COMPUTER NETWORKS AND THE WEB

Computer Practicum EN - WIS - LES

Lecturer: Niki Hrovatin

What is this?



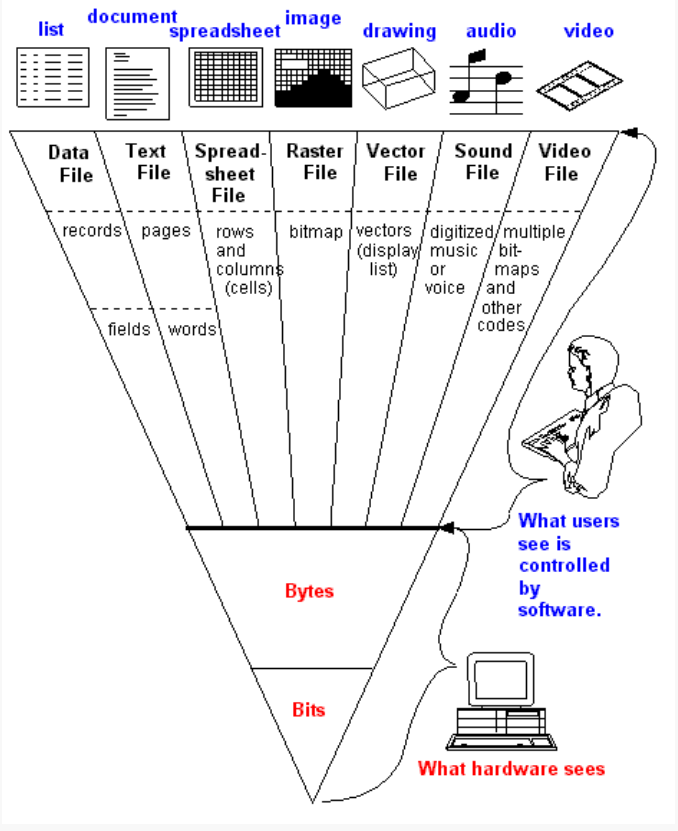
What Is Data Communication?

We have seen that computers process, represent and store data using bits. But how can these data be exchanged between devices?

The word **data** refers to information presented in whatever form is agreed upon by the parties creating and using the data.

The information being communicated may take different forms:

- Text and numbers — messages, documents, numerical values.
- Images and video — photographs, movies, video calls.
- Audio — speech, music, sound recordings.



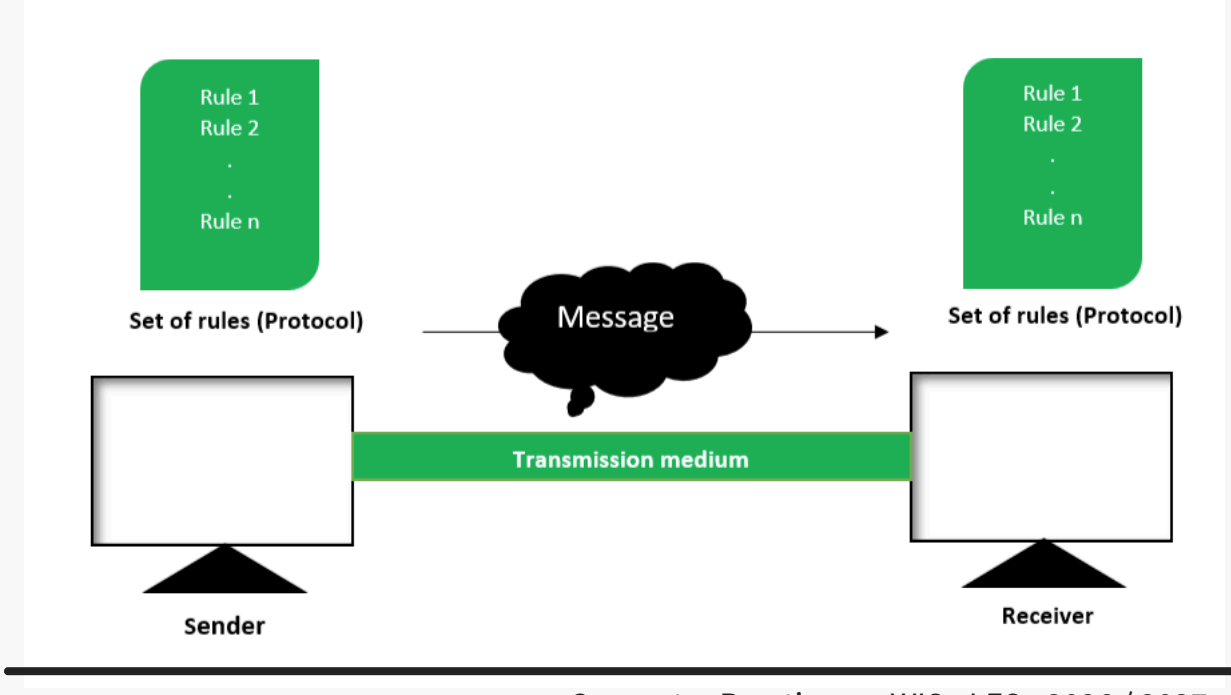
...00101010110010...

What Is Data Communication?

Data communication is the exchange of data between two devices through a **transmission medium**.

In computer networks, data are encoded into bits, transmitted using physical signals, and reconstructed by the receiving device.

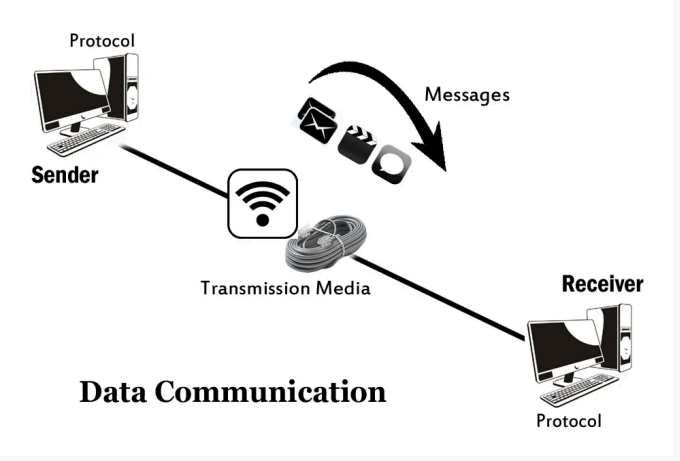
The receiver must be able to interpret the received data correctly.



Components of a Data Communication System

Every data communication system requires five essential components:

1. **Message** — the data being communicated, such as text, numbers, images, audio, or video.
2. **Sender** — the device that originates and transmits the message (e.g., Laptop A).
3. **Receiver** — the device that receives and interprets the message (e.g., Laptop B).
4. **Transmission medium** — the physical path through which signals travel, such as copper cables, optical fibers, or radio waves.
5. **Protocol** — a set of agreed rules specifying how devices communicate, including how messages are formatted, transmitted, and interpreted.



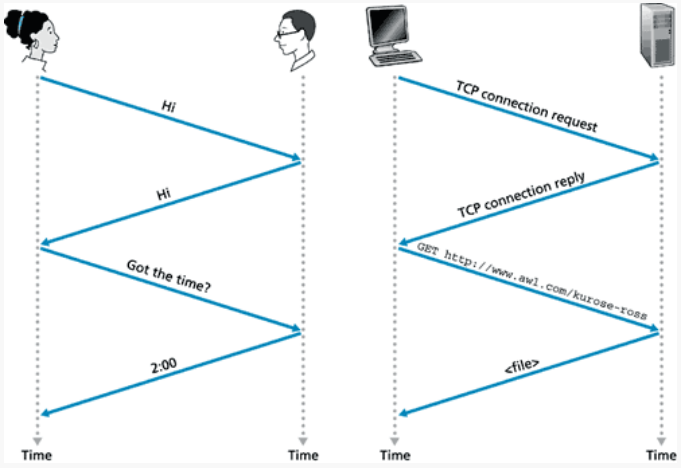
Data Communication

Two devices may be physically connected but unable to communicate without a shared protocol.

A Conversation Is Already a Protocol?

Imagine two people meeting:

- **Alice:** "Hi!"
- **Bob:** "Hi!"
- **Alice:** "How are you?"
- **Bob:** "I am good, thanks! And you?"
- **Alice:** "I am good too!"



They follow an expected **sequence of messages**: greeting, reply, question, answer.

A **protocol** is a set of agreed rules defining **what is communicated, how, and when**.

Bandwidth: How Fast Can Data Be Transmitted?

In computer networks, bandwidth commonly refers to the maximum data transmission rate supported by a communication link.

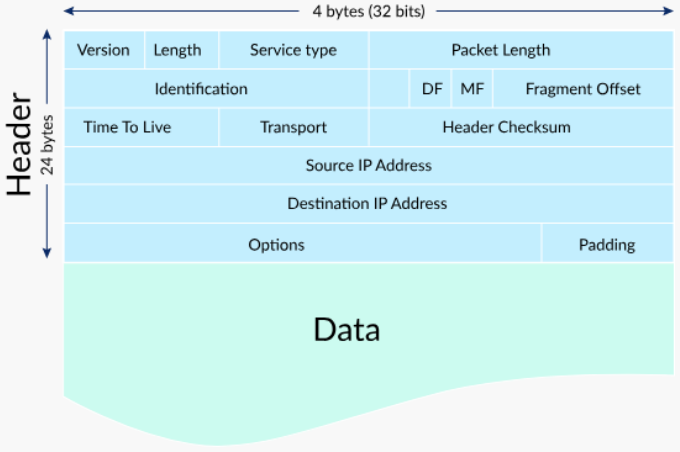
Since computers exchange bits, data rates are measured in bits per second (bit/s or bps).

- 1 kb/s = 1,000 bits per second.
- 1 Mb/s = 1,000,000 bits per second.
- 1 Gb/s = 1,000,000,000 bits per second.

Example: A network connection with a bandwidth of 100 Mb/s can theoretically transmit 100 million bits per second.

This corresponds to 12.5 MB/s, since 1 byte = 8 bits.

However, bandwidth describes a connection's capacity, not necessarily the speed we actually experience.



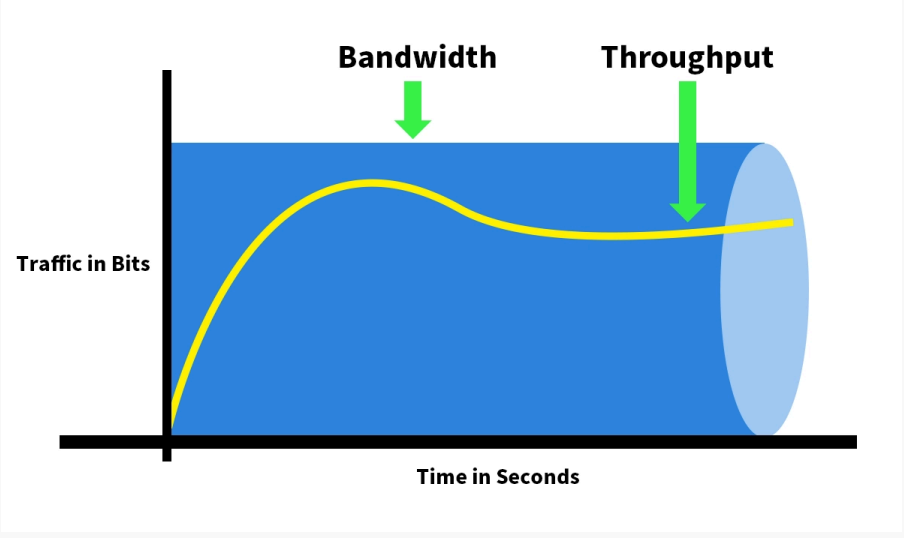
IPv4 packet

Bandwidth vs. Throughput: Capacity vs. Reality

Throughput is the actual rate at which data is transferred through a network, typically measured in bit/s.

Actual throughput may be lower because of:

- Protocol overhead: Additional information such as headers, addresses, and error checks.
- Network congestion: Multiple devices sharing limited network capacity.
- Transmission errors: Lost or corrupted data may need retransmission.



Latency: How Long Does Communication Take?

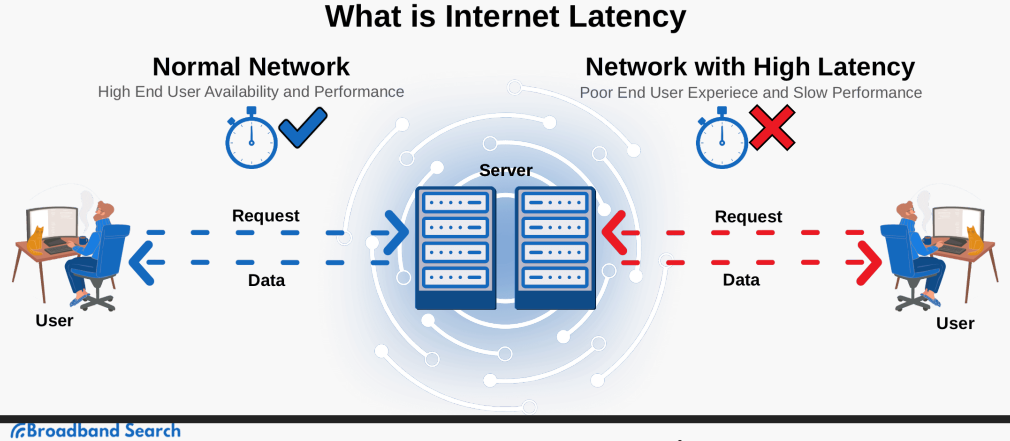
Latency is the time delay between sending data and receiving them at the destination.

Unlike bandwidth and throughput, latency measures time, typically expressed in milliseconds (ms).

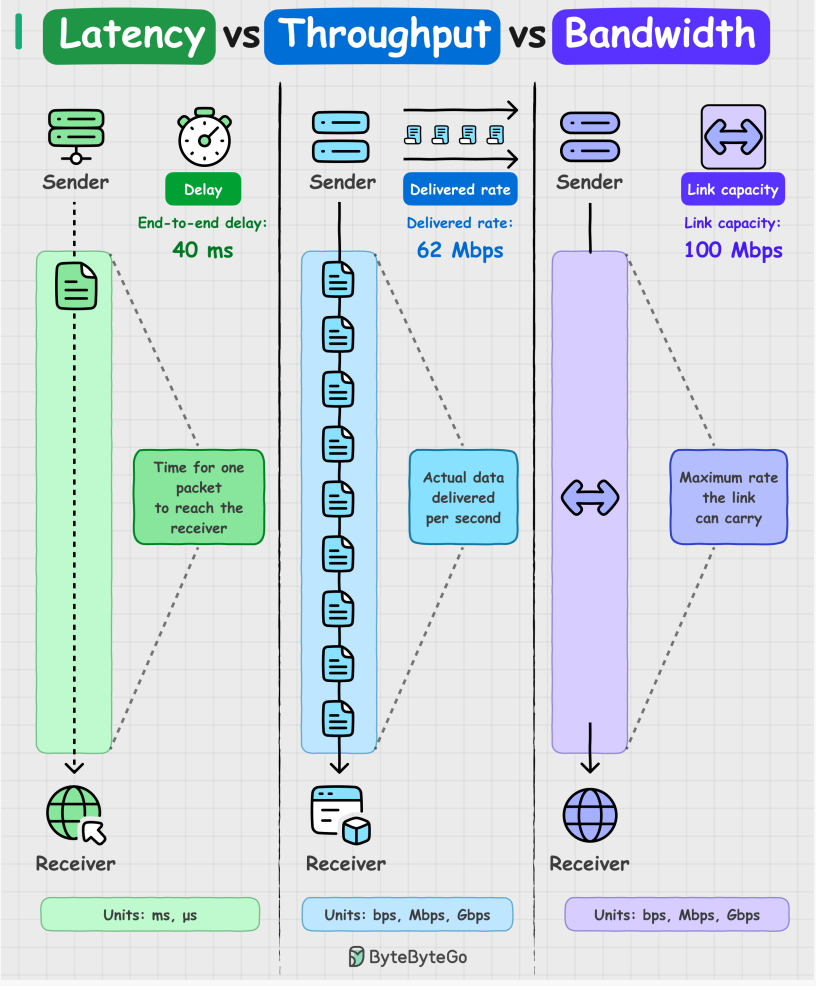
- **One-way latency:** Time required for data to travel from sender to receiver.
- **Round-trip time (RTT):** Time required for a message to travel to another device and for a reply to return.

Latency is influenced by the distance between devices, network equipment, and delays caused by processing or waiting.

A network can have high bandwidth but still feel slow because of high latency.



Latency vs. Throughput vs Bandwidth



Which Direction Can Data Travel?

Simplex - one direction only

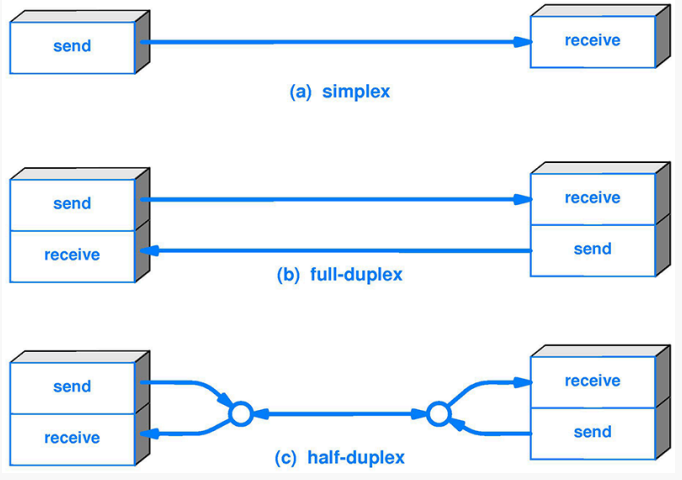
- One device sends; the other only receives.

Half-duplex - both directions, but taking turns

- Both can send and receive, **not at the same time**.

Full-duplex - both directions simultaneously

- Both can send and receive **at the same time**.



Which Direction Can Data Travel?

Simplex - one direction only

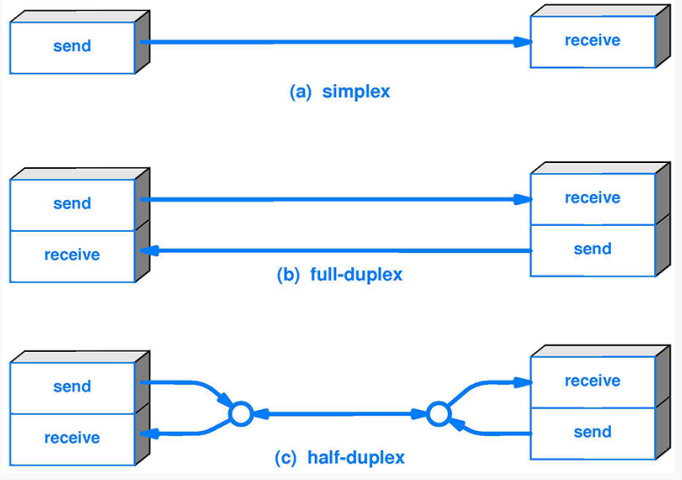
- One device sends; the other only receives.
- Example: a traditional one-way radio broadcast.

Half-duplex - both directions, but taking turns

- Both can send and receive, **not at the same time**.

Full-duplex - both directions simultaneously

- Both can send and receive **at the same time**.



Which Direction Can Data Travel?

Simplex - one direction only

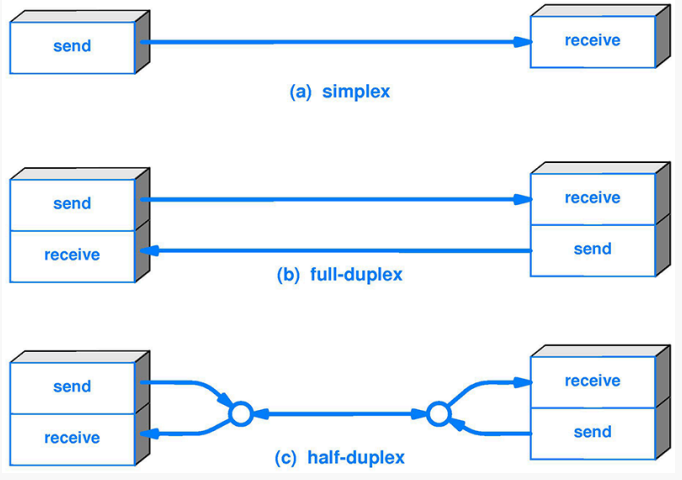
- One device sends; the other only receives.
- Example: a traditional one-way radio broadcast.

Half-duplex - both directions, but taking turns

- Both can send and receive, **not at the same time**.
- Example: two people using walkie-talkies.

Full-duplex - both directions simultaneously

- Both can send and receive **at the same time**.



Which Direction Can Data Travel?

Simplex - one direction only

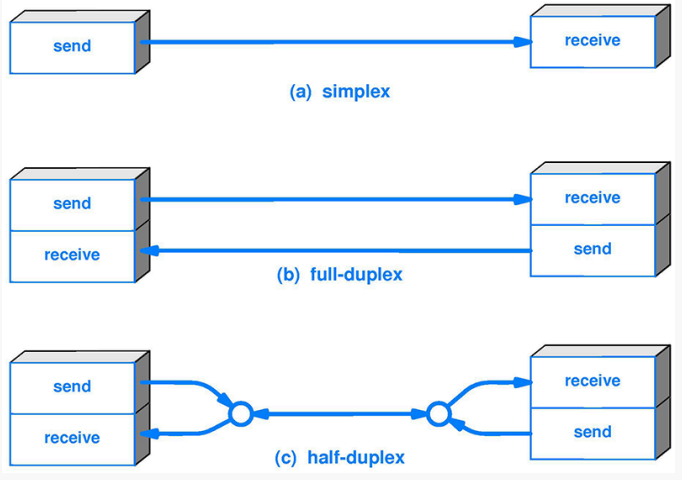
- One device sends; the other only receives.
- Example: a traditional one-way radio broadcast.

Half-duplex - both directions, but taking turns

- Both can send and receive, **not at the same time**.
- Example: two people using walkie-talkies.

Full-duplex - both directions simultaneously

- Both can send and receive **at the same time**.
- Example: a telephone call or switched Ethernet link.



What Is a Computer Network?

A **computer network** is a collection of connected devices that can exchange data and share resources using communication protocols.

- **Nodes (hosts):** endpoints of the network (**senders / receivers**):
  - computers, phones, printers, and servers.
- **Links:** wired or wireless connections between devices.
- **Network equipment:** additional components that support network operation
  - switches, wireless access points, routers, etc....



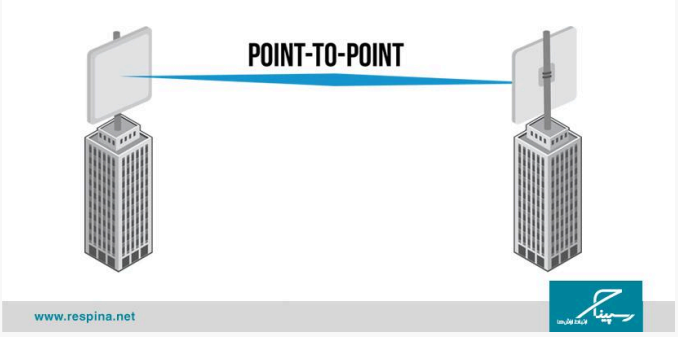
Examples of what a network enables:

- Share a file or a printer.
- Send a message to another computer.
- Access websites and services hosted elsewhere.

A network can work without connecting to the Internet.

Point-to-Point: One Direct Connection

A **point-to-point link** connects exactly **two devices**.

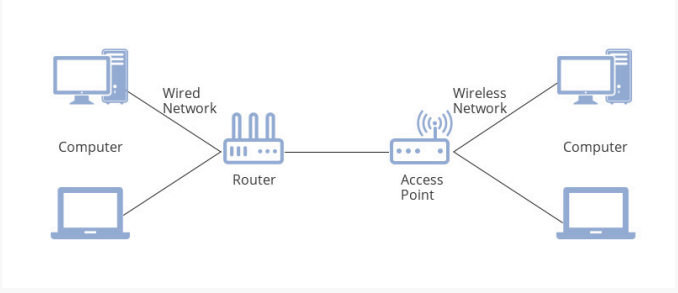


Advantages

- Simple structure and a direct communication path.
- No need to share the link with other attached devices.

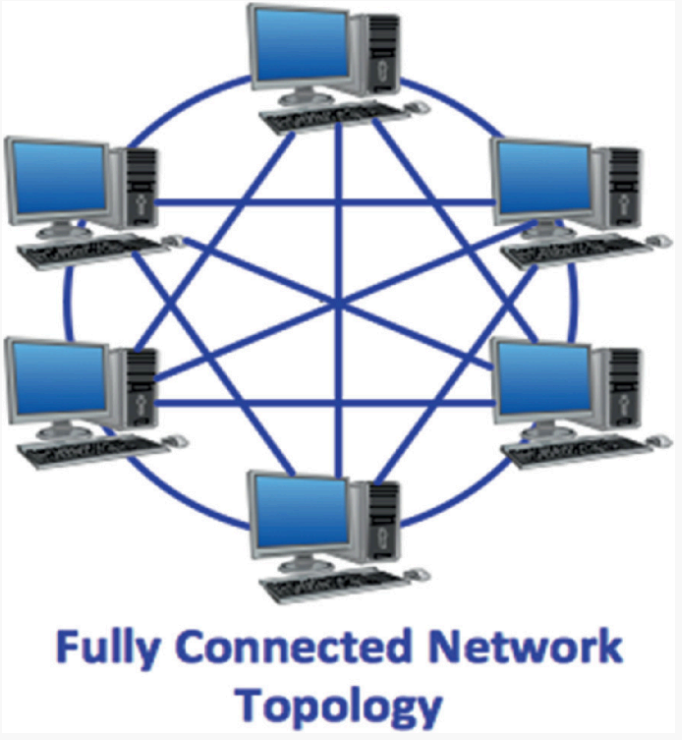
Limitations - No redundancy

- Directly connecting many devices requires many separate links.
- If the only link fails, that direct connection is lost.



Fully Connected Topology

In a **fully connected mesh**, every device has a direct link to every other device.



Advantages

- Multiple alternative routes can exist when a link fails.
- Devices do not depend on a single central connection point.

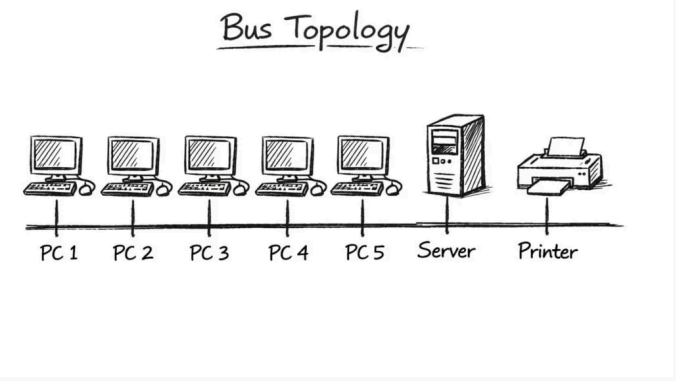
Limitations--High cost

- The number of required links grows very quickly.
- Cabling, interfaces, and maintenance become expensive.

Links required: $L = n(n-1)$, where n are network nodes

Bus Topology: One Shared Medium

In a **bus topology**, devices attach to a shared backbone or communication medium.



Advantages

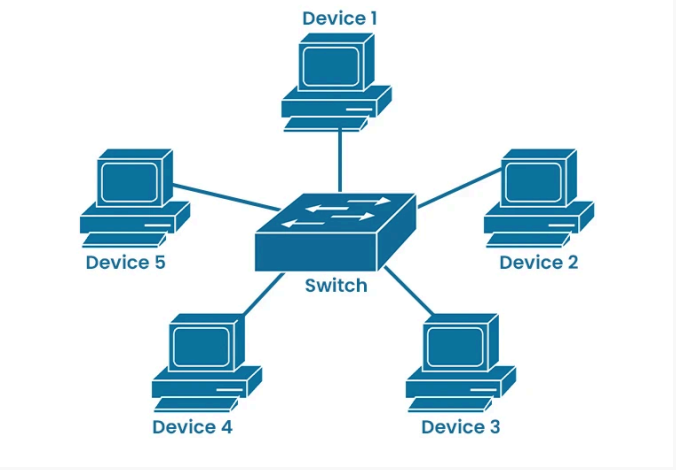
- Simple arrangement with relatively little cabling.
- Suitable for some small or specialized systems.

Limitations - Redundancy and the medium becomes busy

- Devices share the medium and its transmission capacity.
- A backbone fault can affect the entire network.
- Diagnosing faults and expanding the bus can be difficult.

Star Topology: Devices Connect to a Central Point

In a **star topology**, each device has a separate link to a central device, usually a **switch**.



Advantages

- Easy to add or remove devices.
- A failed cable normally affects only its connected device.
- Faults are usually easier to isolate.

Limitations - Single point of failure

- If the central device fails, the whole network fails.

Example: Most modern wired office and classroom LANs.

Hybrid Topology: Combining Network Layouts

A **hybrid topology** combines two or more basic arrangements.

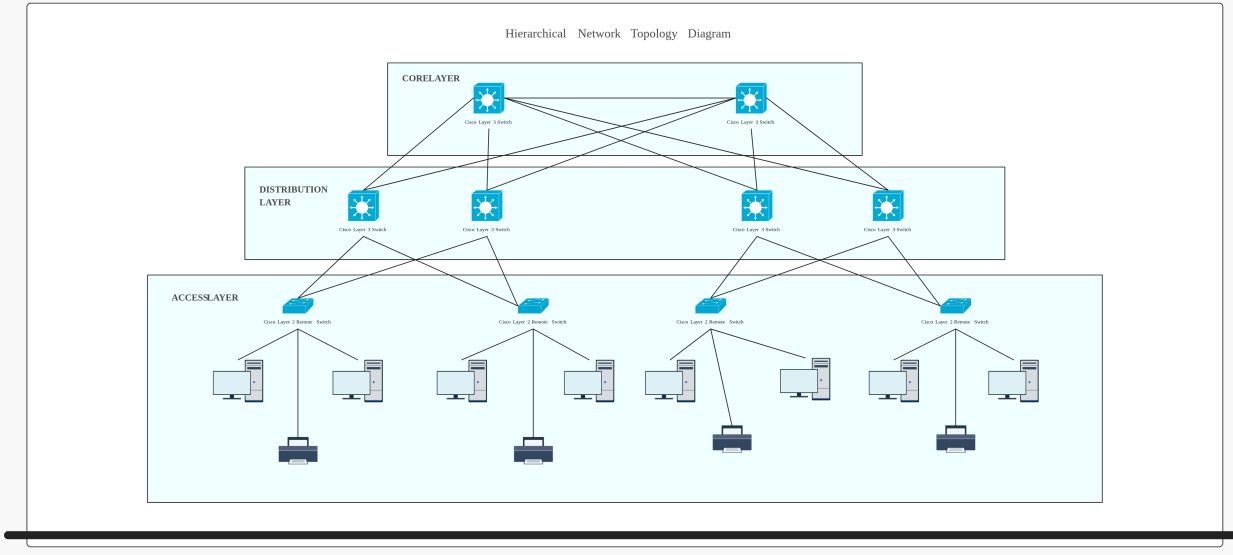
Advantages

- Flexible: different parts can use the most suitable arrangement.
- Easy to extend a network in stages.

Limitations

- Design and troubleshooting become more complex.

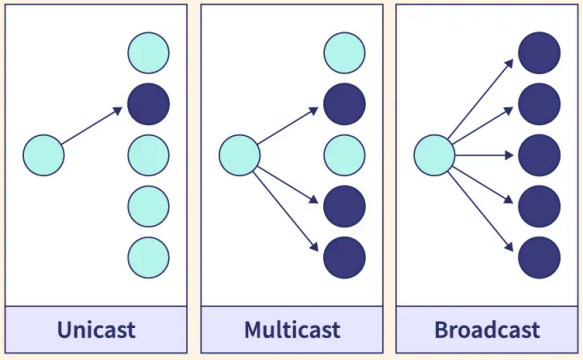
Real networks are often hybrid rather than purely one topology.



Who Is the Message Intended For?

Topology tells us **how devices are connected**. It does not tell us **who receives a message**.

- **Unicast (one-to-one):** A sender addresses one destination.
- **Broadcast (one-to-all):** - A message is transmitted to all potential receivers within its coverage area.
- **Multicast (one-to-group):** a message targets a selected group of receivers.

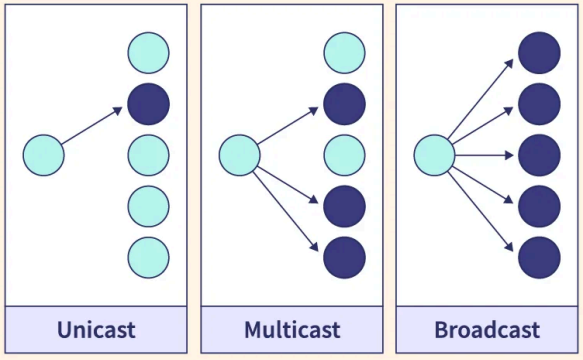


These describe **delivery patterns**, not network shapes.

Who Is the Message Intended For?

Topology tells us **how devices are connected**. It does not tell us **who receives a message**.

- **Unicast (one-to-one):** A sender addresses one destination.
  - Example: A phone call between two people.
- **Broadcast (one-to-all):** - A message is transmitted to all potential receivers within its coverage area.
- **Multicast (one-to-group):** a message targets a selected group of receivers.

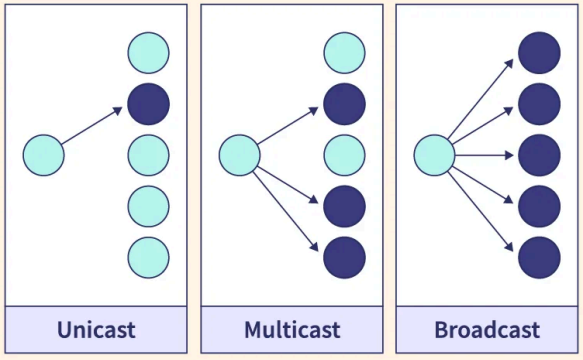


These describe **delivery patterns**, not network shapes.

Who Is the Message Intended For?

Topology tells us **how devices are connected**. It does not tell us **who receives a message**.

- **Unicast (one-to-one):** A sender addresses one destination.
  - Example: A phone call between two people.
- **Broadcast (one-to-all):** - A message is transmitted to all potential receivers within its coverage area.
  - Example: An FM radio station broadcasts a signal that any radio within range can tune into.
- **Multicast (one-to-group):** a message targets a selected group of receivers.

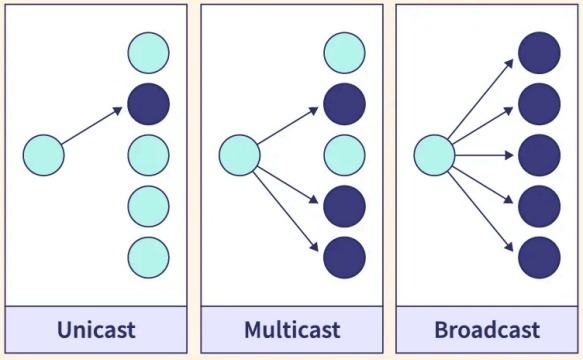


These describe **delivery patterns**, not network shapes.

Who Is the Message Intended For?

Topology tells us **how devices are connected**. It does not tell us **who receives a message**.

- **Unicast (one-to-one):** A sender addresses one destination.
  - Example: A phone call between two people.
- **Broadcast (one-to-all):** - A message is transmitted to all potential receivers within its coverage area.
  - Example: An FM radio station broadcasts a signal that any radio within range can tune into.
- **Multicast (one-to-group):** a message targets a selected group of receivers.
  - Example: Silent disco - A DJ plays music for participants who have selected a particular audio channel, while other participants listen to different DJs.



These describe **delivery patterns**, not network shapes.

Networks Can Cover Very Different Areas

Networks are also classified by their **approximate geographic scope**:

- **PAN (Personal Area Network):** devices around one person (e.g., Bluetooth headphones and phone).
- **LAN (Local Area Network):** a room, building, home, or campus network.
- **MAN (Metropolitan Area Network):** interconnected networks across a city.
- **WAN (Wide Area Network):** connects networks across regions or countries.



How Can Different Networks Work Together?

We have seen that networks can cover different geographical areas, from a few meters to entire continents.

Networks may also differ in their:

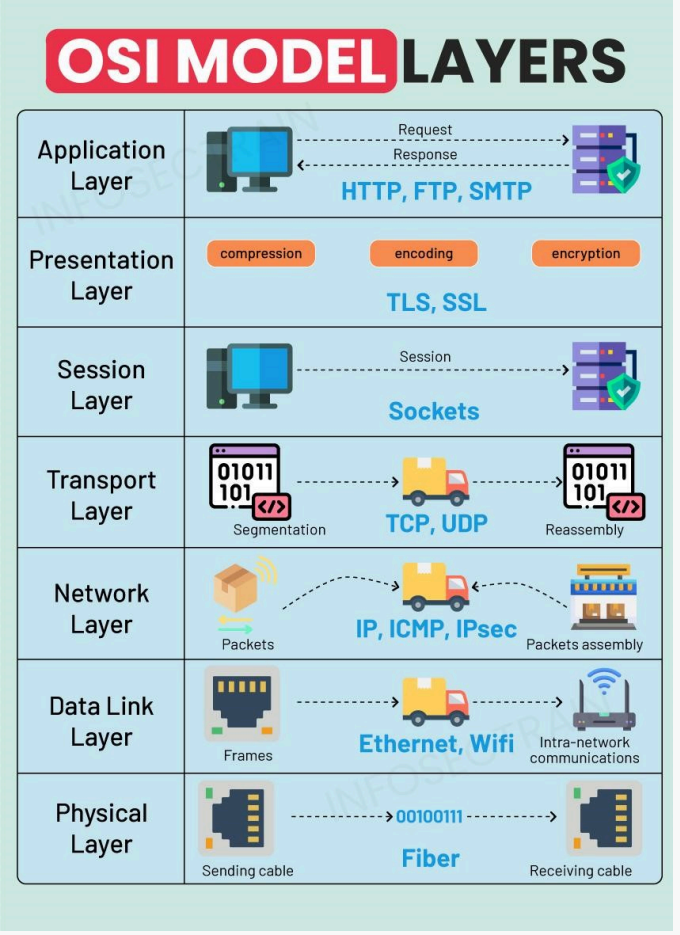
- Transmission media: Copper cables, optical fibers, or wireless signals.
- Communication technologies: Ethernet, Wi-Fi, and others.
- Performance: Transmission speeds, distances, and latency.

However, these different networks must be able to communicate with one another.

- **The solution is a layered architecture.**

Layering divides network communication into several levels, each responsible for a specific task and providing services to the layer above it.

This allows different network technologies to work together through standardized protocols.

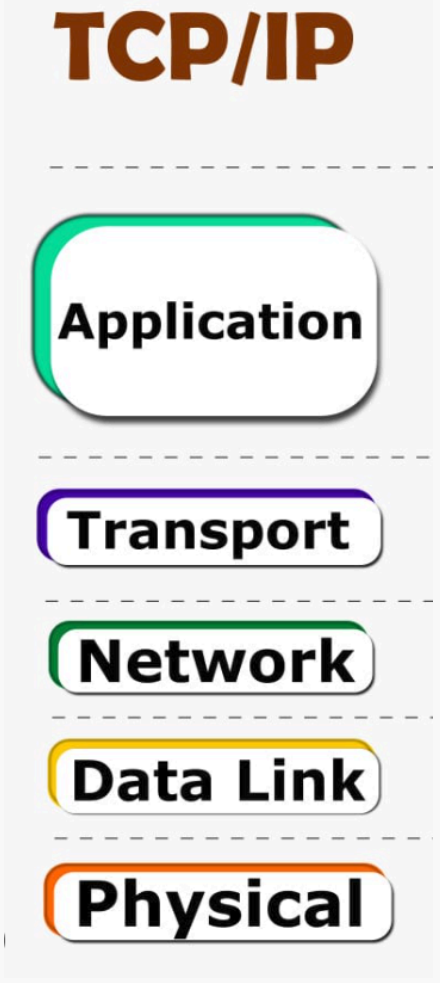


OSI and TCP/IP: Two Ways to Describe Layers

The **OSI model** is a **seven-layer reference model** that helps describe communication functions.

The Internet uses the **TCP/IP protocol suite**, commonly grouped into five layers:

1. **Application:** HTTP and DNS.
2. **Transport:** TCP, UDP, and ports.
3. **Network:** IP addresses and routing.
4. **Data Link:** Ethernet, Wi-Fi, and MAC addresses.
5. **Physical:** signals over copper, fiber, or radio.

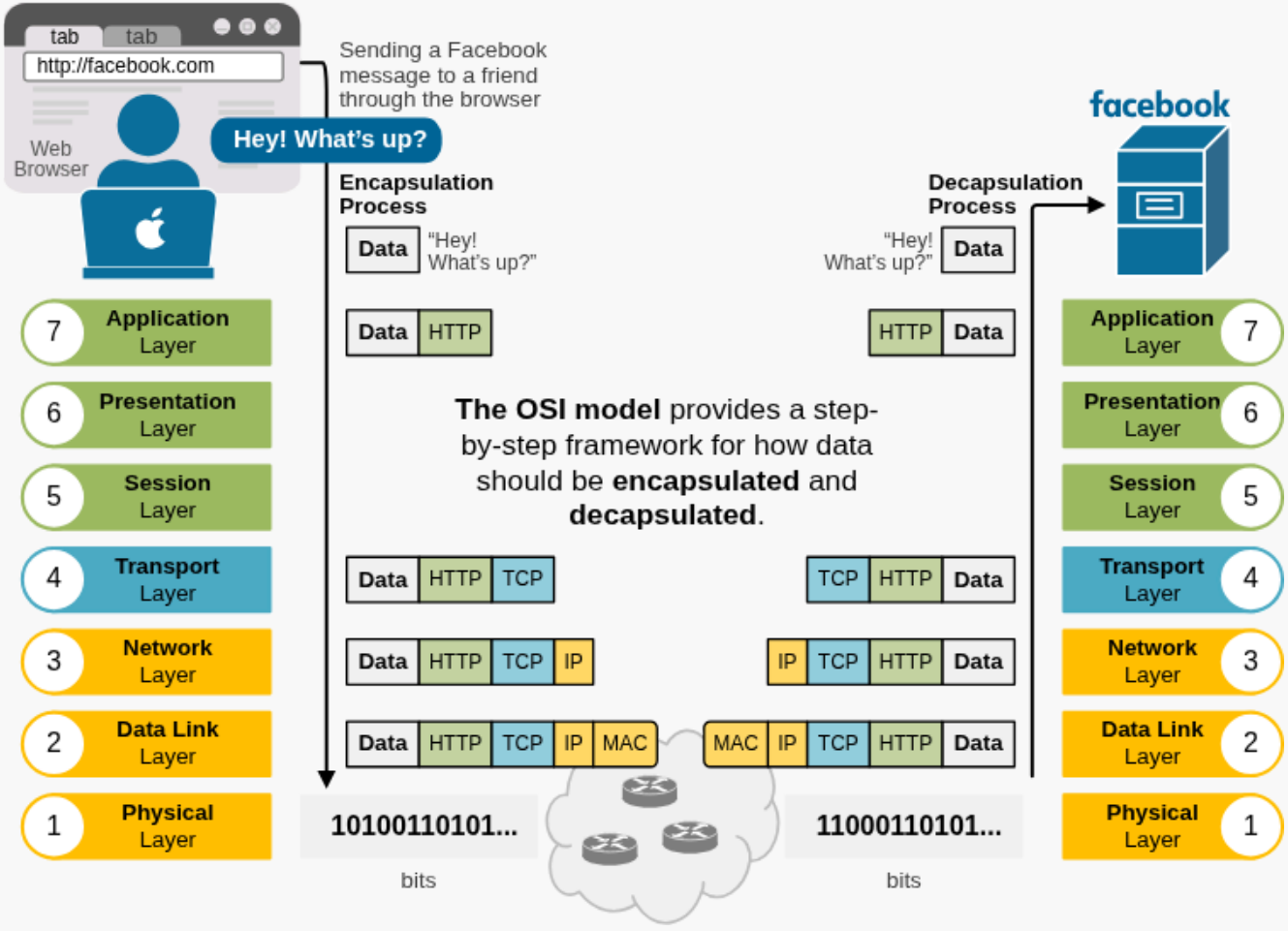


Why is layering important?

Each layer has a specific responsibility and communicates through standardized interfaces. This allows us to **replace or improve individual layers without redesigning the entire communication system**, provided they remain compatible.

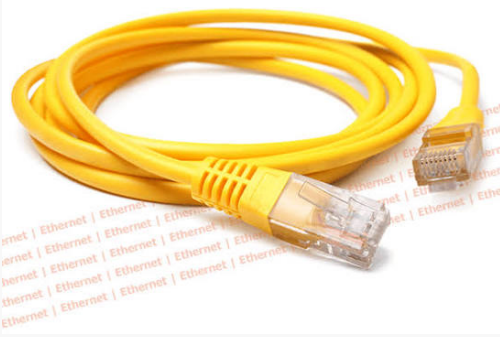
Example: A laptop can switch from Wi-Fi to Ethernet (cable) and continue accessing the same websites using IP, TCP, and HTTP.

OSI and TCP/IP: Two Ways to Describe Layers

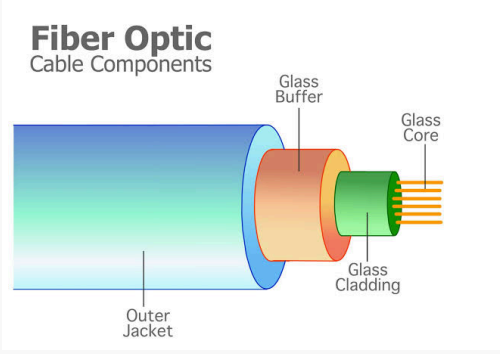


Physical Layer: Moving Bits Through Signals

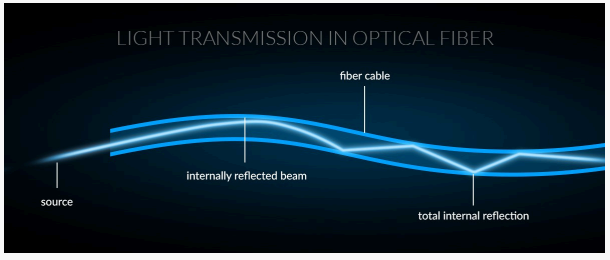
The **physical layer** turns bits into signals that can travel along one connection.



- **Copper cable:** electrical signals.
- **Fiber-optic cable:** light signals.
- **Wireless links (Wi-Fi):** radio signals.



This layer deals with **signal encoding, timing, data rate, range, and noise**.



Sending Bits Through Electrical Signals

How can we transmit a sequence of bits through a copper cable? The simplest approach is to **represent each bit using a different voltage level**.

For example:

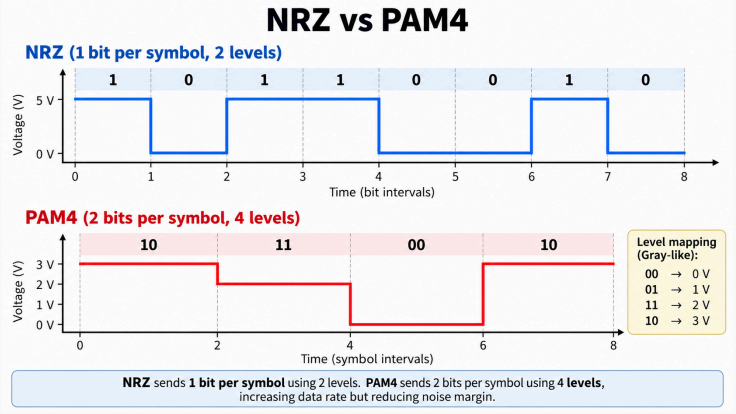
- Bit **0** → 0 V
- Bit **1** → 5 V

To transmit `1 0 1 1 0`, the sender generates the corresponding sequence of voltages.

This simple approach is a form of **line coding**, called *Non-Return-to-Zero (NRZ)*.

Modern high-speed communication systems use more sophisticated **encoding and modulation techniques** to:

- achieve higher throughput
- achieve robustness and synchronization



Serial vs. Parallel Data Transmission

Digital data can be transmitted using two main approaches:

Serial transmission: Bits are transmitted **sequentially**, one after another, over a communication path.

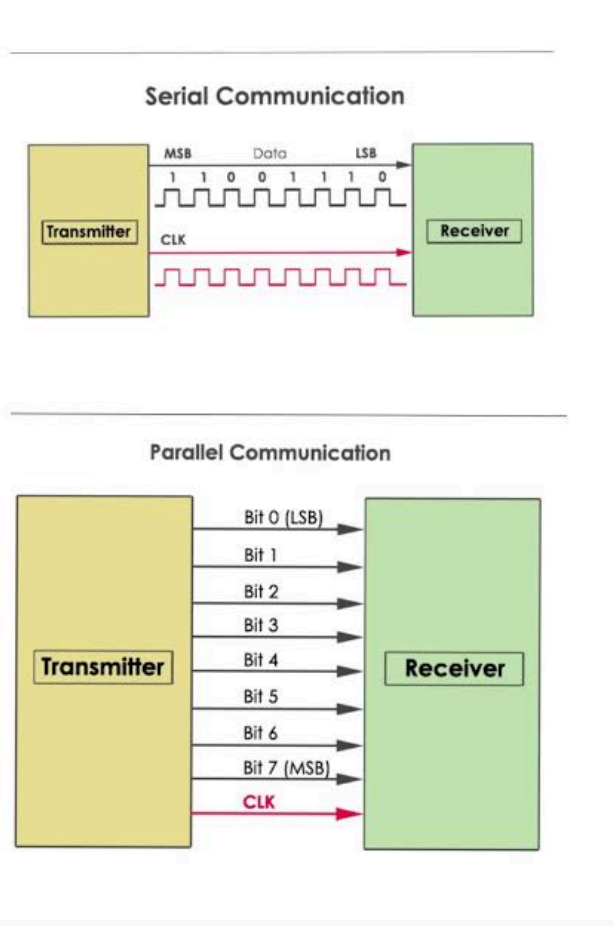
- Requires fewer physical connections.
- Common in modern communication interfaces such as **USB, Ethernet, and PCIe**.

Parallel transmission: Multiple bits are transmitted **simultaneously**, using separate communication lines.

- Can transfer several bits during the same clock interval.
- Requires more connections and careful synchronization.
- Common in internal computer interfaces, such as **memory buses**.

Why is serial transmission so common today?

At high speeds, parallel communication becomes more difficult because signals on different wires may experience different delays and interference.

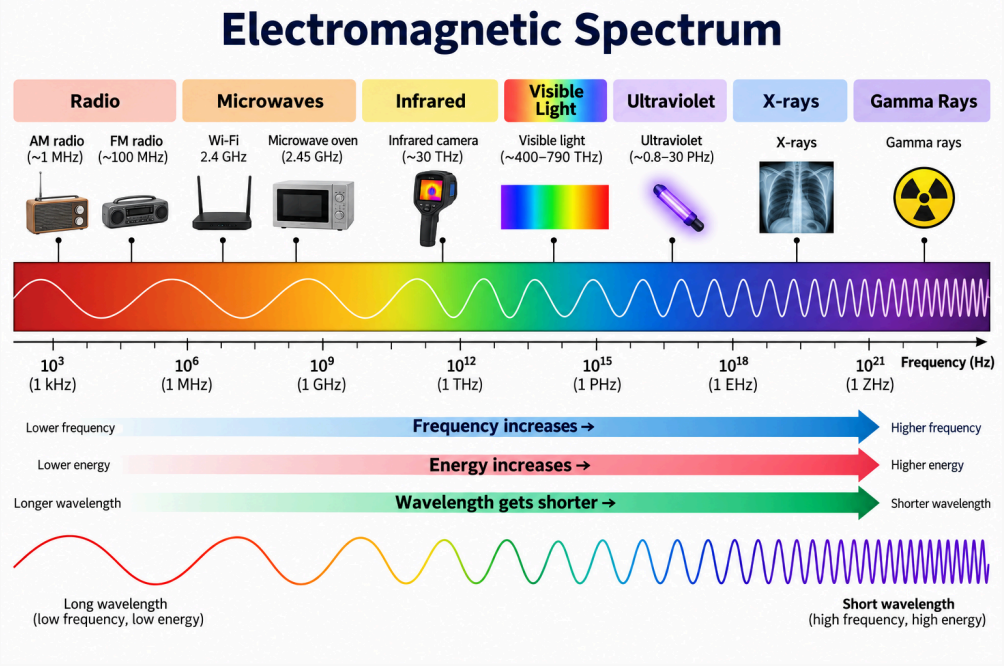


Wireless: Transmitting Data Using Radio Waves

Radio waves are electromagnetic waves that allow information to travel through space without a physical cable.

Different wireless technologies operate at different **radio frequencies**:

- **FM radio:** 88–108 MHz.
- **Wi-Fi:** 2.4, 5, and 6 GHz bands.
- **Cellular networks (4G/5G):** From hundreds of MHz to several GHz, and sometimes higher.



Wireless: Transmitting Data Using Radio Waves

To transmit information, we can use a **carrier wave**, an electromagnetic wave oscillating at a chosen frequency.

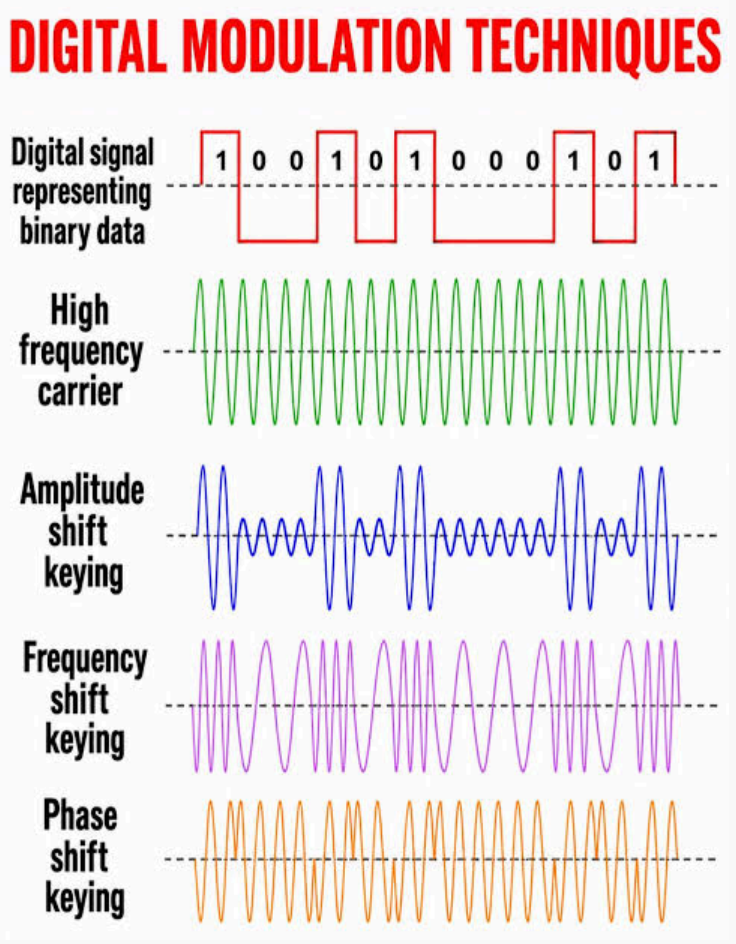
Wi-Fi - radio signal with carrier on 2.4Ghz

Modulation encodes information by changing properties of the carrier wave:

- **Amplitude:** The strength of the wave.
- **Frequency:** How rapidly the wave oscillates.
- **Phase:** The position within its oscillation cycle.

At the receiver, the information is recovered through **demodulation**.

MODEM = MOdulator–DEModulator: A device that modulates signals for transmission and demodulates received signals.

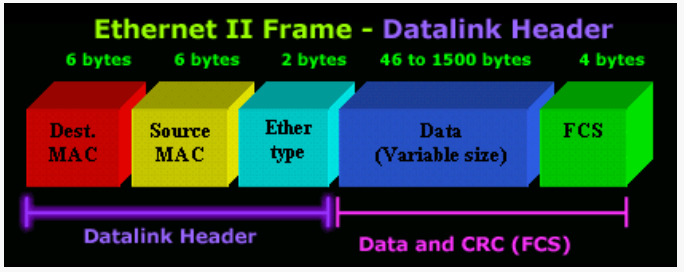


Data Link Layer: Communication on a Local Network

The **data link layer** organizes transmitted bits into **frames** and defines how devices exchange them over a local network (LAN).

Key responsibilities:

1. **Framing:** Organizes bits into frames, adding headers and trailers containing control information.
2. **Local addressing:** Uses **MAC addresses** to identify the sender and intended receiver within a local network.
3. **Medium access control:** Defines rules for when devices can transmit over a shared communication medium (e.g., Wi-Fi).
4. **Error detection:** Detects corrupted frames using mechanisms such as **CRC (Cyclic Redundancy Check)**.
5. **Local delivery:** Enables frames to reach their destination within the local network, for example through **Ethernet switches**.



MAC Addresses and Ethernet Switches

A **MAC address (Media Access Control address)** identifies a network interface for local link-layer delivery.







- A common Ethernet MAC address has **48 bits**.
- Example: `02:1A:2B:3C:4D:5E`.
- A **switch** learns which MAC addresses are reachable on its ports.
- It uses this information to send a frame toward the correct local destination.



Switches connect devices inside an Ethernet LAN.

Network Layer: IP (Internet Protocol)

The **data link layer** enables communication between devices on a local network. But how can we communicate with devices on **other networks**, potentially thousands of kilometers away?

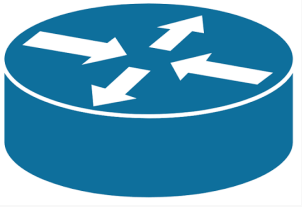
IP (Internet Protocol) enables communication across **interconnected networks and subnetworks**.

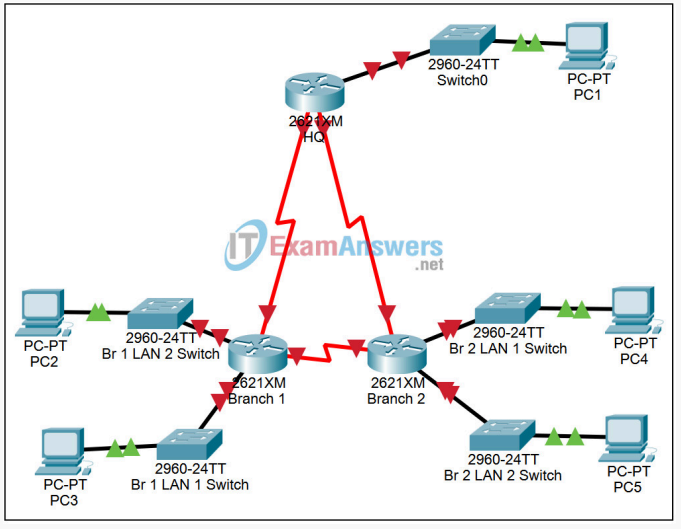
Its main principles are:

- **Packet-based communication:** Data are carried in units called **IP packets**.
- **Addressing:** Each packet identifies its source and destination using **IP addresses**.
- **Routing:** Devices called **routers** forward packets between networks toward their destination.

The Internet is a global network of interconnected networks that communicate using IP.

IP router - device that routes IP packets



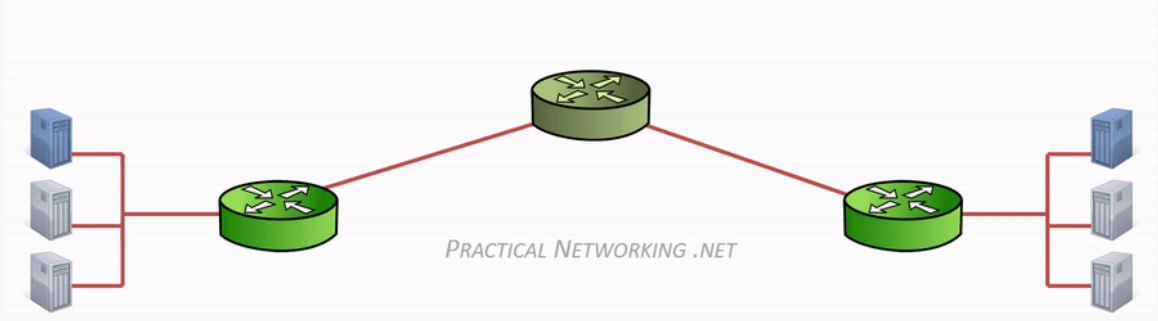


Network Layer: IP Addresses Identify Destinations

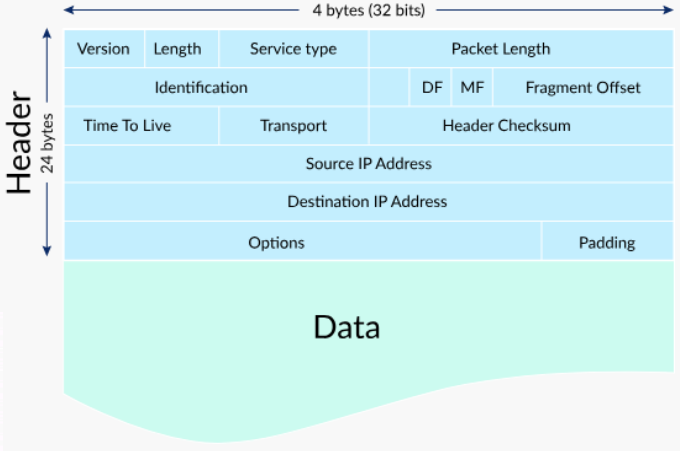
The **network layer** enables data to move between **different networks**.

IP (Internet Protocol) assigns logical addresses and carries packets:

- **IPv4:** 32-bit addresses, e.g., `192.168.1.25`.
- **IPv6:** 128-bit addresses, e.g., `2001:db8::25`.
- Each IP packet has a **source and destination IP address**.



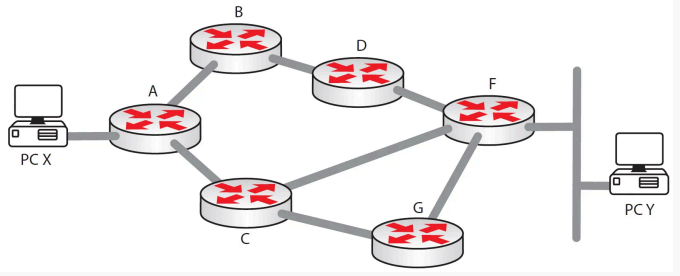
IPv4 packet



Routers: Finding a Path Between Networks

A **router** connects IP networks and forwards packets toward their destination.

- Within one local network, devices can communicate without an Internet connection.
- To reach another network, a device often sends packets to its **default gateway (router)**.
- A packet can pass through **multiple routers** before reaching the destination.



Example: How to reach the webpage: www.u-tokyo.ac.jp

- TRACEROUTE

Switch: mainly local frames. Router: IP packets between networks.

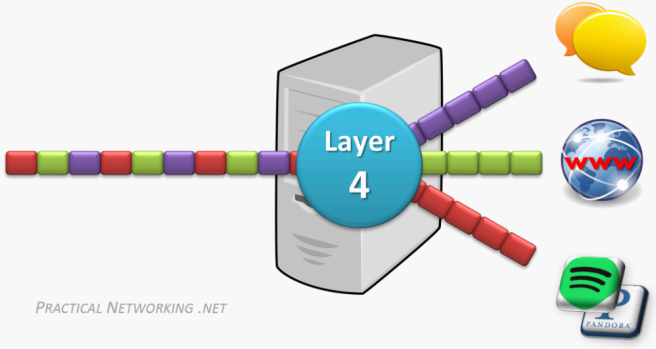
Transport Layer: Delivering Data to Applications

A computer can have **a single IP address** while running multiple network applications simultaneously, such as a web browser, email client, and video call.

But when data arrive at the computer, how does it know which application should receive them?

The **transport layer** solves this problem using **port numbers**:

- **IP address:** Identifies the destination device's network interface.
- **Port number:** Identifies a communication endpoint associated with an application.
- **Operating system:** Uses the transport protocol (TCP/UDP) and port number to deliver incoming data to the appropriate application.

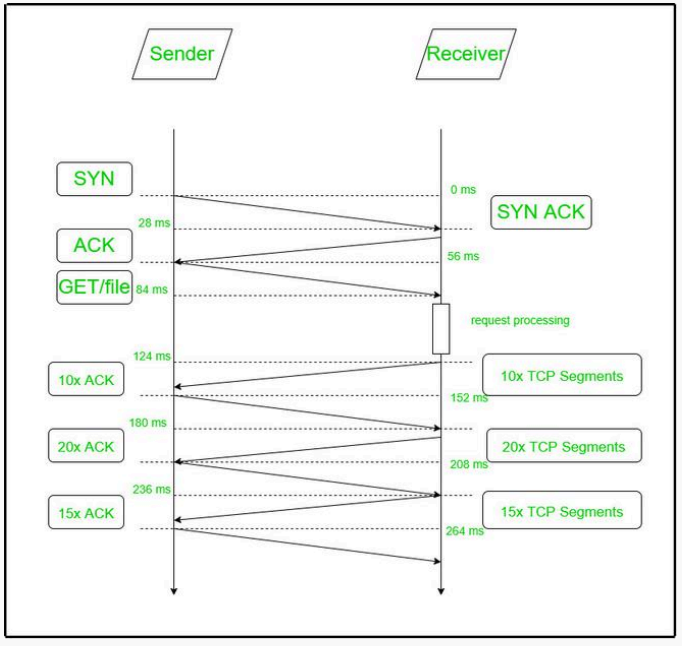


One IP address can support many simultaneous network communications through different ports.

TCP and UDP: Two Transport Approaches

TCP (Transmission Control Protocol)

- Connection-oriented; gives applications a **reliable, ordered byte stream**.
- Detects missing data and uses acknowledgments and retransmissions.
- Used when reliability is key (most of the time)
  - Solves reliability issues of the IP layer



UDP (User Datagram Protocol)

- Sends separate **datagrams** with minimal transport mechanisms.
- Does **not** itself guarantee delivery or order.
- Used when speed is critical (online gaming, torrent, streaming)

Neither is universally "better": applications have different needs.

Encapsulation, decapsulation and the TCP/IP

Encapsulation and **Decapsulation**.

- These terms refer to **how data is moved through the layers from top to bottom when sending and from bottom to top when receiving**.
- The data is handed from layer to layer, each layer adds the information it requires to accomplish its goal
  - Layer 4 will add a TCP header which would include a Source and Destination port
  - Layer 3 will add an IP header which would include a Source and Destination IP address
  - Layer 2 would add an Ethernet header which would include a Source and Destination MAC address



Application Layer: What Do the Messages Mean?

The **application layer** defines the services and message formats used by network applications.

Different services use different protocols:

- Web browsing: **HTTP/HTTPS** — retrieving web pages and other resources.
- Music streaming: Spotify — requesting and streaming music from remote servers.
- Email: SMTP/IMAP — sending and receiving emails.
- Domain resolution: **DNS** — translating domain names into IP addresses.

The same physical network can carry all these services.

Before a browser can contact a website, it often needs to find the server's IP address.

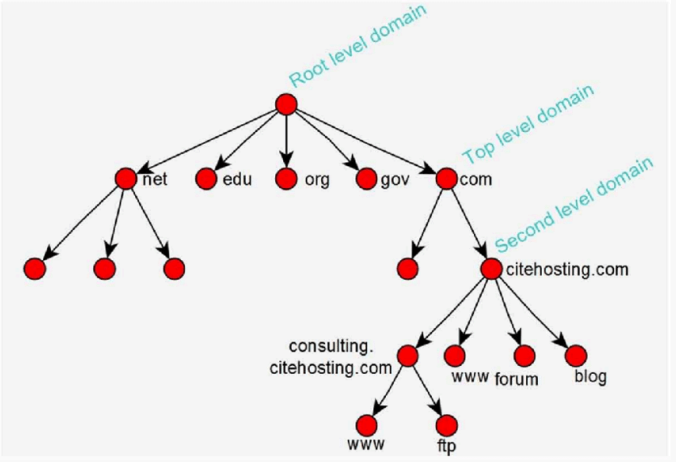
Domain Names: Addresses Humans Can Remember

Computers can communicate using **IP addresses**, but people prefer names such as `www.example.org`.

A **domain name** is a human-readable name in a hierarchical naming system.

In `www.example.org`:

- **`.org`** is the top-level domain (TLD).
- **`example.org`** is a domain name.
- **`www`** identifies a name beneath that domain (commonly used for a website).



One domain name can lead to more than one IP address.

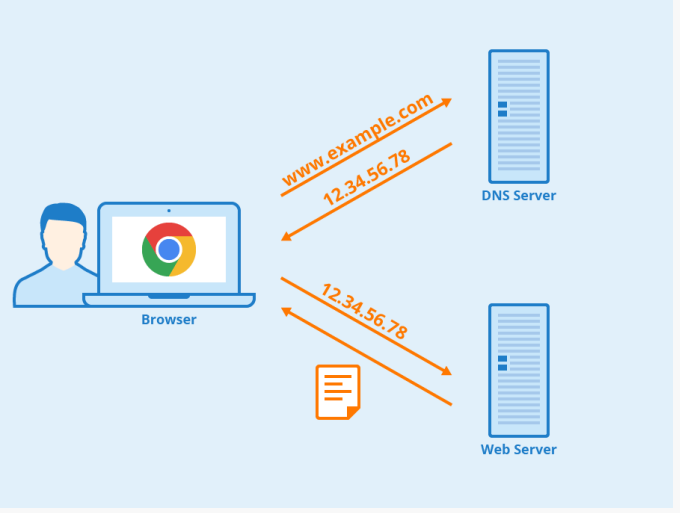
How does the computer translate the name into an address?

DNS: Finding the IP Address Behind a Name

DNS (Domain Name System) helps computers find information associated with domain names.

When you enter `www.example.org`:

1. The browser or operating system needs an IP address.
2. It asks a **DNS resolver** (or uses a saved answer).
3. The resolver returns one or more relevant IP addresses.
4. The browser can then contact the destination server.



DNS answers a naming question; it does not download the webpage.

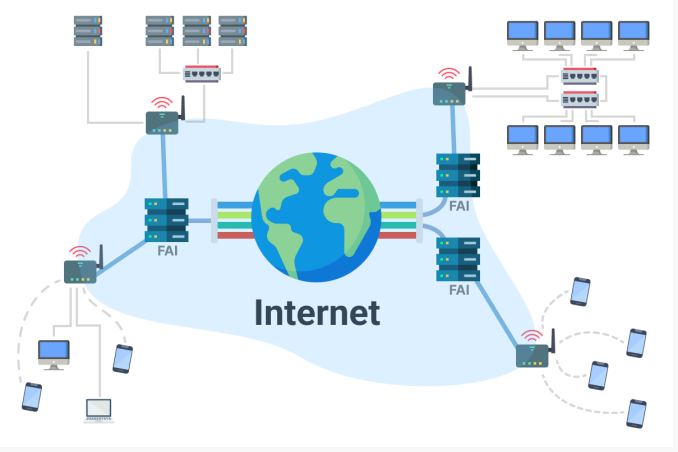
DNS answers may be **cached** so a lookup is not repeated every time.

The Internet and the Web Are Not the Same

The Internet is the global infrastructure of interconnected networks communicating through IP.

The **World Wide Web (WWW)** is a system of interconnected resources, accessible through Web technologies over the Internet.

- The **Internet** carries packets for many different services.
- The **Web** includes websites, webpages, links, images, and Web applications.
- Email, video calls, and online games can use the Internet without being webpages.



The Web runs on the Internet; it is not the Internet itself.

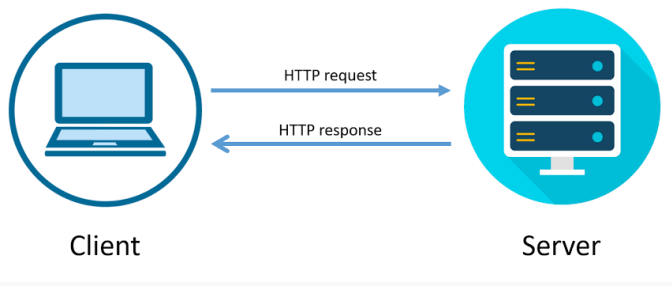
Web Browsers: Clients and Servers

A **web browser** is an application used to navigate the **World Wide Web**, access webpages, and follow links between resources.

Examples: **Google Chrome, Mozilla Firefox, Safari, Microsoft Edge**.

Web browsing follows a **client-server model**:

- **Client (Web browser):** Requests resources, such as webpages, images, and documents.
- **Web server:** A computer program that receives requests and delivers the requested resources.
- **HTTP/HTTPS:** The application-layer protocol used to exchange requests and responses.



Webpages are commonly written in **HTML (HyperText Markup Language)**, which describes their content and structure.

The browser interprets HTML and displays the webpage to the user.

Understanding Web Addresses (URLs)

Every resource on the Web can be identified using a **URL (Uniform Resource Locator)**.

A URL tells the browser **which resource to request and how to access it**.

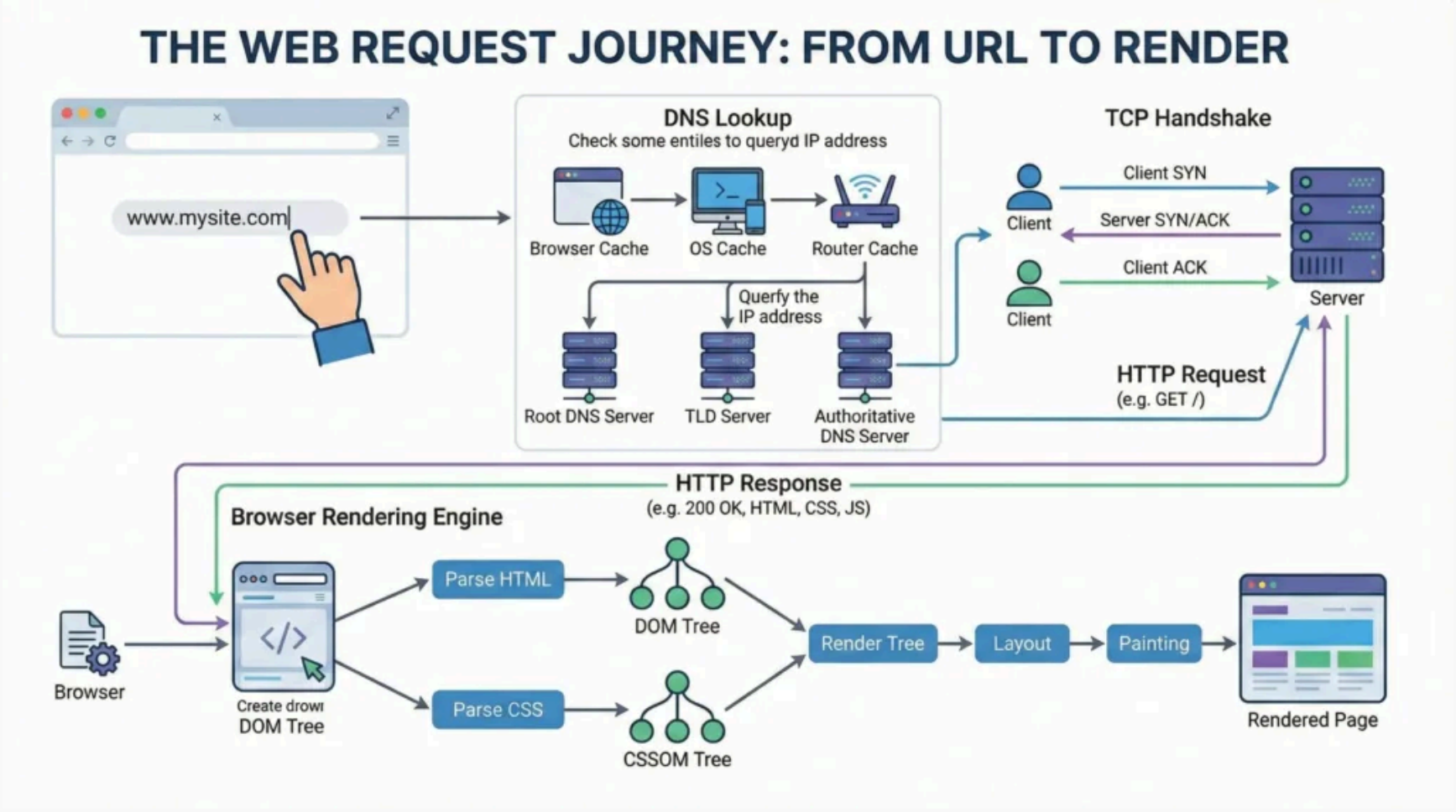
Example: `https://openlibrary.org/search?q=computer`

- **Scheme:** `https` — the protocol used to access the resource.
- **Host:** `openlibrary.org` — the domain name identifying the server.
- **Path:** `search` — the requested resource.
- **Query:** `q=computer` — additional parameters sent with the request.

Webpages also contain **hyperlinks**, allowing users to navigate between resources by following their URLs.

But how does the browser find a server using a domain name instead of an IP address?

What Happens When You Open a Webpage?



What Happens When You Open a Webpage?

Suppose you enter `https://www.example.org/` into your browser.

1. **URL:** The browser identifies the website and the requested resource.
2. **DNS (Domain Name System):** Resolves the domain name into an **IP address**.
3. **Connection:** The browser establishes communication with the Web server using its IP address. For HTTPS, the connection is secured.
4. **HTTP request:** The browser requests the webpage from the server (e.g., using GET).
5. **HTTP response:** The server returns the requested resource, typically an **HTML document**.
6. **Rendering:** The browser interprets the HTML, retrieves additional resources if needed, and **displays the webpage**.

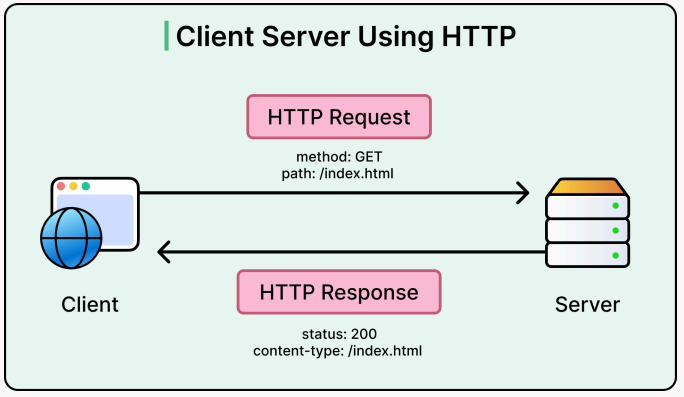
From entering a URL to displaying a webpage, several network protocols work together!

HTTP: The Hypertext Transfer Protocol

HTTP (Hypertext Transfer Protocol) is an **application-layer protocol** that defines how Web clients and servers request and exchange resources.

HTTP follows a **client-server, request-response model** and defines:

- **Request methods:** Specify the action requested by the client.
  - `GET` — retrieve a resource.
  - `POST` — submit data for processing.
  - Other methods include `PUT` and `DELETE`.
- **Response status codes:** Indicate the result of a request.
  - `200 OK` — request successful.
  - `404 Not Found` — resource not found.
  - `500 Internal Server Error` — server encountered an error.
- **Content formats:** Responses can carry **HTML, images, videos, PDF documents, JSON**, and other data.



Important HTTP Response Status Codes

Every HTTP response includes a **three-digit status code** indicating the result of the client's request. http.cat

🟢 2xx – Success

- The request was successful.

🔵 3xx – Redirection:

- The resource has moved or further action is needed.

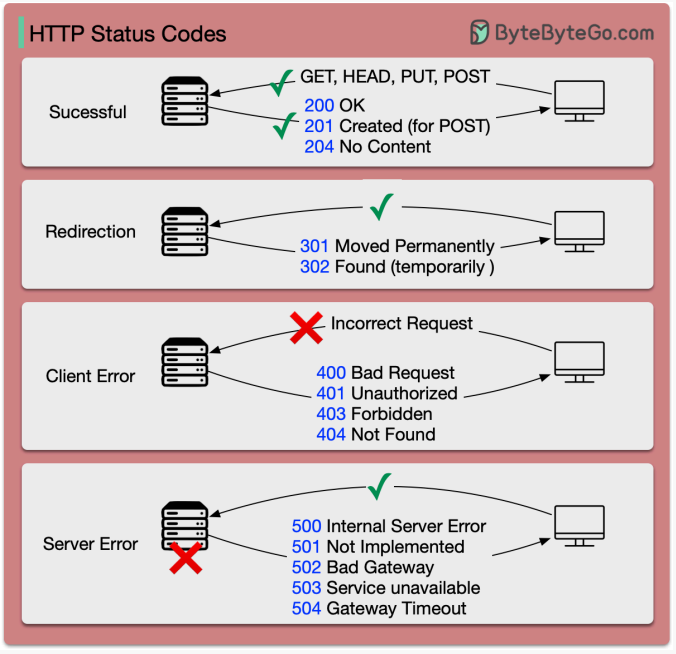
🟠 4xx – Client Error:

- There is a problem with the request or access to the resource.

🔴 5xx – Server Error:

- The server failed to complete the request.

HTTP Code	Status	Description	Test
🟢 200	OK	The server successfully processes the request.	Test
🔵 301	Moved Permanently	The resource has moved to a new permanent URL.	Test
🟠 403	Forbidden	The server denies access to the resource.	Test
🟠 404	Not Found	The requested resource cannot be found.	Test
🔴 500	Internal Server Error	The server encountered an unexpected error.	Test

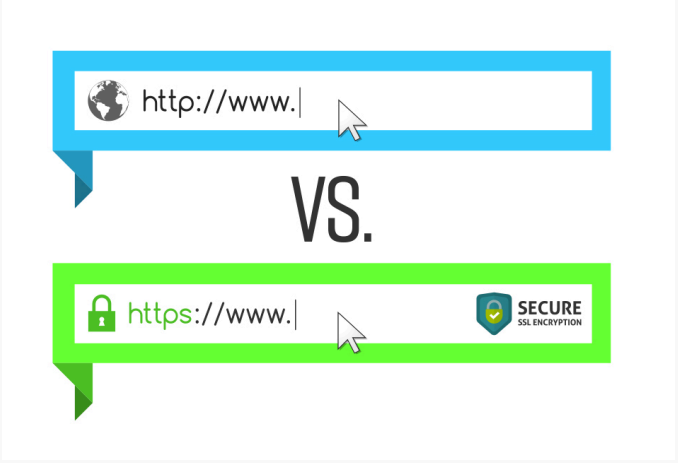


HTTP: The Hypertext Transfer Protocol

HTTP is not secure: It does not provide encryption or protection against intercepted or modified communications.

HTTPS = HTTP protected by TLS (Transport Layer Security), providing:

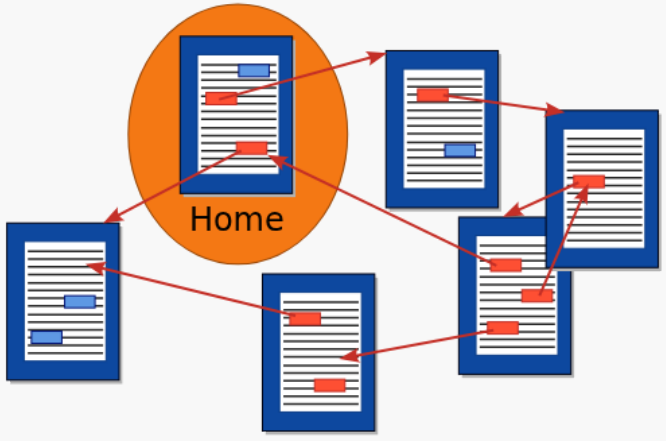
- **Confidentiality:** Encrypts the exchanged data.
- **Integrity:** Protects data against undetected modifications.
- **Server authentication:** Verifies the server's identity using a digital certificate.



Hypertext: Connecting Information Through Links

Traditionally, documents are read in a **linear sequence**, from beginning to end.

Hypertext allows information to be connected through **hyperlinks**, enabling readers to navigate between documents or different parts of the same document.



- A **hyperlink** is a clickable reference to another resource, identified by a URL.
- Clicking a hyperlink allows the browser to navigate to the referenced resource.
- Documents can link to other documents, creating an **interconnected network of information**.

Hyperlinks are what make the World Wide Web a "web" of interconnected information.

HTML: The Language of Webpages

HTML (HyperText Markup Language) is the standard markup language used to describe the **content and structure of webpages**.

HTML uses **tags** to identify different elements:

- `<h1>` Heading, `<p>` Paragraph, `<img>` Image, `<a>` Hyperlink to another resource.

Example HTML:

```
<h1>Wood Materials</h1>
<p>Learn about <a href="oak.html">oak</a>.</p>
```

What the browser displays:

WOOD MATERIALS

Learn about oak.

HTML describes the **structure and meaning** of webpage content. Other technologies, such as **CSS** and **JavaScript**, can control its appearance and interactive behavior.

Search Engines Help Us Find Information on the Web

The Web contains a huge number of webpages and resources.

A **search engine** helps users find relevant webpages by:

- **crawling** the Web and following links,
- **indexing** webpage content,
- returning a **ranked list of results** for a user's query.



Examples: Google, Bing, DuckDuckGo.

A search engine is a **Web service built on top of the Web**.
It is not the same as the Internet or the Web itself.

Famous Webpages in the History of the Web

Some websites became famous not only because of their content, but also because they represent important moments in the history of the **World Wide Web**.

- **1991 — CERN: The First Website**
  The first website explained the World Wide Web project itself.
  Link: https://info.cern.ch/hypertext/WWW/TheProject.html
- **1994 — Yahoo!**
  One of the earliest popular ways to navigate the Web, originally based on a human-curated directory of websites.
  Link: https://www.webdesignmuseum.org/gallery/yahoo-1994
- **1998 — Google**
  A search engine that changed how people find information on the Web.
  Link: https://www.webdesignmuseum.org/gallery/google-1998
- **2001 — Wikipedia**
  A collaborative online encyclopedia built as a large interconnected hypertext system.
  Link: https://www.wikipedia.org/
- **2005 — The Million Dollar Homepage**
  A famous webpage where advertising space was sold pixel by pixel.
  Link: https://milliondollarhomepage.com/

The WWW: Putting Everything Together

The **World Wide Web (WWW)** is a **system of interconnected resources**, accessible through Web technologies over the Internet.

Three fundamental technologies make the Web possible:

- **URL (Uniform Resource Locator):** Identifies a resource and how to access it.
- **HTTP/HTTPS (Hypertext Transfer Protocol):** Defines how Web clients and servers request and exchange resources.
- **HTML (HyperText Markup Language):** Describes the structure of webpages and allows them to contain hyperlinks.

These technologies work together with:

- **Search engines:** Help users discover Web resources
- **Web browsers and servers:** Request, deliver, interpret, and display resources.
- **DNS and IP:** Help locate and communicate with Web servers.
- **The Internet:** Provides the underlying network infrastructure.

The Web is not a single server or network, but a worldwide system of linked information and services accessible through common standards.