path fails, another path is available.
- Provides continuous communication.

## Advantages

- Highly reliable.
- Excellent fault tolerance.
- High security.
- No network congestion.

## Disadvantages

- Very expensive.
- Requires a large number of cables.
- Installation is complex.

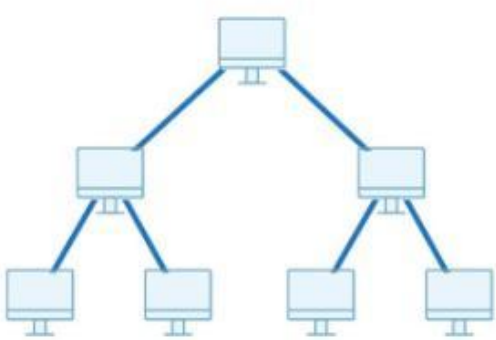
- Difficult to maintain.

## Applications

- Military communication
- Banking networks
- Internet backbone
- Data centers

# 5. Tree Topology

Tree Topology combines Star and Bus topologies in a hierarchical structure.



Tree

## Working

- A root device connects multiple switches.
- Each switch connects several computers.
- Communication flows from the root through branches.

## Advantages

- Easy to expand.
- Supports a large number of devices.
- Easy network management.
- Hierarchical organization.

## Disadvantages

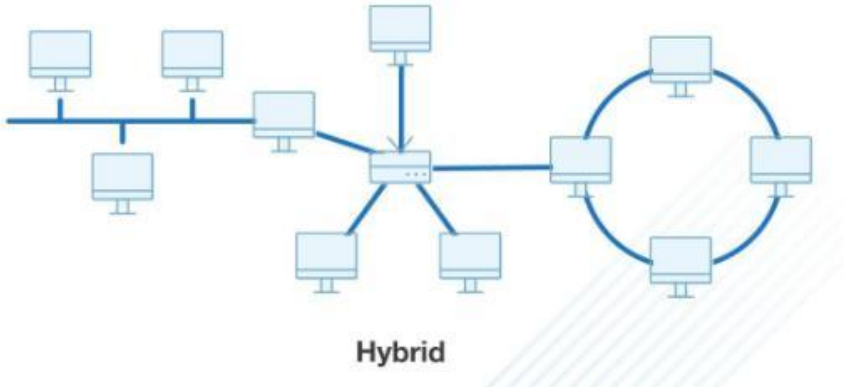
- Root device failure affects large portions of the network.
- More cable required.
- Complex installation.

## Applications

- Universities
- Corporate organizations
- Large campus networks

# 6. Hybrid Topology

Hybrid Topology is formed by combining two or more different topologies, such as Star-Bus or Star-Ring.




## Working

- Different departments may use different topologies.
- These networks are interconnected to form one large network.
- Provides flexibility and scalability.

## Advantages

- Highly flexible.
- Reliable.
- Easy to expand.
- Suitable for large organizations.

## Disadvantages

- High implementation cost.
- Complex design.
- Difficult maintenance.

## Applications

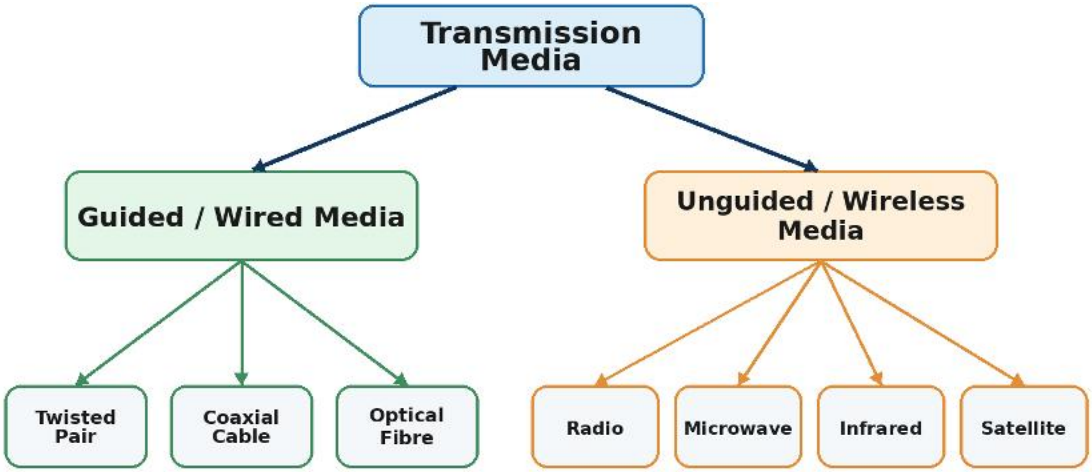
- Multinational companies
- Universities
- Large enterprises
- Industrial networks

# Comparison of Physical Network Topologies

| Topology | Structure | Cost | Reliability | Expansion | Common Use |
|---|---|---|---|---|---|
| **Bus** | Single backbone cable | Low | Low | Difficult | Small LANs |
| **Star** | Central Hub/Switch | Medium | High | Easy | Offices, Schools |
| **Ring** | Circular connection | Medium | Medium | Difficult | Token Ring Networks |
| **Mesh** | Every node connected to every other node | Very High | Very High | Difficult | Military, Data Centers |
| **Tree** | Hierarchical structure | High | High | Easy | Universities, Enterprises |
| **Hybrid** | Combination of topologies | Very High | Very High | Very Easy | Large Organizations |

# Guided Transmission Media

Transmission media provide the path between a sender and receiver. They are classified as guided media, in which signals follow a physical conductor, and unguided media, in which electromagnetic waves travel through air or space. Guided media are widely used for stable, secure and high-capacity fixed connections.




## Types of guided media

- **Twisted-pair cable:** Two insulated copper conductors are twisted together to reduce electromagnetic interference and crosstalk. UTP is inexpensive and common in LANs; STP adds shielding for noisy environments.
- **Coaxial cable:** A central conductor is surrounded by dielectric insulation, a metallic shield and an outer jacket. The common axis and shielding provide better noise resistance than basic twisted pair.
- **Optical fibre:** A glass or plastic core carries light. Cladding confines light by total internal reflection. Fibre provides very high bandwidth, long distance, low attenuation and immunity to electromagnetic interference.

## Factors used to select a medium

- Required data rate and available bandwidth
- Maximum link distance and acceptable attenuation
- Electromagnetic-noise environment
- Installation and maintenance cost
- Security and resistance to tapping
- Physical flexibility, weight and available conduits
- Future expansion and expected service life

**Detailed comparison**

| Feature | Twisted Pair | Coaxial Cable | Optical Fibre |
|---|---|---|---|
| Signal | Electrical | Electrical | Light |
| Relative bandwidth | Medium to high | Medium to high | Very high |
| Typical distance | Short LAN links | Medium | Long |
| Noise immunity | Low to good | Good | Excellent |
| Cost | Low | Medium | Higher installation cost |
| Security | Lower | Moderate | High |
| Installation | Easy | Moderate | Needs skill |
| Applications | Ethernet, telephone | Cable TV, broadband, CCTV | Backbone, telecom, data centre |

## Advantages of guided media

- Provides a defined path and predictable coverage.
- Usually offers greater security and lower interference than open wireless links.
- Supports reliable high-speed full-duplex communication.
- Fibre can carry enormous capacity over long distances.

## Limitations

- Cables restrict mobility and must be physically installed.
- Copper cables can suffer attenuation, corrosion and electromagnetic interference.
- Cable cuts or damaged connectors interrupt communication.
- Fibre termination and testing require specialized equipment and skills.

## Example selection

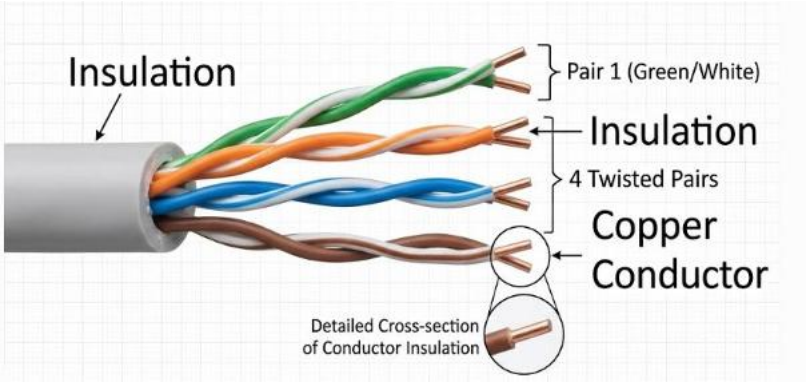
A college may use Cat 6 twisted pair from classroom outlets to floor switches, fibre between buildings and data-centre switches, and coaxial cable for CCTV or cable-television distribution. This mixed design uses each medium where its properties are most suitable.

## Conclusion

Guided media remain essential because they provide dependable, controlled communication paths. Twisted pair is economical, coaxial cable is well shielded, and optical fibre gives the highest capacity and distance. Selection must balance speed, distance, environment, security and cost.

# Twisted-Pair Cable

A twisted-pair cable consists of two individually insulated copper conductors twisted around each other. One conductor carries the signal and the other provides the return path in balanced transmission. Twisting makes both conductors experience nearly the same external interference, allowing the receiver to reject much of the induced noise. Several pairs are commonly placed inside one protective jacket.




## Types

**Unshielded Twisted Pair (UTP):** Contains insulated pairs and an outer jacket without an additional metallic shield. It is light, flexible, inexpensive and widely used for Ethernet and telephone wiring.

**Shielded Twisted Pair (STP):** Adds foil or braided shielding around individual pairs or the complete cable. The shield reduces electromagnetic interference but increases cost, thickness and grounding requirements.

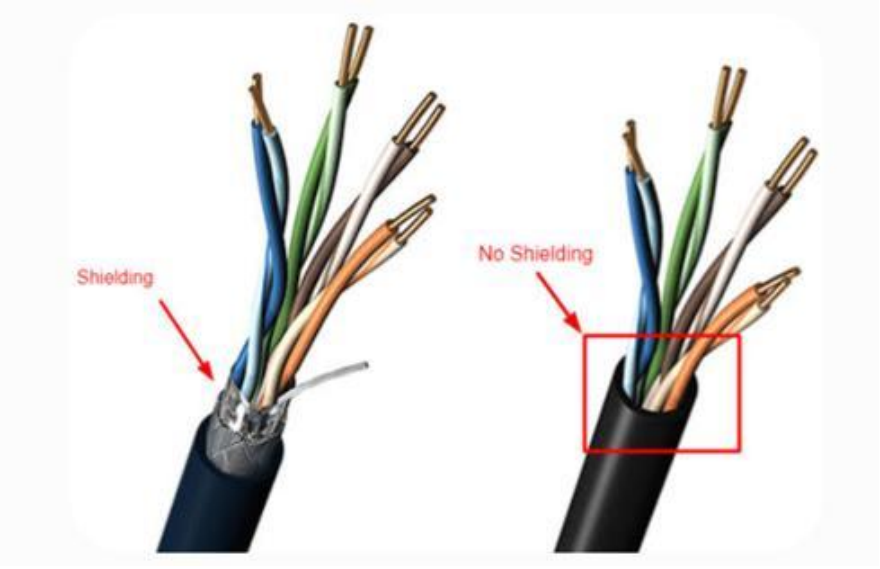



**Figure: Shielding (STP) and No Shielding (UTP)**

## Working principle

Digital data is represented by changing electrical signals applied to the pair. Because the wires are twisted, interference induced in one conductor is also induced in the other. A differential receiver responds mainly to the voltage difference between the conductors and rejects common noise. More twists per unit length generally improve crosstalk control.

## Common categories

| Category | Typical capability / use |
|---|---|
| Cat 3 | Telephone and older low-speed networks |
| Cat 5 | Older Fast Ethernet installations |
| Cat 5e | Common Gigabit Ethernet cabling |
| Cat 6 | Higher performance and reduced crosstalk; Gigabit LANs |
| Cat 6A | Designed for higher-speed Ethernet over standard horizontal distances |
| Cat 7/8 | Shielded, specialized high-performance installations and data centres |

## Connectors and installation

Ethernet twisted-pair links commonly use an eight-position modular connector generally called RJ-45. Correct pair termination, bend radius, cable length, separation from power wiring and testing are important. Poor untwisting at the connector increases crosstalk.

## Advantages

- Low cost and widely available.
- Lightweight, flexible and easy to route through buildings.
- Simple termination and fault replacement.
- Supports voice and high-speed LAN communication.
- UTP does not require complex shielding; STP performs well in noisy locations.

## Disadvantages

- Higher attenuation and shorter distance than optical fibre.
- Susceptible to electromagnetic interference, especially UTP.
- Crosstalk may occur between adjacent pairs.
- Copper can conduct surges and create ground-potential problems.
- Performance depends strongly on cable category and installation quality.

## Applications and example

- Ethernet connections from computers to switches
- Telephone wiring and digital subscriber lines
- IP cameras, access points and voice-over-IP phones
- Building automation and control systems

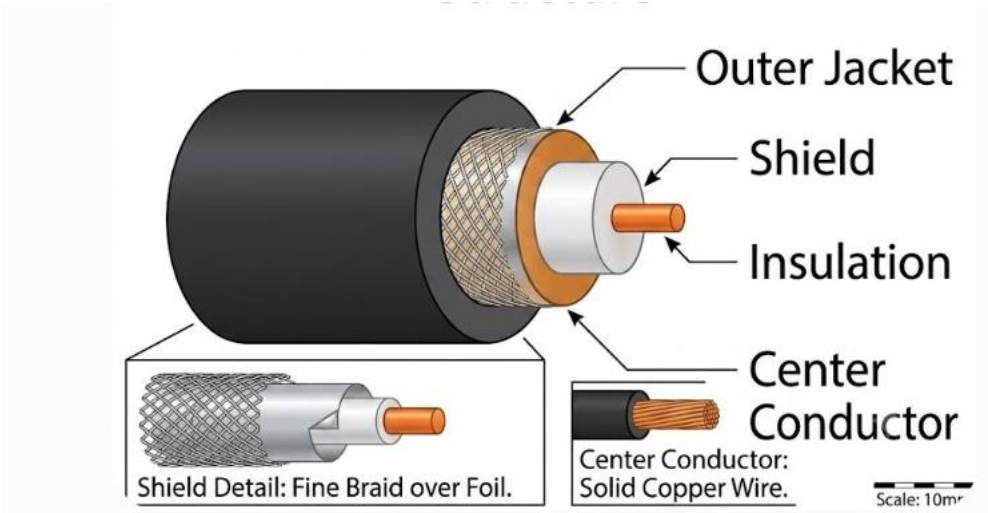
**Example:** A desktop is connected to a floor switch through a Cat 6 UTP cable. Four twisted pairs carry full-duplex Ethernet signals, while standardized connectors permit easy replacement and testing.

## Conclusion

Twisted pair is the most common short-distance copper medium because it combines adequate speed with low cost and simple installation. UTP is preferred in normal offices, whereas STP is useful where electromagnetic interference is strong.

# Coaxial Cable

A coaxial cable is a copper transmission medium in which the central conductor and outer metallic shield share the same geometric axis. The conductor carries the signal, the dielectric maintains spacing and impedance, the shield provides the return path and blocks interference, and the outer jacket protects the cable.




## Parts of a coaxial cable

- **Central conductor:** Solid or stranded copper wire that carries the information-bearing electrical signal.
- **Dielectric insulator:** Non-conducting material that keeps the conductor centered and determines electrical properties such as characteristic impedance.
- **Metallic shield:** Braided wire, foil or both. It provides a return path and shields the internal signal from external electromagnetic noise.
- **Outer jacket:** Protective plastic covering that resists moisture, abrasion and mechanical damage.

## Working principle

The signal travels as an electromagnetic field between the inner conductor and the outer shield. Because the field is substantially confined inside the cable, coaxial cable radiates less energy and receives less external interference than an unshielded pair. Correct impedance matching and proper termination are necessary to prevent reflections and signal distortion.

## Characteristics and types

- Common systems use characteristic impedances such as 50 ohms for radio/data applications and 75 ohms for television/video distribution.
- Connectors include BNC, F-type and N-type depending on the application.
- Coaxial cable supports baseband or broadband signalling and can carry radio-frequency channels.

## Advantages

- Good shielding and better noise immunity than basic twisted pair.
- Supports higher frequencies and longer segments than older unshielded copper links.
- Strong, durable and suitable for outdoor or industrial use when correctly jacketed.
- Can carry multiple television or broadband channels.

## Disadvantages

- Bulkier and less flexible than twisted-pair cable.
- Installation and connector termination require more care.
- More expensive than common UTP for office LANs.
- Cable faults and poor terminations can create reflections.
- Lower bandwidth and greater attenuation than optical fibre for long-distance high-capacity links.

## Applications

- Cable television distribution
- Cable broadband Internet access
- CCTV and video-surveillance systems
- Radio-frequency antenna feeder lines
- Laboratory measurement equipment and older Ethernet systems

## Example

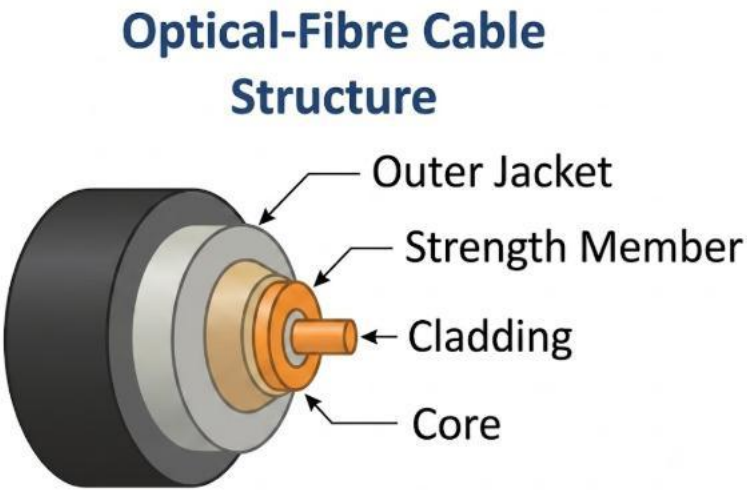
A cable television operator sends many frequency channels through a 75-ohm coaxial distribution network. The metallic shield protects the weak radio-frequency signals from household electrical interference, while an F-type connector joins the cable to the modem or television receiver.

## Conclusion

Coaxial cable offers a useful balance of shielding, bandwidth and mechanical strength. Although modern LANs usually prefer twisted pair or fibre, coaxial cable remains important for television, broadband, video and radio-frequency applications.

# Optical Fibre

Optical fibre is a thin, flexible dielectric waveguide made of glass or plastic that carries information as pulses of light. The central core has a slightly higher refractive index than the surrounding cladding. A coating, strength members and outer jacket protect the fragile glass from moisture, bending and mechanical stress.




Optical fibre is a transmission medium that carries data in the form of light pulses.

It is made of glass or plastic.

An optical fibre contains:

1. Core
2. Cladding
3. Protective coating
4. Outer jacket

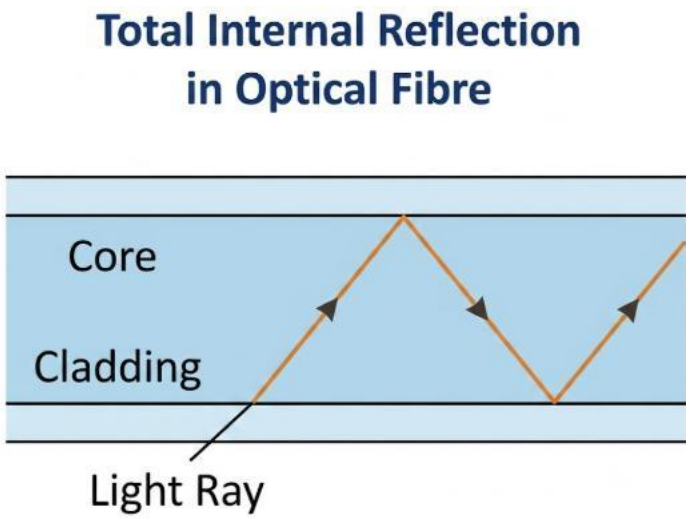
**Core**
The core is the central part through which light travels.

**Cladding**
The cladding surrounds the core and reflects light back into the core.

**Protective Coating**
The protective coating protects the fibre from moisture and physical damage

## Working principle: total internal reflection

When light travels from a medium with a higher refractive index to one with a lower refractive index and strikes the boundary at an angle greater than the critical angle, it is reflected completely back into the first medium. In a fibre, the core has the higher refractive index and the cladding has the lower index. Properly launched light is therefore guided along the core by repeated total internal reflection.

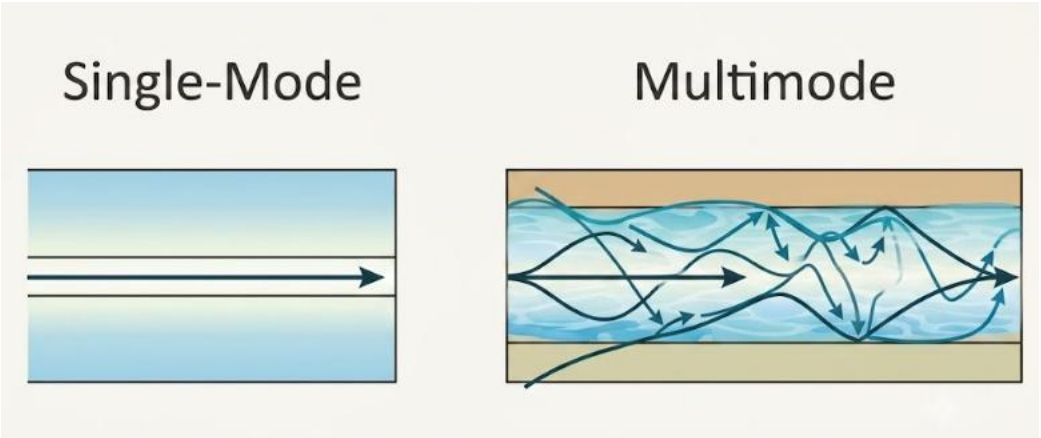



## Optical communication process

- A transmitter converts electrical bits into controlled light pulses using an LED or laser.
- The pulses enter the fibre core through a connector or splice.
- **3.** The core and cladding guide the light over the link.
- At the receiver, a photodetector converts light into an electrical signal.
- **5**Electronic circuits recover timing and reconstruct the transmitted bit stream.

### Types of fibre

- **Single-mode fibre:** Has a very small core and permits essentially one propagation mode. It has low modal dispersion, supports very high capacity and is used for long-distance telecommunications and backbones.
- **Multimode fibre:** Has a larger core and permits many light paths. It is easier to couple but suffers modal dispersion, so it is used for shorter building, campus and data-centre links.



## Losses and dispersion

- Absorption converts some optical energy into heat.
- Scattering redirects light because of microscopic material variations.
- Bending losses occur when the fibre is curved too sharply.
- Dispersion spreads pulses in time and limits the maximum distance at a given data rate.

## Advantages

- Extremely high bandwidth and data-carrying capacity.
- Low attenuation permits long repeater spacing.
- Immune to electromagnetic interference and crosstalk.
- Electrically non-conductive and suitable between buildings.
- Small diameter, low weight and high resistance to corrosion.
- Difficult to tap without disturbing the signal, providing better physical security.

## Disadvantages

- Higher initial cost for optical modules, splicing and test equipment.
- Glass fibre is sensitive to excessive bending and pulling.
- Installation and repair require trained personnel.
- Cannot directly carry electrical power to remote devices.
- Connectors must be clean and accurately aligned.

## Applications

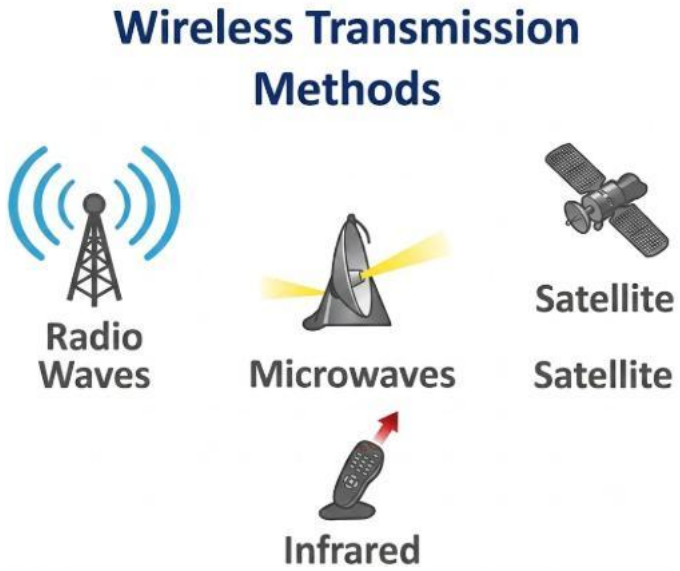
- Internet and telecommunications backbones
- Fibre-to-the-home broadband
- Submarine communication cables
- Data-centre and campus networks
- Cable television distribution
- Medical endoscopy and industrial sensing

## Conclusion

Optical fibre is the preferred medium for high-speed and long-distance communication. Its use of light, total internal reflection, low attenuation and immunity to electromagnetic noise gives it major advantages over copper, especially in modern backbones and data centres.

# Wireless Transmission

Wireless or unguided transmission sends information through free space using electromagnetic waves rather than a continuous physical cable. Antennas convert electrical signals into electromagnetic radiation and convert received waves back into electrical form. Wireless systems provide mobility and rapid deployment, but the shared medium introduces interference, security and propagation challenges.




## Radio-wave transmission

Radio waves can be omnidirectional and may penetrate walls, depending on frequency and building material. They are used for broadcasting, Wi-Fi, Bluetooth, mobile communication and Internet-of-Things devices. Their broad coverage simplifies mobility, but signals may interfere and can be received by unauthorized users unless encryption is used.

## Microwave transmission

Microwaves are higher-frequency, usually directional waves. Terrestrial microwave links use aligned antennas on towers or rooftops and generally require line of sight. They provide high-capacity point-to-point links and support cellular backhaul. Obstacles, antenna misalignment and rain at some frequency bands can reduce quality.

## Infrared transmission

Infrared communication is used over short distances and normally does not pass through walls. It may use direct line of sight or reflections from room surfaces. Television remote controls, sensors and short-range device links are familiar examples. Its limited room confinement can reduce interference and improve privacy, but sunlight and obstacles may disrupt communication.

## Satellite communication

A ground station transmits an uplink signal to a satellite. A transponder receives, shifts frequency, amplifies and retransmits the signal as a downlink to another region. Satellites provide broad coverage for broadcasting, navigation, remote-area Internet, weather observation and global communication. Long propagation distance can introduce delay, and launch and operation are expensive.

**Comparison**

| Method | Direction / range | Typical applications | Main limitation |
|---|---|---|---|
| Radio waves | Often omnidirectional; short to wide area | Broadcast, Wi-Fi, Bluetooth, mobile | Interference and security |
| Terrestrial microwave | Directional; long line-of-sight links | Backhaul, building links | Obstacles and alignment |
| Infrared | Short range; room scale | Remote controls, sensors | Cannot pass through walls |
| Satellite | Very wide or global coverage | TV, navigation, remote Internet | Cost and propagation delay |

## General advantages

- Supports mobility and roaming users.
- Fast deployment where cable installation is difficult.
- Can cover remote, temporary or moving locations.
- Enables broadcasting from one transmitter to many receivers.
- Scales conveniently by adding wireless clients within coverage.

## General disadvantages

- Shared spectrum causes interference and variable performance.
- Signals can be intercepted, so authentication and encryption are essential.
- Buildings, terrain, distance and weather affect propagation.
- Available spectrum and transmit power are regulated.
- Wireless capacity may be lower or less predictable than a dedicated fibre link.

## Example

A university uses Wi-Fi radio links for students, point-to-point microwave between two nearby buildings where trenching is impossible, infrared remote controls in classrooms and satellite service as backup connectivity for a remote field station.

## Conclusion

Wireless transmission complements guided media by providing mobility, flexibility and wide coverage. Radio, microwave, infrared and satellite systems have different propagation properties; the correct choice depends on range, directionality, bandwidth, obstacles, delay, security and cost.

# UNIT 2: DATA LINK LAYER

## Data Link Layer and Design Issues

The Data Link Layer is the second layer of the OSI reference model. It accepts raw bit streams from the Physical Layer and converts them into reliable, well-structured frames for delivery between two directly connected nodes. Its responsibility is therefore node-to-node delivery rather than end-to-end delivery across an entire internetwork.

A physical link may introduce noise, duplicate frames, lose frames or allow a fast sender to overwhelm a slow receiver. The Data Link Layer uses framing, addressing, error control, flow control and medium-access rules to hide many of these problems from the Network Layer.

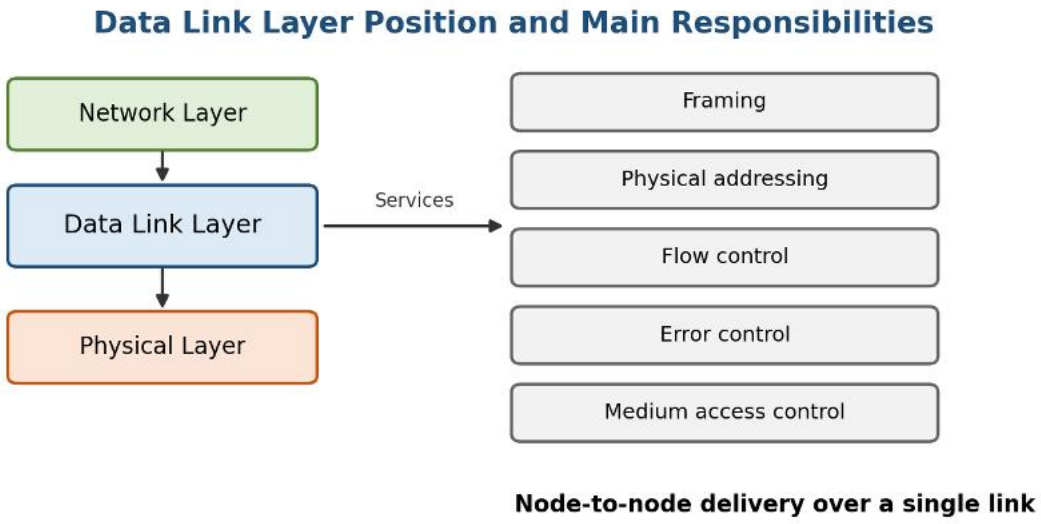



*Figure: Position and major responsibilities of the Data Link Layer*

### Services provided to the Network Layer

**Unacknowledged connectionless service:** Frames are sent without connection establishment or acknowledgements. It is suitable for highly reliable media and real-time traffic where occasional loss is preferable to delay.

**Acknowledged connectionless service:** Each frame is individually acknowledged. Lost or damaged frames are retransmitted, making the service useful on unreliable wireless links.

**Acknowledged connection-oriented service:** A logical connection is established, frames are numbered and delivered exactly once in sequence, and the connection is released after transfer.

### Major design issues

**1. Framing:** The continuous stream of bits must be divided into distinguishable units called frames. The receiver must recognize the beginning and end of every frame even when the data contains patterns similar to delimiters.

**2. Physical addressing:** A frame normally carries source and destination MAC addresses so that a device on a shared LAN can determine whether the frame is intended for it.

**3. Flow control:** The sender must not transmit faster than the receiver can process or buffer frames. Stop-and-wait and sliding-window methods regulate the number of outstanding frames.

**4. Error control:** The layer detects damaged frames and recovers from loss, corruption or duplication using error-detecting codes, acknowledgements, sequence numbers, timers and retransmissions.

**5. Access control:** When many stations share one broadcast channel, rules are needed to decide which station may transmit. ALOHA, CSMA and collision-free protocols solve this problem.

**6. Link management:** The layer may establish, maintain and release a logical link, negotiate parameters and report link failures.

## Example

When a laptop sends a frame to a switch through Ethernet, the Data Link Layer places the network-layer packet inside an Ethernet frame, adds source and destination MAC addresses, calculates a frame check sequence and transmits the frame. The receiving interface checks the frame, removes the header and trailer, and passes the packet upward only if the frame is valid.

## Advantages

- Provides orderly node-to-node transfer over an imperfect physical medium.
- Detects corruption and prevents uncontrolled receiver-buffer overflow.
- Supports both point-to-point and shared broadcast links.
- Separates link-specific details from network-layer routing.

## Limitations

- Reliability mechanisms add header, acknowledgement and retransmission overhead.
- A reliable link cannot by itself guarantee end-to-end delivery across multiple networks.
- Complex protocols require buffers, timers and additional processing.

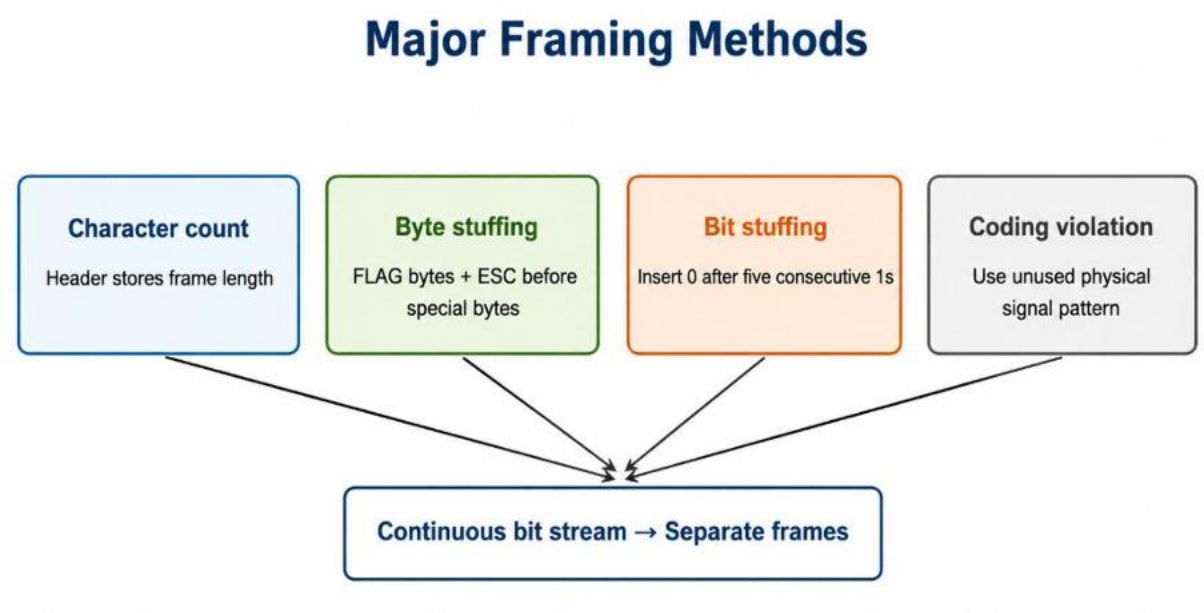
## Conclusion

The Data Link Layer is the immediate manager of a communication link. By solving framing, addressing, flow, error and channel-access problems, it converts an unreliable physical connection into a service that the Network Layer can use effectively.

# Framing

Framing is the process of dividing a continuous sequence of transmitted bits into identifiable blocks called frames. Each frame usually contains a header, a payload and a trailer. The header carries control information such as addresses and sequence numbers, while the trailer often contains an error-detecting field.

Without framing, the receiver would see only an uninterrupted stream of zeros and ones and would not know where one message ends and the next begins. A good framing method must identify boundaries, remain transparent to arbitrary user data, detect malformed frames and introduce limited overhead.




**1. Character count**

The first field in the frame header specifies the total number of bytes in the frame. After reading the count, the receiver knows where the next frame begins. For **example**, a count value of 5 may indicate one count byte followed by four data bytes.

**Advantage:** Simple and has small overhead.

**Limitation:** If the count field is corrupted, the receiver loses synchronization and may interpret several later frames incorrectly.

**2. Byte-oriented framing and byte stuffing**

Special flag bytes mark the beginning and end of a frame. If the same flag occurs inside the payload, the sender inserts an escape byte before it. If an escape byte itself occurs in the payload, it is also escaped. The receiver removes these inserted escape bytes before delivering the data.

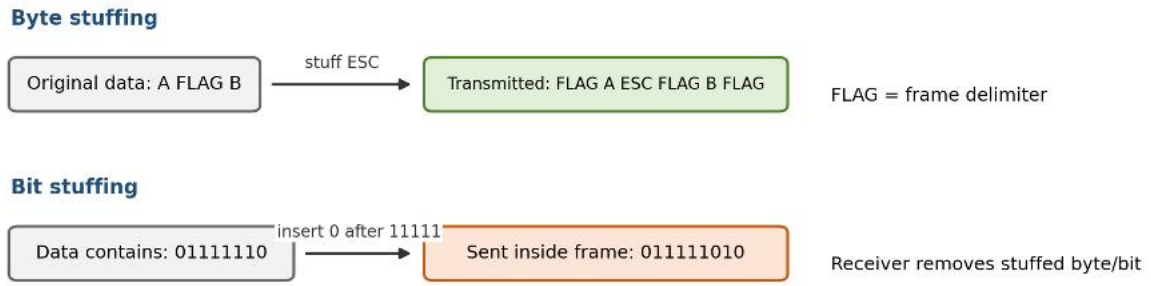



*Figure: Examples of byte stuffing and bit stuffing*

**Example**: Suppose FLAG is the delimiter and ESC is the escape byte. Data A FLAG B is transmitted as FLAG A ESC FLAG B FLAG. The receiver recognizes the outer FLAG bytes as boundaries and restores the original payload.

**Advantages:** Easy to implement in character-oriented systems and fully transparent to ordinary data bytes.

**Limitations:** The method depends on a defined byte size and may add many escape bytes when special values occur frequently.

### 3. Bit-oriented framing and bit stuffing

A special bit pattern, commonly 01111110, is used as the frame flag. To prevent this pattern from appearing accidentally in the payload, the sender inserts a 0 after every sequence of five consecutive 1s. The receiver removes a 0 that follows five consecutive 1s. This method is used by bit-oriented protocols such as HDLC.

**Example:** Payload bits 01111110 become 011111010 inside the frame because a zero is inserted after the fifth consecutive 1. The inserted bit is removed at the receiver.

**Advantages:** Independent of character encoding and suitable for arbitrary binary data.

**Limitations:** Requires bit-level processing and produces variable overhead depending on the data pattern.

### 4. Physical-layer coding violations

Some physical encodings contain signal patterns that are not used for normal data symbols. These unused patterns can serve as frame boundaries. Because the delimiter cannot occur in valid encoded data, no stuffing is necessary. The technique is efficient but is available only when the physical coding scheme provides unused symbols.

## Comparison of framing methods

| Method | Boundary information | Main advantage | Main limitation |
|---|---|---|---|
| Character count | Length field | Very simple | Corrupted count loses synchronization |
| Byte stuffing | Flag and escape bytes | Suitable for byte-oriented links | Extra bytes and byte-code dependence |
| Bit stuffing | Flag bit pattern and inserted zeros | Transparent to any bit pattern | Bit-processing overhead |
| Coding violation | Unused signal symbol | No stuffing overhead | Depends on physical encoding |

## Conclusion

Framing is essential because it gives structure to an otherwise continuous bit stream. Bit stuffing is widely used in bit-oriented protocols, byte stuffing is common on character-oriented links, and coding violations are attractive when supported by the physical layer.

# Error Detection and Correction

**Introduction**

Imagine sending an important email, only to find out the data got corrupted halfway through transmission. In digital communication, even a single wrong bit can change everything.

That's where error detection and correction steps in — ensuring your data remains accurate and reliable across noisy channels, making it a core concept every CS student must master.

According to IEEE research, communication systems lose up to 15% of data accuracy without robust error correction protocols. Modern systems, such as TCP/IP and satellite networks, rely heavily on these techniques to achieve near-perfect reliability.

## What is Error Correction and Detection?

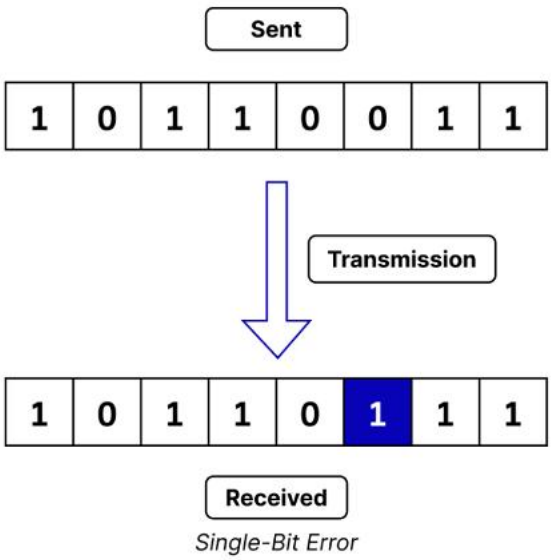
Error correction is a method used to identify and correct errors in data communications or storage that have not been retransmitted. The technique is essential for maintaining data accuracy, especially in areas where it is challenging or costly to resend the data. Additionally, this process allows the receiving side to identify and correct errors that may have been introduced during the signal's transmission. By allowing error correction, digital communication systems become less vulnerable; thus, the data received can be considered reliable and trustworthy.

On the other hand, error detection refers to the methods and techniques used to identify errors that may occur during the transmission or storage of data. The primary goal is to ensure that the data received matches what was originally sent. Error detection identifies the presence of errors; it plays a vital role in maintaining data integrity in communication systems.

## Types of Errors in Computer Networks

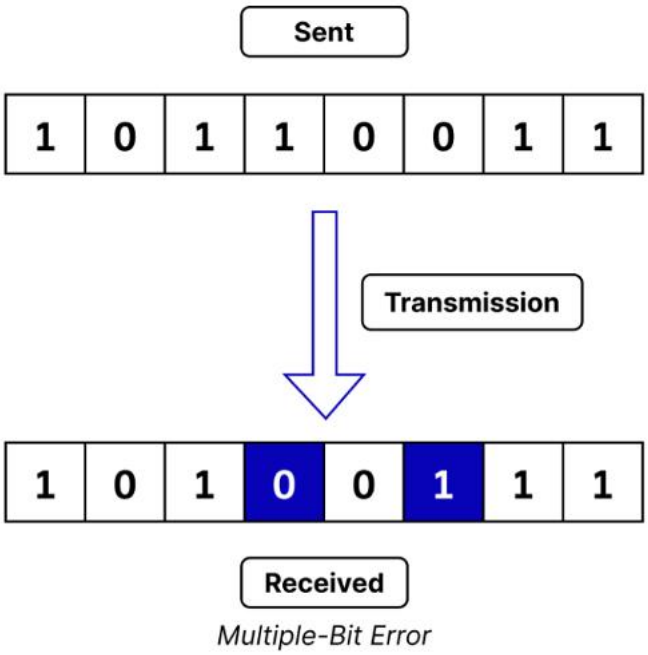
Here are the types of errors in computer networks

**1. Single-Bit Error:** This type of error occurs when one bit of a transmitted data unit is altered, leading to corrupted data.



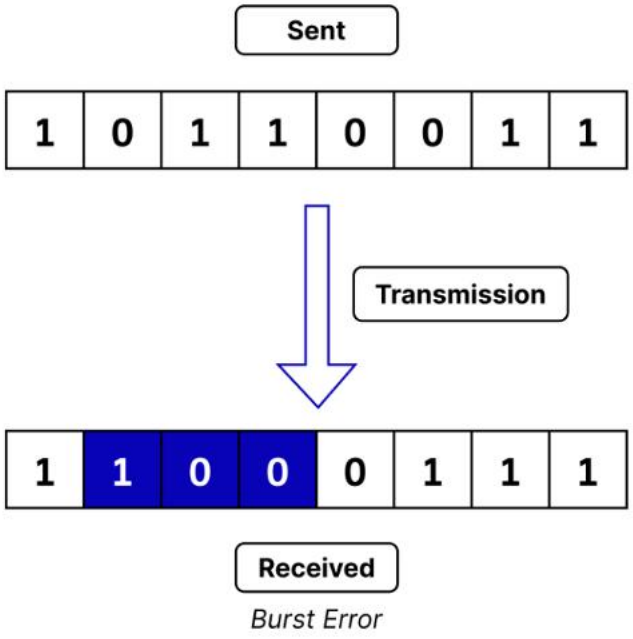
*Single-Bit Error*

**2. Multiple-Bit Error:** This type of error occurs when more than one bit is affected. While rarer than single-bit errors, they can occur in high-noise environments.



*Multiple-Bit Error*

**3. Burst Error:** This type of error occurs when a sequence of consecutive bits is flipped, resulting in several adjacent bits being incorrect.

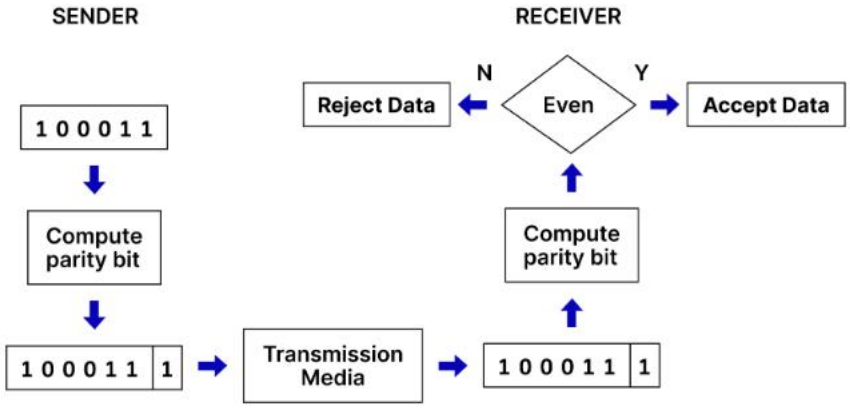


*Burst Error*

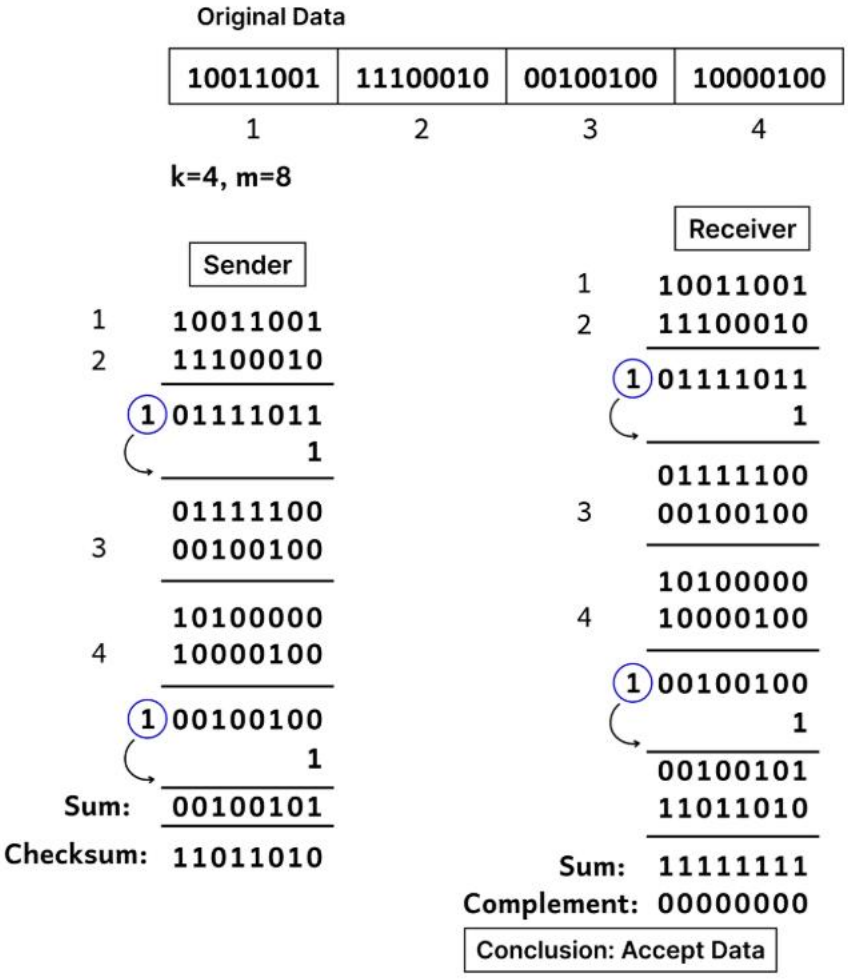
# Error Detection Techniques

Error detection techniques are essential in data transmission and storage to ensure data integrity. Here are some common methods:

**1. Parity Bits:** A simple method that adds a single bit to data to ensure the total number of 1s is even (even parity) or odd (odd parity).




**2. Checksums:** A mathematical sum of data values calculated before transmission and verified at the destination. If the checksum doesn't match, an error is detected.



**3. Cyclic Redundancy Check (CRC):**

- CRC is based on binary division, unlike checksum which uses addition; redundant CRC bits are appended so the transmitted data becomes exactly divisible by a predefined generator polynomial.
- At the receiver, the data is divided using the same polynomial; a zero remainder means the data is accepted, while a non-zero remainder indicates transmission errors.
- Highly effective in detecting single-bit, multiple-bit, and burst errors, making it more reliable than parity checks and checksum methods.
- Widely used in communication protocols such as Ethernet, HDLC, and USB due to its strong error detection capability.
- Provides only error detection, not error correction; its effectiveness depends on the chosen generator polynomial.
- More computationally complex than simple parity or checksum methods and introduces additional processing overhead.

# CRC Working

We have given dataword of length n and divisor of length k.

Step 1: Append (k-1) zero's to the original message

Step 2: Perform modulo 2 division

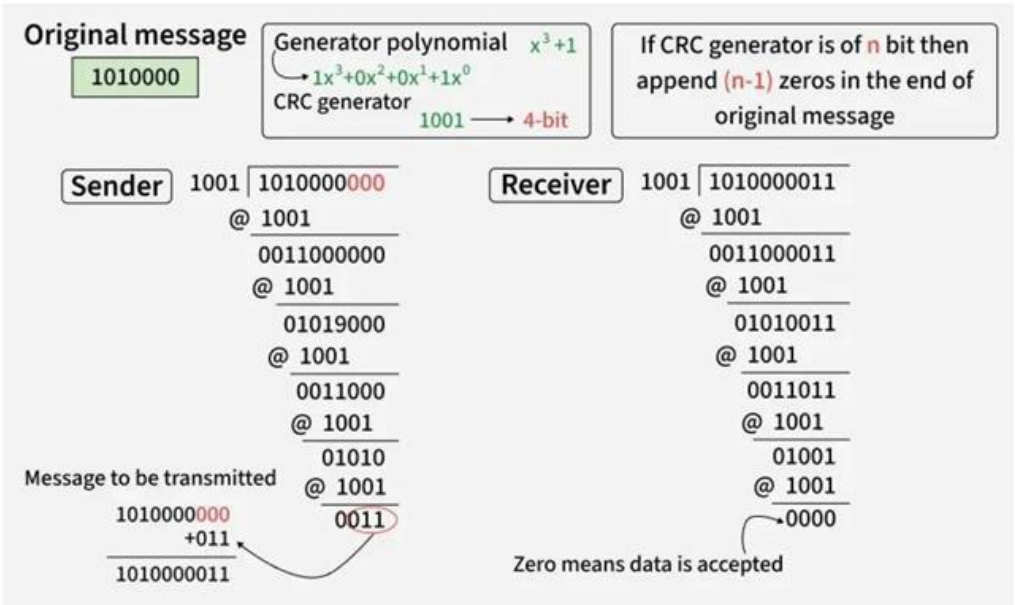
Step 3: Remainder of division = CRC

Step 4: Code word = Data with append k-1 zero's + CRC

Note:

- CRC must be k-1 bits
- Length of Code word = n+k-1 bits

Example: Let's data to be send is 1010000 and divisor in the form of polynomial is x3+1. CRC method discussed below.




**4. Checksums with Hash Functions:** The advanced checksum methods implement hash functions that are of cryptographic nature, such as SHA-256, to be the ultimate guards of data integrity, especially those that are in secure communications.

# Error Correction Methods in Computer Networks

Error correction in computer networks is mainly achieved through the pivotal methods that describe the manner in which errors that have been detected are handled and corrected. These primary techniques are Automatic Repeat Request (ARQ), Forward Error Correction (FEC), and Hybrid Schemes. In essence, these methods employ different sets of error-correcting codes and protocols to ensure data integrity, even in situations affected by noise or inherently unreliable conditions.

## 1. Automatic Repeat Request (ARQ)

ARQ is a feedback-driven error correction technique, whereby the receiver is allowed to request data resending in the event of errors being found. The process is facilitated with the help of error-detection codes, acknowledgements (ACKs), negative acknowledgements (NAKs), and timeouts, which collectively form a reliable communication system. Examples of ARQ protocols are:

- **Stop-and-Wait ARQ:** The sender transmits one frame at a time and waits for an acknowledgement before sending the next. If an error is detected or no acknowledgement is received, the frame is resent.
- **Go-Back-N ARQ:** The sender can transmit several frames before needing an acknowledgement. If an error is detected in a frame, all subsequent frames are resent from the erroneous one.
- **Selective Repeat ARQ:** Only the specific frames that were received with errors are retransmitted, improving efficiency over Go-Back-N.

ARQ is widely used in protocols where retransmission is feasible, such as TCP/IP networking.

## 2. Forward Error Correction (FEC)

The FEC achieves this by adding redundancy to the data transmitted through the use of error-correcting codes. In this way, the receiver can detect and correct errors on its own without having to request the data again. This is an indispensable operation in real-time or high-latency situations where retransmission cannot be afforded, such as in cases involving voice over IP, video streaming, or satellite broadcasting.

Key FEC code types include:

- **Lock Codes:** In these codes, the data is chopped into blocks of a specific size, and additional bits are added for error correction. These examples comprise the Hamming

code and the Reed–Solomon code. Block codes are powerful in correcting random errors and are used in devices for storage and protocols for communication.

- **Convolutional Codes:** They examine the data one bit at a time, and, besides, the parity information for the current bits is also generated from the past ones. Most of the time, the Viterbi algorithm is used for decoding the convolutional codes, which are generally employed in the communications of the wireless and deep-space sectors.
- **Repetition Codes:** The simplest form, where each bit is transmitted multiple times. While easy to implement, repetitive codes are inefficient and mainly used in scenarios where simplicity is critical.
- **Burst Error-Correcting Codes:** Specialised codes designed to correct errors that occur in clusters (bursts), rather than isolated single bits.

FEC is also fundamental to **ECC memory** (Error-Correcting Code memory), which utilises error-correcting codes (often Hamming or Reed–Solomon) to detect and correct single-bit errors in RAM modules, thereby ensuring reliability in mission-critical systems.

### 3. Hybrid Automatic Repeat Request (Hybrid ARQ)

Hybrid ARQ combines the strengths of ARQ and FEC, offering a more robust solution. Data is transmitted with FEC parity information, and if the receiver cannot correct all errors, it requests retransmission using ARQ. There are two main approaches:

- Continuously transmit with FEC; request retransmission only if FEC fails.
- Initially transmit without FEC; send additional parity information upon request.

Hybrid ARQ is particularly useful in wireless and high-speed communication systems, where it strikes a balance between efficiency and reliability.

**Key Takeaways So Far**

- Backward correction is most effective for reliable, low-latency networks.
- Forward correction is crucial when retransmission isn't possible.
- Hybrid systems combine modern IoT and mobile communication.

## Error Correction Techniques

Following are the error correction techniques of computer networks:

### 1. Single-bit Error Detection

One extra bit can detect the errors but not correct them.

### 2. Hamming Code

R.W. Hamming invented it, it detects and corrects single-bit errors by adding redundant bits.

Hamming code places parity bits in positions that are powers of two: 1, 2, 4, 8 and so on. Data bits occupy the remaining positions. Each parity bit checks a selected set of positions. At the receiver, failed parity checks form a binary syndrome whose value identifies the erroneous bit position. The receiver flips that bit to correct the error.

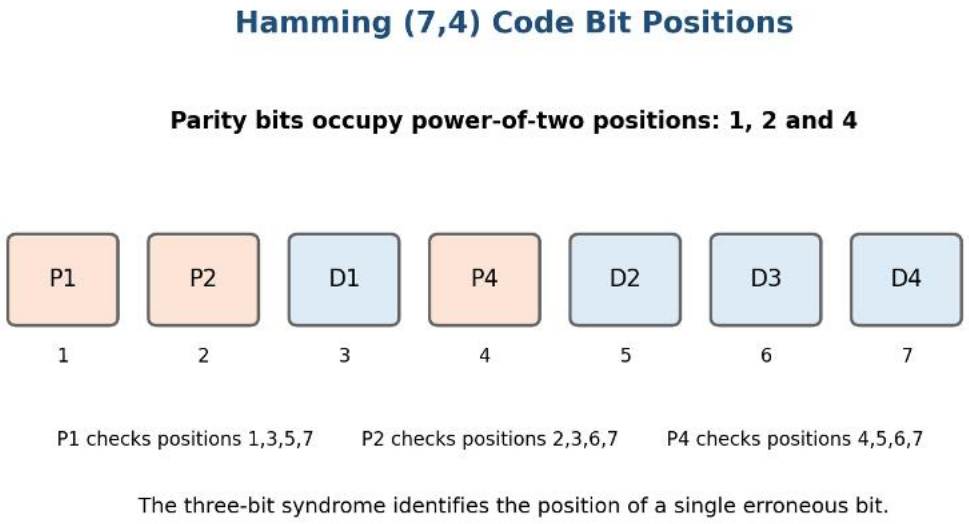



***Figure: Bit arrangement in Hamming (7,4) code***

For m data bits, the required number r of parity bits must satisfy $2^r \geq m + r + 1$. For four data bits, three parity bits are sufficient because $2^3 = 8$ and $4 + 3 + 1 = 8$.

### 3. Parity Bits

Parity bits are appended to binary data such that the total count of 1s is even or odd.

**Even Parity**

- When the total count of 1s is even, the parity bit is 0.
- When the total count of 1s is odd, the parity bit is 1.

**Odd Parity**

- When the total count of 1s is even, the parity bit is 1.
- If the count of 1s is odd, then the parity bit would be 0.

## Practical Implementation of Error Detection and Correction

One of the initial steps in introducing error detection and correction techniques into real-world computer networks is to create a plan and perform calculations. The error detection and correction techniques are basically made practical through the usage of specific methods, which are mentioned below, along with some considerations about their implementation:

### Single Parity Check

A single parity check is just a straightforward error detection method. Each data unit receives a parity bit, which ensures that the entire data unit has either even parity (an even number of

1s) or odd parity (an odd number of 1s). For example, if even parity is chosen and the data is found to contain an odd number of 1s, the parity bit is set to 1, ensuring the whole data has an even number of 1s. The parity is also calculated at the receiver's side. If it does not match, an error is detected. This method, although simple, is still unable to detect all types of errors, particularly those in which two bits have changed their values.

## Two-Dimensional Parity Check

A two-dimensional parity check builds upon the single parity check and arranges the data in a matrix. In this new concept, parity check bits are computed not only for each row but also for each column, thus giving a more powerful error detection feature. For example, if both the row and column parity checks fail, the intersection identifies the bit that causes the error. Nevertheless, this method still requires more redundant bits and has some weaknesses, such as failing to detect specific multi-bit errors.

## Hamming Code and Redundant Bits

The Hamming code is one of the robust error correction methods that can double the detection and correction of single-bit errors. To achieve this, it progressively inserts several redundant bits (parity-check bits) into the original data. The bits that are placed at the positions related to powers of two (e.g., positions 1, 2, 4, 8, etc.) are just like r1 bit, r2 bit, r4 bit, and so on.

Each parity check bit (r1, r2, r4, etc.) covers specific positions in the data. For example:

- The r1 bit covers all positions whose binary representation has a 1 in the least significant bit.
- The r2 bit covers positions with a 1 in the second least significant bit.
- The r4 bit covers positions with a 1 in the third least significant bit.

During transmission, these parity bits are calculated using even parity or odd parity rules. At the receiver's end, the same calculations are performed. If an error is detected, the combination of failed parity bits pinpoints the exact location of the erroneous bit, which can then be corrected.

## Comparison of Error Detection and Correction

| Error Detection | Error Correction |
|---|---|
| The error detection operation is to determine the existence of errors | The error correction operation is to correct errors without retransmission |
| It is often more efficient (less overhead) | This can be more overhead and complicated |
| t is easier to implement | It is more complex as a result of other coding schemes |
| It has lower latency (only requires checking) | It contains higher latency (requires decoding and correction) |
| The error detection is applied in networking (e.g., TCP, UDP) | The error correction is applied in storage systems, error-prone environments (e.g., CDs, DVDs) |
| Examples of Error detection are Parity Check, CRC, Checksum | Examples of Error correction are Hamming Code, Reed-Solomon, Turbo Codes |
| This cannot fix errors, only detects them | It is limited to specific types and numbers of errors |
| It ensures data integrity during transmission | It ensures reliable data retrieval and storage |

## Advantages of Error Detection

- Less complex to implement with lower computational requirements.
- Less time-consuming since it processes faster as it only identifies the errors but does not correct them.
- Generally needs less extra data than error correction techniques.
- Can readily identify errors if there is data transmission.

## Disadvantages of Error Detection

- Detects errors but not correct them, requiring retransmission.
- Does not detect all types of errors, particularly when errors are compound.
- Deps on the assumption that retransmission will correct problems.

## Advantages of Error Correction

- They not only correct errors but also ensure that the data is more reliable and has integrity.
- They use less retransmission, which is favorable for bandwidth-constrained environments.

- Provide more error resiliancy especially where the channel is noisy.

## Disadvantages of Error Correction

- More complicated to implement, using sophisticated algorithms and coding methods.
- Methods the use of additional bits for correction, which will increase overall data size.
- Demands more processing time since decoding and correction of errors are needed.
- Can only correct a limited amount of errors, beyond which data integrity is at risk.

## Applications of Error Detection and Correction

- Internet Communication
- Deep-Space Telecommunications
- Satellite Broadcasting
- Data Storage
- Error-Correcting Memory (ECC Memory)
- Wireless and High-Speed Communication

## Conclusion

To sum up, error detection and correction in computer networks are the backbone of reliable computer networks. Knowledge of different error types and the methods to address them enables network designers to create systems that can maintain data integrity even when data loss occurs. The use of these methods is projected to continue growing in tandem with technological progress, ensuring that data transmission remains safe and efficient.

# Elementary data link protocols

# Simplex Protocol

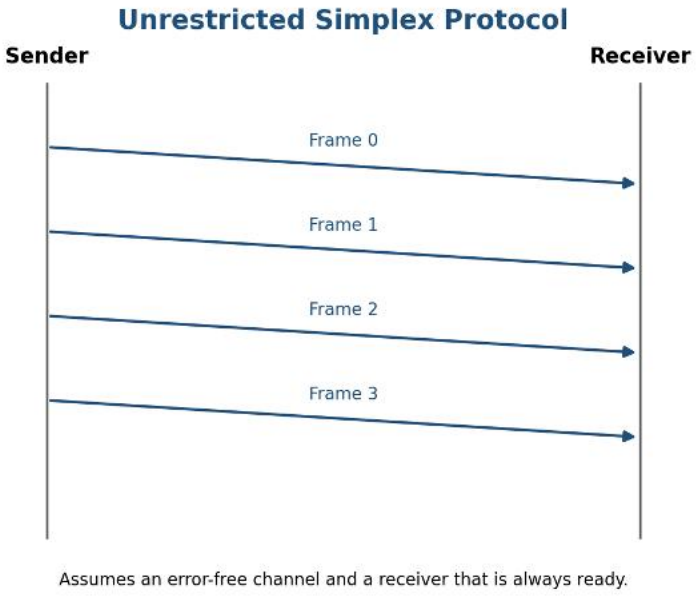



*Figure: Frame transfer in an unrestricted simplex protocol*

## Assumptions

- Communication is simplex: data travels only from sender to receiver.
- The communication channel never loses, duplicates or damages a frame.
- The receiver is always ready and can process incoming frames infinitely fast.
- The sender always has data available.

No flow control or error control is required

## Sender operation

- Obtain a packet from the Network Layer.
- Place the packet in the information field of a frame.
- Send the frame to the Physical Layer.
- Immediately obtain the next packet and repeat forever.

## Receiver operation

- Wait for a frame to arrive from the Physical Layer.
- Extract the packet from the received frame.
- Deliver the packet to the Network Layer.
- Return to the waiting state.

## Simplified pseudocode

**Sender:**

While true → from_network_layer(packet) → frame.info = packet → to_physical_layer(frame).

**Receiver:**

while true → wait_for_event(frame_arrival) → from_physical_layer(frame) → to_network_layer(frame.info).

## Example

Consider a sensor connected to an ideal internal bus where the receiving controller always consumes data immediately and the link is assumed error-free. The sensor can send frames continuously without acknowledgements or waiting.

## Advantages

- Extremely simple and has the smallest protocol overhead.
- Maximum possible throughput on an ideal channel because no acknowledgements or waiting are used.
- Useful as a conceptual starting point for understanding more realistic protocols.

## Limitations

- Unrealistic assumption that the receiver has unlimited processing speed and buffer space.
- Cannot recover from damaged, lost or duplicate frames.
- Provides no acknowledgement, sequence numbering or flow control.
- Not suitable for real communication networks.

**Conclusion**The unrestricted simplex protocol is mainly a theoretical baseline. It shows the minimum actions required to transfer frames and makes clear why practical protocols need flow control and error control.

# Stop-and-Wait Protocol for an Error-Free Channel

**Need for flow control**

Even when the channel is error-free, the sender may generate frames faster than the receiver can process them. If frames arrive continuously, the receiver's buffers may overflow. Stop-and-wait solves this problem by allowing only one unacknowledged frame at a time.

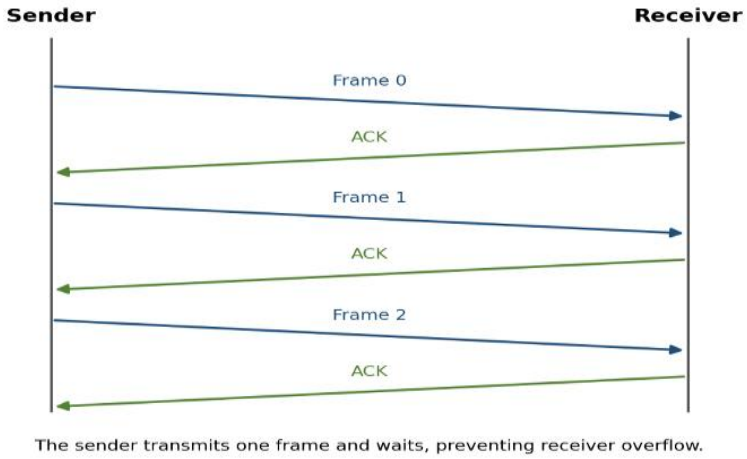



***Figure: Operation of stop-and-wait on an error-free channel***

## Working

The sender obtains one packet, builds a frame and transmits it.

The sender stops and waits for an acknowledgement or permission frame from the receiver.

The receiver accepts the frame, passes its packet upward and sends an acknowledgement.

After receiving the acknowledgement, the sender transmits the next frame.

## Why it works without sequence numbers

The assumed channel does not lose or duplicate frames or acknowledgements. Therefore, every acknowledgement corresponds to the single outstanding frame, and sequence numbers are unnecessary. The acknowledgement mainly acts as a readiness signal for flow control.

## Efficiency

If Tt is the frame transmission time and Tp is the one-way propagation time, let a = Tp/Tt. Ignoring acknowledgement transmission time, utilization is approximately $\eta = 1/(1 + 2a)$. The protocol is efficient on short, slow links but wastes capacity on long-delay or high-speed links because the sender remains idle while waiting.

## Example

Suppose a frame takes 1 ms to transmit and one-way propagation delay is 5 ms. Then a = 5 and utilization is about 1/(1 + 10) = 9.09%. Although reliable on the assumed ideal channel, most of the link remains idle.

## Advantages

- Simple to understand and implement.
- Prevents a fast sender from overflowing a slow receiver.
- Needs only one frame buffer at the sender and receiver.

## Limitations

- Low channel utilization on links with a large bandwidth-delay product.
- The sender cannot transmit a second frame while the first is in transit.
- The basic error-free version cannot handle lost or damaged frames or acknowledgements.

## Conclusion

Stop-and-wait introduces effective flow control by synchronizing every transmission with an acknowledgement. Its simplicity is valuable, but its one-frame window makes it inefficient on long or fast links and motivates sliding-window protocols.

# Stop-and-Wait Protocol for a Noisy Channel

A noisy channel may damage or lose data frames and acknowledgements. A practical stop-and-wait protocol therefore adds error detection, acknowledgements, a retransmission timer and a one-bit sequence number. This reliable method is called Stop-and-Wait ARQ, where ARQ means Automatic Repeat reQuest.

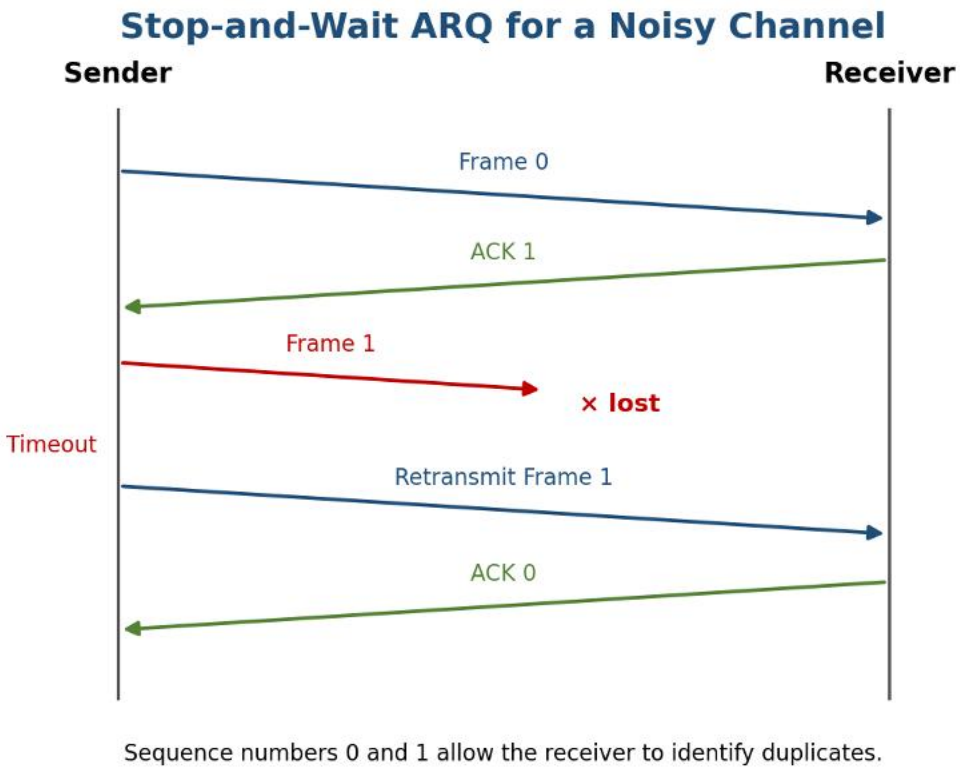



***Figure: Stop-and-Wait ARQ on a noisy channel***

## Sender operation

- Assign sequence number 0 or 1 to the new frame.
- Transmit the frame and start a timer.
- If the correct acknowledgement arrives before timeout, stop the timer, toggle the sequence number and send the next packet.

- If the timer expires or a negative acknowledgement is received, retransmit the same frame with the same sequence number.

## Receiver operation

- Check the frame using its error-detecting code.
- If the frame is damaged, discard it and optionally send a negative acknowledgement.
- If the frame has the expected sequence number, deliver its packet and toggle the expected number.
- If the frame is a duplicate, discard its data but repeat the acknowledgement so that the sender can continue.

## Why one-bit sequence numbering is sufficient

Only one frame can be outstanding, so the receiver needs to distinguish the current frame from a retransmission of the immediately previous frame. Alternating sequence numbers 0 and 1 are therefore sufficient.

## Lost acknowledgement example

The receiver correctly accepts Frame 0 and sends ACK 1, but the acknowledgement is lost. The sender's timer expires and it retransmits Frame 0. The receiver is now expecting Frame 1, recognizes Frame 0 as a duplicate, does not deliver it again, and repeats ACK 1.

## Advantages

- Provides reliable ordered delivery on a noisy point-to-point link.
- Uses very small sequence-number space and limited buffering.
- Duplicate detection prevents repeated delivery to the Network Layer.

## Limitations

- Poor utilization on high-delay links because only one frame is outstanding.
- Timer value must be chosen carefully; too short causes unnecessary retransmission and too long delays recovery.
- Retransmitting an entire frame may be costly when frames are large.

## Conclusion

Stop-and-Wait ARQ extends basic flow control into reliable communication. Sequence numbers, acknowledgements and timeouts solve loss and duplication, but the one-frame window still limits performance.

# Sliding-Window Protocols

# One-Bit Sliding Window

**Sliding-window concept**

A sliding-window protocol permits several frames to be in transit before acknowledgements arrive. Frames carry sequence numbers from a finite sequence-number space. The sender window contains sequence numbers that may be transmitted but are not yet acknowledged, while the receiver window contains sequence numbers that it is currently prepared to accept.

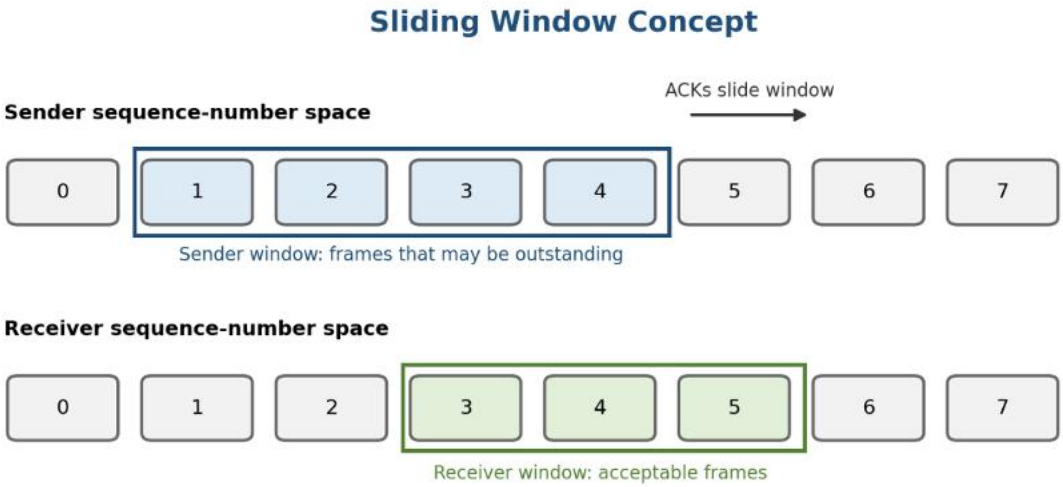



***Figure: Sender and receiver windows in a sliding-window protocol***

## Important terms

**Sender window:** The set of frames that may be outstanding. As acknowledgements arrive, the lower edge moves forward and new sequence numbers become usable.

**Receiver window:** The set of sequence numbers acceptable at the receiver.

**Cumulative acknowledgement:** An acknowledgement may confirm all frames up to a particular sequence number.

**Piggybacking:** In full-duplex communication, an acknowledgement is temporarily delayed and inserted into the header of an outgoing data frame, reducing separate ACK traffic.

# One-bit sliding-window protocol

The one-bit sliding-window protocol uses sequence numbers 0 and 1 and a window size of one at each station. It is essentially a full-duplex form of stop-and-wait. Both stations may send data, and each frame carries a sequence number and an acknowledgement for the opposite direction.

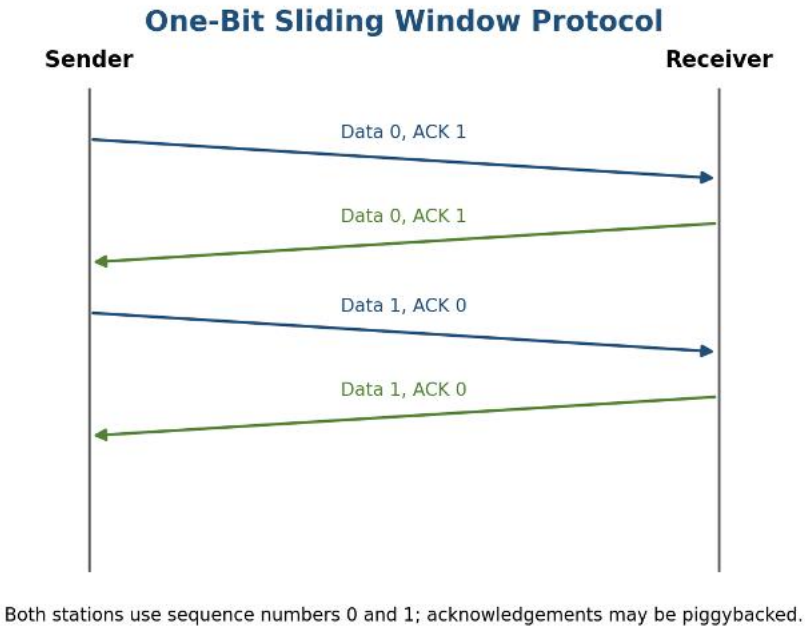



***Figure: Typical exchange in a one-bit sliding-window protocol***

# Working

- Station A sends a frame with sequence number 0 and includes the acknowledgement expected by Station B.
- Station A waits until it receives a frame or separate ACK that confirms its Frame 0.
- Station B accepts only the expected sequence number, delivers it once and returns an acknowledgement.
- After acknowledgement, the sender toggles its next sequence number between 0 and 1.
- If a frame or acknowledgement is lost, timeout causes retransmission; the receiver identifies duplicates by sequence number.

# Piggybacking considerations

Piggybacking saves bandwidth because one frame carries both data and acknowledgement information. However, an acknowledgement cannot be delayed indefinitely. A short acknowledgement timer is used; if no reverse data is ready before it expires, a separate acknowledgement is sent.

# Advantages

Supports reliable bidirectional communication.

Piggybacking reduces the number of separate control frames.

Sequence numbers prevent duplicate delivery.

The general sliding-window framework can be enlarged to improve utilization.

# Limitations

With a one-frame window, utilization is still similar to stop-and-wait.

Simultaneous initial transmissions may cause duplicate frame exchanges in some implementations, although correctness is preserved.

Piggybacking delay must be controlled to avoid unnecessary sender timeouts.

## Conclusion

Sliding windows combine reliability, flow control and pipelining. The one-bit version introduces the key ideas of numbered frames, windows and piggybacked acknowledgements, while larger windows provide the performance needed on modern links.

# Go-Back-N ARQ

Go-Back-N ARQ is a pipelined sliding-window protocol in which the sender may transmit up to N frames without waiting for individual acknowledgements. The receiver normally has a window size of one and accepts only the next expected frame. If a frame is lost or damaged, later frames are discarded and the sender retransmits the missing frame and all subsequent outstanding frames.

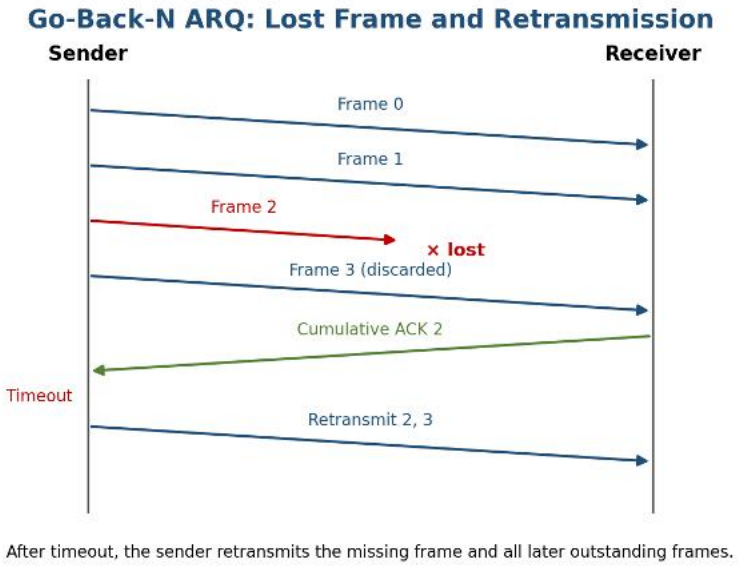



***Figure: Go-Back-N recovery after a lost frame***

## Sender operation

- Maintain variables for the oldest unacknowledged frame and the next sequence number.
- Transmit new frames while the next sequence number is inside the sender window.
- Start or maintain a timer for the oldest outstanding frame.
- Process cumulative acknowledgements and slide the window forward.
- On timeout, retransmit the oldest unacknowledged frame and every later frame already sent.

**Receiver operation**

- Accept a frame only when its sequence number equals the next expected value.
- Deliver the accepted packet and increment the expected sequence number.
- Discard damaged, duplicate and out-of-order frames.
- Return a cumulative acknowledgement indicating the next frame expected or the highest frame received in order.

**Sequence-number and window-size rule**

With an m-bit sequence-number field, the sequence space contains $2^m$ values. The maximum safe sender-window size is $2^m - 1$. Leaving one unused sequence number prevents a delayed old frame from being confused with a new frame after numbering wraps around.

**Example**

Assume a sender window of four frames. Frames 0, 1, 2 and 3 are sent, but Frame 2 is lost. The receiver accepts 0 and 1, discards Frame 3 because it is expecting 2, and repeatedly acknowledges the last in-order position. When the timer expires, the sender retransmits Frames 2 and 3.

**Advantages**

Much better link utilization than stop-and-wait because several frames are in transit.

Receiver design is simple and requires little buffering because out-of-order frames are discarded.

Cumulative acknowledgements reduce control traffic.

Well suited when the error rate is low.

**Disadvantages**

A single error may cause many correctly transmitted later frames to be retransmitted.

Performance becomes poor on noisy links or when the sender window is large.

The sender must buffer every outstanding frame and manage timers and window boundaries.

**Applications**

The idea is used in pipelined reliable links where receiver simplicity and low error rates make cumulative retransmission economical. It also provides the conceptual basis for cumulative acknowledgement mechanisms in more complex protocols.

**Conclusion**

Go-Back-N achieves high utilization through pipelining while keeping the receiver simple. Its main cost is unnecessary retransmission after an isolated loss, which Selective Repeat avoids.

# Selective Repeat ARQ

Selective Repeat ARQ is a sliding-window protocol that retransmits only frames that are actually lost or damaged. The receiver accepts and buffers correct out-of-order frames. When the missing frame arrives, buffered frames are placed in sequence and delivered to the Network Layer.

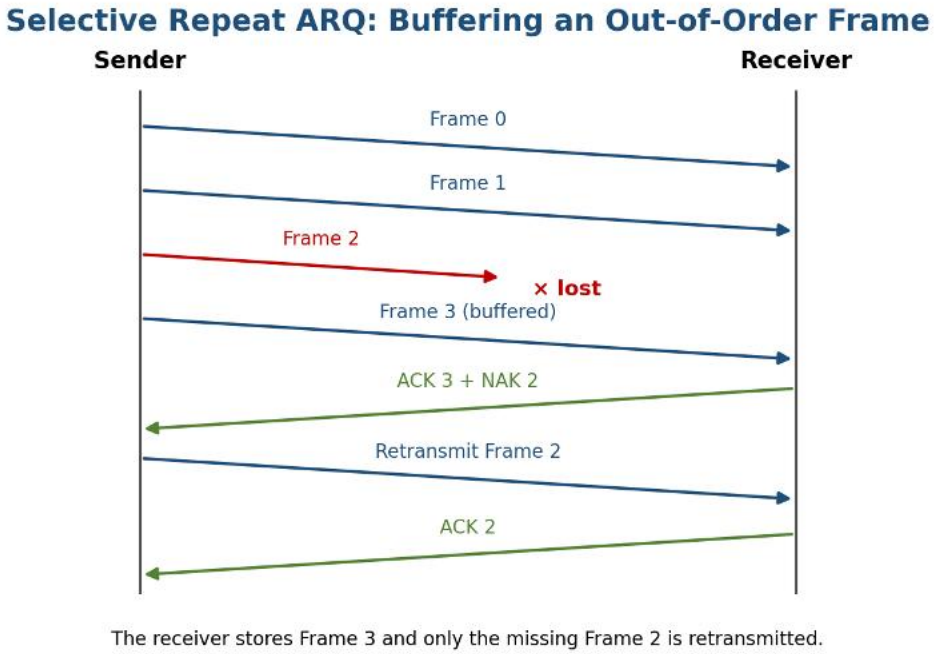



***Figure: Selective Repeat with out-of-order buffering***

## Sender operation

- Transmit any frame whose sequence number lies within the sender window.
- Keep a separate copy and usually a separate timer for each outstanding frame.
- Mark individually acknowledged frames as received.
- Retransmit only a frame whose timer expires or for which a negative acknowledgement is received.
- Slide the window when all frames from its lower edge have been acknowledged.

## Receiver operation

- Accept any correct frame whose sequence number lies within the receiver window.
- Buffer an accepted out-of-order frame and send its individual acknowledgement.
- Discard frames outside the window but repeat acknowledgements when necessary.
- When the lowest missing frame arrives, deliver it and all consecutively buffered frames, then slide the receiver window.

## Window-size rule

For an m-bit sequence-number field, the sender and receiver windows must normally be no larger than $2^{m-1}$. The window is limited to half the sequence space so that a delayed old frame cannot fall inside the current receive window after sequence numbers wrap around.

## Example

Frames 0, 1, 2 and 3 are sent and Frame 2 is lost. The receiver accepts 0 and 1, buffers Frame 3 and acknowledges it, and requests or waits for Frame 2. The sender retransmits only Frame 2. After receiving it, the receiver delivers Frames 2 and 3 in order.

## Go-Back-N and Selective Repeat comparison

| Feature | Go-Back-N | Selective Repeat |
|---|---|---|
| Receiver window | Usually 1 | More than 1 |
| Out-of-order frames | Discarded | Buffered |
| Acknowledgements | Mostly cumulative | Individual/selective |
| Retransmission | Missing frame and all later outstanding frames | Only missing or damaged frames |
| Receiver complexity | Low | Higher |
| Buffer requirement | Low at receiver | Higher at receiver |
| Best environment | Low error rate | Moderate/high error rate or long links |

## Advantages

- Uses bandwidth efficiently when errors occur because correct frames are not resent.
- Suitable for long-delay, high-speed and moderately noisy links.
- Individual acknowledgements provide precise recovery information.

## Disadvantages

- Requires larger receiver buffers and logic for reordering.
- Maintaining a timer for each outstanding frame is more complex.
- Careful window sizing is required to avoid sequence-number ambiguity.

## Conclusion

Selective Repeat provides the most efficient retransmission strategy among the elementary sliding-window protocols, but it trades bandwidth efficiency for additional memory and protocol complexity.

# Example Data Link Protocols:

# Data Link Protocols

Data Link Protocols are communication protocols that operate at the **Data Link Layer (Layer 2)** of the OSI model. They provide **reliable node-to-node communication** by performing framing, flow control, error detection, error correction, and link management. The commonly used Data Link Protocols are:

## 1. SDLC (Synchronous Data Link Control)

**SDLC (Synchronous Data Link Control)** is the first bit-oriented data link protocol developed by **IBM** for synchronous communication between computers and terminals. It provides reliable communication through error detection, flow control, and frame synchronization.

### Features

- Bit-oriented protocol
- Supports synchronous communication
- Error detection using CRC
- Flow control
- Supports point-to-point and multipoint communication

### Advantages

- Reliable data transmission
- Efficient error detection
- High-speed communication

### Applications

- IBM Mainframe Networks
- Banking Systems
- Enterprise Networks

## 2. HDLC (High-Level Data Link Control)

**HDLC (High-Level Data Link Control)** is an ISO standard bit-oriented protocol derived from SDLC. It provides reliable communication over point-to-point and multipoint links.

### Features

- Bit-oriented protocol
- Uses CRC for error detection
- Supports full-duplex and half-duplex communication
- Provides flow control
- Supports sliding window protocol

### Advantages

- Reliable communication
- High efficiency
- Widely used in WANs

### Applications

- WAN communication
- Router-to-router communication

- Leased line networks

# 3. SLIP (Serial Line Internet Protocol)

**SLIP (Serial Line Internet Protocol)** is a simple protocol used to transmit **IP packets** over serial communication links.
It is one of the earliest protocols used for Internet connectivity over dial-up telephone lines.

## Features

- Supports only IP protocol
- Simple implementation
- No authentication
- No error detection

## Advantages

- Easy to implement
- Low overhead

## Limitations

- No authentication
- No dynamic IP address assignment
- Supports only IP

## Applications

- Early dial-up Internet connections
- Serial communication

# 4. PPP (Point-to-Point Protocol)

**PPP (Point-to-Point Protocol)** is a standard protocol used to establish a direct communication link between two devices over a point-to-point connection.
PPP is an improved version of SLIP.

## Features

- Authentication using PAP and CHAP
- Error detection using CRC
- Supports multiple network layer protocols
- Dynamic IP address assignment
- Reliable communication

## Advantages

- Secure communication
- Supports authentication
- Widely used

## Applications

- Internet Service Providers (ISP)
- DSL connections
- VPN connections
- Dial-up Internet

# 5. LCP (Link Control Protocol)

**LCP (Link Control Protocol)** is a component of PPP used to establish, configure, test, maintain, and terminate a point-to-point communication link.

## Functions

- Establishes the communication link
- Configures link parameters

- Tests link quality
- Authenticates users
- Terminates the connection

### Applications

- PPP communication
- WAN links

## 6. LAP (Link Access Procedure)

**LAP (Link Access Procedure)** is a family of data link protocols derived from HDLC and used in X.25 and ISDN networks to provide reliable communication.
Common versions include **LAPB (Link Access Procedure Balanced)** for X.25 and **LAPD (Link Access Procedure for the D-channel)** for ISDN.

### Features

- Error detection and correction
- Flow control
- Reliable frame transmission
- Bit-oriented protocol

### Applications

- X.25 networks
- ISDN communication
- Packet-switched networks

## 7. NCP (Network Control Protocol)

**NCP (Network Control Protocol)** is a component of PPP that configures and enables different **Network Layer protocols** after the link has been established by LCP.
Each network protocol has its own NCP, such as:

- IPCP (Internet Protocol Control Protocol) for IPv4
- IPv6CP for IPv6

### Functions

- Configures network layer parameters
- Assigns IP addresses
- Enables communication for network layer protocols

### Applications

- PPP Internet connections
- Broadband communication
- VPNs

## Comparison of Data Link Protocols

| Protocol | Full Form | Main Purpose | Key Features | Applications |
|---|---|---|---|---|
| **SDLC** | Synchronous Data Link Control | Reliable synchronous communication | Bit-oriented, CRC, flow control | IBM Networks |
| **HDLC** | High-Level Data Link Control | Reliable WAN communication | Bit-oriented, sliding window, CRC | WAN, Routers |
| **SLIP** | Serial Line Internet Protocol | Transfer IP over serial links | Simple, IP only, no authentication | Early Dial-up Internet |
| **PPP** | Point-to-Point Protocol | Point-to-point communication | Authentication, CRC, multi-protocol support | DSL, VPN, ISP |

| Protocol | Full Form | Main Purpose | Key Features | Applications |
|---|---|---|---|---|
| **LCP** | Link Control Protocol | Establish and manage PPP links | Link setup, testing, termination | PPP |
| **LAP** | Link Access Procedure | Reliable communication in X.25/ISDN | Error control, flow control | X.25, ISDN |
| **NCP** | Network Control Protocol | Configure network layer over PPP | IP configuration, protocol negotiation | PPP, Internet |

# Medium Access Sublayer and the Channel Allocation Problem

**Introduction**

On a point-to-point link, only two devices use the medium and access is straightforward. On a broadcast LAN or wireless network, many independent stations share the same channel. If two stations transmit simultaneously, their signals may interfere. The Medium Access Control, or MAC, sublayer coordinates access to this shared medium.

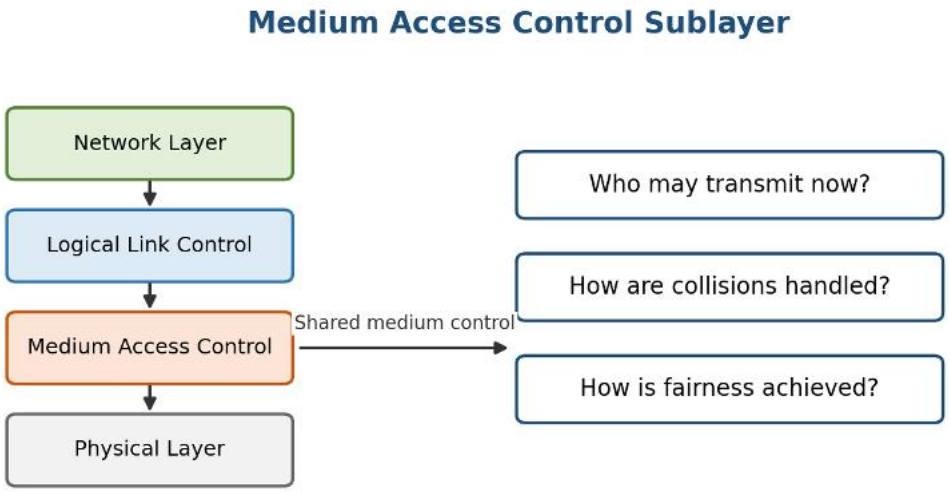



***Figure : Position and responsibilities of the MAC sublayer***

# The channel allocation problem

The channel allocation problem asks how a single communication channel should be divided among competing stations. A good method should provide high utilization, low delay, fairness, stability under heavy load and limited control overhead.

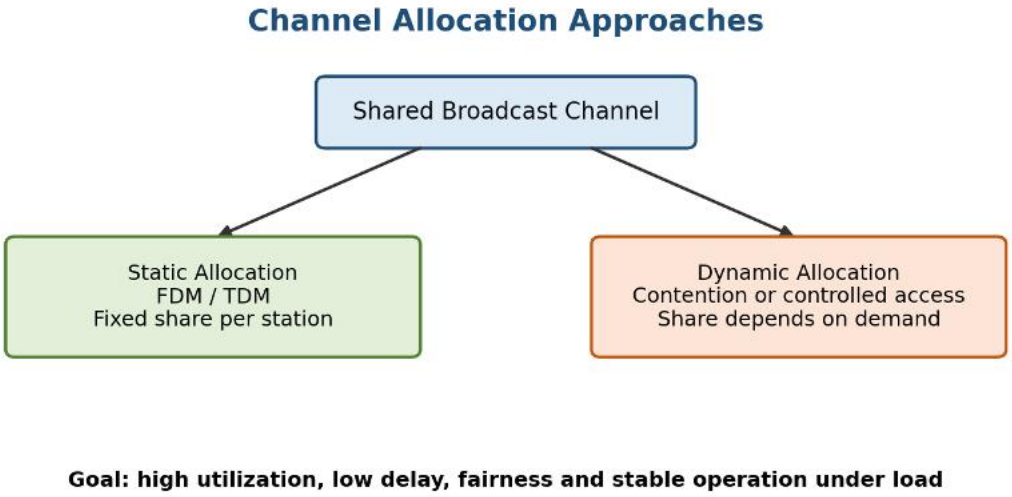



*Figure: Static and dynamic channel allocation*

**1. Static allocation**

The channel is permanently divided into fixed subchannels. Frequency Division Multiplexing assigns a frequency band to each station, and Time Division Multiplexing assigns recurring time slots. Static allocation is predictable and collision-free but wastes capacity when a station has no traffic because its reserved share remains unused.

**2. Dynamic allocation**

The channel is allocated according to current demand. Stations contend for access or follow a controlled reservation/arbitration rule. Dynamic methods use the channel efficiently for bursty computer traffic but must handle collisions, coordination delay or reservation overhead.

**Common assumptions in dynamic channel models**

- Independent stations generate frames unpredictably.
- A single shared channel is available for communication.
- A collision occurs when two frames overlap and both become unusable.
- Time may be continuous or divided into slots.
- Stations may be able to sense the carrier before transmitting or may have no carrier-sense capability.

## Types of multiple-access protocols

**Random access or contention:** Stations compete without a fixed schedule. Examples are ALOHA and CSMA.

**Controlled access or collision-free:** Stations reserve or win permission before data transmission. Examples are bit-map reservation, token passing and binary countdown.

**Channelization:** The resource is separated by frequency, time or code, as in FDMA, TDMA and CDMA.

## Static versus dynamic comparison

| Criterion | Static allocation | Dynamic allocation |
|---|---|---|
| Allocation | Fixed in advance | Based on current demand |

| Criterion | Static allocation | Dynamic allocation |
|---|---|---|
| Collisions | None | Possible in contention methods |
| Bursty traffic | Inefficient | Usually efficient |
| Delay at low load | May wait for assigned slot | Often small |
| Control complexity | Low after setup | Medium to high |
| Examples | FDM, TDM | ALOHA, CSMA, reservation |

## Conclusion

The MAC sublayer turns one shared physical channel into a usable communication service for many stations. Static allocation is simple and predictable, while dynamic allocation better matches bursty network traffic and forms the basis of ALOHA, CSMA and collision-free protocols.

# ALOHA Protocols

ALOHA is a random-access protocol developed for packet radio communication. Stations transmit frames over a shared channel without central scheduling. If frames overlap, a collision occurs and the affected stations wait random times before retransmitting. The two principal versions are Pure ALOHA and Slotted ALOHA.

## A. Pure ALOHA

In Pure ALOHA, a station transmits immediately whenever it has a frame. After transmission it waits for an acknowledgement. If no acknowledgement arrives before timeout, the station assumes a collision, waits a random backoff interval and retransmits.

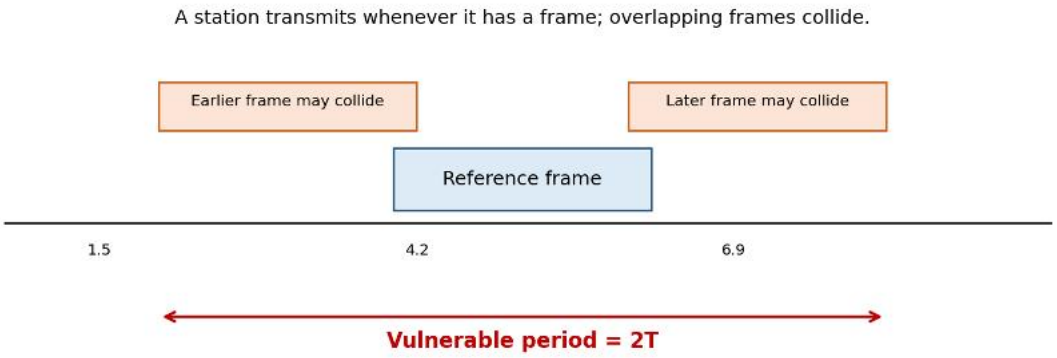



***Figure: Vulnerable period in Pure ALOHA***

Let T be the frame transmission time. A reference frame is destroyed by any other frame that begins within T before its start or within T after its start. Therefore, the vulnerable period is 2T.

If G is the average number of transmission attempts per frame time, the successful throughput is $S = G\, e^{-2G}$. The maximum occurs at G = 0.5, giving Smax = 1/(2e) ≈ 0.184, or 18.4 percent of channel capacity.

## B. Slotted ALOHA

Slotted ALOHA divides time into equal slots of one frame duration. A station may begin transmission only at a slot boundary. Frames either overlap completely within a slot or do not overlap, reducing the vulnerable period to T. Stations must therefore be time-synchronized.

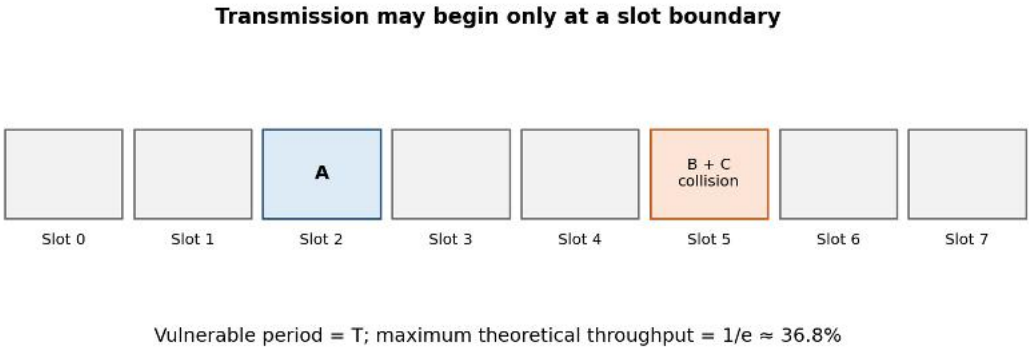



***Figure: Transmission slots in Slotted ALOHA***

The throughput is $S = G\, e^{-G}$. The maximum occurs at G = 1 and equals 1/e ≈ 0.368, or 36.8 percent. Slotted ALOHA therefore doubles the theoretical maximum efficiency of Pure ALOHA.

## Operational steps

- A station creates a frame.
- It transmits immediately in Pure ALOHA or waits for the next slot in Slotted ALOHA.
- If an acknowledgement arrives, transmission is complete.
- If timeout occurs, the station chooses a random backoff time to avoid repeated simultaneous retransmissions.
- The frame is transmitted again after the backoff.

## Comparison

| Feature | Pure ALOHA | Slotted ALOHA |
|---|---|---|
| Transmission start | Any time | Only at slot boundary |
| Synchronization | Not required | Required |
| Vulnerable period | 2T | T |
| Maximum throughput | 18.4% | 36.8% |
| Delay at very low load | Very small | May wait up to one slot |
| Implementation | Simpler | More efficient but synchronized |

## Advantages

- Decentralized and simple; no carrier sensing is required.
- Works in environments such as long-delay radio or satellite channels where sensing may be difficult.
- Performs acceptably when traffic is light and transmissions are infrequent.

## Limitations

- Many collisions and low maximum channel utilization.
- Performance becomes unstable or delay becomes very large under heavy offered load.
- Random retransmission and acknowledgements add delay.
- Slotted ALOHA requires global slot synchronization.

## Conclusion

ALOHA established the principle of random access. Slotted ALOHA improves efficiency by aligning transmissions to slots, but carrier-sense and controlled-access protocols are preferred when higher utilization is required.

# Carrier Sense Multiple Access Protocols

## Basic principle

Carrier Sense Multiple Access improves random access by requiring a station to listen to the channel before transmitting. The rule is often summarized as "listen before talk." If the channel is idle, a station may transmit; if it is busy, the station follows a persistence strategy. Collisions can still occur because two distant stations may both sense an idle channel before either signal reaches the other.

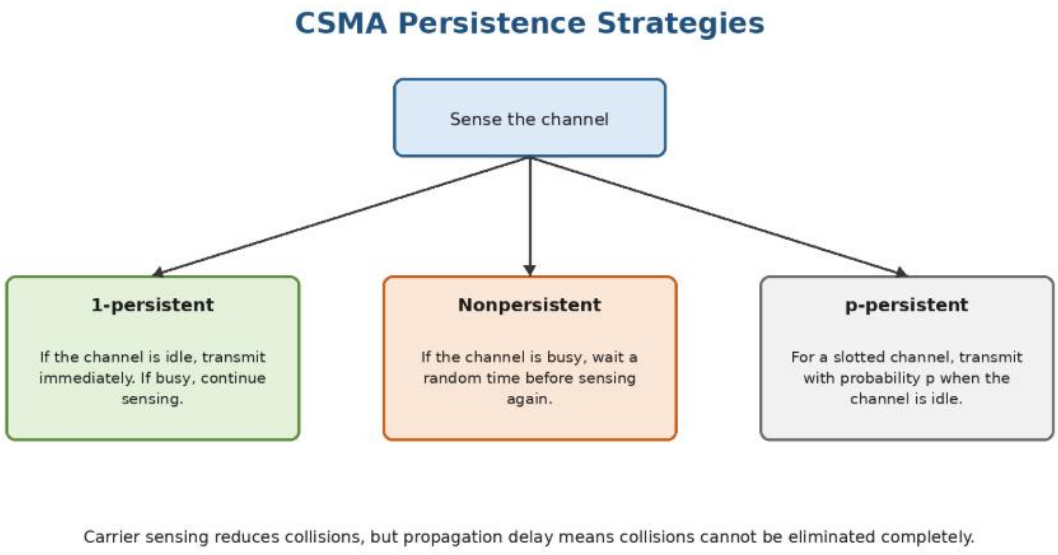



***Figure: Persistence strategies used by CSMA***

# 1. 1-persistent CSMA

A ready station continuously senses the channel. It transmits immediately with probability 1 when the channel becomes idle. If a collision occurs, it waits a random time and retries. This method gives low delay at light load, but several waiting stations may all transmit as soon as the current transmission ends, causing a collision.

# 2. Nonpersistent CSMA

If the channel is busy, the station does not keep sensing continuously. It waits a random interval and then senses again. This reduces the chance that many stations transmit together, improving performance under high load, but it may leave the channel idle unnecessarily and increases delay.

# 3. p-persistent CSMA

This method is used with slotted channels. When the channel is idle, the station transmits in the next slot with probability p and defers one slot with probability $1 - p$. It repeats the decision while the channel remains idle. Selecting p balances collision probability against idle slots.

# 4. CSMA with Collision Detection

CSMA/CD was used by traditional shared, half-duplex Ethernet. A station senses, begins transmission when idle and continues listening while it transmits. If it detects a collision, it immediately stops, sends a jam signal so that every station notices the collision, and waits according to binary exponential backoff before retrying.

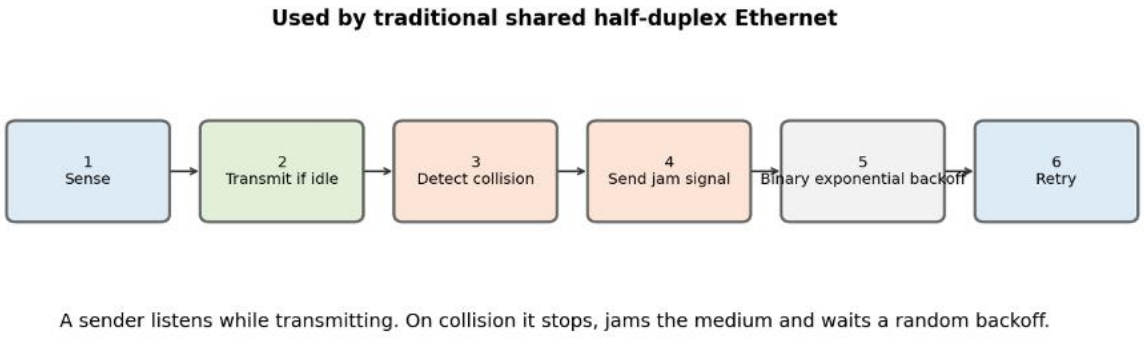



***Figure: Main steps in CSMA/CD***

Binary exponential backoff chooses a random number of slot times from a range that increases after repeated collisions. This spreads retransmissions and stabilizes the network under load. Collision detection requires a minimum frame length so that a sender is still transmitting when a worst-case collision signal returns.

# 5. CSMA with Collision Avoidance

Wireless stations cannot reliably detect collisions while transmitting because their own signal is much stronger than an incoming signal. IEEE 802.11 therefore uses CSMA/CA. A station senses the medium, waits an interframe space, selects a random backoff counter and transmits when the counter reaches zero. Positive acknowledgements confirm delivery. Optional RTS/CTS control frames reduce hidden-station collisions.

## Comparison of CSMA variants

| Protocol | Busy-channel action | Main benefit | Main issue |
|---|---|---|---|
| 1-persistent | Keep sensing; transmit immediately when idle | Low delay at light load | Many waiting stations may collide |
| Nonpersistent | Wait randomly, then sense again | Fewer collisions under load | Extra idle time and delay |
| p-persistent | Transmit with probability p in an idle slot | Adjustable trade-off | Needs slotted channel and suitable p |
| CSMA/CD | Detect and abort collision | Avoids wasting full frame time | Not practical on wireless; obsolete in switched full-duplex Ethernet |
| CSMA/CA | Backoff before transmit; use ACK/RTS/CTS | Designed for wireless | Cannot guarantee collision-free operation |

## Advantages

Much higher utilization than ALOHA because transmissions usually avoid a busy channel.

Distributed operation without a central scheduler.

Persistence and backoff rules adapt to traffic conditions.

## Limitations

Propagation delay allows residual collisions.

Performance degrades as the number of active stations grows.

Hidden and exposed terminal problems complicate wireless carrier sensing.

Collision detection and avoidance require additional timing and control mechanisms.

## Conclusion

CSMA reduces random-access collisions by sensing the carrier. Persistence controls when waiting stations retry, CSMA/CD shortens wired collisions, and CSMA/CA uses backoff and acknowledgements to reduce wireless collisions.

# Collision-Free Protocols

Contention protocols permit simultaneous attempts and therefore may suffer collisions. Collision-free protocols coordinate stations before data transmission so that at most one station uses the channel at a time. They provide predictable operation under heavy load, although reservation or arbitration introduces control overhead.

## 1. Bit-map or reservation protocol

Assume N stations numbered 0 to N − 1. A contention period contains N reservation slots. Station i transmits a 1 in slot i if it has a frame and otherwise leaves 0. After the reservation map is known, every station that reserved transmits one data frame in numerical order. Because the order is known in advance, no data collisions occur.

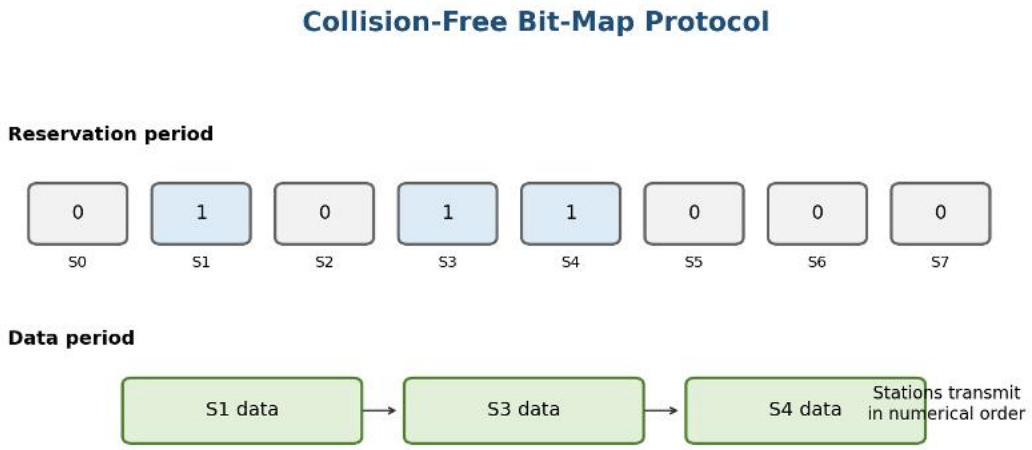



***Figure: Reservation and data periods in the bit-map protocol***

**Advantages:** Collision-free, fair and efficient when many stations are active.

**Limitations:** Each cycle requires N reservation bits even when only a few stations have data, and a station may wait for its reservation slot.

## 2. Binary countdown

Every ready station broadcasts its binary station address from the most significant bit to the least significant bit while monitoring the channel. A 1 is treated as dominant. If a station sends 0 but observes 1, it withdraws. The station with the numerically highest address remains and transmits its frame. Arbitration bits may be considered part of the winning frame, reducing separate reservation overhead.

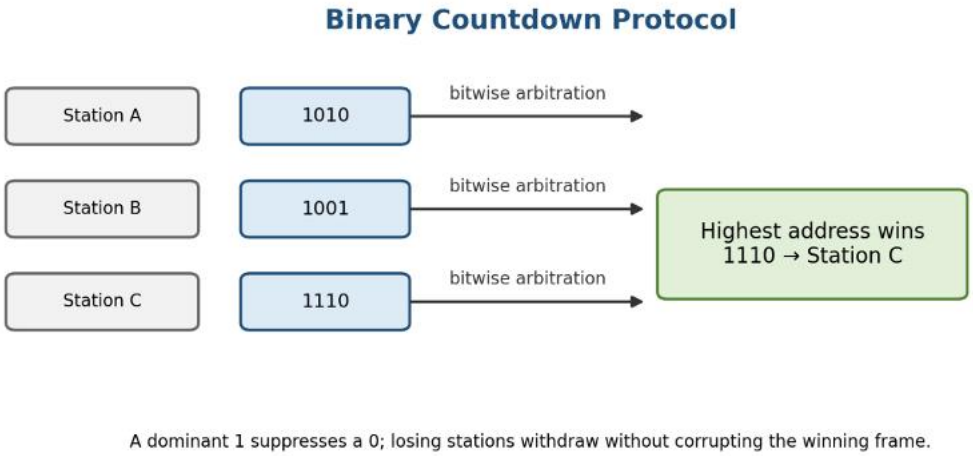



***Figure: Bitwise arbitration in binary countdown***

**Advantages:** Fast distributed arbitration and no destructive collision; efficient when addresses are short.

**Limitations:** Fixed priority may be unfair because high-numbered stations win repeatedly unless priorities are rotated.

## 3. Token passing

A special control frame called a token circulates among stations in a logical order. Only the station holding the token may transmit. After sending its permitted data or when a holding timer expires, it passes the token to the next station. Token Ring and token bus systems are classic examples.

**Advantages:** Collision-free, fair access and bounded waiting time, making it useful for deterministic traffic.

**Limitations:** Token loss, duplication or station failure requires recovery procedures; token circulation adds delay at light load.

## 4. Adaptive tree walk

Stations are organized conceptually as leaves of a binary tree. A control process probes groups of stations. If no station in a group is ready, the group is skipped; if exactly one is ready, it transmits; if a collision indicates several active stations, the group is divided and its subgroups are tested recursively. The method combines contention and reservation and performs well when the active set varies.

## Comparison

| Protocol | Coordination method | Strength | Weakness |
|---|---|---|---|
| Bit-map | One reservation bit per station | Fair and simple | Reservation overhead proportional to N |
| Binary countdown | Bitwise address arbitration | Fast and no destructive collision | Priority bias |
| Token passing | Permission token circulates | Bounded delay and fairness | Token management complexity |
| Adaptive tree | Recursive group splitting | Adapts to number of active stations | More complex control procedure |

## Collision-free versus contention access

At light load, contention methods often provide lower delay because a station can transmit without waiting for a reservation cycle. Under heavy load, repeated collisions waste capacity and collision-free methods become more stable and predictable. Practical systems sometimes combine both approaches.

## Conclusion

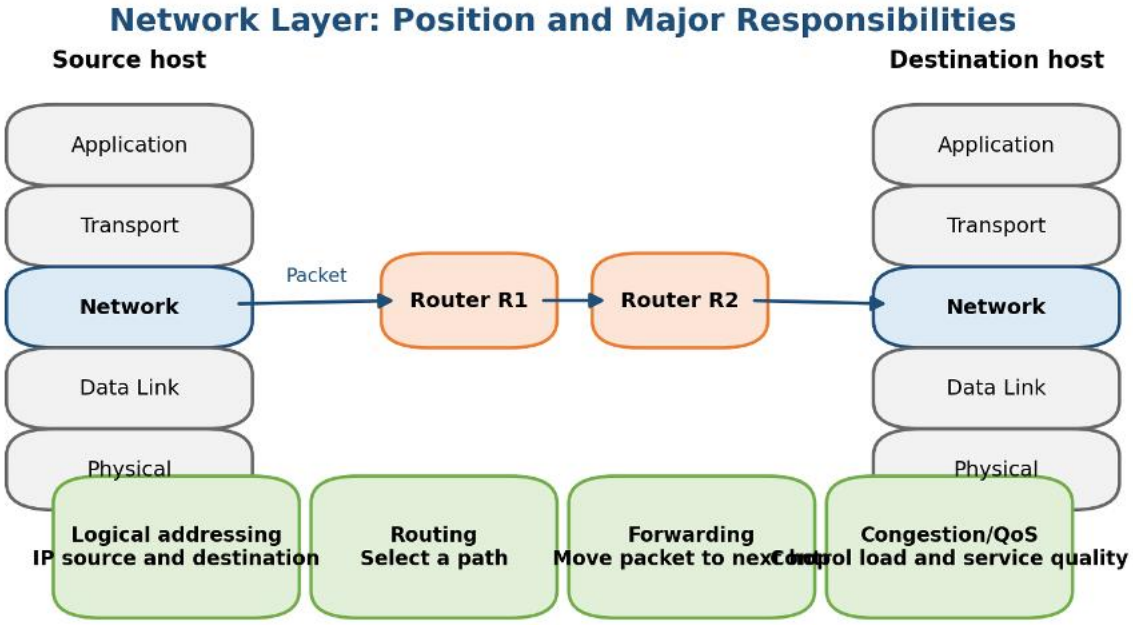
Collision-free protocols exchange control information to prevent destructive simultaneous data transmission. Bit maps provide explicit reservations, binary countdown performs distributed priority arbitration, and tokens provide orderly permission. Their predictable behavior is especially valuable under heavy or time-sensitive traffic.

# UNIT 3: THE NETWORK LAYER

## Network Layer and Design Issues

The Network Layer is the third layer of the OSI reference model. Its central task is to deliver packets from a source host to a destination host, even when the two hosts are connected through many routers and heterogeneous networks. The layer provides logical addressing, selects routes, forwards packets, handles network-wide congestion and supports service quality required by transport-layer applications.

Unlike the Data Link Layer, which transfers frames over one link, the Network Layer considers the complete end-to-end path through an internetwork. A packet may cross Ethernet, optical-fibre, wireless and wide-area links, but the network-layer addressing and forwarding process provides a uniform delivery service.

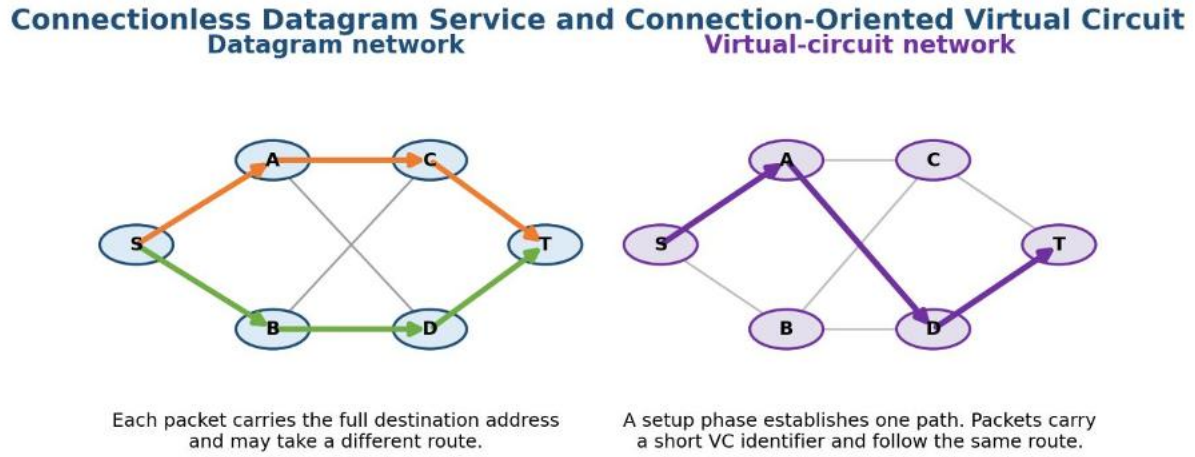



## Major design issues

1. **Service to the Transport Layer:** The service should hide router technology, link types and internal topology. Applications should use a uniform addressing and delivery interface.
2. **Store-and-forward packet switching:** A router normally receives the complete packet, checks the header and places it in an output queue before forwarding it to the next hop.
3. **Logical addressing:** Every source and destination requires a network-layer address. In the Internet this is an IPv4 or IPv6 address.
4. **Routing and forwarding:** Routing computes suitable paths and builds forwarding tables. Forwarding is the local action of moving each arriving packet to the selected output interface.
5. **Connection model:** The designer must choose between a connectionless datagram service and a connection-oriented virtual-circuit service.

6. **Packet size and fragmentation:** Different networks support different maximum frame sizes. The network layer may fragment a large packet or require the source to use path-MTU information.
7. **Congestion and resource allocation:** When offered traffic exceeds link or buffer capacity, queues grow and packets are lost. Congestion-control and QoS mechanisms must allocate resources fairly.
8. **Error reporting and diagnostics:** The network layer reports unreachable destinations, expired hop counts and other delivery problems through control protocols such as ICMP.
9. **Scalability and internetworking:** Routing tables and update traffic must remain manageable as the network grows to millions of destinations and many administrative domains.

## Datagram and virtual-circuit implementation




**Datagram network**: No setup is performed before data transfer. Every packet contains the complete destination address and is routed independently. Routers maintain destination-based forwarding tables but do not store per-flow connection state. Packets may take different routes, arrive out of order, be duplicated or be lost. The Internet Protocol is the best-known datagram service.

**Virtual-circuit network:** A route is established before the first data packet is sent. Every router on that route creates an entry that maps an incoming interface and virtual-circuit identifier to an outgoing interface and new identifier. Data packets carry the short identifier instead of a complete address. All packets normally follow the same route until the virtual circuit is released.

| Feature | Datagram network | Virtual-circuit network |
|---|---|---|
| **Connection setup** | Not required | Required before data transfer |
| **Router state** | No per-connection state | VC table entry at every router |
| **Packet header** | Full destination address | Short VC identifier |

| Feature | Datagram network | Virtual-circuit network |
|---|---|---|
| **Route** | May differ for each packet | Usually fixed for the connection |
| **Failure effect** | Later packets may be rerouted | VC must usually be re-established |
| **Ordering** | Not guaranteed | Normally preserved |
| **Typical example** | Internet Protocol | ATM, Frame Relay, MPLS label-switched path |

## Example

Suppose a student sends a five-packet file from a hostel network to a cloud server. In a datagram network, packets 1 and 2 may travel through Router R1 while packets 3 to 5 use R2 after a routing change. The receiver may obtain them in the order 1, 3, 2, 4, 5, and the transport layer restores the sequence. In a virtual-circuit network, a path is set up first and all five packets carry the same VC identifier along that path.

## Advantages

- Layering hides physical-network differences from applications and the Transport Layer.
- Datagram service is robust because packets can be rerouted after a link or router failure.
- Virtual circuits provide predictable ordering and can reserve resources for a flow.
- Logical addressing allows packets to cross many independent link technologies.

## Limitations

- Datagram delivery is best effort; reliability, sequencing and recovery must be supplied by higher layers when required.
- Virtual circuits add setup delay and per-connection state, and a failed router can break many active circuits.
- Large routing tables, congestion and incompatible MTUs make global packet delivery complex.
- Security problems such as spoofed addresses and denial-of-service traffic require additional controls.

## Applications

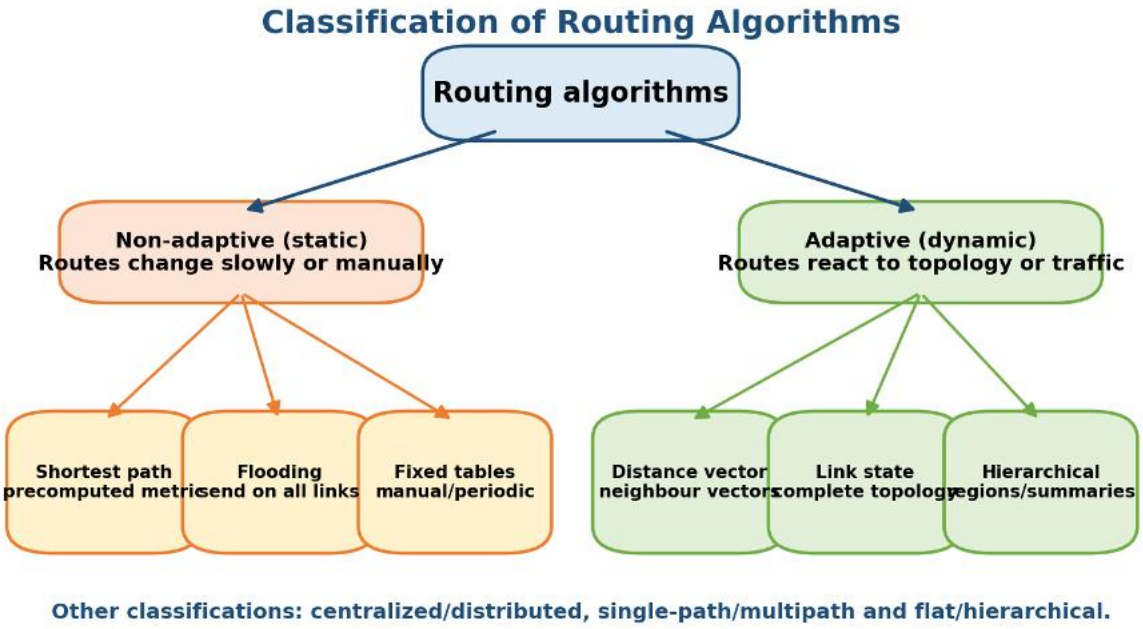
- Internet packet delivery
- Enterprise routing and VPNs
- Data-centre fabrics
- Mobile and wireless core networks
- Cloud interconnection and software-defined WANs

## Conclusion

The Network Layer converts a collection of individual links into a packet-delivery system. Its most important design choices concern the service offered to the Transport Layer, the use of datagrams or virtual circuits, route selection, congestion management, addressing and internetworking.

# Routing Algorithms: Requirements and Classification

A routing algorithm is the procedure used by routers to determine the path or next hop for packets. The algorithm uses a model of the network, routing metrics and information exchanged among routers to build a routing table. A good algorithm must not merely find a route; it must find routes that remain useful as traffic and topology change.




## Properties of a good routing algorithm

- **Correctness:** Every valid destination should be reached without persistent loops.
- **Simplicity:** The control logic and messages should not consume excessive processor, memory or bandwidth.
- **Robustness:** The network should continue operating after failures, overload and incorrect updates.
- **Stability:** After a change, routing information should converge to a consistent state without long oscillations.
- **Fairness:** No user or flow should be permanently denied network resources.
- **Optimality:** Paths should minimize a selected metric such as cost, delay or hop count.
- **Scalability:** Table size and control overhead should grow slowly enough for large networks.
- **Fast convergence:** Routers should discover failures and agree on new routes quickly.

## Routing metrics

A metric converts a link or path into a numerical cost. The selected route may minimize hop count, monetary cost, delay, loss probability or a composite metric. A bandwidth-sensitive protocol may assign low cost to high-capacity links, whereas a delay-sensitive network may measure queueing and transmission delay.

| Metric | Meaning | Possible use |
|---|---|---|
| **Hop count** | Number of routers crossed | Simple small networks |
| **Administrative cost** | Configured preference | Policy and link-type selection |
| **Delay** | Propagation, transmission and queueing time | Interactive traffic |
| **Bandwidth** | Available or nominal link capacity | High-volume traffic |
| **Reliability** | Failure rate or packet loss | Critical services |
| **Load** | Current utilization or queue length | Adaptive traffic distribution |

## Classification

Non-adaptive or static routing uses routes that are computed in advance and change only manually or at long intervals. It is simple and predictable but cannot respond quickly to failures or congestion. Examples are fixed shortest-path tables and static default routes.

Adaptive or dynamic routing changes routes according to current information. Routers exchange updates and recalculate paths after a topology or metric change. Distance-vector, link-state and path-vector protocols are adaptive. Their benefit is responsiveness, but they require control traffic and must prevent loops and instability.

Centralized routing uses a controller with a global network view. Distributed routing lets routers compute paths by exchanging information. Single-path routing selects one next hop, whereas multipath routing divides traffic across several equal or unequal paths. Flat routing treats all routers alike; hierarchical routing organizes them into areas or domains.

## Example

A campus may use a static default route from a small laboratory router to the core. The core routers run a dynamic protocol. If one fibre link fails, the dynamic routers recalculate an alternative path, while the laboratory continues sending all external packets to its unchanged default gateway.

## Advantages of adaptive routing

- Automatically responds to link and router failures.
- Can balance traffic across alternative paths.
- Supports large and changing networks.
- Reduces the amount of manual route maintenance.

## Limitations

- Routing updates consume bandwidth and processing power.
- Slow convergence can create temporary loops or black holes.
- Rapid metric changes may cause route oscillation.
- Poorly selected metrics can move traffic onto a theoretically short but overloaded path.

## Conclusion

Routing algorithms translate network topology and policy into forwarding decisions. Their quality is judged by correctness, stability, convergence, robustness, fairness and path cost. Static routing suits simple stable situations, while adaptive routing is necessary for large networks that change frequently.

# Shortest Path Routing

## Concept

Shortest-path routing represents the network as a weighted graph. Routers are vertices, communication links are edges and each edge has a non-negative cost. The cost may represent distance, delay, hop count, monetary cost or a configured combination. The selected path is the route whose total edge cost is minimum.

Dijkstra's algorithm computes shortest paths from one source to every other vertex when all edge weights are non-negative. It maintains a permanent set of vertices whose shortest distance is known and a tentative distance for every remaining vertex.

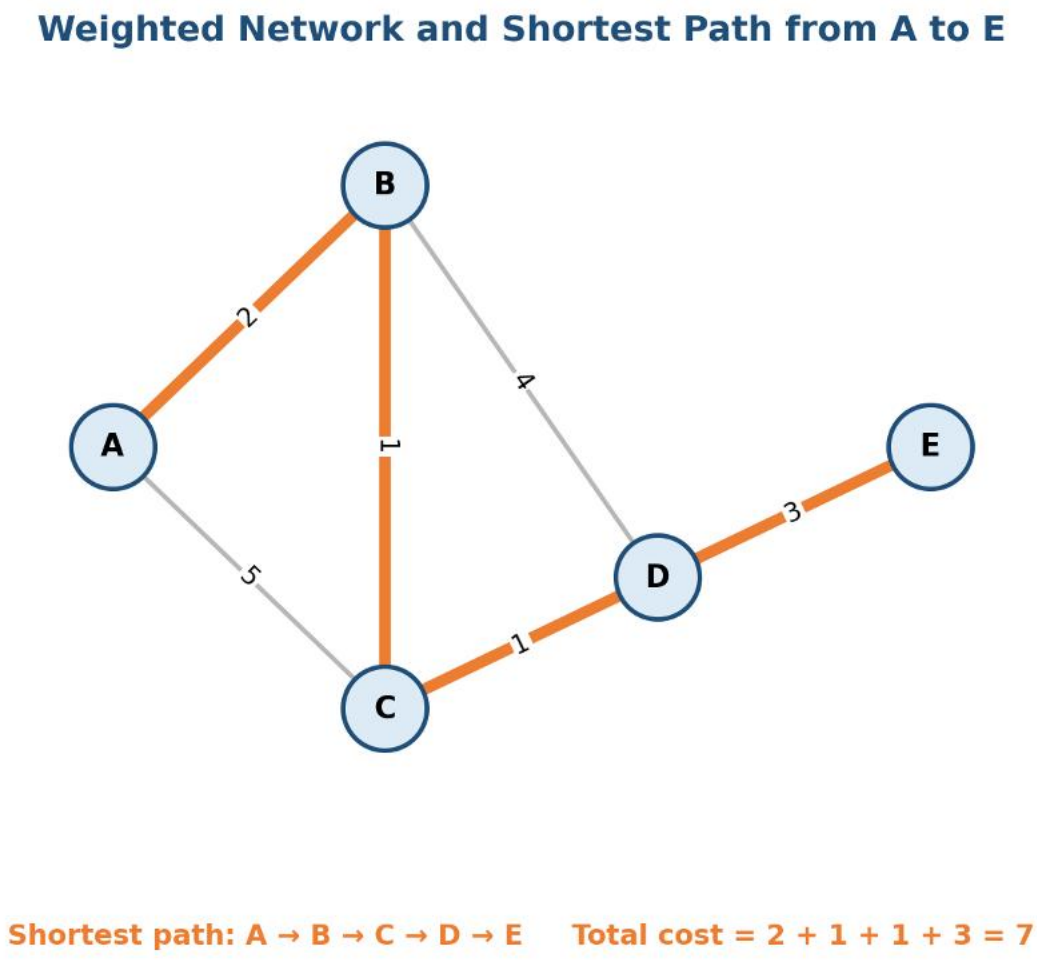



Figure: Weighted network and shortest route from A to E

### Dijkstra algorithm

**1.** Assign distance 0 to the source and infinity to every other vertex. Mark all vertices temporary.

**2.** Select the temporary vertex having the smallest tentative distance and make it permanent.

**3.** For every temporary neighbour v of the selected vertex u, calculate distance(source,u) + cost(u,v). If this value is smaller than the present distance of v, replace the distance and record u as predecessor.

**4.** Repeat selection and relaxation until the destination becomes permanent or all reachable vertices are processed.

**5.** Recover the route by following predecessor entries from the destination back to the source.

### Worked example

For Figure. the source is A. Link costs are A-B=2, A-C=5, B-C=1, B-D=4, C-D=1, C-E=7 and D-E=3.

| Iteration | Permanent set | Tentative distances (predecessor) |
|---|---|---|
| **Initial** | {A} | B=2(A), C=5(A), D=∞, E=∞ |
| **Choose B** | {A,B} | C=min(5,2+1)=3(B); D=6(B); E=∞ |
| **Choose C** | {A,B,C} | D=min(6,3+1)=4(C); E=10(C) |
| **Choose D** | {A,B,C,D} | E=min(10,4+3)=7(D) |
| **Choose E** | {A,B,C,D,E} | Final distance to E = 7 |

Following predecessors gives E←D←C←B←A. Therefore the shortest path is A-B-C-D-E and its total cost is 7.

## Routing-table use

A router runs the algorithm with itself as the source. The first edge on each resulting shortest path becomes the next-hop entry in its forwarding table. Link-state protocols such as OSPF distribute link information and apply Dijkstra's algorithm independently at every router.

## Advantages

- Produces optimal paths for the selected non-negative metric.
- Deterministic and easy to verify.
- A single run finds routes from the source to all destinations.
- Works well with a link-state database and supports equal-cost paths.

## Limitations

- Every router needs an accurate view of the topology and link costs.
- Frequent topology changes can require repeated computation.

- The shortest path for one metric may not satisfy delay, policy or bandwidth requirements.
- Dynamic load-dependent costs can cause oscillation if many routers change routes at the same time.

## Applications

- OSPF and IS-IS interior routing
- Road and railway route planning
- Data-centre traffic engineering
- Shortest-delay paths in communication networks
- Network design and failure analysis

## Conclusion

Shortest-path routing converts route selection into a graph problem. Dijkstra's algorithm repeatedly fixes the closest temporary vertex and relaxes its neighbours, producing an optimal routing tree for all non-negative link costs.

# Flooding

**Definition**

Flooding is a routing method in which a router forwards an incoming packet on every outgoing link except the link on which the packet arrived. No routing table or path calculation is required. Because many copies are created, at least one copy reaches every reachable destination, provided that links are operating.

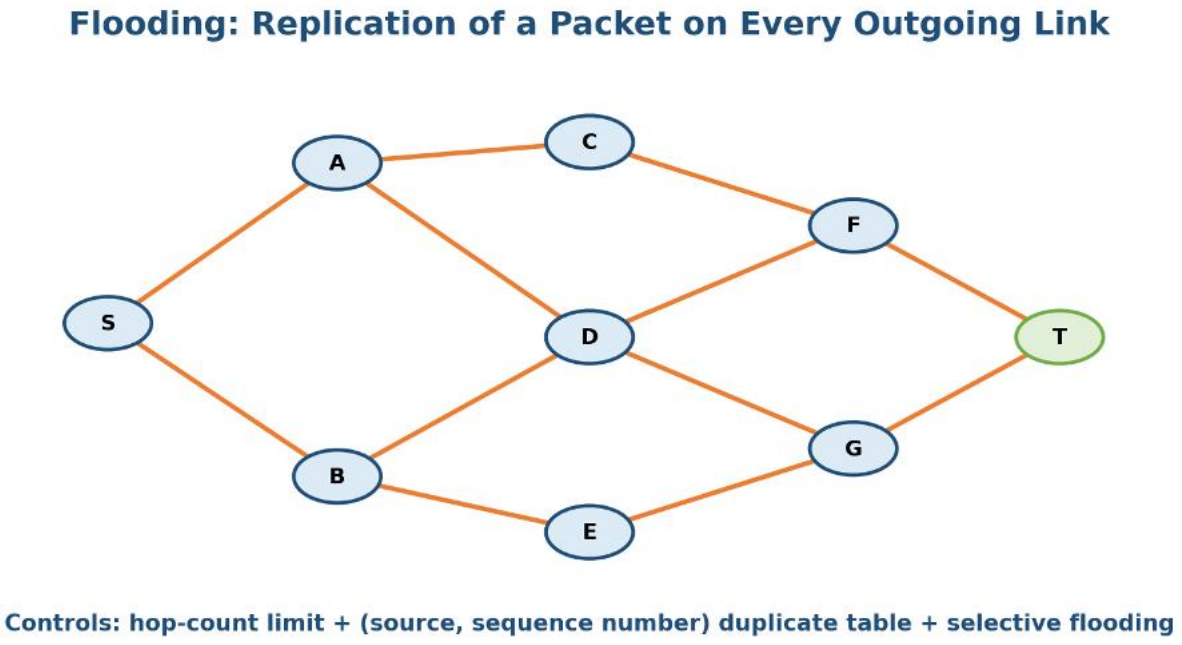



**Figure: Flooding and packet replication through a network**

**Working**

**1.** The source creates a packet containing source address, sequence number and usually a hop-count or time-to-live value.

**2.** The first router makes a separate copy for every eligible outgoing link.

**3.** Each receiving router repeats the process, producing a wave of copies through the topology.

**4.** A destination accepts the first valid copy and discards later duplicates.

**5.** Routers suppress loops and endless replication using controlled-flooding rules.

## Methods for controlling flooding

**Hop-count limit:** The source sets a maximum number of hops. Every router decrements it, and a packet is discarded when the value reaches zero.

**Sequence-number table:** Each router remembers recently seen pairs of source address and sequence number. A repeated packet is discarded instead of being flooded again.

**Selective flooding:** A router forwards only on links that approximately lead toward the destination, rather than on every link.

**Spanning-tree flooding:** Copies travel only on a loop-free spanning tree, so every router receives one copy.

**Reverse-path forwarding:** A packet is forwarded only if it arrived on the interface that the router would use to reach the source.

## Example

In Figure, source S sends one packet. Routers A and B each create copies, and the copies continue through C, D, E, F and G. Destination T may receive one copy through F and another through G. The duplicate table ensures that only the first copy is processed.

## Advantages

- Extremely robust: a copy can use any surviving route.
- No route discovery or routing table is needed.
- The first arriving copy follows a minimum-delay path under equal processing conditions.
- Naturally supports broadcast and route-discovery messages.
- Useful when topology information is incomplete or rapidly changing.

## Limitations

- Creates an enormous number of duplicate packets and wastes bandwidth.
- Consumes router buffers and processing time, possibly causing congestion.
- Without strict controls, packets loop indefinitely.
- Not suitable for ordinary high-volume unicast traffic.
- Duplicate suppression requires storage of recent packet identifiers.

## Applications

- Link-state routing advertisements
- Emergency and military networks
- Wireless ad-hoc route discovery
- Broadcast in small networks
- Distribution where maximum reliability is more important than bandwidth efficiency

## Conclusion

Flooding sacrifices efficiency for robustness and simplicity. Controlled flooding uses hop counts and sequence numbers to prevent infinite copies, while selective, spanning-tree and reverse-path methods reduce unnecessary traffic.

# Hierarchical Routing

**Need for hierarchy**

In a flat network every router may require one entry for every destination or prefix. As the network grows, routing tables, update messages and computation become too large. Hierarchical routing solves this scaling problem by dividing routers into regions, areas, clusters or autonomous systems and summarizing information between them.

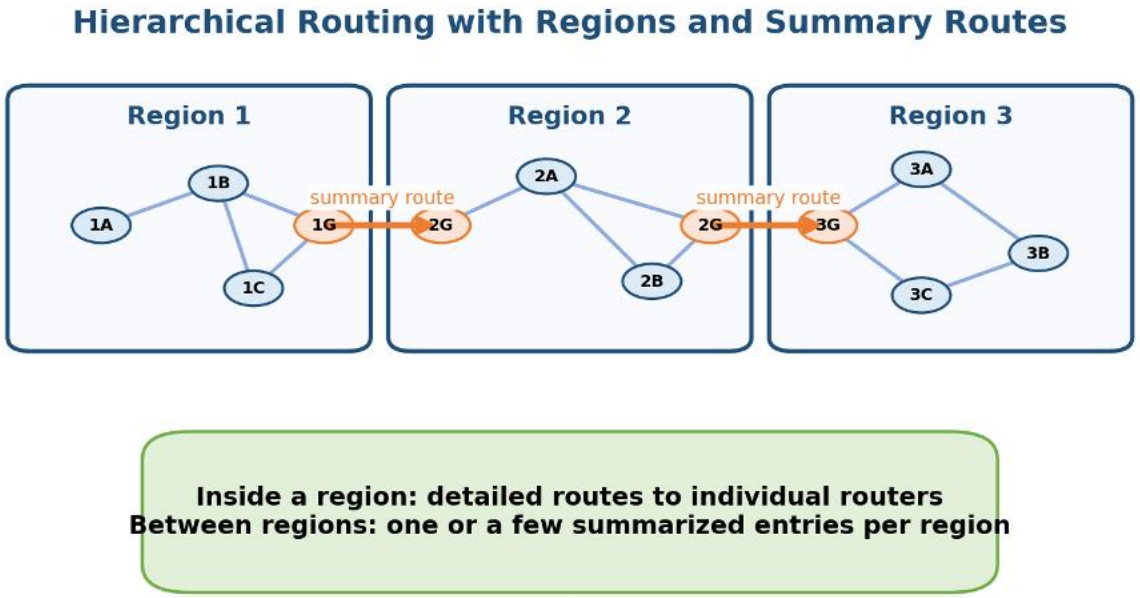



**Figure: Hierarchical routing with regions and summarized inter-region routes**

## Working principle

**1.** Routers are grouped according to geography, administration or topology.

**2.** A router keeps detailed routes for destinations inside its own region.

**3.** For a remote region, it keeps one or a small number of summary entries instead of routes to every remote router.

**4.** Border or gateway routers exchange reachability information between regions.

**5.** A packet first follows an intra-region route to a gateway, then an inter-region route, and finally an intra-region route inside the destination region.

## Table-size example

Assume a network has 1000 routers divided into 10 regions of 100 routers each. A flat scheme may need approximately 999 destination entries at every router. A simple two-level hierarchy may require about 99 entries for routers in the local region plus 9 entries for the other regions,

or roughly 108 entries. The exact value depends on topology and summarization, but the reduction is substantial.

## Route aggregation

Address blocks are assigned so that many destinations share a common prefix. A border router advertises the common prefix rather than every individual subnet. For example, four networks 10.20.0.0/24 to 10.20.3.0/24 can be summarized as 10.20.0.0/22 when they are contiguous and correctly aligned.

## Advantages

- Greatly reduces routing-table size.
- Limits routing-update traffic to the affected area.
- Improves convergence because local failures need not be advertised globally.
- Supports administrative policies and independent management.
- Makes very large internetworks such as the Internet practical.

## Limitations

- Summarized information may produce a path that is not globally shortest.
- Poor address planning prevents efficient aggregation.
- Border routers have greater complexity and can become bottlenecks.
- A failure hidden by a summary can cause temporary black holes.
- Multiple levels increase configuration and troubleshooting difficulty.

## Applications

- OSPF areas and backbone Area 0
- IS-IS levels
- Autonomous systems and BGP on the Internet
- Enterprise campus core-distribution-access design
- Large mobile and service-provider networks

## Conclusion

Hierarchical routing trades a small amount of path optimality for major gains in scalability. Detailed routes remain local, while summary routes represent remote regions, reducing table size, control traffic and convergence scope.

# Broadcast Routing

Broadcast routing delivers one packet from a source to every host or router in a specified network. The objective is complete delivery with minimum duplicate traffic and without persistent loops. It is used for discovery, routing control, service announcements and distribution of common information.

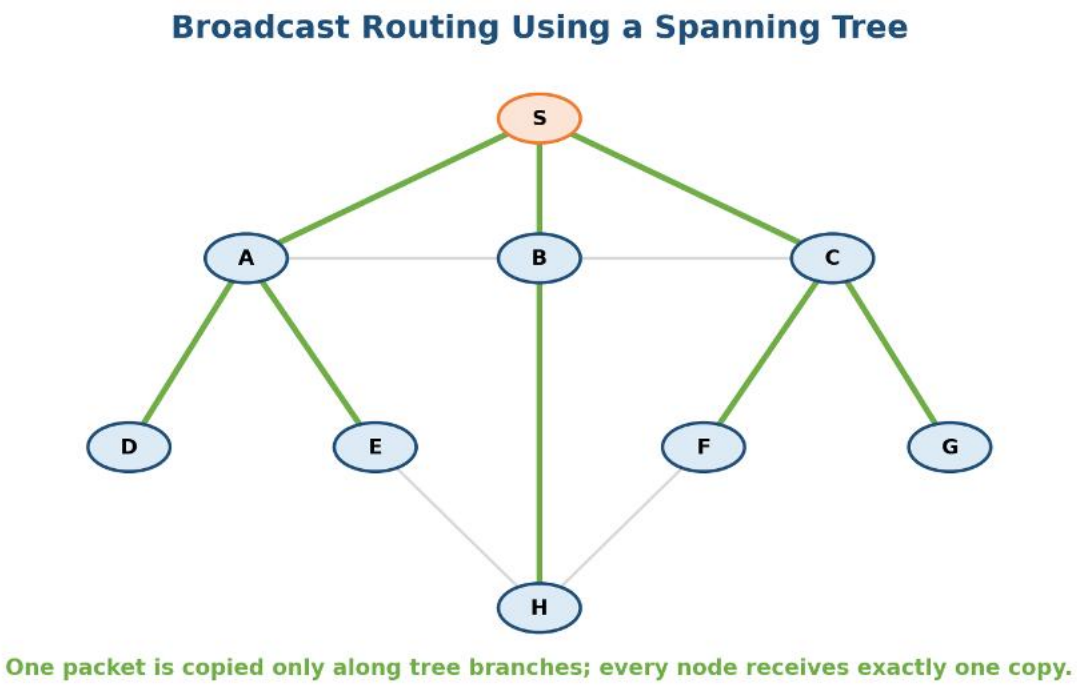



Figure: Broadcast forwarding over a spanning tree

## Broadcast-routing methods

1. **Separate unicast packets:** The source sends one packet to each destination. It is easy to understand but wastes source bandwidth and requires the source to know every destination.
2. **Flooding:** Every router forwards a copy on all links except the incoming link. Delivery is robust but uncontrolled duplicates are expensive.
3. **Multi-destination routing:** One packet carries a list or bitmap of destinations. A router divides the list among output links and creates copies only where paths separate.
4. **Spanning-tree broadcast:** Routers agree on a loop-free tree covering all nodes. A packet received on the tree is copied only to child branches, so each node receives one copy.
5. **Reverse-path forwarding (RPF):** A router accepts a broadcast from source S only when it arrives on the interface that lies on the router's shortest path back to S. Accepted packets are then forwarded on other interfaces.

## Example

In Figure, S is the root of a spanning tree. S sends copies to A, B and C. Router A forwards to D and E, B forwards to H, and C forwards to F and G. Extra physical links are not used for this broadcast; therefore every node receives exactly one copy and no loop is formed.

## Comparison

| Method | Duplicate traffic | Knowledge required | Main limitation |
|---|---|---|---|
| **Separate unicasts** | High near source | All destinations | Poor scalability |
| **Flooding** | Very high | Almost none | Packet explosion |
| **Multi-destination** | Moderate | Destination set and routes | Large header/processing |
| **Spanning tree** | Low | Tree construction | Tree failure/recalculation |
| **RPF** | Low to moderate | Route back to source | May still create some duplicates |

## Advantages

- Efficiently distributes identical information to all nodes.
- Spanning trees and RPF prevent loops without one full unicast per receiver.
- Useful for network discovery and control protocols.
- Can exploit router replication so the source sends only one packet.

## Limitations

- Broadcast traffic consumes resources at hosts that do not need the data.
- Large broadcast domains can suffer broadcast storms.
- Routers normally limit broadcast across administrative boundaries.
- Authentication is necessary because forged broadcasts can disrupt many systems.

## Applications

- ARP requests inside IPv4 LANs
- Routing and topology announcements
- DHCP discovery within a broadcast domain
- Service discovery
- Emergency notifications in controlled networks

## Conclusion

Broadcast routing is a one-to-all delivery problem. Flooding guarantees reachability but creates duplicates, while spanning trees and reverse-path forwarding provide efficient loop-controlled distribution.

# Multicast Routing

Multicast routing delivers a packet to a selected group of receivers rather than to one receiver or every host. A multicast group is identified by a group address, and hosts may join or leave dynamically. Routers replicate a packet only at points where routes to group members separate.

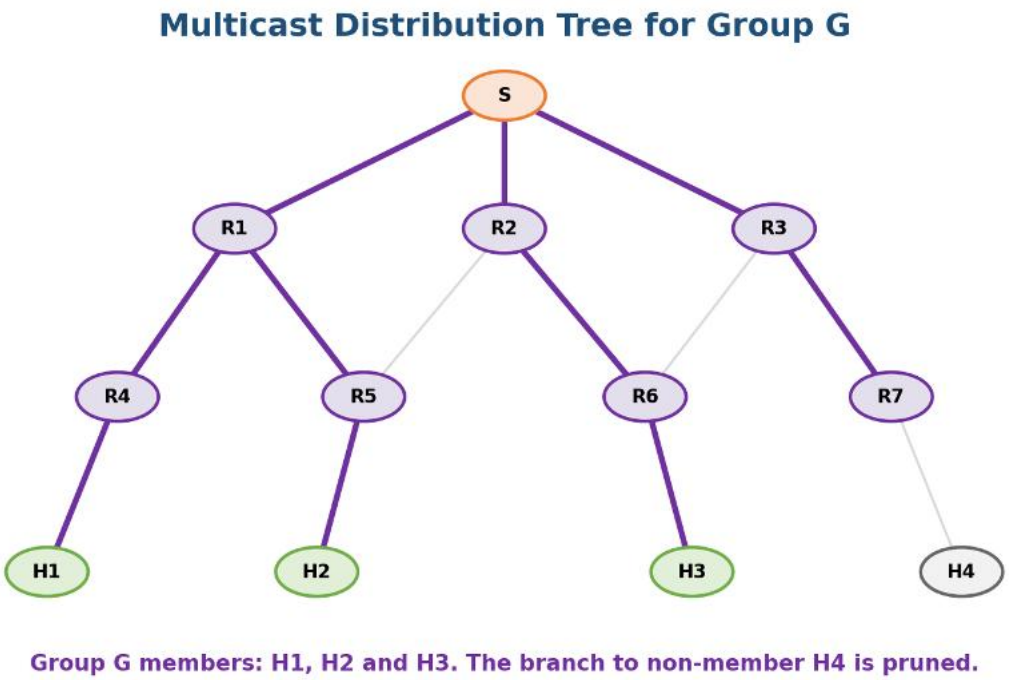



Figure : Multicast distribution tree with a pruned non-member branch

## Main operations

1. **Group membership management:** Hosts inform a local router that they wish to receive traffic for group G. IPv4 commonly uses IGMP, while IPv6 uses Multicast Listener Discovery.
2. **Tree construction:** Routers build a distribution tree that connects the source or a shared rendezvous point to networks containing group members.
3. **Packet replication:** A router creates copies only when two or more outgoing branches are required.
4. **Pruning and grafting:** Branches without interested receivers are removed. If a receiver later joins, the required branch is reattached.
5. **Loop prevention:** Reverse-path checks and routing state ensure that packets do not circulate indefinitely.

## Types of multicast trees

Source-based shortest-path tree: A separate tree is rooted at each source and usually gives low delay, but routers may need substantial state for many sources. Shared tree: All sources send toward a common core or rendezvous point, reducing state but sometimes producing a longer path. Dense-mode protocols initially flood and prune; sparse-mode protocols explicitly join the tree where receivers exist.

## Example

In Figure: H1, H2 and H3 are members of group G, while H4 is not. Source S sends one packet. Routers duplicate it only along branches leading to H1, H2 and H3. The R7-H4 branch is pruned. Compared with three separate unicast streams, the common links carry only one copy.

## Unicast, broadcast and multicast comparison

| Property | Unicast | Broadcast | Multicast |
|---|---|---|---|
| **Receiver set** | One destination | All hosts in domain | Members of a group |
| **Source copies** | One per receiver | Usually one | Usually one |
| **Router replication** | No | At broadcast branches | Only toward group members |
| **Efficiency for group data** | Low | Wasteful for non-members | High |
| **Typical use** | Web request | Local discovery | Live class/video feed |

## Advantages

- Saves source and link bandwidth for one-to-many communication.
- Scales better than transmitting a separate unicast stream to each receiver.
- Non-members do not receive the packet when pruning is correct.
- Supports dynamic membership and distributed applications.

## Limitations

- Routers must maintain group and tree state.
- Deployment across different administrative domains is complex.
- Reliability, ordering and congestion control for many receivers are difficult.
- Security requires control of who may send to or join a group.

## Applications

- Live online lectures and IPTV
- Financial-market data feeds
- Video conferences
- Software and content distribution
- Replicated databases and service discovery

## Conclusion

Multicast routing is an efficient one-to-selected-many service. Membership protocols identify receivers, routers build source or shared trees, and pruning ensures that packet copies travel only toward active group members.

# Distance-Vector Routing

Distance-vector routing is a distributed adaptive algorithm in which every router maintains a vector containing its current best distance to each destination and the corresponding next hop. A router periodically or after a change sends its complete vector to immediate neighbours. It does not initially know the full topology.

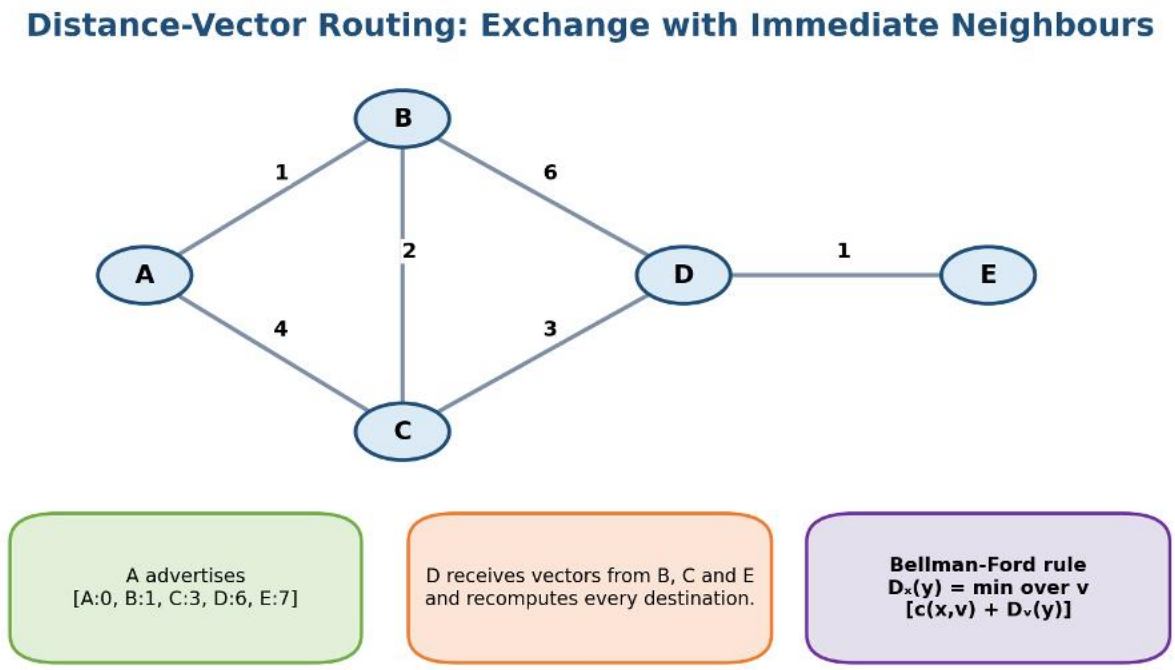



**Figure: Distance-vector exchange among neighbouring routers**

## Bellman-Ford principle

For router x, destination y and neighbour v, let c(x,v) be the direct-link cost and Dv(y) be neighbour v's advertised distance to y. Router x computes:

$$D_x(y) = \min_v \{ c(x,v) + D_v(y) \}$$

The neighbour that gives the minimum value becomes the next hop. Repeated exchanges propagate reachability information through the network until no table changes occur; this condition is convergence.

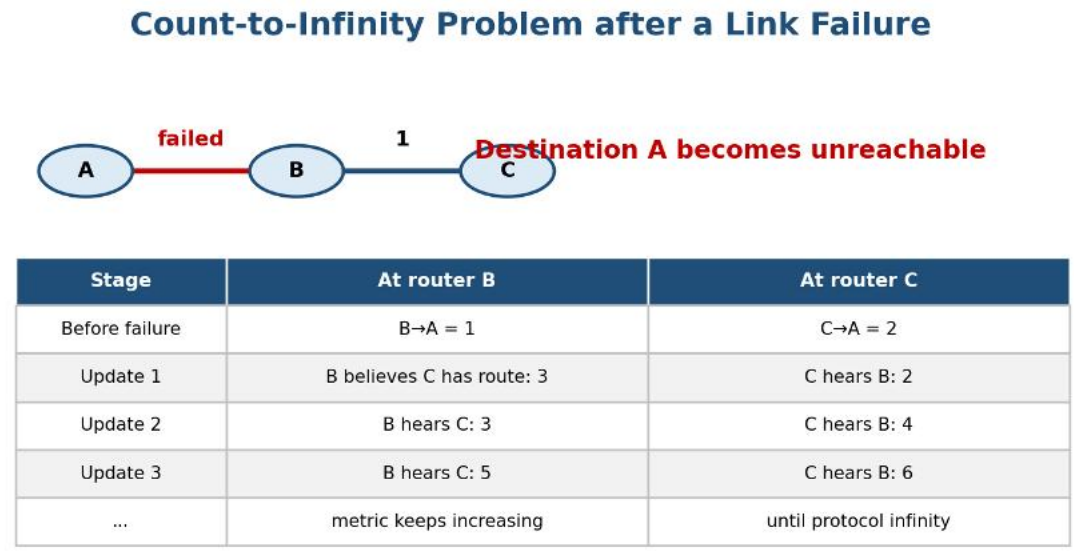
## Algorithm steps

1. Initialize distance to self as 0, distance to each directly connected neighbour as the link cost, and all other distances as infinity.
2. Send the current vector to every neighbour periodically and immediately after an important change.
3. When a vector is received from neighbour v, add the cost of the x-v link to every advertised destination value.
4. For each destination choose the smallest result among all neighbours and update the next hop.
5. Continue until all routers have consistent vectors.

## Worked update example

In Figure 3.9 router D has direct costs D-C=3, D-B=6 and D-E=1. Suppose C advertises distance 3 to A, B advertises distance 1 to A and E has no route to A. D compares via C: 3+3=6; via B: 6+1=7; via E: infinity. Therefore D selects next hop C and distance 6 to A.

## Count-to-infinity problem




| Stage | At router B | At router C |
|---|---|---|
| Before failure | B→A = 1 | C→A = 2 |
| Update 1 | B believes C has route: 3 | C hears B: 2 |
| Update 2 | B hears C: 3 | C hears B: 4 |
| Update 3 | B hears C: 5 | C hears B: 6 |
| ... | metric keeps increasing | until protocol infinity |

Remedies: split horizon, poisoned reverse, triggered updates and hold-down timers.

**Figure: Count-to-infinity after the A-B link fails**

Distance-vector routers know what neighbours advertise but not the complete route behind the advertisement. After destination A becomes unreachable, B may believe that C still has a route to A, while C's route actually passes through B. They repeatedly increase the metric and create a temporary routing loop. Bad news therefore travels slowly.

## Solutions and improvements

- **Split horizon:** Do not advertise a route back on the interface from which it was learned.
- **Poisoned reverse:** Advertise that route back with an infinite metric, clearly indicating that the neighbour must not use it.
- **Triggered updates:** Send an update immediately after a significant metric or reachability change instead of waiting for the periodic timer.
- **Hold-down timer:** Temporarily reject suspicious claims of a better route after a failure.
- **Finite infinity:** Protocols use a small maximum metric; RIP treats 16 hops as unreachable.
- **Path information:** Path-vector protocols carry the sequence of administrative domains, allowing loop detection.

## Advantages

- Distributed and conceptually simple.
- Routers exchange information only with neighbours.
- Requires less topology memory than a full link-state database.
- Automatically adapts to many network changes.

## Limitations

- May converge slowly after failures.
- Can form temporary loops and suffer count-to-infinity.
- Periodic full-vector updates consume bandwidth in large networks.
- A simple hop metric may choose a slow or congested path.

## Applications

- Routing Information Protocol (RIP)
- Early ARPANET routing
- Small educational and enterprise networks
- Foundation for distributed Bellman-Ford methods
- Conceptual basis for path-vector routing

## Conclusion

Distance-vector routing computes routes through repeated neighbour-to-neighbour Bellman-Ford updates. It is simple and distributed, but loop-prevention techniques are essential because routers do not possess a complete topology map.

# Congestion-Control Algorithms

Congestion occurs when the amount of traffic offered to a network approaches or exceeds the capacity of links, router processors or buffers. Packets accumulate in queues, delay and jitter rise, buffers overflow, retransmissions add more load and useful throughput may eventually fall. This severe condition is called congestion collapse.

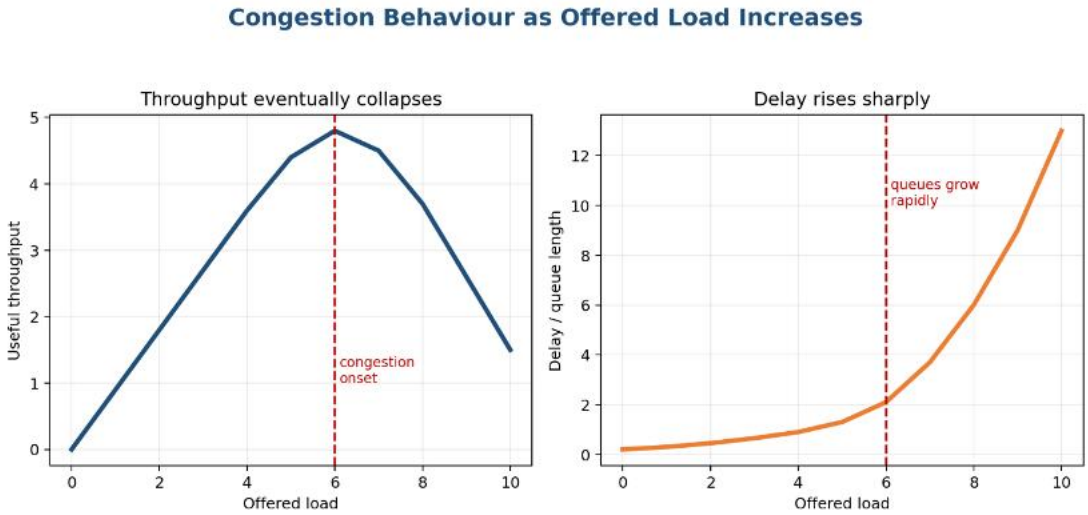



Figure: Throughput and delay as offered load increases

## Major causes

- Many input links sending simultaneously to one slower output link.
- Insufficient router buffer space or slow packet processing.
- Bursty traffic and synchronized sources.
- Routing that concentrates flows on the same path.

- Unnecessary retransmissions caused by long delay or packet loss.
- Link or router failure that shifts traffic onto fewer remaining resources.

## Congestion control versus flow control

Flow control prevents one sender from overwhelming one receiver. Congestion control protects the network as a whole from excessive aggregate load. A receiver may be fast while routers in the middle are congested; therefore end-system flow control alone cannot solve network congestion.

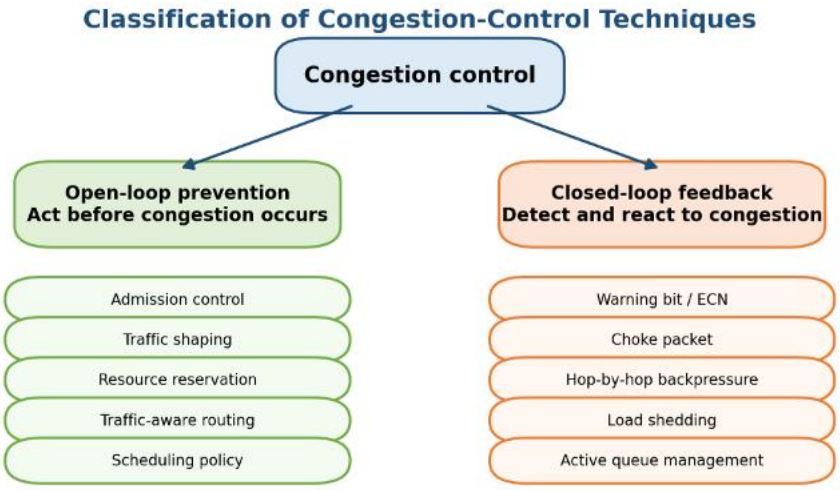



**Figure: Open-loop and closed-loop congestion-control techniques**

## Open-loop congestion prevention

- **Admission control:** A new flow or virtual circuit is accepted only when sufficient bandwidth and buffers are available.
- **Traffic shaping and policing:** Leaky-bucket and token-bucket rules limit the rate and burst size entering the network.
- **Traffic-aware routing:** Routes are chosen to avoid heavily loaded links, while damping prevents oscillation.
- **Resource reservation:** Capacity and buffer space are reserved for admitted traffic.
- **Scheduling and discard policy:** Routers decide service order and which packets may be dropped first when buffers fill.
- **Retransmission and acknowledgement policy:** Timers and acknowledgement strategies are chosen to avoid unnecessary duplicate traffic.
- **Closed-loop feedback algorithms**
- **Backpressure:** A congested router asks upstream routers to reduce forwarding, and the signal may propagate toward the source.
- **Choke packet:** The router sends a control packet to the source identifying a congested destination or flow. The source temporarily lowers its rate.
- **Explicit Congestion Notification:** A router marks packets instead of dropping them. The receiver informs the sender, which reduces its sending rate.
- **Hop-by-hop choke:** Every router on the return path reduces the traffic immediately, giving faster relief on long paths.
- **Load shedding:** When overload is severe, selected packets are discarded. Policy may protect real-time, control or high-priority packets.

- **Active queue management:** A router drops or marks some packets before the buffer becomes full. RED is a classic example intended to signal congestion early.

## Traffic shaping

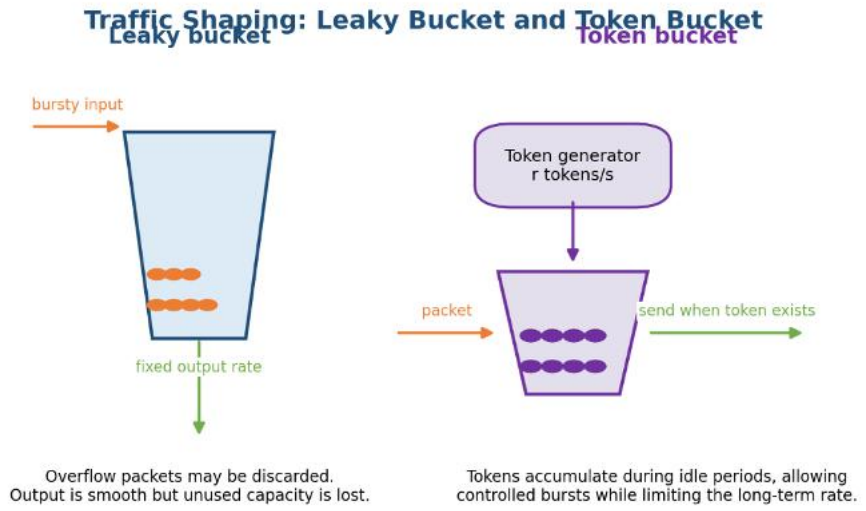



**Figure: Leaky-bucket and token-bucket traffic shaping**

The leaky bucket produces a nearly constant output rate. A burst enters a finite bucket, but packets leave at a fixed rate; overflow is discarded. The token bucket creates tokens at rate r and stores up to b tokens. A packet can be sent only by consuming the required tokens. It limits long-term rate while allowing a controlled burst after an idle period.

## Example

A router output link can transmit 1000 packets per second, but several inputs suddenly offer 1800 packets per second. Its queue grows and begins dropping packets. ECN marks packets, TCP sources reduce their sending windows, and a token-bucket policer limits a non-responsive video source. The offered load falls below capacity and the queue returns to a stable size.

## Advantages

- Prevents congestion collapse and improves useful throughput.
- Controls delay and packet loss.
- Allocates resources more fairly among flows.
- Protects real-time and high-priority services.

## Limitations

- Feedback arrives after congestion has begun.
- Aggressive traffic-aware routing can oscillate.
- Admission and reservation reduce flexibility and require state.
- Incorrect dropping or policing can unfairly damage a flow.
- Some sources do not respond to congestion signals.

## Conclusion

Congestion control combines prevention, feedback and traffic conditioning. Admission, shaping and scheduling reduce the probability of overload, while ECN, choke messages, active queue management and load shedding restore stability after congestion is detected.

# Quality of Service

Quality of Service (QoS) is the ability of a network to provide predictable and differentiated performance to traffic classes or flows. Best-effort delivery attempts to carry every packet but gives no guarantee. QoS mechanisms control resource allocation so that applications such as voice, video, industrial control and critical transactions receive acceptable performance.

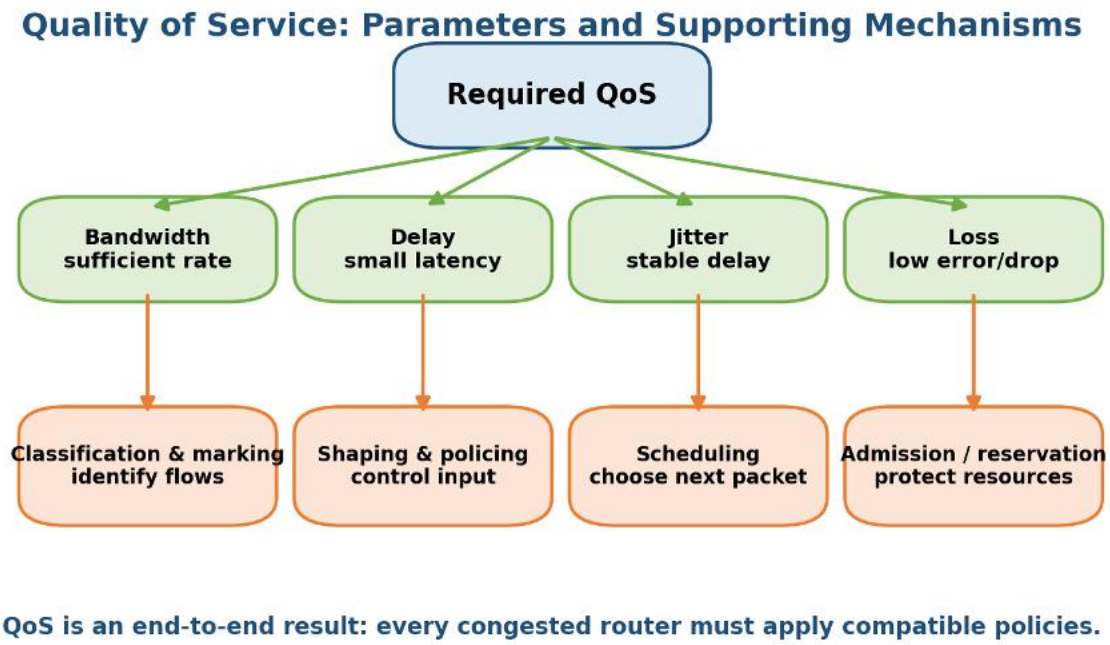



**Figure: QoS parameters and mechanisms**

## QoS parameters

- **Bandwidth or throughput:** The data rate available to a flow. A high-definition stream needs sustained capacity.
- **Delay or latency:** The time from source to destination, including processing, queueing, transmission and propagation.
- **Jitter:** Variation in packet delay. Real-time audio requires small jitter to avoid irregular playback.
- **Packet loss or error rate:** The fraction of packets that are corrupted or discarded. File transfer tolerates delay better than loss, while live voice may tolerate a small loss better than a late packet.
- **Availability and reliability:** The probability that service remains reachable and correctly forwards traffic.
- **Priority:** The relative importance assigned to a class when resources are limited.

## QoS techniques

1. **Overprovisioning:** Install more capacity than normal peak demand. It is simple but costly and cannot eliminate every burst.
2. **Classification and marking:** Packets are identified by address, port, protocol or application and marked with a class, for example the IPv4/IPv6 DSCP field.
3. **Traffic shaping and policing:** Token-bucket or leaky-bucket rules make input traffic conform to a profile. Shaping delays excess traffic; policing may remark or drop it.

4. **Buffering:** A receiver or router temporarily stores packets to absorb jitter, but larger buffers increase delay.
5. **Packet scheduling:** FIFO serves arrival order; priority queuing serves high classes first; round robin rotates among queues; weighted fair queuing allocates a configured share to each class.
6. **Admission control:** A new real-time flow is rejected when its requirements would violate guarantees already given to admitted flows.
7. **Resource reservation:** Bandwidth, queues and scheduling parameters are reserved along the path. Integrated Services and RSVP are classic approaches.
8. **Differentiated Services:** Routers apply per-hop behaviour to marked traffic classes rather than storing detailed state for every individual flow.
9. **Traffic engineering:** Paths are selected so that demand is distributed across available links and important flows avoid bottlenecks.

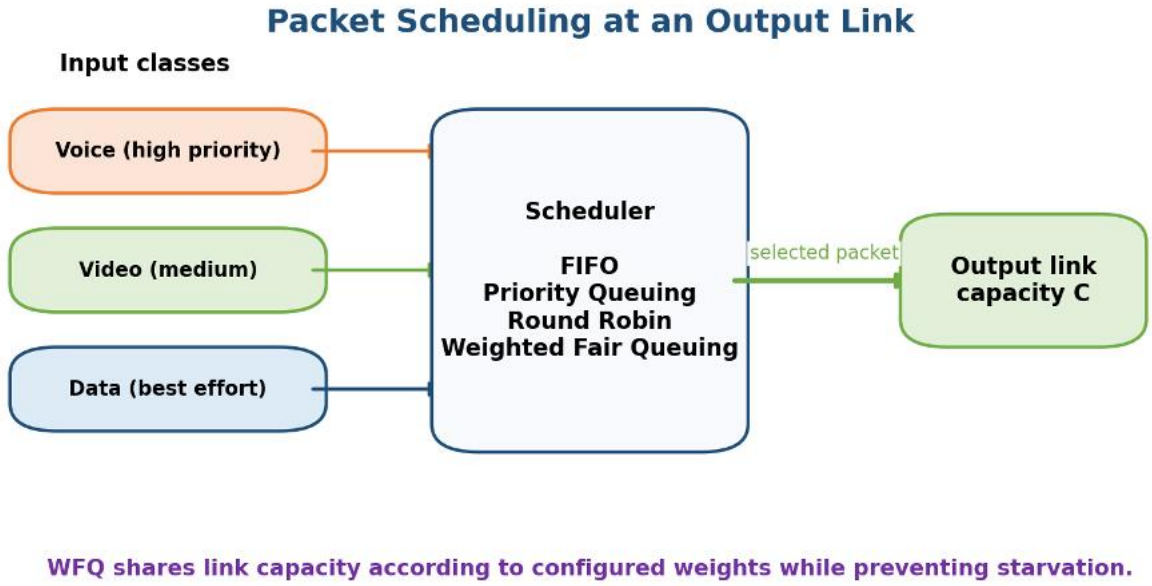



**Figure: Packet scheduling for voice, video and best-effort queues**

## Worked example

A college has a 100-Mbit/s internet link. During an online examination, the administrator reserves 20 Mbit/s for the examination portal and authentication, gives interactive voice a high-priority low-latency queue, assigns video classes a weighted queue and allows downloads to use remaining best-effort capacity. Token-bucket policers prevent any student device from consuming the entire link.

## Advantages

- Provides predictable performance to delay-sensitive applications.
- Prevents one traffic class from monopolizing shared resources.
- Improves user experience for voice and video.
- Supports service-level agreements and business-critical applications.
- Uses available capacity more intelligently than simple equal treatment.

## Limitations

- True end-to-end guarantees require cooperation across every network on the path.
- Strict priority can starve lower classes unless carefully limited.

- Classification, policing and scheduling increase configuration complexity.
- Reserved capacity may remain unused while best-effort traffic is waiting.
- QoS does not create bandwidth; it only manages and prioritizes available resources.

## Applications

- Voice over IP and video conferences
- Telemedicine and industrial control
- Online examinations and banking transactions
- Cloud service-level agreements
- Enterprise WAN and multimedia streaming

## Conclusion

QoS converts application requirements into measurable network treatment. Classification, shaping, scheduling, admission control, reservation and differentiated services work together to control bandwidth, delay, jitter and loss.

# Internetworking

Internetworking is the process of connecting two or more independent networks so that hosts on one network can communicate with hosts on another. The component networks may use different physical media, frame formats, speeds, maximum packet sizes and administrative policies. Routers provide a common network-layer service across these differences.

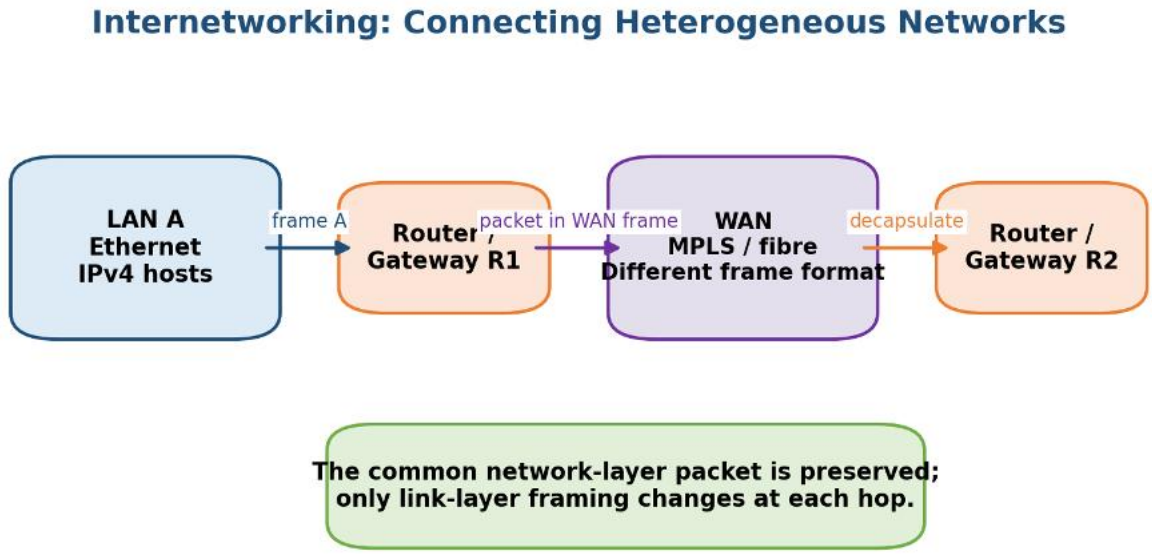



**Figure: Connecting heterogeneous networks through routers and gateways**

## Major internetworking issues

**Addressing:** Every destination requires a globally or administratively unique network-layer address and a scalable way to aggregate prefixes.

**Routing between domains:** Routers must select paths across networks owned by different organizations and obey policy as well as technical cost.

**Different maximum transmission units:** A packet acceptable on one network may be too large for the next link.

**Packet lifetime:** Hop count or time-to-live prevents a packet from circulating forever after a routing loop.

**Protocol differences:** Encapsulation, tunnelling, translation or dual-stack operation may be required when network-layer protocols differ.

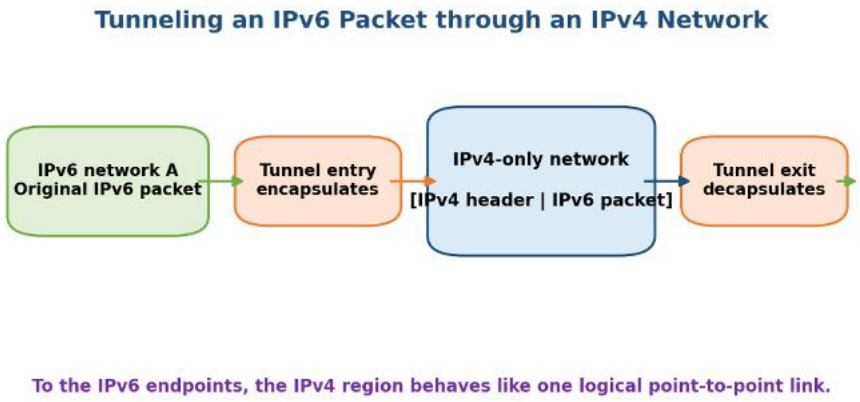
**Error reporting:** Control messages must cross the internetwork and identify the point or reason for failure.

**Quality and security:** Different networks may provide different bandwidth, delay, filtering and trust levels.

## Encapsulation at each hop

A router removes the incoming link-layer frame, examines the common network-layer packet, updates necessary fields and places the packet inside a new frame suitable for the outgoing link. Ethernet MAC addresses are therefore local to one link, while the IP source and destination addresses normally remain end to end.

## Tunnelling




Tunnelling carries a packet of one protocol through a network that does not directly support it. The tunnel entry encapsulates the original packet inside an outer packet understood by the intermediate network. The tunnel exit removes the outer header and restores the original packet. VPNs, IPv6 transition and overlay networks commonly use tunnelling.

## Fragmentation and reassembly

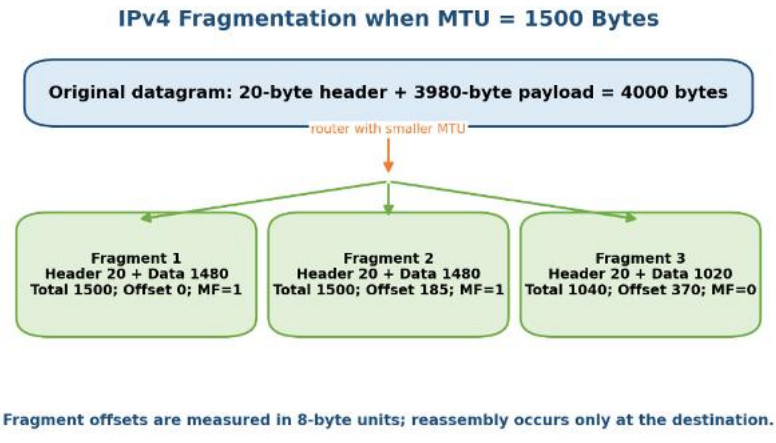


**Figure: IPv4 fragmentation for a 1500-byte MTU**

The maximum packet that a link can carry is its MTU. In IPv4, a router may divide a larger datagram into fragments. Each fragment receives its own IP header, the same identification value, a fragment offset and a More Fragments flag. Only the final destination reassembles the original datagram. If any fragment is lost, the complete upper-layer packet may need retransmission.

**Worked example**: A 4000-byte IPv4 datagram contains a 20-byte header and 3980 bytes of data. On a link with MTU 1500, each non-final fragment can carry 1480 bytes because fragment data except the last must be a multiple of 8 bytes. The three fragments have total lengths 1500, 1500 and 1040 bytes, with offsets 0, 185 and 370.

# Advantages

- Allows independently designed networks to communicate.
- Preserves investment in existing link technologies.
- Tunnelling provides gradual protocol migration and secure overlays.
- Hierarchical addressing and routing support global scale.

# Limitations

- Protocol translation and tunnelling add headers, processing and troubleshooting complexity.
- Fragmentation increases overhead and loss sensitivity.
- Different policies and security rules may block otherwise valid traffic.
- End-to-end QoS is difficult across independent networks.
- Incorrect MTU discovery can create connectivity problems.

# Applications

- The global Internet
- Enterprise branch and cloud interconnection
- Virtual private networks
- IPv6 transition tunnels
- Mobile networks and data-centre overlays

# Conclusion

Internetworking creates a common packet-delivery service across heterogeneous networks. Routers change local framing, tunnelling crosses incompatible regions, and fragmentation or path-MTU methods handle different packet-size limits.

# The Network Layer in the Internet

**Internet network-layer architecture**

The Internet Network Layer provides a connectionless, best-effort datagram service. Hosts and routers use the Internet Protocol to address and forward packets across multiple networks. IP does not guarantee delivery, ordering, duplicate suppression or a fixed delay. Reliability, when needed, is provided by transport protocols such as TCP and by applications.

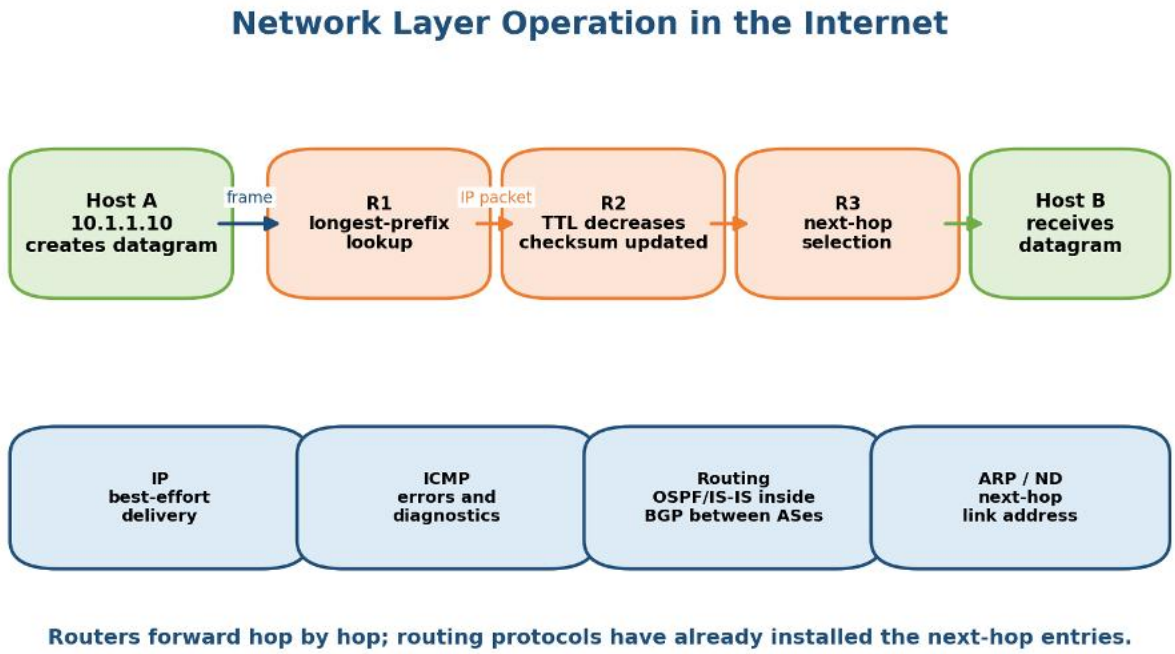



**Figure: Packet forwarding and supporting protocols in the Internet**

## IPv4 datagram

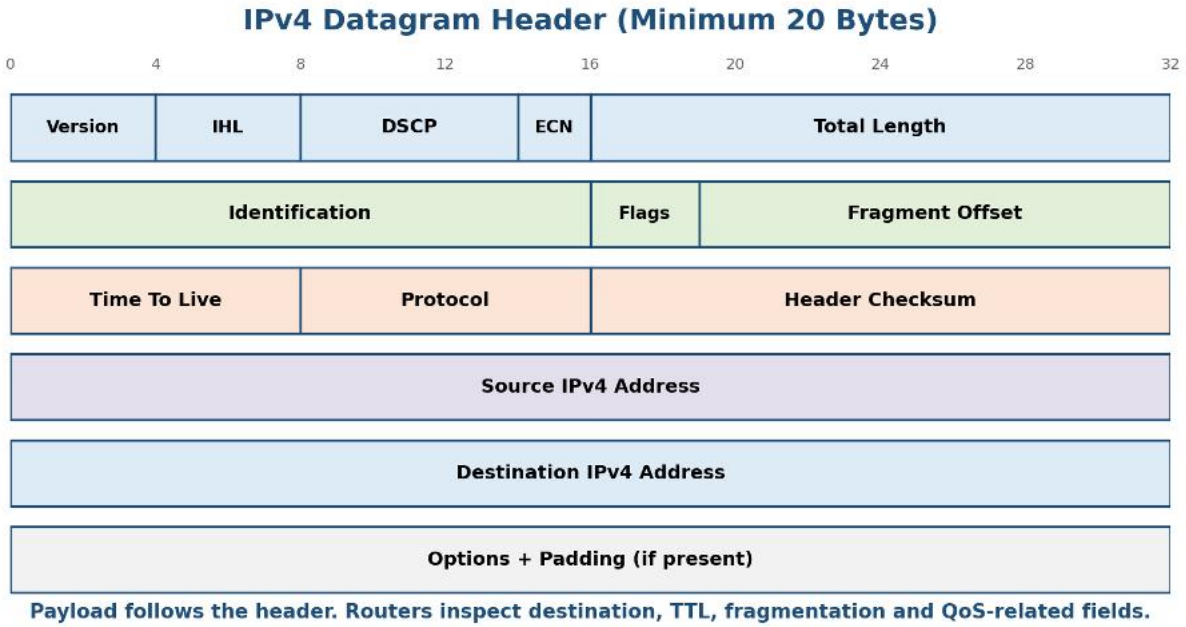



**Figure: IPv4 datagram header**

**Version:** Identifies IPv4.

**IHL:** Gives the header length in 32-bit words; the minimum is 5 words or 20 bytes.

**DSCP and ECN:** Carry service-class and explicit-congestion information.

**Total Length:** The size of the entire datagram, header plus data, up to 65,535 bytes.

**Identification, Flags and Fragment Offset:** Support fragmentation and reassembly.

**Time To Live:** Decreased by every router; the packet is discarded at zero to stop endless loops.

**Protocol:** Identifies the upper-layer payload, such as TCP, UDP or ICMP.

**Header Checksum:** Detects errors in the IPv4 header and is recomputed at every router because TTL changes.

**Source and Destination Addresses:** Provide 32-bit logical addresses used for delivery and return communication.

**Options and Padding:** Optional control information, rarely used in ordinary forwarding.

# IPv4 addressing and CIDR

An IPv4 address contains a network prefix and a host part. Classless Inter-Domain Routing writes the prefix length after a slash, for example 192.168.10.0/24. A router compares the destination address with routing-table prefixes and selects the longest matching prefix. CIDR allows variable-size networks and route aggregation.

Example: If a forwarding table contains 10.0.0.0/8 through R1, 10.20.0.0/16 through R2 and 10.20.30.0/24 through R3, destination 10.20.30.45 is sent through R3 because /24 is the longest and most specific match.

# Internet control and address-support protocols

**ICMP:** Reports errors and provides diagnostics. Echo Request/Reply supports ping; Time Exceeded supports traceroute; Destination Unreachable reports delivery problems.

**ARP and IPv6 Neighbour Discovery:** Resolve the next-hop IP address to a link-layer address on a local network. They operate at the boundary between network and link layers.

**DHCP:** Supplies a host with an IP address, prefix, default gateway, DNS server and lease information.

**NAT:** Translates private internal addresses and ports to public values at an edge device. It conserves IPv4 addresses but weakens transparent end-to-end addressing.

### Routing in the Internet

The Internet is hierarchical. Within one autonomous system, an Interior Gateway Protocol such as OSPF or IS-IS computes routes using technical metrics. Between autonomous systems, BGP advertises reachable prefixes and selects policy-compliant paths using attributes such as AS path and local preference. Forwarding remains hop by hop using the chosen next-hop entries.

### IPv6

IPv6 uses 128-bit addresses, a fixed 40-byte base header, extension headers for optional information and no router fragmentation. It eliminates broadcast in favour of multicast and anycast, and simplifies forwarding by removing the IPv4 header checksum. Hosts use Neighbour Discovery and can configure addresses through router advertisements or DHCPv6.

**IPv4 and IPv6: Major Network-Layer Differences**

| Feature | IPv4 | IPv6 |
|---|---|---|
| **Address size** | 32 bits | 128 bits |
| **Base header** | Variable 20-60 bytes | Fixed 40 bytes |
| **Router fragmentation** | Allowed | Not allowed |
| **Checksum** | Header checksum | No header checksum |
| **Broadcast** | Supported | No broadcast; multicast/anycast |
| **Address notation** | 192.0.2.10 | 2001:db8::10 |

IPv6 expands the address space and simplifies router processing while using extension headers for optional functions.

**Figure: Major differences between IPv4 and IPv6**

# End-to-end example

1. Host A compares destination B with its own prefix. Because B is remote, A selects the default gateway.
2. ARP or Neighbour Discovery obtains the gateway's link-layer address, and A places the IP datagram in a local frame.
3. Each router removes the incoming frame, decrements TTL or Hop Limit, performs a longest-prefix lookup and creates a new outgoing frame.
4. Interior and interdomain routing protocols have already installed the forwarding entries used by these lookups.
5. The final router resolves B's local link address and delivers the datagram. The transport layer at B processes the payload.

# Advantages

- Open, vendor-independent packet delivery across heterogeneous networks.
- Connectionless service is robust and supports rerouting.
- CIDR and hierarchical routing provide global scalability.
- IPv6 provides an enormous address space and simpler base-header processing.

# Limitations

- Best-effort IP gives no guarantee of delivery, order, bandwidth or delay.
- IPv4 address scarcity caused widespread NAT and operational complexity.
- Routing attacks or misconfiguration can affect large parts of the Internet.
- Fragmentation, MTU problems and filtering can create difficult failures.
- End-to-end QoS and multicast deployment remain challenging across many providers.

# Conclusion

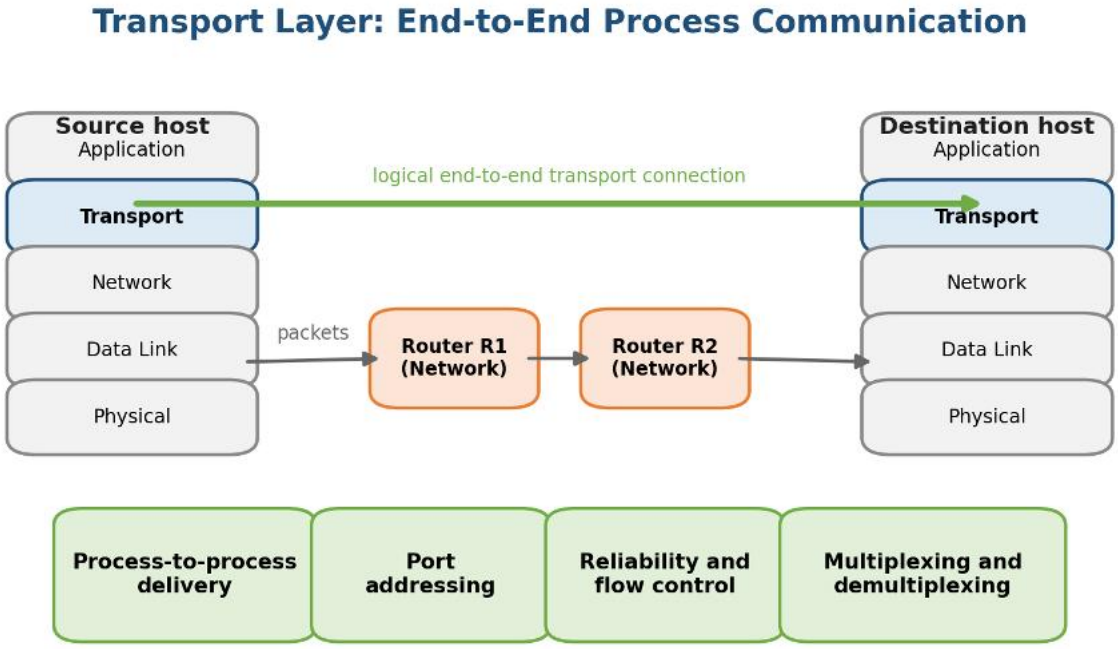
The Internet Network Layer is built around connectionless IP datagrams, hierarchical addressing and hop-by-hop forwarding. ICMP and address-resolution protocols support operation, OSPF and BGP establish routes, and IPv6 extends the architecture with 128-bit addressing and a simplified base header.

## UNIT 4: THE TRANSPORT LAYER

# Transport Layer Services

The Transport Layer is Layer 4 of the OSI model and the second end-to-end layer of the TCP/IP architecture. It provides logical communication between application processes running on different hosts. The Network Layer delivers packets from host to host, whereas the Transport Layer identifies the exact sending and receiving processes and, when required, supplies reliability, ordering, flow control and congestion control.

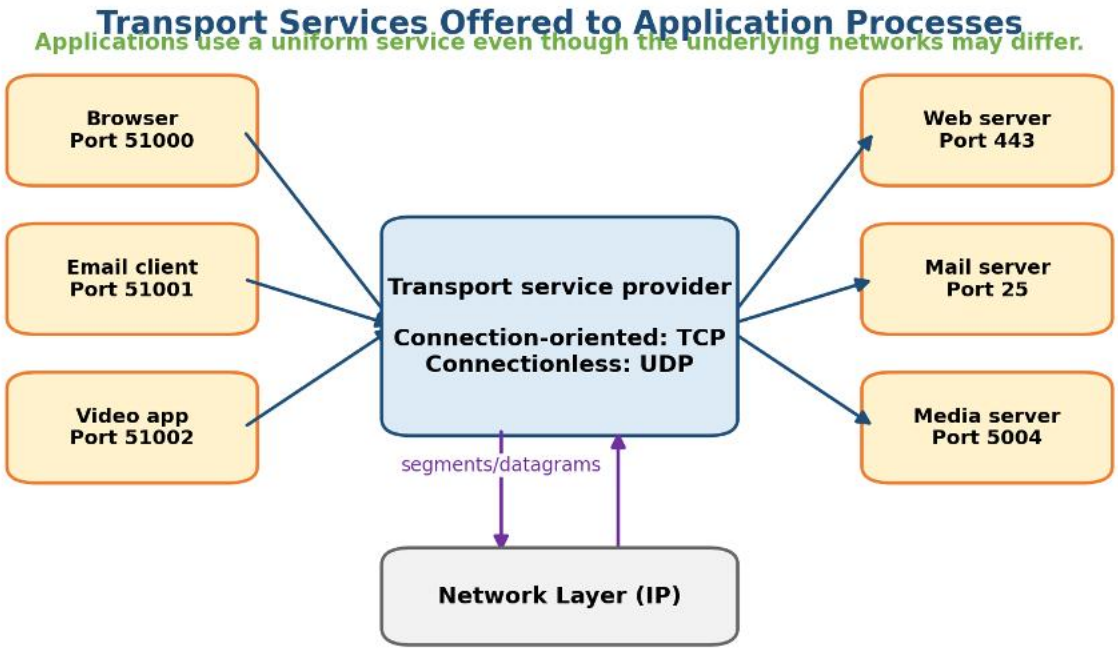
A transport entity is the software or firmware that implements a transport protocol. It may be part of the operating-system kernel, a networking library or a specialized communication subsystem. Applications access it through sockets or transport service access points.




## Main transport services

1. **Process-to-process delivery:** Port numbers identify the source and destination applications. A complete endpoint is commonly represented by an IP address and a port number.
2. **Segmentation and reassembly:** Large application messages are divided into transport segments. At the receiver, segments are reordered and combined before delivery to the application.
3. **Connection control:** A service may be connection-oriented, as in TCP, or connectionless, as in UDP. Connection-oriented service establishes and releases state at both endpoints.

4. **Reliable delivery:** Sequence numbers, acknowledgments, retransmission timers and checksums detect loss, duplication and corruption.
5. **Ordered delivery:** Out-of-order segments are buffered and presented to the application in the original byte order.
6. **Flow control:** The sender is prevented from overwhelming a slow receiver. TCP uses a receiver-advertised sliding window.
7. **Congestion control:** The sender adapts its transmission rate when the network shows signs of overload.
8. **Multiplexing and demultiplexing:** Many application streams share the Network Layer, and arriving segments are separated using port numbers and connection identifiers.
9. **Full-duplex communication:** Both endpoints can send and receive data simultaneously over one transport connection.
10. **Quality-related support:** Real-time transport protocols add timestamps, source identifiers and loss information required by audio and video applications.




## Port addressing and sockets

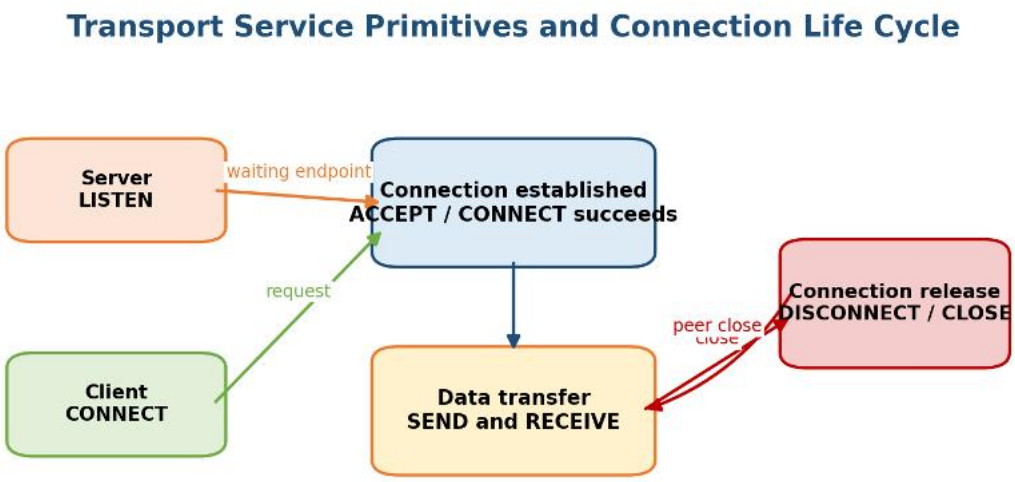
An IP address identifies a host interface, but a host may run many applications at the same time. A port number identifies one application endpoint within that host. The pair (IP address, port number) is called a socket address. A TCP connection is uniquely identified by source IP, source port, destination IP and destination port. This four-item combination allows thousands of clients to use the same server port simultaneously.

| Port category | Range | Typical meaning |
|---|---|---|
| **Well-known ports** | 0-1023 | Standard server services such as HTTP, HTTPS, DNS and SMTP |
| **Registered ports** | 1024-49151 | Assigned applications and vendor services |
| **Dynamic/private ports** | 49152-65535 | Temporary client-side or private application ports |

## Transport service primitives

A service primitive is an operation that an application invokes at the boundary between the application and transport layers. The exact names differ among operating systems, but the conceptual operations are similar.

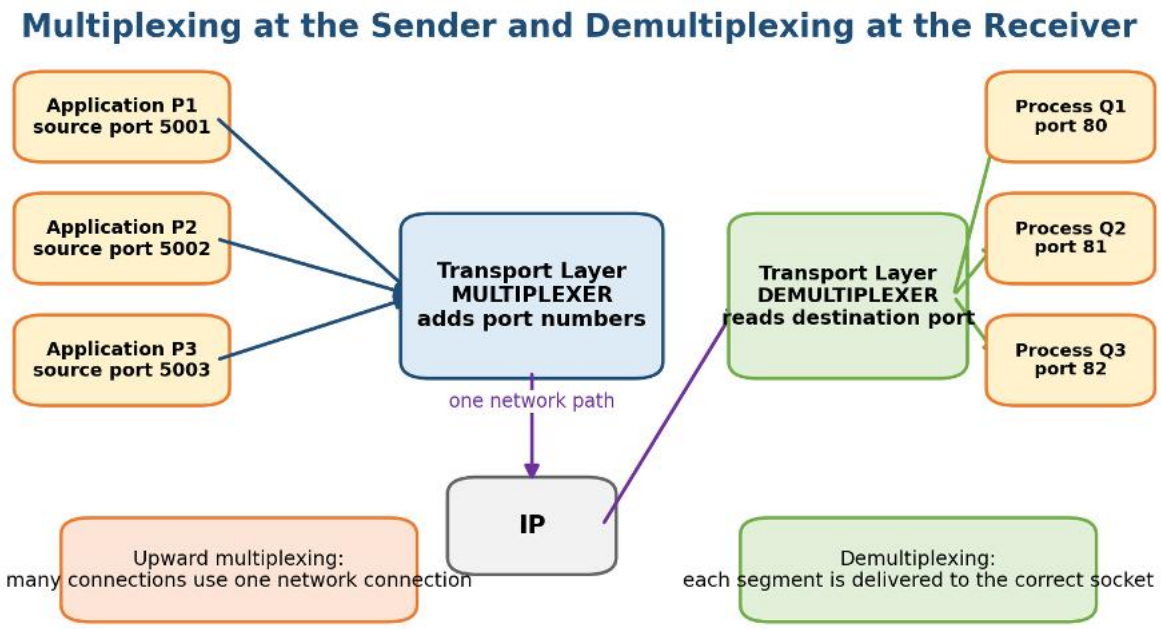
| Primitive | Purpose | Typical socket equivalent |
|---|---|---|
| **LISTEN** | Wait for an incoming connection request | listen() |
| **CONNECT** | Actively request a connection to a remote endpoint | connect() |
| **ACCEPT** | Accept a pending incoming connection | accept() |
| **SEND** | Transfer application bytes or a message | send()/write() |
| **RECEIVE** | Obtain bytes or a message from the peer | recv()/read() |
| **DISCONNECT** | Release the connection | close()/shutdown() |




A transport service interface hides packet routing and exposes operations meaningful to applications.

## Multiplexing and demultiplexing

At the sender, the transport layer accepts data from many processes and adds a header containing source and destination ports. This is multiplexing. At the receiver, the transport layer examines the destination port and other connection fields and delivers the data to the correct socket. This is demultiplexing. In upward multiplexing, several transport connections may share one network path. In downward multiplexing, one high-rate transport connection may be distributed over several network paths when the architecture permits it.




## Example

A student laptop can simultaneously browse a secure website, send email and attend an online class. The browser may use source port 51000 and destination port 443, the mail client may use source port 51001 and destination port 587, and the media application may use UDP ports assigned for its RTP session. All packets use the same laptop IP address, but the transport headers keep the three conversations separate.

## Advantages

- Provides a uniform process-to-process interface independent of router and link technologies.
- Allows many applications to share one host and one network connection safely.
- Can provide reliable, ordered and full-duplex communication.
- Separates application design from packet routing and link-level details.
- Supports both dependable data transfer and low-delay datagram service.

## Limitations

- Reliability mechanisms add headers, state, buffering and delay.

- End-to-end recovery cannot repair every application-level error or semantic failure.
- Encryption and authentication are not automatically supplied by TCP or UDP.
- Port numbers alone do not identify a user and can be targeted by scanning or denial-of-service attacks.
- Real-time applications may need additional protocols because ordinary TCP retransmission can cause excessive delay.

## Applications

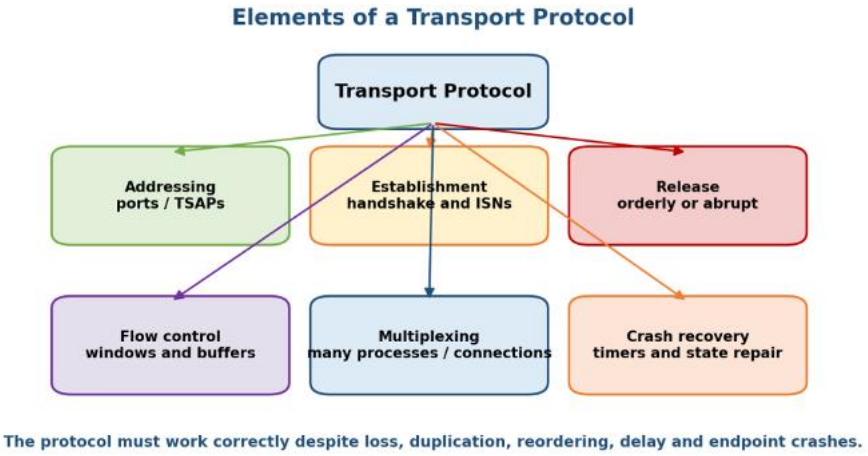
- Web and cloud services
- Email and file transfer
- Database connections
- Voice and video communication
- Online games and DNS
- Industrial and Internet of Things communication

## Conclusion

The Transport Layer converts host-to-host packet delivery into an application-oriented service. Port addressing, segmentation, multiplexing, connection control, reliability, flow control and congestion control allow independent processes to communicate efficiently across an internetwork.

# Elements of Transport Protocols

A transport protocol operates only at the end hosts, but it must cope with an unreliable and variable-delay network. Packets may be lost, duplicated, reordered or delayed, and either endpoint may crash. Therefore, a correct transport protocol needs mechanisms for naming endpoints, establishing fresh connections, controlling data, releasing state and recovering from failures.




## 1. Addressing

Applications are identified by Transport Service Access Points (TSAPs), implemented in the Internet as port numbers. A server normally uses a stable, well-known or registered port,

while a client obtains a temporary port. A name server or directory may translate a service name into an address and port. The transport header carries the source and destination ports so that the receiver can demultiplex incoming traffic.

Example: a web server listens on TCP port 443. Client A may connect from port 52000 and Client B from port 52001. Although both connections end at the same server port, their four-tuples differ, so TCP keeps their byte streams separate.

## 2. Connection establishment

A connection-oriented transport protocol must confirm that both endpoints are ready and must choose initial sequence numbers. A simple two-message exchange is unsafe because an old delayed connection request could reappear and create a false connection. TCP therefore uses a three-way handshake. The final acknowledgment proves that the initiator received the responder's current sequence number, so both sides agree that the exchange is fresh.


Sequence numbers and the third message distinguish a current request from an old duplicate.

## 3. Connection release

Connection release may be abrupt or orderly. Abrupt release immediately removes state and may discard data still in transit. Orderly release treats each direction independently: an endpoint can stop sending while continuing to receive. Because a release message or its acknowledgment can be lost, timers and retransmissions are required. TCP uses FIN and ACK exchanges and keeps the active closer in TIME-WAIT to absorb delayed duplicates.

## 4. Flow control and buffering

The sender and receiver may operate at different speeds. The receiver therefore advertises available buffer space, and the sender limits the amount of unacknowledged data. Buffer organization may use fixed-size buffers, variable-size buffers or circular byte buffers. The protocol must decide how much data to accumulate before sending, how to avoid tiny packets and how to prevent a fast sender from exhausting receiver memory.

- **Sender buffer:** holds bytes that have been accepted from the application but not yet acknowledged.

- **Receiver buffer:** stores out-of-order bytes and data not yet read by the application.
- **Window:** specifies how much data can remain outstanding.
- **Piggybacking:** places acknowledgments in reverse-direction data segments to reduce overhead.
- **Silly-window prevention:** avoids repeated transmission of very small amounts of data.

## 5. Multiplexing

Upward multiplexing maps several transport connections onto one network connection or one IP service. It reduces network-level setup and allows efficient sharing. Downward multiplexing distributes one transport connection across multiple network connections to obtain more bandwidth or resilience, but it complicates ordering. Internet TCP normally uses one IP path at a time, while advanced multipath transports can use several paths.

## 6. Crash recovery

A host crash destroys volatile connection state while the peer may continue believing the connection exists. After restart, old segments can still be present in the network. Recovery strategies use sequence spaces, maximum segment lifetimes, retransmission timers, keepalive messages, stable storage or application-level transactions. No general transport protocol can guarantee that a non-idempotent operation was executed exactly once when acknowledgments and endpoint state are lost; applications often use transaction identifiers and duplicate suppression.

## Worked example

Assume a client sends a payment request carrying transaction ID T908. The server processes it but crashes before the acknowledgment reaches the client. The client cannot determine from transport information alone whether the operation occurred. After timeout it may retransmit. To prevent a duplicate payment, the application stores T908 in durable storage and returns the earlier result instead of performing the operation again. This illustrates why crash recovery extends beyond the transport header.

## Advantages of well-designed protocol elements

- Prevent old duplicate packets from creating false connections.
- Protect receiver buffers and adapt to different endpoint speeds.
- Allow thousands of simultaneous application conversations.
- Support orderly shutdown without losing accepted data.
- Limit damage caused by loss, delay, reordering and crashes.

## Limitations and challenges

- More state and control messages increase implementation complexity.
- Timeout values are difficult to select in networks with changing delay.
- Crash recovery cannot always distinguish completed work from lost work.
- Large windows improve throughput but consume buffer memory.
- Middleboxes and address translation can interfere with transport connection state.

## Conclusion

The elements of a transport protocol solve end-to-end problems that link protocols cannot solve. Correct addressing, fresh connection setup, controlled buffering, safe release, multiplexing and crash-aware design are necessary for dependable process communication.

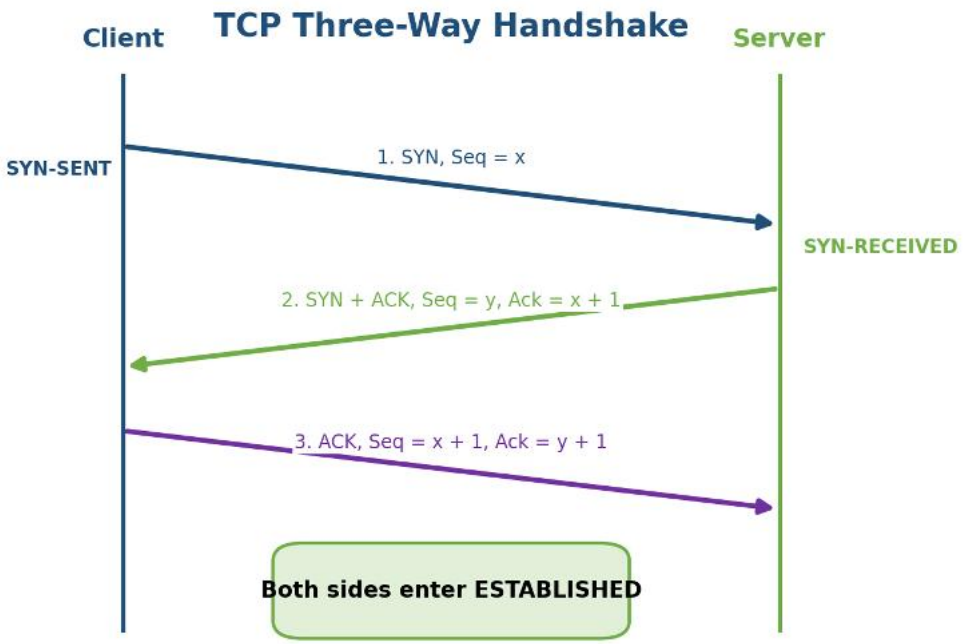
# Connection Management

Connection management consists of the procedures used to establish, maintain and release a connection-oriented transport association. The endpoints must agree on addresses, initial sequence numbers, options, buffer limits and readiness before application data is accepted. They must also remove state safely after both directions of data transfer are complete.

## Problems during establishment

- **Loss:** a connection request or reply may disappear, so timers and retransmissions are needed.
- **Duplication:** a retransmitted request may create more than one connection unless duplicates are recognized.
- **Delayed old packets:** a request from an earlier connection may arrive after that connection has ended.
- **Simultaneous open:** both endpoints may actively request a connection at nearly the same time.
- **Sequence synchronization:** both sides must learn the other side's current initial sequence number.
- **Resource exhaustion:** half-open connections consume memory and can be abused in SYN-flood attacks.

## TCP three-way handshake




**1. SYN:** The active opener selects initial sequence number x and sends a segment with SYN = 1 and Seq = x.

2. **SYN + ACK:** The passive opener allocates connection state, selects initial sequence number y and sends SYN = 1, ACK = 1, Seq = y and Ack = x + 1.
3. **Final ACK:** The client acknowledges the server sequence using ACK = 1, Seq = x + 1 and Ack = y + 1. Both endpoints then enter ESTABLISHED.

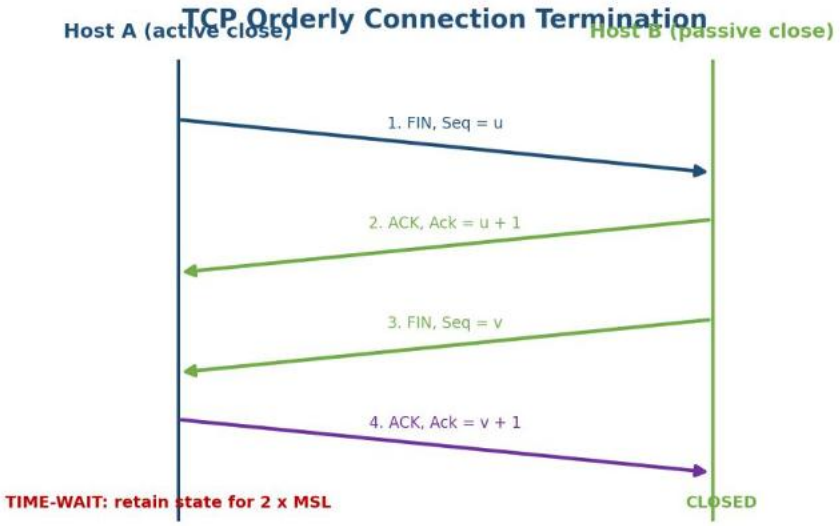
The SYN flag consumes one sequence number even when the segment carries no application data. The exchange also negotiates options such as Maximum Segment Size, window scaling, selective acknowledgment permission and timestamps.

## State transitions

| Endpoint action | Important states | Meaning |
|---|---|---|
| **Server waits** | CLOSED -> LISTEN | Ready to receive a SYN |
| **Client opens** | CLOSED -> SYN-SENT | SYN transmitted |
| **Server replies** | LISTEN -> SYN-RECEIVED | SYN received and SYN-ACK sent |
| **Handshake completes** | SYN-SENT/SYN-RECEIVED -> ESTABLISHED | Data transfer can begin |
| **Active close** | ESTABLISHED -> FIN-WAIT-1 -> FIN-WAIT-2 | Local FIN sent and acknowledged |
| **Passive close** | ESTABLISHED -> CLOSE-WAIT -> LAST-ACK | Remote FIN received; application later closes |
| **Final wait** | TIME-WAIT -> CLOSED | Old duplicates expire and final ACK can be retransmitted |

## Orderly release

TCP provides a full-duplex byte stream, so each direction is closed separately. When Host A has no more data, it sends FIN. Host B acknowledges the FIN but may continue sending. When B finishes, it sends its own FIN, and A acknowledges it. Thus, the normal termination uses four segments, although the ACK and FIN may sometimes be combined.




## Purpose of TIME-WAIT

**1.** It allows the active closer to retransmit the final ACK if the peer retransmits its FIN.

2. It prevents delayed segments from an old connection from being accepted by a later connection using the same socket pair.
3. The waiting period is traditionally twice the assumed maximum segment lifetime, often described as 2 x MSL.

## Abrupt termination and reset

A segment with RST = 1 aborts a connection immediately. It is used when a segment arrives for a nonexistent connection, when an application requests an abort or when a serious protocol error occurs. Data not yet delivered may be lost. Therefore, FIN-based orderly release is preferred for normal operation.

## Worked example

Suppose the client chooses x = 12000 and the server chooses y = 45000. The client sends SYN Seq 12000. The server replies SYN-ACK Seq 45000, Ack 12001. The client sends ACK Seq 12001, Ack 45001. If the client then sends 600 bytes starting at Seq 12001, the next expected acknowledgment is 12601. These values show that SYN consumes one sequence number and acknowledgments name the next byte expected.

## Advantages

- Synchronizes both sequence spaces before data transfer.
- Rejects delayed duplicate connection requests.
- Negotiates capabilities and flow-control options.
- Supports independent half-closes and orderly data completion.
- TIME-WAIT protects future connections from old segments.

## Limitations

- Three-way setup introduces at least one round-trip delay before ordinary data transfer.
- Connection state consumes memory at both endpoints.
- Half-open SYN state can be exploited in denial-of-service attacks.
- Lost FIN or ACK segments extend connection cleanup.
- TIME-WAIT may temporarily retain many socket records on busy servers.

## Conclusion

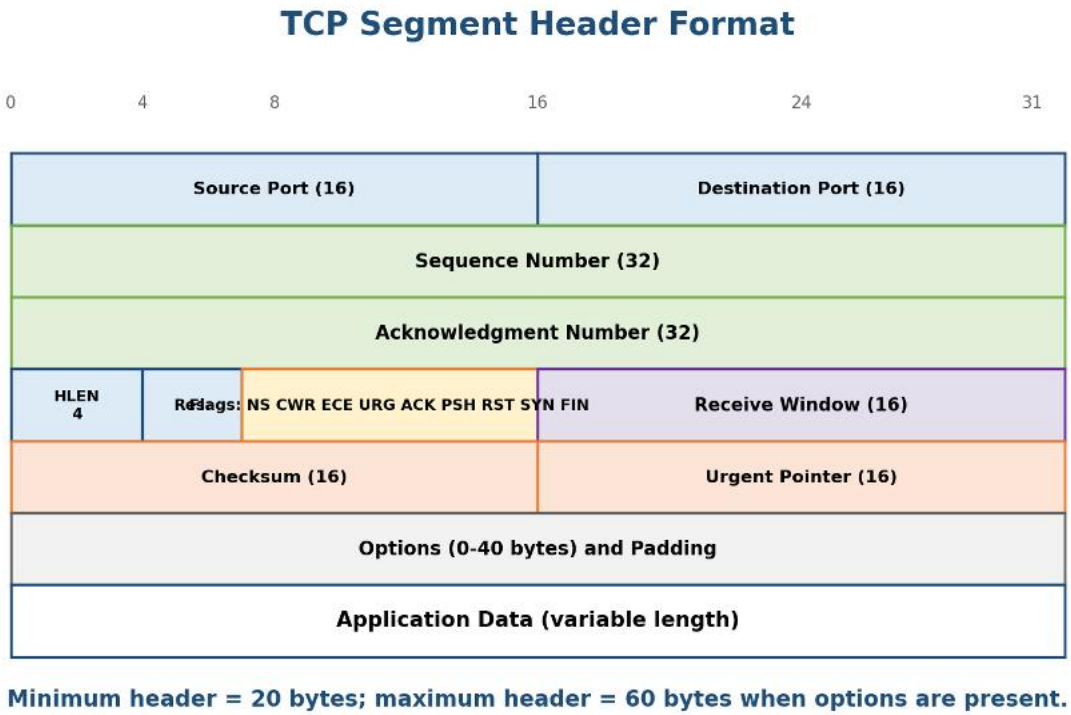
Connection management ensures that a TCP byte stream begins with synchronized, fresh state and ends without confusing old and new traffic. The three-way handshake, FIN exchange, retransmission timers and TIME-WAIT together handle loss, duplication, delay and full-duplex shutdown.

# Transmission Control Protocol (TCP)

Transmission Control Protocol is a connection-oriented, reliable, full-duplex, byte-stream transport protocol used by the Internet. TCP accepts a continuous stream of bytes from an application, groups them into segments, and delivers the same byte stream to the peer application in order. It runs over IP, which is best effort, so TCP itself must detect loss, duplication, corruption and reordering.

- Connection-oriented: a handshake establishes endpoint state before normal transfer.
- Reliable: missing bytes are retransmitted and duplicates are discarded.
- Byte-stream oriented: message boundaries are not preserved.
- Full duplex: both directions have independent sequence numbers and buffers.
- Flow controlled: receiver window prevents buffer overflow.
- Congestion controlled: congestion window limits load placed on the network.
- Point to point: one TCP connection has exactly two endpoints.

## TCP segment header

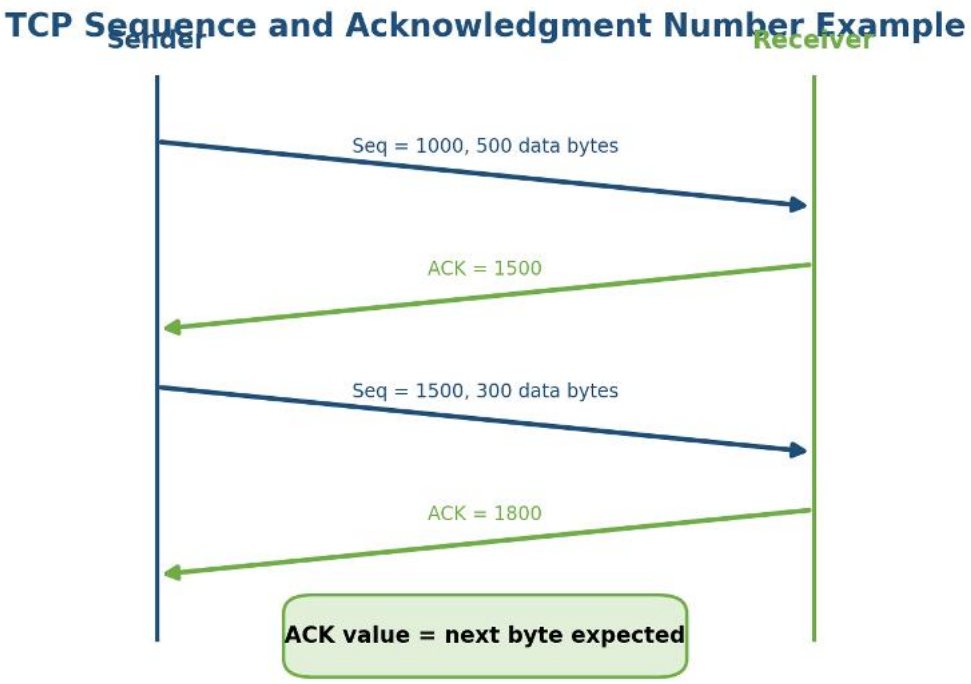



| Field | Size | Function |
|---|---|---|
| **Source and destination ports** | 16 bits each | Identify sending and receiving application endpoints |

| Field | Size | Function |
|---|---|---|
| **Sequence number** | 32 bits | Number of the first data byte in this segment; also carries the initial sequence number in a SYN |
| **Acknowledgment number** | 32 bits | Next byte expected from the peer when ACK is set |
| **Header length** | 4 bits | TCP header size in 32-bit words |
| **Flags** | Control bits | SYN, ACK, FIN, RST, PSH, URG and congestion-related controls |
| **Window** | 16 bits | Receiver-advertised available byte space |
| **Checksum** | 16 bits | Detects corruption in pseudoheader, TCP header and data |
| **Urgent pointer** | 16 bits | Marks end of urgent data when URG is set |
| **Options** | 0-40 bytes | MSS, window scale, timestamps, SACK and padding |

## Sequence numbers and acknowledgments

TCP numbers every byte. If a segment has sequence number S and carries L data bytes, it covers bytes S through S + L - 1. A cumulative acknowledgment A means that all bytes below A have been received in order and the receiver now expects byte A. SYN and FIN each consume one sequence number. Duplicate acknowledgments indicate that a gap may exist.



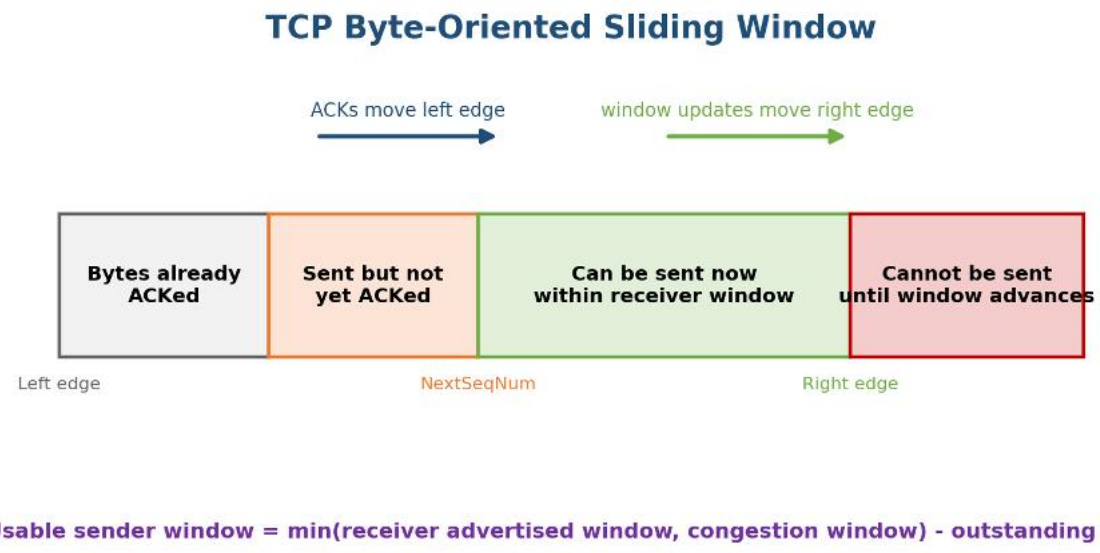

## Error-control mechanisms

**1. Checksum:** The receiver verifies the TCP checksum and discards a corrupted segment.

2. **Retransmission timeout:** The sender estimates round-trip time and retransmits data whose timer expires.
3. **Fast retransmit:** Three duplicate acknowledgments normally trigger retransmission before the timer expires.
4. **Duplicate detection:** Sequence numbers allow the receiver to discard repeated bytes.
5. **Reordering:** Segments that arrive out of order may be buffered until the missing range arrives.
6. **Selective acknowledgment:** When negotiated, the receiver can report non-contiguous blocks already received, reducing unnecessary retransmission.

## Flow control

The receiver advertises a window rwnd containing the number of additional bytes it can currently accept beyond the acknowledgment number. The sender may have at most that amount of unacknowledged data in flight, subject also to congestion control. A zero window stops ordinary data but the sender periodically transmits a small probe so that a lost window update does not deadlock the connection.



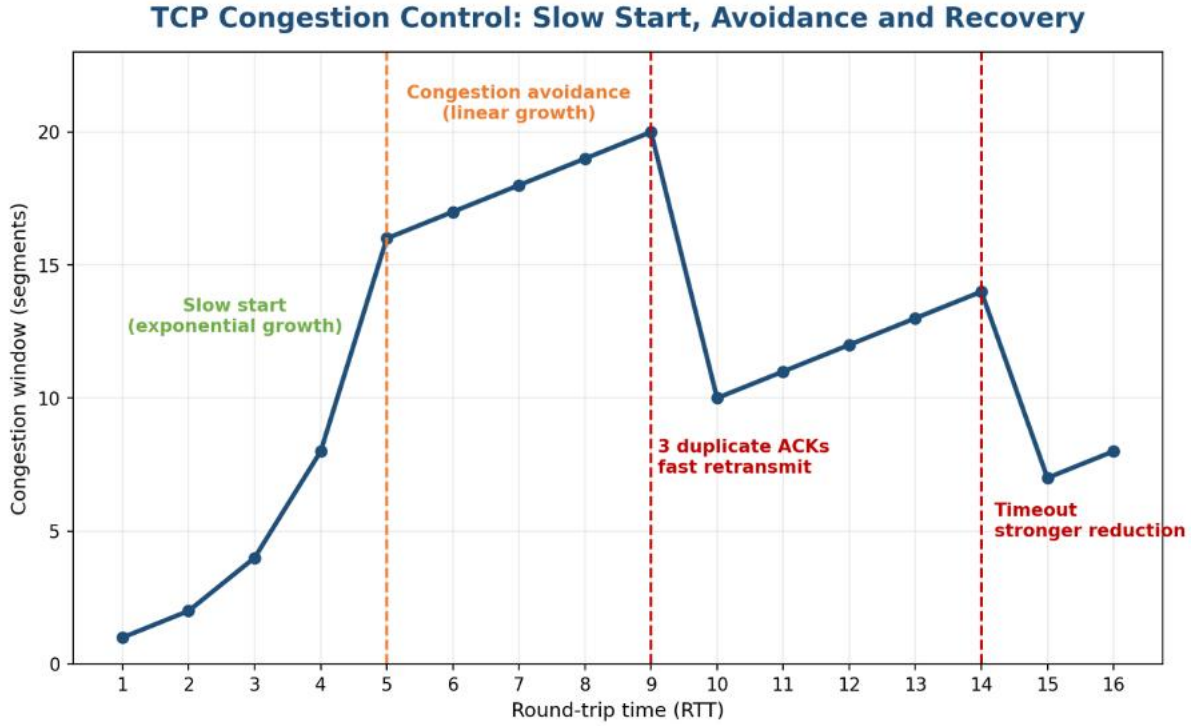

**Usable sender window = min(receiver advertised window, congestion window) - outstanding data**

Effective sending limit = minimum of receiver window (rwnd) and congestion window (cwnd). Receiver flow control protects the destination host; congestion control protects the network.

## Congestion control

1. **Slow start:** cwnd begins small and increases approximately exponentially, commonly by about one maximum segment per acknowledgment, until a threshold or congestion event is reached.
2. **Congestion avoidance:** Above the slow-start threshold, cwnd grows approximately linearly to probe for additional capacity.
3. **Fast retransmit:** Three duplicate acknowledgments indicate likely loss, so the missing segment is retransmitted immediately.

4. **Fast recovery:** After duplicate-ACK loss, the sender reduces its window but may avoid returning to the smallest initial rate.
5. **Timeout response:** A timeout is treated as stronger congestion; the threshold is reduced and slow start restarts from a small window.
6. **Modern refinements:** Implementations may use advanced congestion controllers, pacing, explicit congestion notification and improved loss recovery while preserving network fairness principles.




## Worked flow-control example

The receiver acknowledges byte 10,000 and advertises rwnd = 6,000 bytes. The sender congestion window is cwnd = 4,000 bytes and 1,500 bytes are already outstanding. The maximum permitted flight is min(6,000, 4,000) = 4,000 bytes. Therefore, the sender can transmit 4,000 - 1,500 = 2,500 additional bytes. Even though the receiver has more buffer space, network congestion control is the tighter limit.

## Advantages

- Reliable and ordered byte-stream delivery.
- Automatic adaptation to receiver speed and network congestion.
- Efficient use of bandwidth through sliding windows and cumulative acknowledgments.
- Widely implemented and supported through the socket interface.
- Suitable for applications in which correctness is more important than occasional delay.

## Limitations

- Connection setup and retransmission add latency.
- Head-of-line blocking delays later bytes behind a missing byte range.
- Message boundaries are not preserved; applications must define record framing.

- Per-connection state and buffers consume resources.
- Ordinary TCP is not ideal for loss-tolerant real-time media because retransmission may arrive too late.

## Applications

- HTTP/HTTPS web access
- Email transfer and access
- File transfer and secure shell
- Database sessions
- Remote administration
- Reliable cloud and enterprise APIs

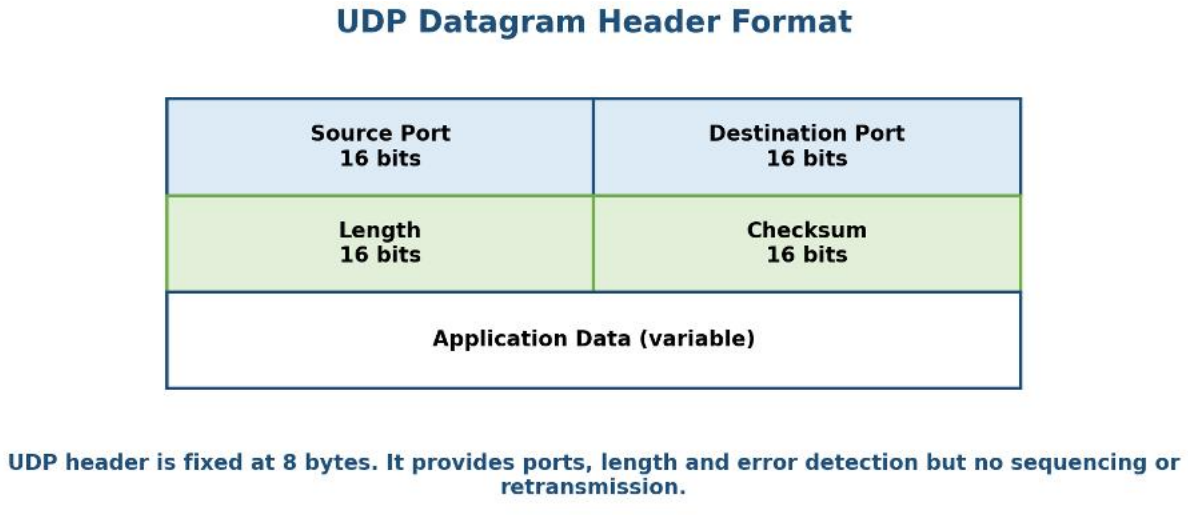
## Conclusion

TCP builds a reliable, ordered and congestion-aware byte stream on top of unreliable IP delivery. Its header, sequence space, acknowledgments, sliding windows, retransmission logic and congestion algorithms cooperate to provide dependable Internet communication.

# User Datagram Protocol (UDP)

User Datagram Protocol is a connectionless, message-oriented transport protocol. It provides process-to-process delivery using port numbers and detects corruption using a checksum, but it does not establish a connection, guarantee delivery, retransmit lost datagrams, preserve arrival order or perform built-in flow and congestion control. Each application message is normally carried as one UDP datagram, subject to the IP packet-size limits.

## UDP header




UDP header is fixed at 8 bytes. It provides ports, length and error detection but no sequencing or retransmission.

| Field | Size | Function |
|---|---|---|
| **Source port** | 16 bits | Identifies the sending process; may be zero in limited cases where no reply is expected |
| **Destination port** | 16 bits | Identifies the receiving process |

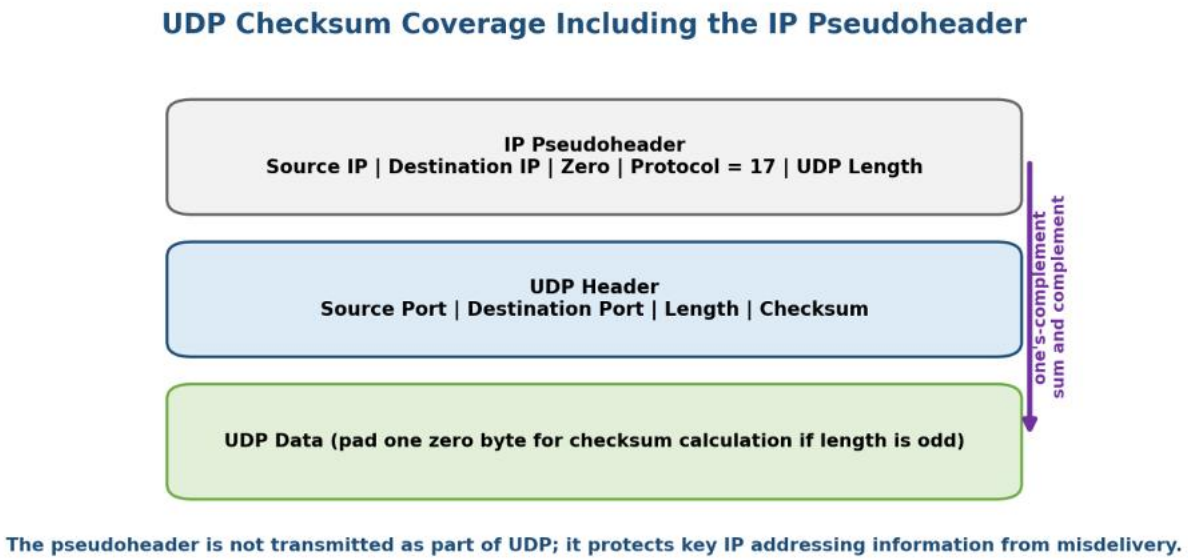
| Field | Size | Function |
|---|---|---|
| **Length** | 16 bits | Total UDP header plus data length; minimum value is 8 |
| **Checksum** | 16 bits | Detects corruption in the UDP datagram and selected IP-layer fields |
| **Data** | Variable | One application message or part defined by the application protocol |

## Operation

**1.** The sender application gives a message and destination socket address to UDP.
**2.** UDP adds source port, destination port, total length and checksum.
**3.** The datagram is passed to IP and may be fragmented by the network layer if it is too large.
**4.** At the receiver, IP passes the datagram to UDP.
**5.** UDP verifies the checksum and uses the destination port to select a socket.
**6.** If no process is bound to the destination port, the host may return an ICMP Port Unreachable message.
**7.** The application receives the complete datagram; UDP does not combine it with neighbouring datagrams.

## UDP checksum

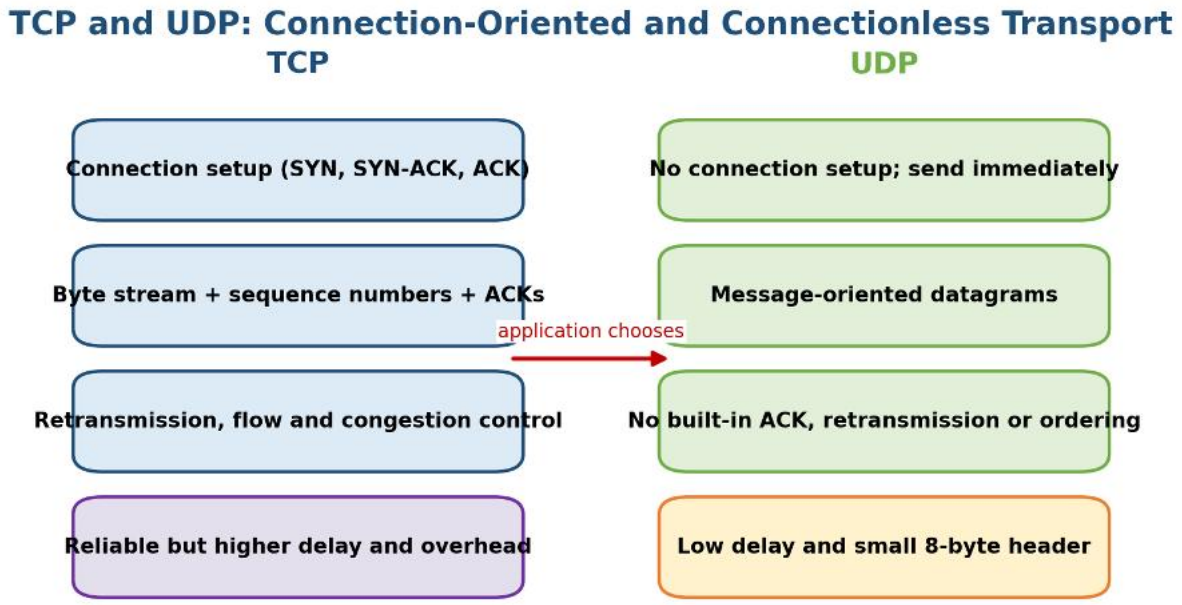
The UDP checksum is computed using the one's-complement sum of a pseudoheader, the UDP header and the data. The pseudoheader includes source and destination IP addresses, the protocol number and UDP length. It is included only in checksum calculation and is not delivered as part of the UDP datagram. Including these fields helps detect a datagram delivered to the wrong host or protocol.



## Numerical example

An application sends 1,200 bytes through UDP. The UDP header adds 8 bytes, so the UDP length field is 1,208. With IPv4 and no IP options, the IP header adds 20 bytes, producing a 1,228-byte IP packet. If the path MTU is 1,500 bytes, no IP fragmentation is needed. If the application instead sends 3,000 bytes, the resulting IP packet is larger than the MTU and fragmentation or application-level packetization becomes necessary.

## TCP and UDP comparison




| Feature | TCP | UDP |
|---|---|---|
| **Service** | Connection-oriented byte stream | Connectionless datagrams |
| **Header size** | 20-60 bytes | 8 bytes |
| **Reliability** | ACKs, timers and retransmission | No built-in reliability |
| **Ordering** | Ordered bytes | No order guarantee |
| **Flow control** | Receiver window | Not provided |
| **Congestion control** | Provided by TCP algorithms | Application must behave responsibly |
| **Message boundaries** | Not preserved | Preserved per datagram |
| **Broadcast/multicast** | Not supported by standard TCP | Can be used with IP multicast/broadcast where permitted |

| Feature | TCP | UDP |
|---|---|---|
| **Typical use** | Web, email, files, databases | DNS, voice, video, games, telemetry |

## Advantages

- Very small fixed header and no handshake delay.
- Preserves message boundaries.
- Supports one-to-many communication through multicast where the network permits it.
- Applications can implement only the reliability or timing mechanisms they need.
- Suitable for short request-response exchanges and real-time traffic.

## Limitations

- Datagrams can be lost, duplicated, corrupted or reordered.
- No automatic rate adaptation to receiver capacity.
- No built-in congestion control; careless applications can overload the network.
- Large datagrams may be fragmented, and loss of one fragment loses the entire datagram.
- Applications must implement acknowledgments, sequencing or security when required.

## Applications

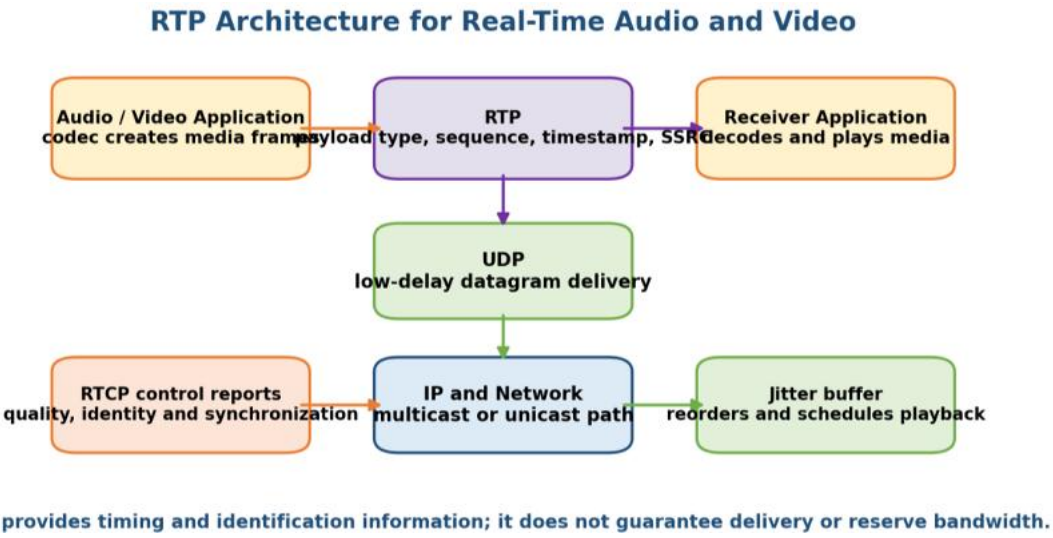
- Domain Name System queries
- DHCP
- Voice over IP and RTP media
- Online games
- Sensor telemetry
- Network management
- Streaming protocols with application-level recovery

## Conclusion

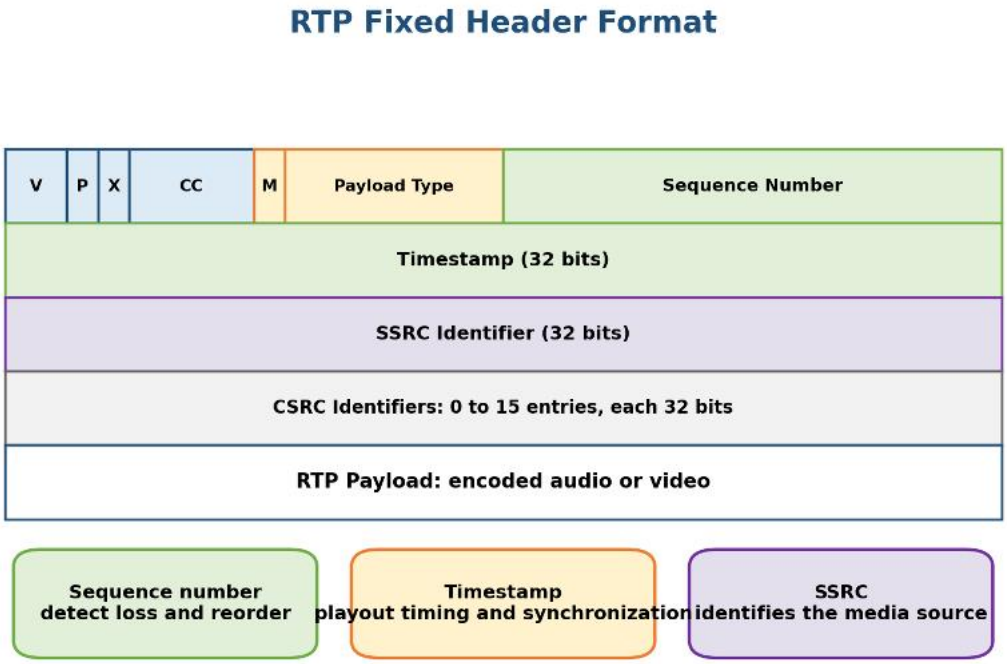
UDP offers the minimum transport functions needed for process addressing and datagram integrity. Its low overhead and immediate transmission make it valuable for short transactions and real-time applications, provided the application accepts loss or adds its own recovery and congestion behavior.

# Real-time Transport Protocol (RTP)

Real-time Transport Protocol carries real-time media such as encoded audio and video over packet networks. RTP is commonly transported in UDP datagrams and is used for both unicast and multicast sessions. It adds information needed to identify the media format and source, detect missing or out-of-order packets, reconstruct timing and synchronize playback. RTP itself does not guarantee delivery, retransmit lost packets, reserve bandwidth or prevent congestion.




## RTP header fields




| Field | Purpose |
|---|---|
| **Version (V)** | Identifies the RTP format version; the widely used version is 2 |
| **Padding (P)** | Indicates padding bytes at the end of the packet |
| **Extension (X)** | Indicates an additional profile-defined header extension |

| Field | Purpose |
|---|---|
| **CSRC count (CC)** | Number of contributing-source identifiers following the fixed header |
| **Marker (M)** | Profile-specific event marker, such as a frame boundary |
| **Payload type (PT)** | Identifies the encoding or media format according to the session profile |
| **Sequence number** | Increments by one for each RTP packet; detects loss and restores order |
| **Timestamp** | Represents the sampling instant of the first payload unit and controls playout timing |
| **SSRC** | Synchronisation source identifier for the packet stream |
| **CSRC list** | Identifies original contributing sources when a mixer combines streams |

## Working of RTP

**1.** The application captures audio or video and encodes it with a codec.
**2.** Media data is divided into packets according to the selected RTP payload format.
**3.** RTP adds payload type, sequence number, timestamp and source identifier.
**4.** UDP and IP carry the RTP packet across the network.
**5.** The receiver checks sequence numbers to detect missing and out-of-order packets.
**6.** The receiver uses timestamps and a local clock to schedule playout.
**7.** A jitter buffer temporarily stores packets, reorders them and absorbs delay variation.
**8.** The decoder reconstructs the media; lost packets may be concealed, ignored or recovered by application-specific methods.

## Sequence number and timestamp

The sequence number is a packet counter and normally increments once per RTP packet. It does not represent time. The timestamp represents the media sampling timeline and increases according to the media clock rate. Several packets belonging to one video frame can have the same timestamp, while consecutive audio packets have timestamps separated by the number of audio samples carried.

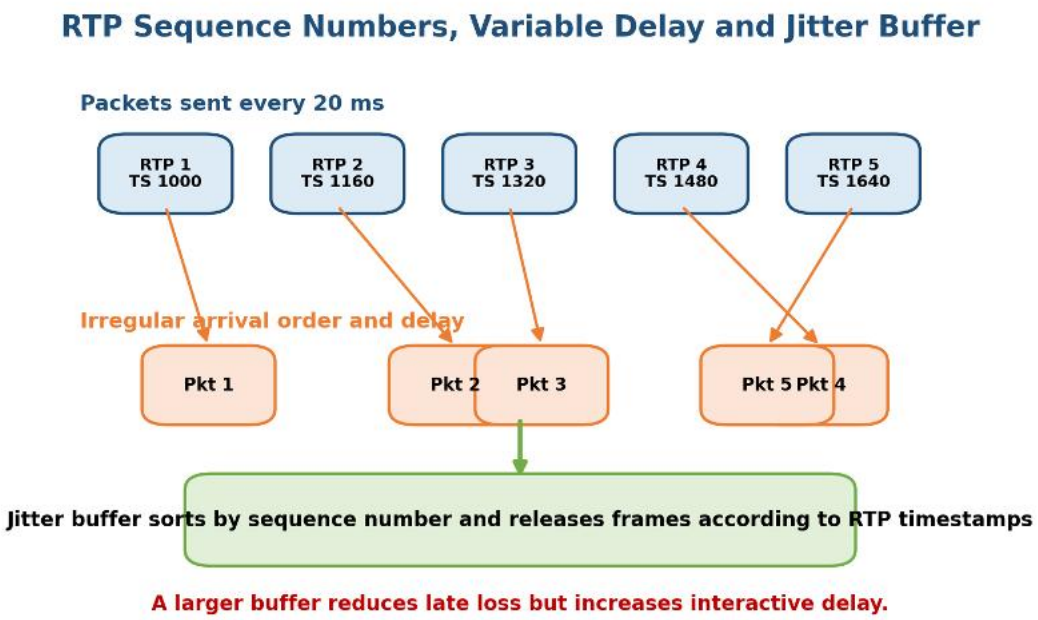



*Figure: RTP packet delay variation and jitter-buffer operation*

## Worked audio example

Consider 8-kHz audio packetized into 20-ms blocks. Each packet contains 8,000 x 0.020 = 160 samples. If the first packet has sequence number 4200 and timestamp 10000, the next packets normally use sequence numbers 4201, 4202 and 4203, with timestamps 10160, 10320 and 10480. If packet 4202 arrives after 4203, the sequence numbers reveal the reordering. The receiver uses timestamps to play blocks at 20-ms intervals after an initial buffering delay.

## Mixers and translators

An RTP mixer combines several input streams, for example multiple conference speakers, and creates a new synchronized stream. The mixer uses its own SSRC and may list the original sources as CSRC values. A translator forwards or converts media without combining sources; it may change encoding, packet size or network addressing while preserving the logical session.

## Advantages

- Provides standard packet sequence and media timing information.
- Supports many audio and video encodings through payload profiles.
- Works with both unicast and multicast delivery.
- Identifies sources and supports mixers in conferences.
- Allows receivers to build jitter buffers and synchronize playback.

## Limitations

- Does not guarantee packet delivery, order or bandwidth.
- Does not perform congestion control by itself.
- Security requires additional protection such as Secure RTP and key management.
- Profile and payload-format agreement is required between endpoints.

- Jitter buffering trades lower late-packet loss for higher end-to-end delay.

## Applications

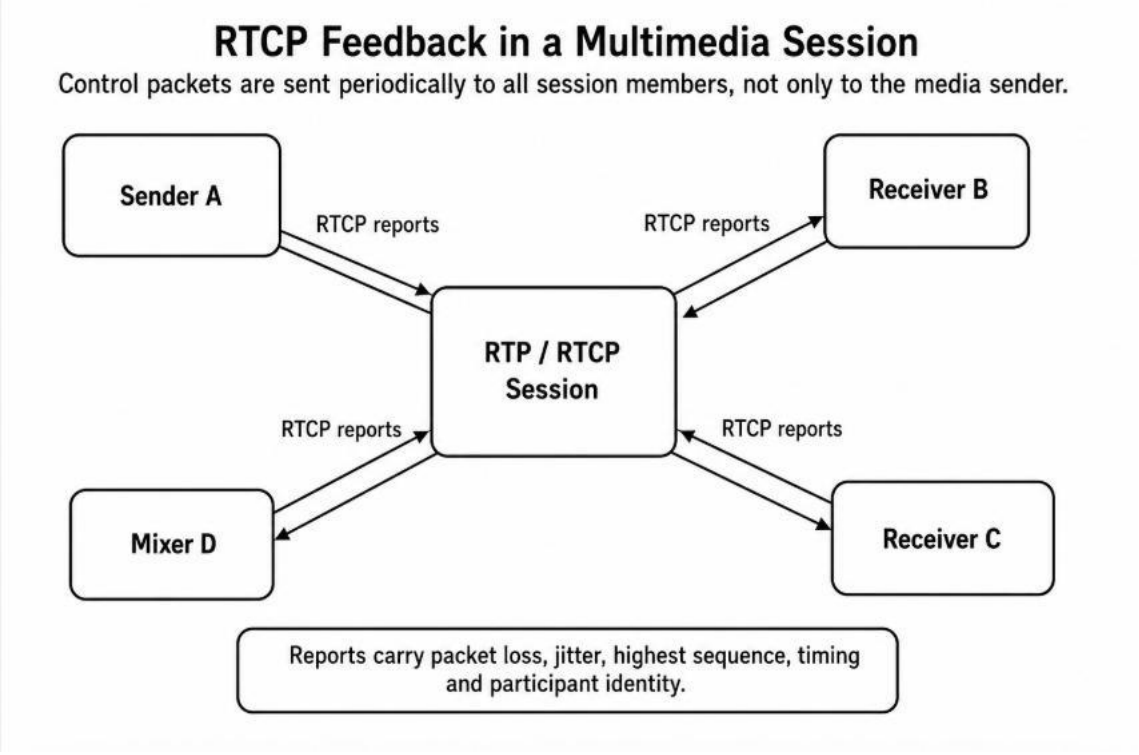
- Voice over IP
- Video conferencing
- Interactive distance learning
- IP television and live streaming
- Telemedicine media
- Multiplayer voice chat
- Web real-time communication media paths

## Conclusion

RTP adds the sequence, timing and source information that real-time media needs but ordinary UDP lacks. By combining RTP headers with codecs, jitter buffers and RTCP feedback, applications can deliver interactive audio and video despite variable packet delay and occasional loss.

# Real-time Transport Control Protocol (RTCP)

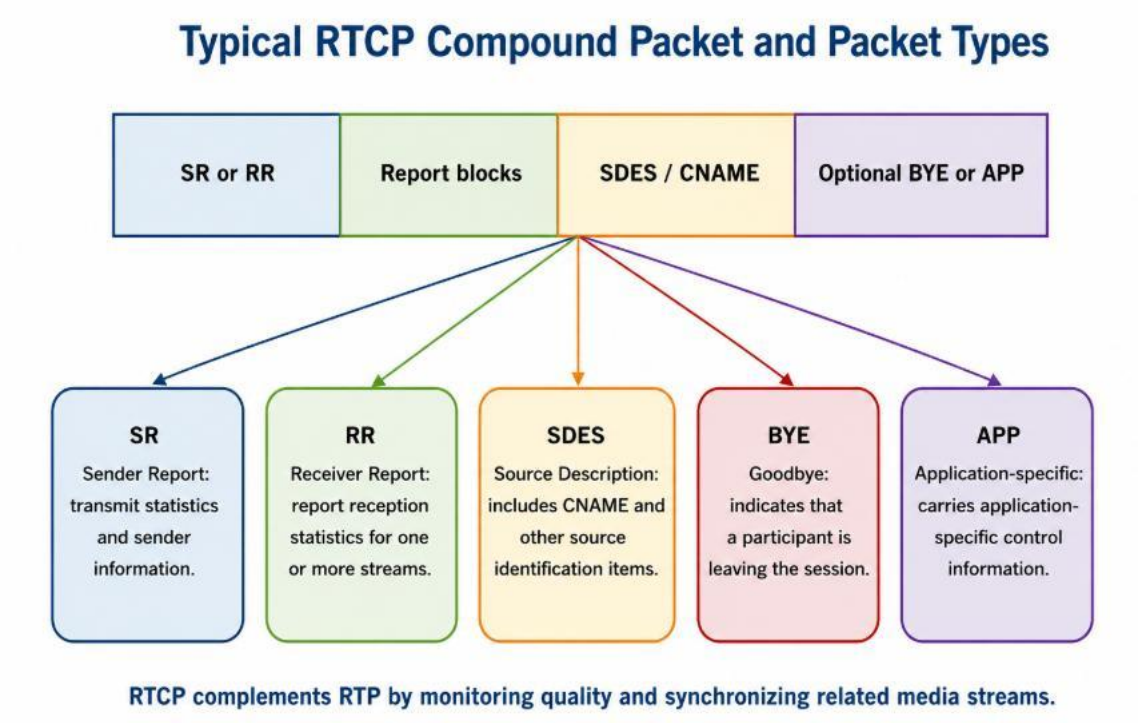
Real-time Transport Control Protocol is the control companion of RTP. RTP carries media packets, while RTCP periodically exchanges information about reception quality, sender timing, participant identity and session membership. RTCP packets are normally sent to all members of the RTP session using a control port separate from the media port. RTCP does not carry ordinary audio or video payload.



## Main functions

1. **Quality feedback:** Reports contain packet loss, extended highest sequence number, interarrival jitter and timing information. A sender or conference controller can use these values to adapt encoding or diagnose a path.
2. **Source identification:** Source Description packets carry a canonical name and optional identity information so that media streams can be associated with participants.
3. **Inter-media synchronization:** Sender Reports map an RTP timestamp to an absolute wall-clock timestamp, allowing audio and video streams with different RTP clock rates to be synchronized.
4. **Session participation:** Periodic reports indicate that a source is active. BYE packets announce that a participant is leaving.
5. **Scalable control traffic:** The reporting interval increases as the group grows so that RTCP consumes only a small controlled fraction of session bandwidth.
6. **Application-specific control:** APP packets allow experiment or application-defined information when permitted by the session profile.

## RTCP packet types




| Type | Name | Main contents and use |
|---|---|---|
| **SR** | Sender Report | RTP/NTP time mapping, sender packet count, octet count and reception report blocks |

| Type | Name | Main contents and use |
|---|---|---|
| **RR** | Receiver Report | Reception report blocks from a participant that has not recently sent RTP media |
| **SDES** | Source Description | CNAME and optional participant descriptive items |
| **BYE** | Goodbye | Indicates that one or more sources are leaving the session |
| **APP** | Application-defined | Control data specific to an application or experiment |
| **Feedback extensions** | Early/extended feedback | Profile-dependent loss, codec or congestion feedback in modern real-time systems |

## Sender Report fields

- **SSRC of sender:** identifies the RTP source described by the report.
- **NTP timestamp:** absolute wall-clock time at which the report is generated.
- **RTP timestamp:** media-clock value corresponding to the same instant as the NTP timestamp.
- **Sender packet count:** total RTP data packets sent by the source.
- **Sender octet count:** total RTP payload octets sent.
- **Reception report blocks:** statistics describing other sources heard by the reporter.

## Reception report information

- **Fraction lost:** recent packet-loss fraction during the reporting interval.
- **Cumulative packets lost:** long-term difference between expected and received packet counts.
- **Extended highest sequence number:** highest received sequence number together with wrap count.
- **Interarrival jitter:** statistical estimate of variation in packet transit time.
- **Last SR and delay since last SR:** values used to estimate round-trip time between participants.

## Round-trip time example

Receiver B obtains a Sender Report from A. Later B sends a Receiver Report containing the middle 32 bits of A's NTP timestamp as LSR and the delay since receiving that report as DLSR. When A receives the Receiver Report at local time A_now, it estimates round-trip time as A_now - LSR - DLSR, with all values interpreted in the defined timestamp units. The subtraction removes the time B intentionally held the report before responding.

## Audio-video synchronization example

A conference endpoint sends audio with an 8-kHz RTP clock and video with a 90-kHz RTP clock. The RTP timestamps cannot be compared directly. Sender Reports for both streams provide pairs of values: one NTP time and one RTP timestamp. The receiver maps both media timelines to the same wall clock and can present the corresponding audio sample and video frame together, maintaining lip synchronization.

## RTCP bandwidth control

Every participant should not transmit frequent control packets in a large multicast group because control traffic would grow with group size. RTCP therefore calculates a randomized reporting interval from the number of participants, the average RTCP packet size, session bandwidth and whether the participant is an active sender. The design traditionally allocates only a small fraction of session bandwidth to RTCP and divides part of it among active senders.

## Advantages

- Makes packet loss and jitter visible to endpoints and administrators.
- Enables audio-video synchronization across independent RTP streams.
- Identifies participants and active sources.
- Supports adaptive codec, rate and error-protection decisions.
- Scales report frequency according to session population.

## Limitations

- Feedback is periodic and may not react instantly to sudden congestion.
- Reports consume additional bandwidth and processing.
- RTCP observes quality but does not itself retransmit media or reserve network resources.
- Large sessions require careful interval calculation and feedback suppression.
- Reports can reveal participant and network information unless protected by secure real-time protocols.

## Applications

- Conference quality monitoring
- Audio-video synchronization
- Adaptive streaming and codecs
- Participant identification
- Troubleshooting voice and video networks
- Real-time session analytics

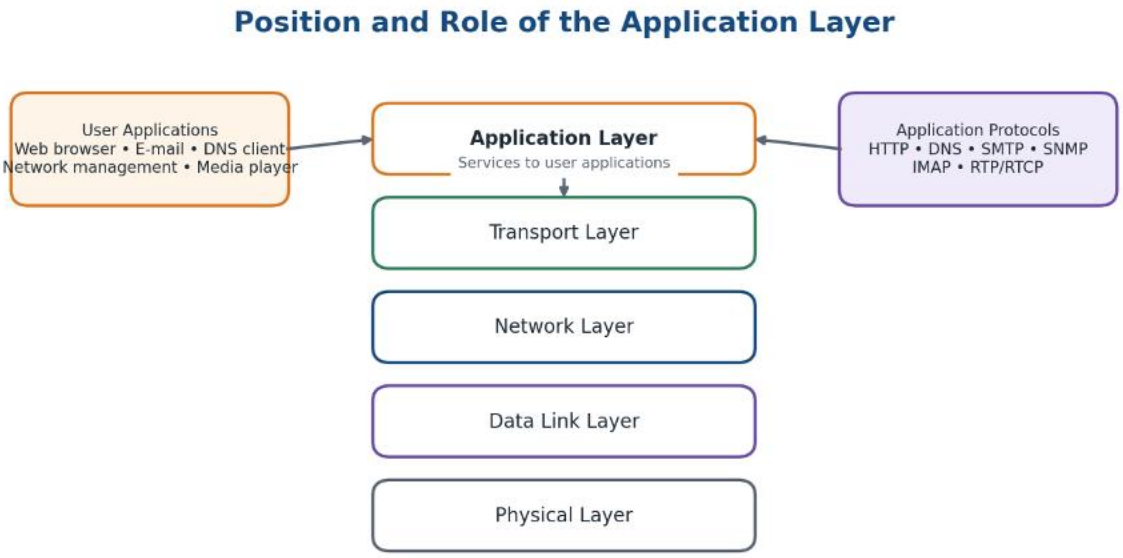
## Conclusion

RTCP completes the RTP framework by supplying control, quality and synchronization information. Sender and receiver reports, source descriptions, membership notices and bandwidth-scaled reporting allow multimedia applications to understand and adapt to real network conditions.

# UNIT 5 — THE APPLICATION LAYER

## Application Layer:
## Services, Processes and Architectures

The application layer is the highest layer of the TCP/IP protocol suite and the layer closest to the end user. It provides communication services directly to network applications such as web browsers, e-mail clients, name-resolution software, network-management tools and media players. It does not mean that the entire application is part of the network protocol; rather, the application-layer protocol defines how processes on different hosts exchange messages.




### Major Responsibilities

- Identifying the communicating processes and the remote service required.
- Defining message types, message syntax, field meanings and the order of exchange.
- Providing services such as name resolution, web access, file transfer, e-mail and network management.
- Choosing a suitable transport service: reliable TCP, low-overhead UDP, or a secure transport such as TLS over TCP.
- Supporting authentication, representation, session state and application-specific error reporting when needed.

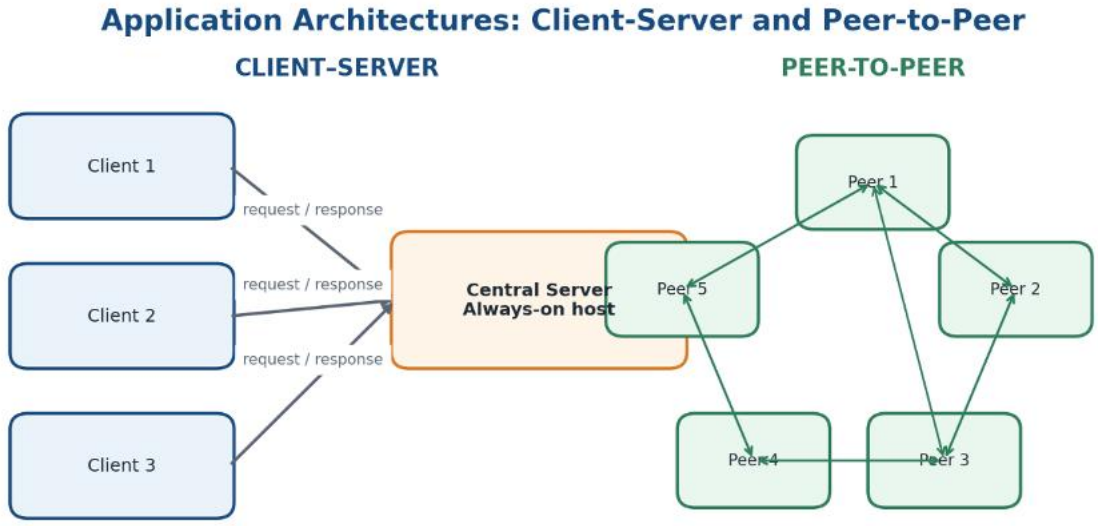
### Application Processes and Sockets

A process is a running program. Two application processes communicate through sockets. A socket is the software interface between an application process and the transport layer. A

server process normally listens at a well-known port, while a client process uses a temporary port. The pair consisting of IP address and port number identifies a transport endpoint.

**Example:** When a browser connects to https://www.example.com, DNS first supplies the server IP address. The browser then opens a transport connection from a temporary client port to the server port, commonly TCP port 443 for HTTPS. HTTP messages are exchanged through the socket.

## Application Architectures




| Feature | Client–Server | Peer-to-Peer |
|---|---|---|
| Control | Central server controls data or service | Peers share responsibility |
| Availability | Server is usually always on | Peers may join and leave |
| Scalability | Server can become a bottleneck | Capacity can grow with peers |
| Management | Easier to secure and administer | More difficult to coordinate |
| Examples | Web, e-mail, DNS, cloud applications | File sharing, some real-time systems |

## Common Application-Layer Protocols

| Protocol | Main Purpose | Common Transport / Port |
|---|---|---|
| DNS | Maps names to addresses and other records | UDP or TCP, port 53 |
| HTTP / HTTPS | Transfers web resources and API data | TCP 80 / TCP 443; modern HTTP may use QUIC |
| SMTP | Transfers electronic mail | TCP 25, 587 or 465 depending on role |
| POP3 / IMAP | Retrieves and synchronizes mail | TCP 110 / 143; secure variants 995 / 993 |
| SNMP | Monitors and controls network devices | UDP 161 and traps to 162 |
| RTP / RTCP | Carries and monitors real-time media | Usually over UDP |

## Advantages

- Provides standardized communication between heterogeneous applications.
- Allows application developers to use transport and network services without handling physical transmission details.
- Supports modular design: protocols can evolve independently of the lower layers.
- Enables a wide variety of internet services through well-defined message formats.

## Limitations and Design Challenges

- Different applications require different combinations of reliability, delay, bandwidth and security.
- Application protocols may expose privacy or authentication risks when not secured.
- State management, scalability and compatibility become complex in distributed systems.
- Application performance still depends on lower-layer congestion, delay and packet loss.

## Applications

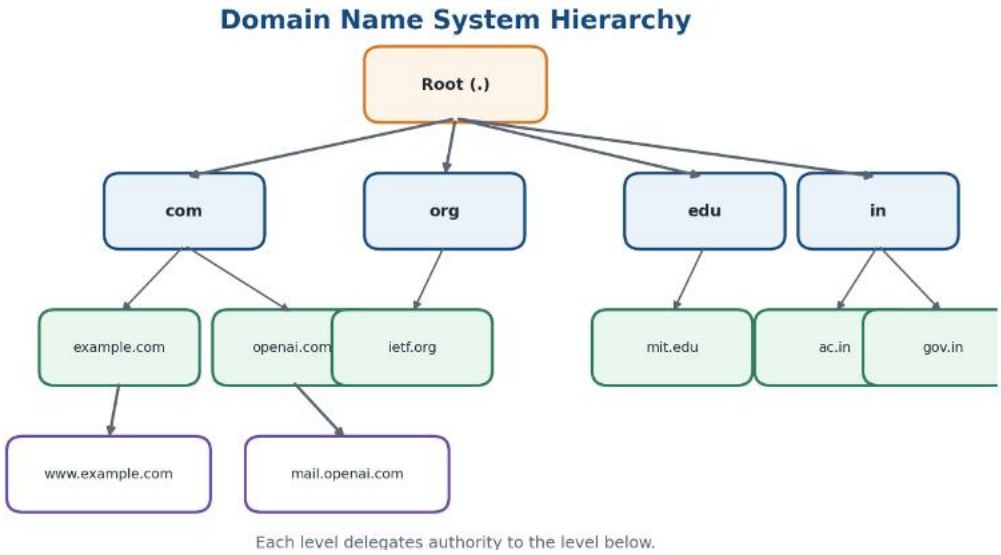
- Web and cloud services
- Electronic mail and collaboration
- Domain-name resolution
- Network monitoring
- Streaming and real-time communication
- Remote administration and file transfer

## Conclusion

The application layer converts user requirements into structured network communication. Its protocols specify what messages mean and how applications cooperate, while lower layers provide end-to-end transport and packet delivery.

# Domain Name System (DNS)

The Domain Name System is a distributed and hierarchical naming system that translates human-readable domain names into information required by network applications, most importantly IP addresses. Users prefer names such as www.college.edu, whereas routers deliver packets using numerical IP addresses. DNS also stores mail-server, alias, service and administrative information.




## DNS Namespace and Zones

- The root is represented by a dot and is the highest point in the namespace.
- Top-level domains include generic domains such as .com, .org and .edu and country-code domains such as .in.
- A domain is a subtree of the namespace. A zone is the portion of that namespace administered by a particular authority.
- Delegation allows a parent zone to transfer responsibility for a child domain to another set of authoritative name servers.

## DNS Components

| Component | Function |
|---|---|
| Stub resolver | Client-side library that sends queries to a configured resolver |
| Recursive resolver | Finds the complete answer and caches it for clients |
| Root name server | Directs a resolver toward the correct top-level-domain servers |
| TLD name server | Directs the resolver toward authoritative servers for a domain |

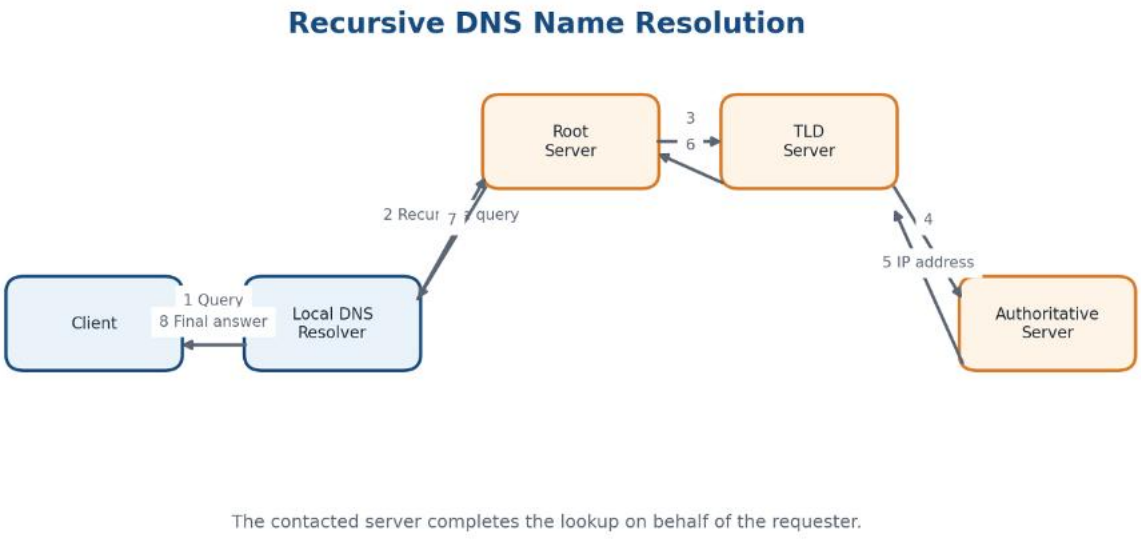
| Component | Function |
|---|---|
| Authoritative server | Stores definitive records for its zone |

## Resource Records

DNS information is stored as resource records. A record has the conceptual fields Name, Time to Live, Class, Type and Value. The TTL controls how long a resolver may cache the record.

| Record Type | Purpose | Example Value |
|---|---|---|
| A | Maps a name to an IPv4 address | 192.0.2.10 |
| AAAA | Maps a name to an IPv6 address | 2001:db8::10 |
| NS | Identifies an authoritative name server | ns1.example.com |
| CNAME | Creates an alias for a canonical name | www → server1.example.com |
| MX | Identifies mail exchangers and priority | 10 mail.example.com |
| PTR | Supports reverse lookup | Address → host name |
| TXT | Carries verification or policy text | SPF or ownership data |
| SOA | Contains zone authority and timing data | Primary server, serial, refresh values |

## Recursive Resolution



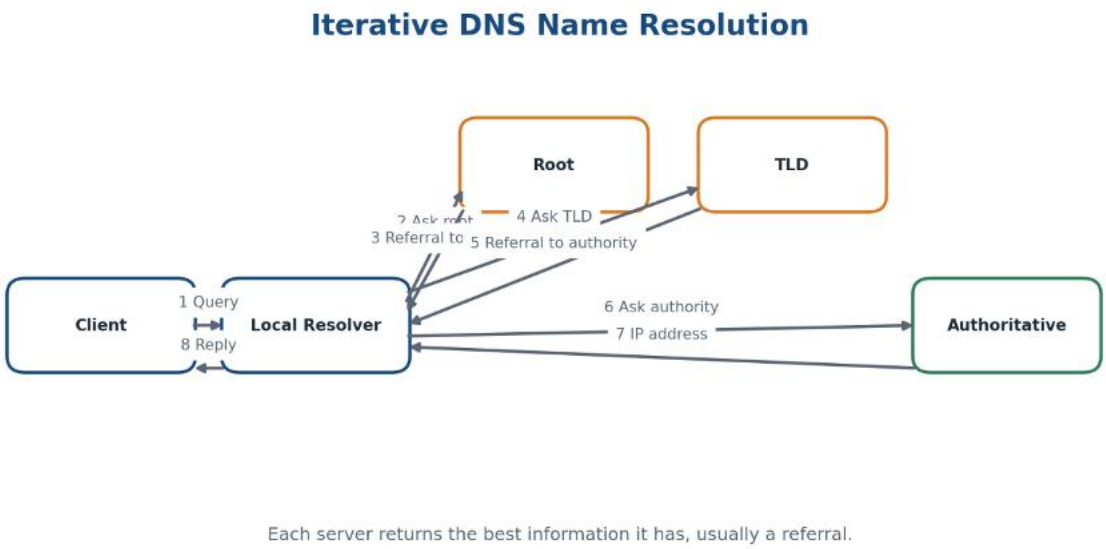

**1.** The client sends a query to its local recursive resolver.

**2.** The resolver checks its cache. If no valid answer exists, it contacts the DNS hierarchy.

**3.** The required servers cooperate until the authoritative answer is found.

**4.** The resolver returns the answer to the client and stores it for the duration of the TTL.

## Iterative Resolution




In iterative resolution, each name server returns either the answer or a referral to a server closer to the authoritative source. Recursive resolvers commonly use iterative queries while resolving names on behalf of clients.

## Worked Example: Resolving www.example.com

**1.** The browser asks the local resolver for www.example.com.

**2.** The resolver asks a root server and receives a referral to .com TLD servers.

**3.** It asks a .com server and receives the authoritative server for example.com.

**4.** It asks that authoritative server and receives the A or AAAA record.

**5.** The resolver caches the record and returns the IP address to the browser.

## DNS Caching and Reliability

Caching reduces lookup delay and decreases load on authoritative servers. DNS is replicated across multiple servers for fault tolerance. UDP is commonly used for ordinary queries because of its low overhead, while TCP is used for large responses, zone transfers and situations requiring reliable transfer.

## Advantages

- Human-friendly names hide changing numerical addresses.
- Distributed administration avoids a single global database bottleneck.
- Caching improves performance and scalability.
- Multiple record types support web, e-mail and service discovery.
- Replication provides high availability.

## Limitations and Security Issues

- Cached data can become temporarily stale until TTL expiry.
- DNS spoofing and cache poisoning can redirect users.
- Distributed denial-of-service attacks can target resolvers or authoritative servers.
- Misconfigured delegation and records can make services unreachable.
- DNSSEC can provide authenticity and integrity, but deployment and key management add complexity.

## Applications

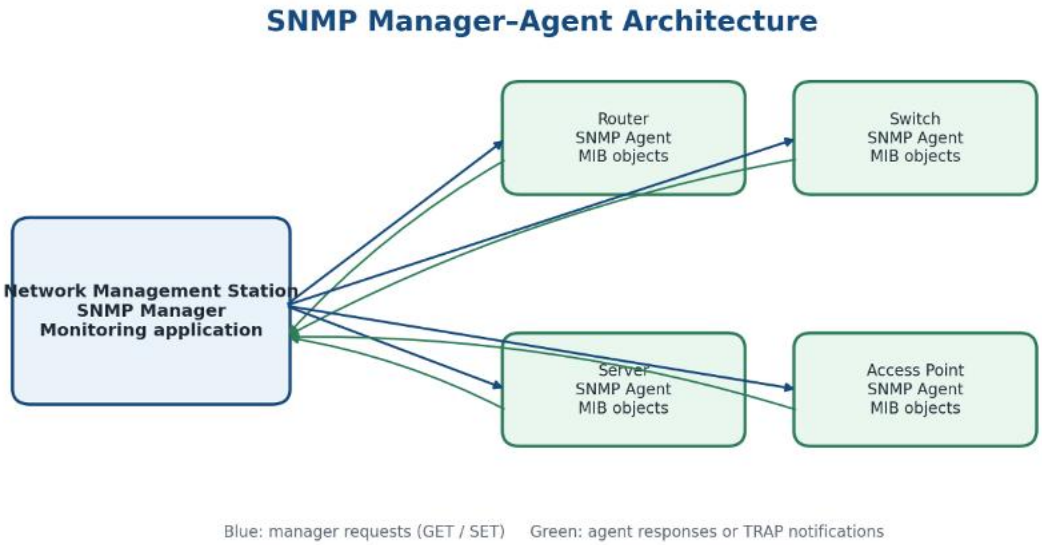
- Website access
- Electronic-mail routing through MX records
- Reverse address lookup
- Load distribution using multiple address records
- Service discovery and domain verification

## Conclusion

DNS is the naming foundation of the internet. Its hierarchy, delegation, replication and caching allow billions of names to be resolved efficiently without relying on one central server.

# Simple Network Management Protocol (SNMP)

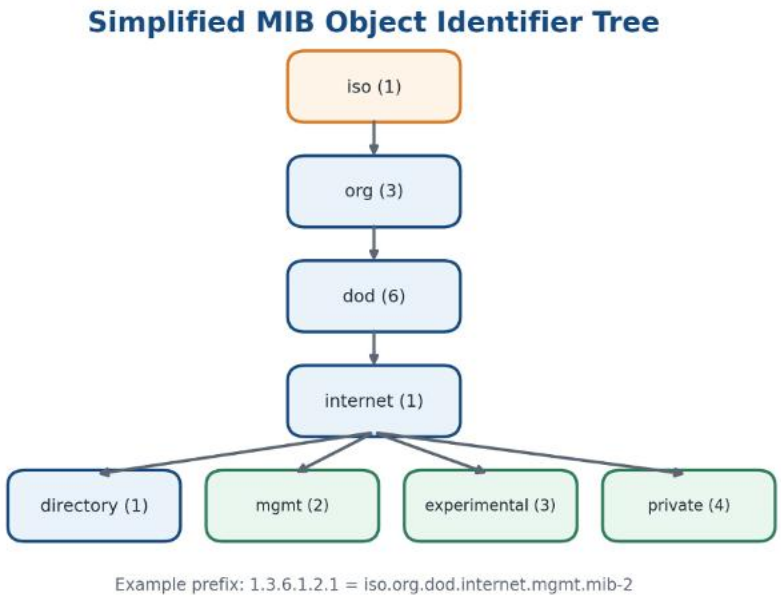
SNMP is an application-layer protocol used to monitor and manage network devices. It allows a management station to read operational variables, change selected configuration variables and receive event notifications from routers, switches, servers, printers, wireless access points and other managed devices.



## SNMP Components

| Component | Description |
|---|---|
| Manager / Network Management Station | Runs monitoring software, sends requests, stores statistics and displays alarms |
| Agent | Software on a managed device that provides access to management variables |
| Managed device | Router, switch, server, printer, sensor or other network element |
| MIB | Logical database of managed objects identified by object identifiers |
| SMI | Rules for naming, typing and encoding management information |

## Management Information Base and Object Identifiers




Every managed object has an object identifier, or OID, positioned in a global tree. For example, an interface counter may record the number of octets received on a port. A manager requests the OID, and the agent returns the current value.

## SNMP Operations

| Operation | Purpose |
|---|---|
| GET | Read the value of one or more objects |
| GET-NEXT | Read the next object in MIB order |
| GET-BULK | Efficiently retrieve a block of variables |
| SET | Change a writable variable |
| RESPONSE | Return requested values or an error status |
| TRAP | Agent sends an unsolicited event notification |

| Operation | Purpose |
|---|---|
| INFORM | Acknowledged notification, generally between management entities |

## Working Example: Monitoring a Router Interface

1. The manager periodically sends GET or GET-BULK requests for input octets, output octets, errors and interface status.
2. The router agent reads the corresponding MIB objects and returns the values.
3. The manager computes traffic rate by comparing counters across sampling intervals.
4. If the interface goes down, the agent immediately sends a link-down trap.
5. The management application raises an alarm and may notify the administrator.

## SNMP Versions

| Version | Characteristics |
|---|---|
| SNMPv1 | Basic operations and community-string security; simple but weak protection |
| SNMPv2c | Improved performance, error handling and bulk retrieval; still uses community strings |
| SNMPv3 | Adds user-based authentication, integrity protection, encryption and access control |

## Advantages

- Vendor-independent monitoring using standardized objects.
- Low protocol overhead and easy deployment.
- Supports periodic polling and immediate event notification.
- Allows centralized visibility across many devices.
- Extensible MIBs support vendor-specific and application-specific objects.

## Limitations

- Excessive polling can consume bandwidth and device CPU.
- UDP delivery does not guarantee that every request or trap arrives.
- Older versions provide weak security.
- Different vendors may implement private MIBs inconsistently.
- Raw counters require interpretation and correlation to identify root causes.

## Applications

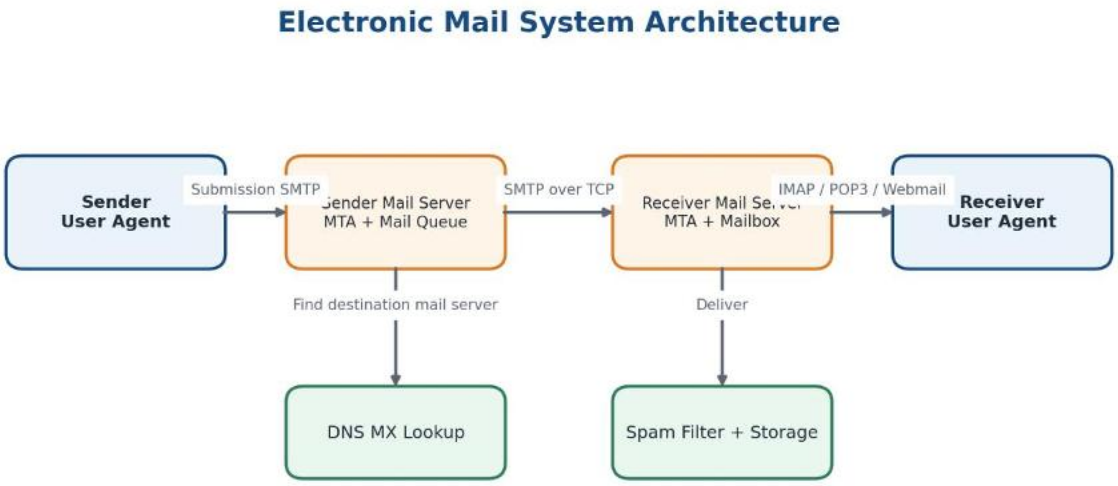
- Interface and bandwidth monitoring
- CPU, memory and environmental monitoring
- Fault and alarm management
- Inventory and configuration tracking
- Service-level reporting and capacity planning

## Conclusion

SNMP provides a simple framework for observing and controlling network elements. The manager-agent model, MIB and standardized operations make it a core protocol for network-management systems, especially when SNMPv3 security is used.

# Electronic Mail: Architecture, SMTP, MIME, POP3 and IMAP

Electronic mail is a store-and-forward application that allows users to exchange text and multimedia messages. A complete e-mail system includes user agents, mail servers, message-transfer agents, mailboxes, queues, DNS mail records and access protocols.




## Major Components

| Component | Role |
|---|---|
| User Agent | Creates, sends, reads, replies to and organizes messages |
| Mail Submission Agent | Accepts outgoing mail from an authenticated user |
| Mail Transfer Agent | Transfers messages between mail servers using SMTP |
| Mail Delivery Agent | Places accepted messages into the recipient mailbox |
| Mailbox | Persistent storage for received mail |
| Mail Queue | Temporarily stores messages awaiting successful delivery |
| DNS MX Record | Identifies the receiving mail servers for a domain |

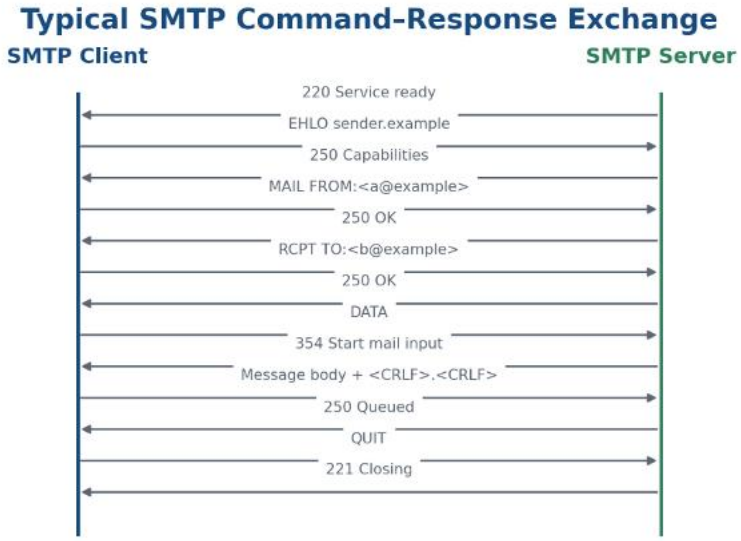
## Sending and Delivery Process

1. The sender composes a message in a user agent.
2. The user agent submits the message to the sender mail server using authenticated SMTP.

3. The sending server queries DNS for the recipient domain's MX records.
4. The sending server opens a TCP connection to the selected receiving mail server and transfers the message using SMTP.
5. The receiving server checks policy and spam rules, then stores accepted mail in the recipient mailbox.
6. The receiver reads or synchronizes the mailbox using IMAP, POP3 or a webmail interface.

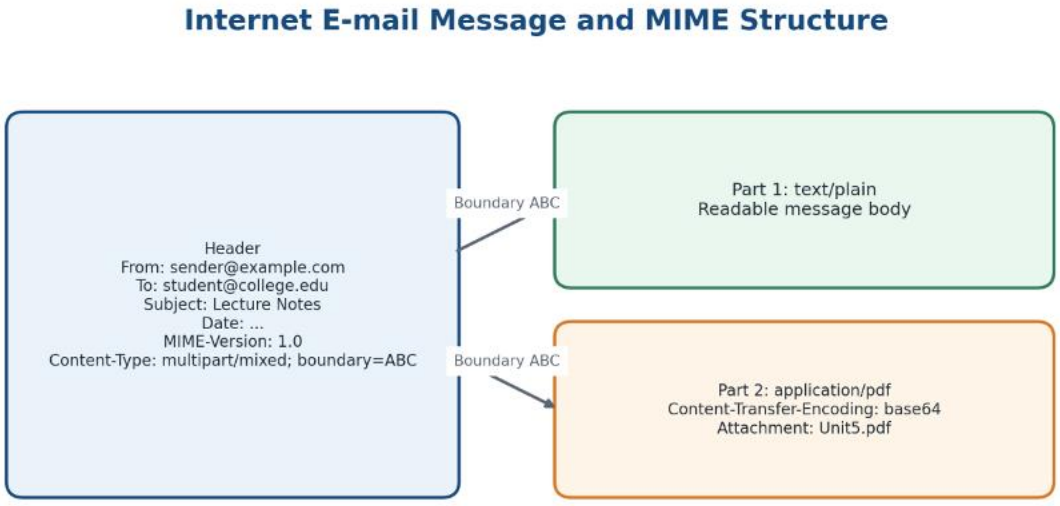
## SMTP Protocol

SMTP is a text-based, command-response protocol for message submission and server-to-server transfer. It uses a reliable TCP connection. The envelope commands identify the sender and recipient, while the DATA command begins transfer of the message content.

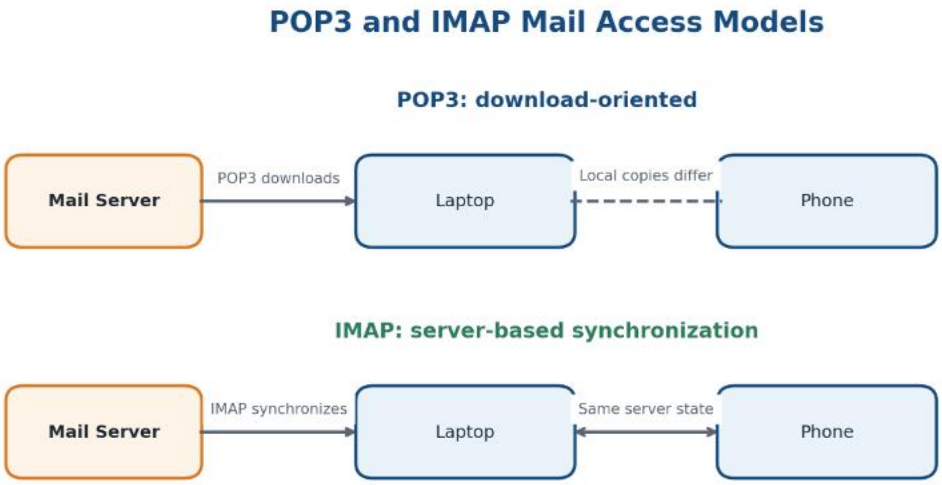



## Internet Message Format and MIME

An Internet mail message has a header and body separated by a blank line. The header contains fields such as From, To, Subject and Date. MIME extends the format so that non-ASCII text, images, audio, video and attachments can be carried. It defines content types, multipart boundaries and transfer encodings such as base64.



# POP3 and IMAP




| Feature | POP3 | IMAP |
|---|---|---|
| Primary model | Download messages to a client | Keep mailbox state on the server |
| Multiple devices | Limited synchronization | Designed for synchronized access |
| Folders | Mostly local organization | Server-side folders and flags |
| Offline use | Strong after download | Supported through local cache |
| Server storage | Can remove mail after download | Usually retains mail on server |
| Suitable for | Single-device or simple retrieval | Modern multi-device access |

## Example

A student sends project.pdf to a lecturer. The sender's client constructs a multipart MIME message containing a text part and a PDF attachment encoded as base64. SMTP transfers the message to the college mail server identified by its MX record. The lecturer's phone and laptop both show the same message and read status because they synchronize through IMAP.

## Advantages

- Asynchronous communication: sender and receiver need not be online together.
- Store-and-forward delivery across independent organizations.
- Supports attachments and rich content through MIME.
- Messages can be archived, searched and accessed from multiple devices.
- SMTP, MIME, POP3 and IMAP are widely interoperable standards.

## Limitations and Security Concerns

- Delivery may be delayed, rejected or routed to spam.
- Sender addresses can be forged unless authentication policies are used.

- Attachments can carry malware or expose confidential information.
- Metadata is visible to intermediate servers even when message content is encrypted.
- Mailbox storage and synchronization require continuous administration and security.

## Applications

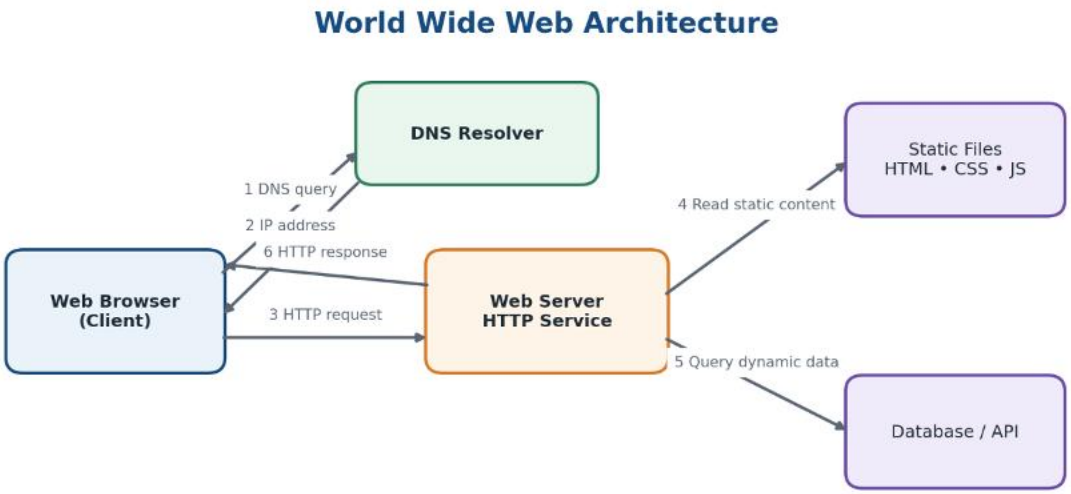
- Personal and institutional communication
- Business workflow and notifications
- Document exchange
- Account verification and password recovery
- Mailing lists and automated alerts

## Conclusion

Electronic mail combines SMTP-based transfer, DNS-based routing, MIME content representation and POP3/IMAP mailbox access. Its layered design allows different products and organizations to exchange messages reliably.

# The World Wide Web

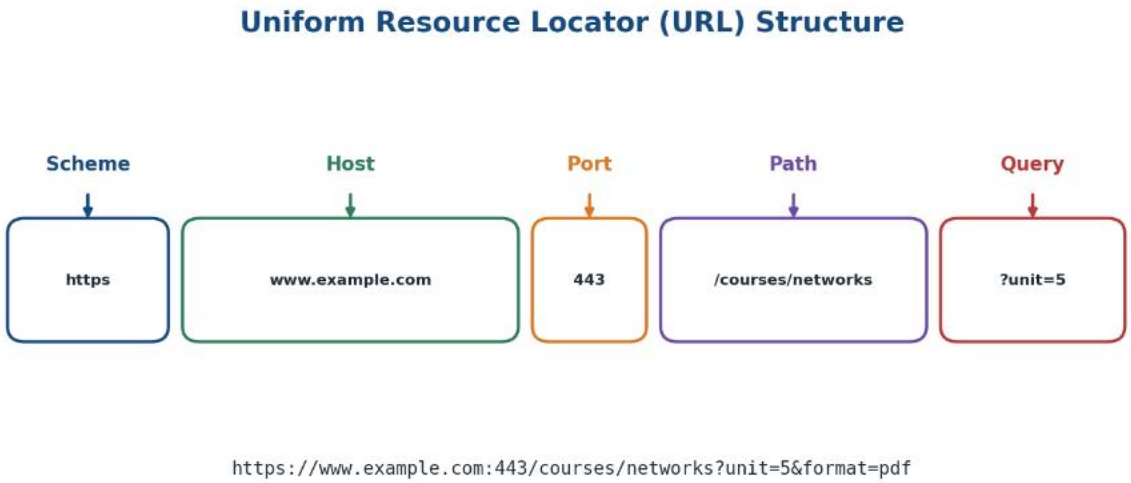
The World Wide Web is a distributed hypermedia system that allows users to access interlinked resources over the internet. A web resource may be an HTML document, image, video, script, style sheet, downloadable file or API response. The web primarily uses URLs to identify resources, DNS to locate servers and HTTP to transfer representations.



## Core Web Concepts

| Concept | Meaning |
|---|---|
| Web browser | Client application that requests, interprets and displays resources |
| Web server | Application that stores or generates resources and answers HTTP requests |
| Web page | Hypertext document, generally written in HTML |
| Website | Related collection of web resources under a domain |
| Hyperlink | Reference connecting one resource to another |
| URL | Address describing how and where a resource can be accessed |
| HTML | Markup language for page structure |
| CSS and JavaScript | Presentation and interactive behavior |

## Uniform Resource Locator




A URL can contain a scheme, host, optional port, path, query string and fragment. The scheme identifies the access protocol, the host is resolved through DNS, the path identifies a resource, and the query supplies parameters to server-side logic.

## How a Web Page Is Retrieved

1. The user enters a URL or follows a hyperlink.
2. The browser parses the URL and resolves the host name through DNS.
3. It establishes a transport connection to the server. HTTPS additionally establishes TLS security.
4. The browser sends an HTTP request for the selected resource.

5. The server returns a response containing a status, headers and content.
6. The browser parses HTML and issues additional requests for CSS, JavaScript, images, fonts and media.
7. The browser builds the document structure, applies styles, runs scripts and renders the page.

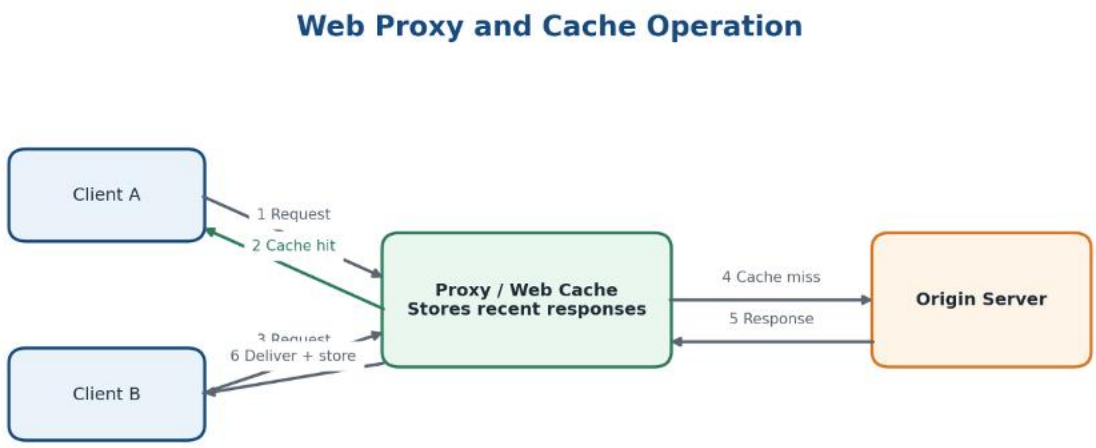
## Static and Dynamic Web Pages

| Static Page | Dynamic Page |
|---|---|
| Stored as a fixed file | Generated or personalized at request time |
| Same content for most users | May depend on user, database or current state |
| Simple, fast and easy to cache | Supports interaction and transactions |
| Example: syllabus PDF page | Example: student portal dashboard |

## Cookies and Web Sessions

HTTP is stateless, but many applications require login sessions and user preferences. A server can send a Set-Cookie header. The browser stores the cookie and includes it in later requests to the matching site. The cookie may hold a session identifier, while sensitive state remains on the server. Secure, HttpOnly and SameSite attributes reduce common attacks.

## Web Caching and Content Delivery




A cache stores reusable responses close to users. On a cache hit, the resource is returned without contacting the origin server. On a miss, the cache retrieves the resource, forwards it and may store it. Content Delivery Networks extend this idea using many geographically distributed edge servers.

## Advantages

- Platform-independent access through standard browsers.
- Hyperlinks make distributed information easy to navigate.
- Supports text, graphics, audio, video and interactive applications.
- Open standards enable worldwide interoperability.
- Caching and CDNs provide large-scale content distribution.

## Limitations and Challenges

- Web applications face privacy, tracking and security risks.
- Performance depends on DNS, network delay, server load and resource size.

- Different browsers and devices may render content differently.
- Dynamic pages require secure server-side code and database protection.
- Accessibility and usability must be intentionally designed.

## Applications

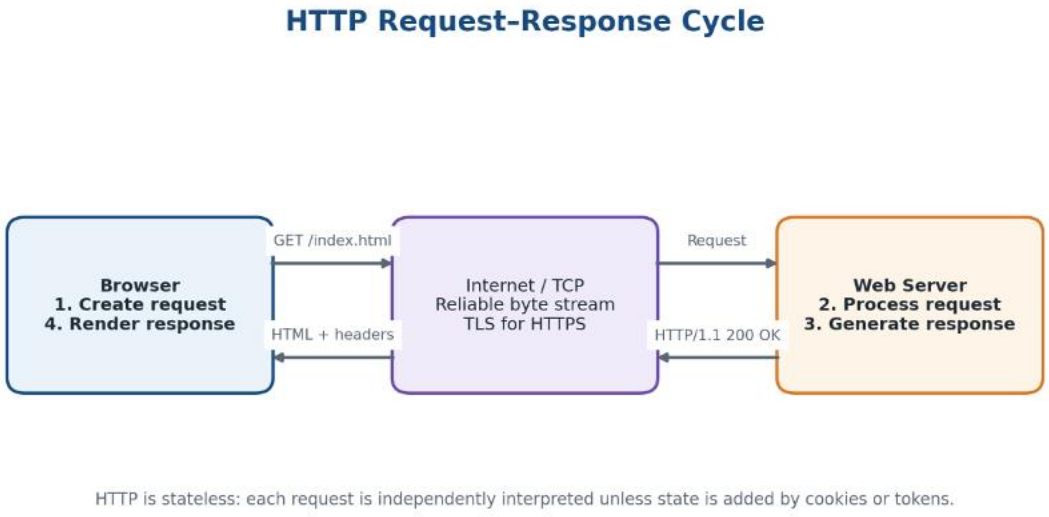
- Information publishing and search
- Electronic commerce and banking
- Online education
- Social networking and collaboration
- Cloud applications and web APIs
- Entertainment and media delivery

## Conclusion

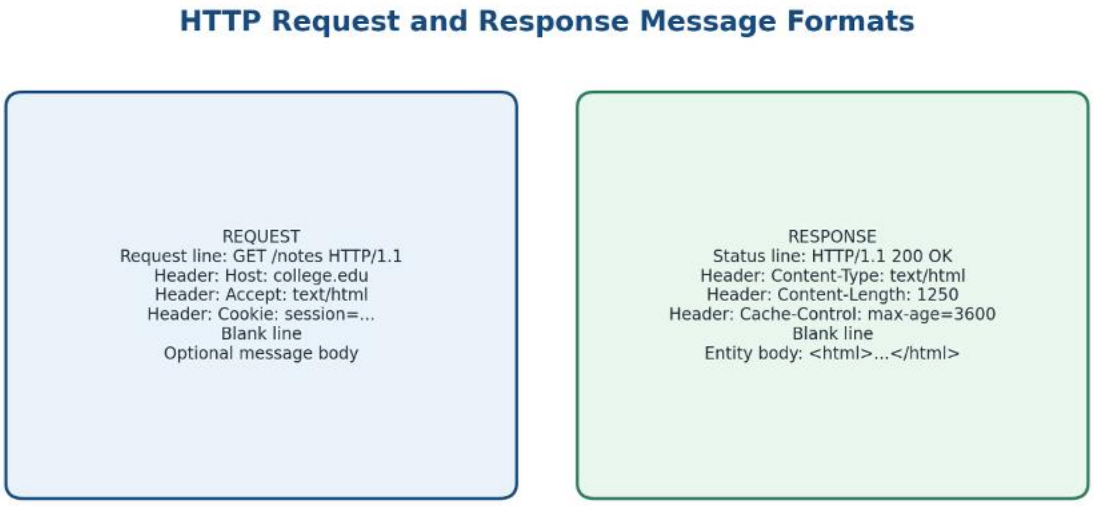
The web combines resource identification, name resolution, transfer protocols and browser rendering into a universal information system. Its open architecture has made the browser a common interface for documents, services and applications.

# Hypertext Transfer Protocol (HTTP)

HTTP is an application-layer request-response protocol used to transfer web resources and API data. A client sends a request containing a method, target and headers; a server returns a response containing a status code, headers and an optional representation. HTTP is stateless: the server interprets each request independently unless additional mechanisms such as cookies, tokens or server sessions are used.



## HTTP Message Format




## Common HTTP Methods

| Method | Use | Typical Property |
|---|---|---|
| GET | Retrieve a representation | Safe and generally cacheable |
| HEAD | Retrieve headers only | Useful for metadata checks |
| POST | Submit data or create subordinate processing | May change server state |
| PUT | Create or replace a resource at a known URI | Intended to be idempotent |
| PATCH | Partially modify a resource | Carries a change description |
| DELETE | Remove a resource | Intended to be idempotent |
| OPTIONS | Discover communication options | Used in capability and CORS checks |

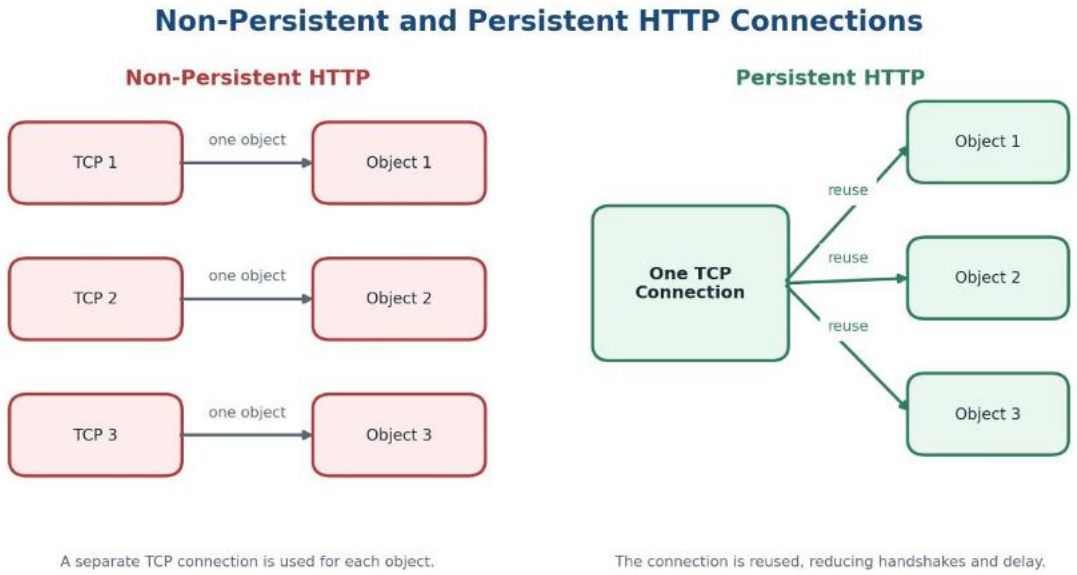
## Status Code Classes

| Class | Meaning | Examples |
|---|---|---|
| 1xx | Informational | 100 Continue |
| 2xx | Successful | 200 OK, 201 Created, 204 No Content |
| 3xx | Redirection | 301 Moved Permanently, 304 Not Modified |
| 4xx | Client error | 400 Bad Request, 401 Unauthorized, 404 Not Found |
| 5xx | Server error | 500 Internal Server Error, 503 Service Unavailable |

## Example HTTP Transaction

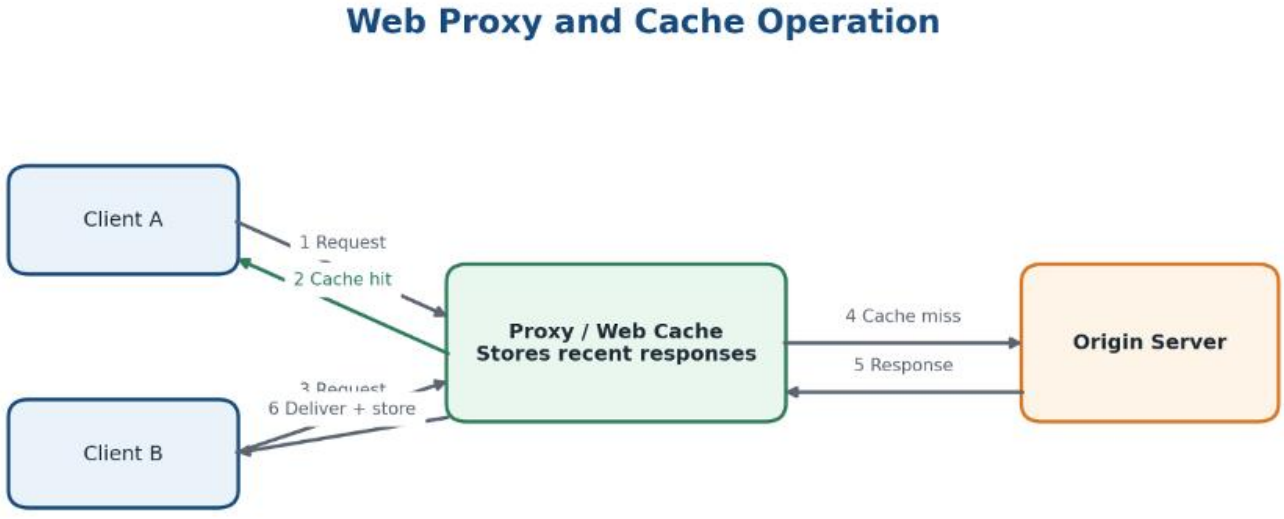
Request: GET /unit5/notes HTTP/1.1 with Host: college.edu and Accept: application/pdf. If the document exists and the user is authorized, the server may respond HTTP/1.1 200 OK with Content-Type: application/pdf and the file bytes. If the resource does not exist, it may return 404 Not Found.

## Connection Models




Non-persistent HTTP uses a separate transport connection for each object. Persistent HTTP reuses a connection for multiple requests and responses, reducing handshake overhead and latency. Modern HTTP versions also support multiplexing so several exchanges can progress efficiently over a connection.

## Caching and Conditional Requests



Response headers such as Cache-Control and Expires tell clients and intermediaries how a representation may be reused. Validators such as ETag and Last-Modified support conditional requests. If the cached copy is still valid, the server can respond 304 Not Modified without retransmitting the body.

## HTTPS

HTTPS is HTTP protected by TLS. TLS authenticates the server using a certificate, negotiates cryptographic keys, encrypts transferred data and detects modification. HTTPS protects credentials and content in transit, although endpoint security and correct certificate validation remain essential.

## HTTP Versions — Conceptual Comparison

| Version | Important Idea |
|---|---|
| HTTP/1.0 | Basic request-response; non-persistent operation was common |
| HTTP/1.1 | Persistent connections, improved caching and host-based virtual sites |
| HTTP/2 | Binary framing, multiplexed streams and compressed headers |
| HTTP/3 | Runs over QUIC, reducing some connection and loss-related delays |

## Advantages

- Simple and extensible header-based design.
- Works through common network infrastructure and supports many media types.
- Stateless operation assists scaling and load balancing.
- Caching reduces delay and bandwidth use.
- HTTPS provides confidentiality, integrity and server authentication.

## Limitations

- Statelessness requires extra mechanisms for sessions.
- Large headers and many objects can create overhead.
- Caching rules and invalidation can be complex.
- Application vulnerabilities remain possible even with HTTPS.
- Performance depends on transport behavior, congestion and server processing.

## Applications

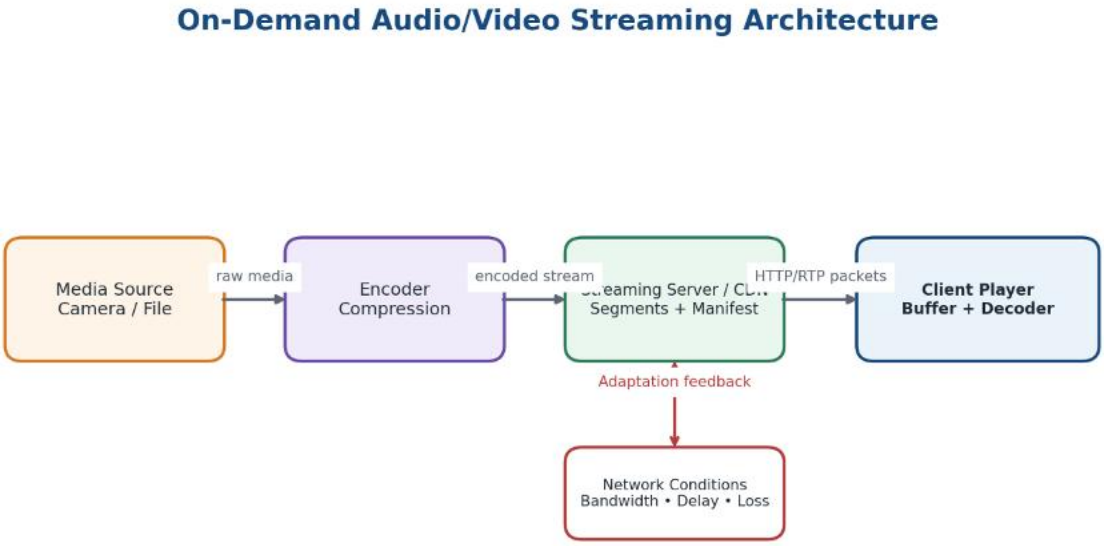
- Web pages and websites
- REST-style APIs
- Cloud and mobile back ends
- Software updates and file downloads
- Adaptive media streaming
- Internet of Things control services

## Conclusion

HTTP is the common transfer protocol of the web and many modern distributed applications. Its methods, status codes, headers, persistent connections, caching and TLS security provide a flexible framework for exchanging resources.

# Streaming Audio and Video

Streaming delivers audio or video so that playback can begin before the entire media object has been downloaded. The application must handle high data rates, variable network delay, packet loss and user expectations for smooth playback. Streaming can be on-demand, where stored content is selected by the user, or live, where content is captured and delivered with a small delay.



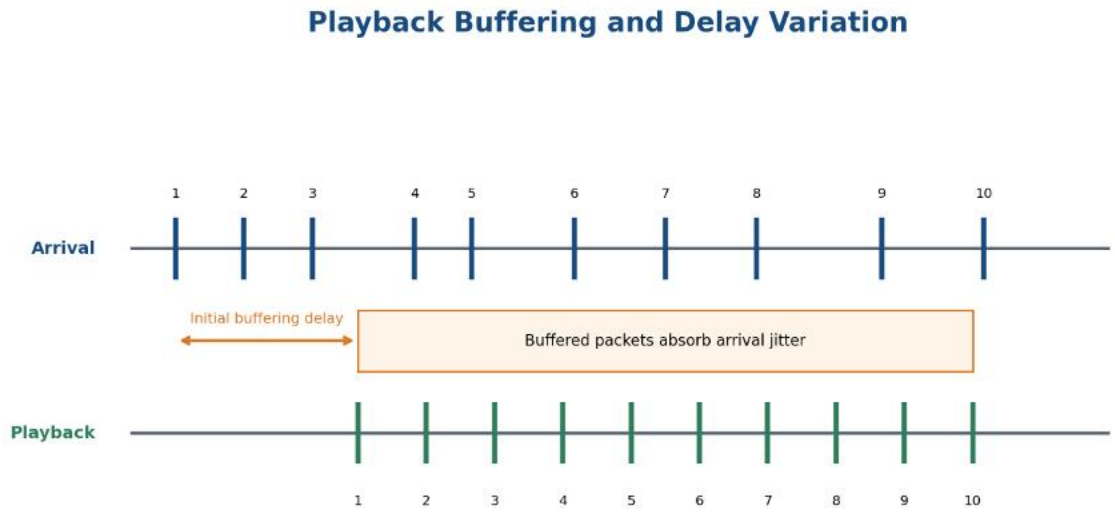

## Media Characteristics and Requirements

- Bandwidth: encoded media requires a sustained data rate.
- End-to-end delay: especially important for live and interactive media.
- Jitter: packets do not always arrive at uniform intervals.
- Packet loss: small losses may be concealed, but repeated loss degrades quality.
- Synchronization: audio and video streams must remain aligned.
- Scalability: popular streams may need to serve millions of viewers.

## Compression and Encoding

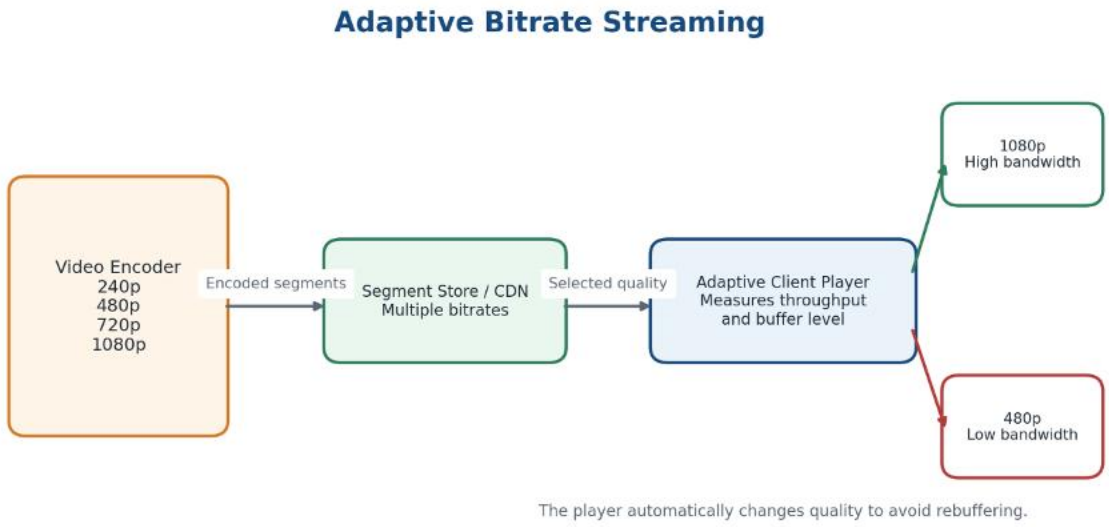
Raw audio and video are too large for practical internet delivery. An encoder removes redundancy and represents the media at one or more bitrates. Compression may be lossy, trading some fidelity for a major reduction in data rate. The client decoder reconstructs media frames for playback.

## Buffering and Jitter Control



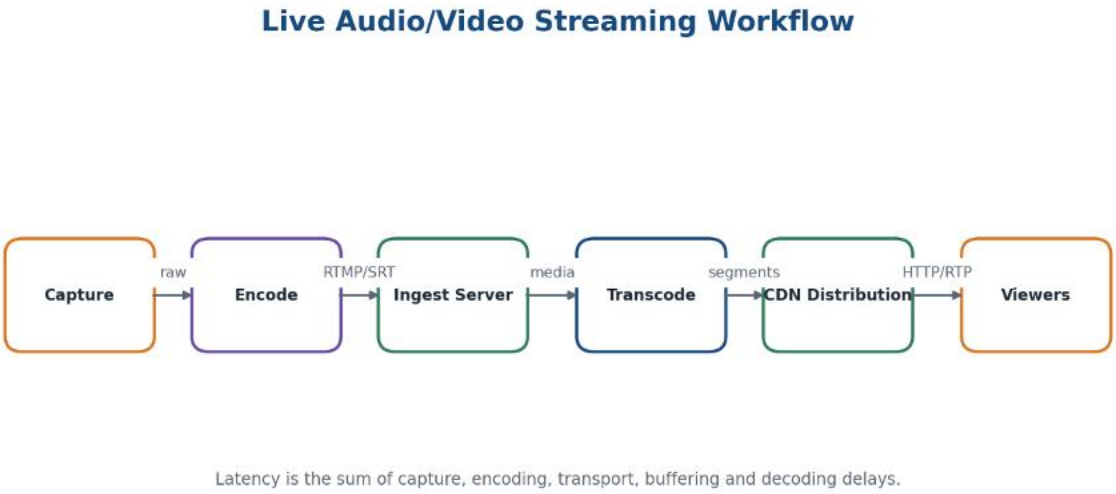

The player stores arriving media in a playback buffer and waits for an initial buffering interval. This allows packets that experience different network delays to be played at regular time intervals. If the buffer becomes empty, playback stalls; if it grows too large, live-stream latency increases.

## Adaptive Bitrate Streaming




- The source is encoded at several quality levels and divided into short segments.
- A manifest tells the player which versions and segments are available.
- The player estimates available throughput and observes buffer occupancy.
- It requests a higher bitrate when conditions are good and a lower bitrate when bandwidth falls.
- The objective is to maximize quality while avoiding rebuffering.

# Live Streaming




Live streaming captures media, encodes it immediately, sends it to an ingest service, may transcode it into several bitrates, distributes segments through a CDN and plays them at viewers. Low latency requires shorter segments and smaller buffers, but this leaves less protection against jitter.

## Protocols and Delivery Approaches

| Approach | Characteristics | Typical Use |
|---|---|---|
| HTTP segment streaming | Uses ordinary web requests, caches and CDNs; adaptive bitrate is common | Large-scale on-demand and live distribution |
| RTP over UDP | Carries timestamped real-time media with sequence numbers | Interactive voice/video and managed real-time sessions |
| RTCP | Reports reception quality, loss, jitter and timing alongside RTP | Monitoring and synchronization |
| WebRTC-style real-time delivery | Low-latency peer or server-assisted media with congestion control | Video meetings and browser communication |

## Worked Example

A student watches a recorded lecture. The player initially requests a medium-quality segment and buffers several seconds. When measured throughput rises, it selects 1080p segments. When the student moves to a weak Wi-Fi area, throughput falls and the buffer starts decreasing, so the player switches to 480p before the buffer becomes empty. Playback continues with lower image quality but without interruption.

## Advantages

- Playback starts without waiting for the complete file.
- Adaptive streaming can match changing network capacity.
- CDNs make large-scale delivery practical.
- Users can seek, pause and resume on-demand content.
- Live streaming supports remote events, education and broadcasting.

## Limitations and Challenges

- High and sustained bandwidth is required for good quality.
- Network congestion causes quality reduction, delay or rebuffering.
- Encoding and CDN infrastructure can be expensive.
- Live streams face a trade-off between low latency and smooth playback.
- Digital rights, privacy and content protection must be addressed.

## Applications

- Online lectures and training
- Movies and music services
- Live sports and news
- Video conferencing and telemedicine
- Security-camera monitoring
- Social-media live broadcasts

## Conclusion

Streaming systems combine compression, transport, buffering, adaptation and distributed delivery. Effective streaming manages the trade-off among quality, startup delay, rebuffering and live latency under changing network conditions.