WEEK 4 - PART 3

DECODING NETWORK TRAFFIC -

HOW DATA FLOWS OVER THE INTERNET

LINNAEUS
UNIVERSITY

SUMMER 2024

TABLE OF CONTENTS

WHAT IS A NETWORK?

A network refers to a collection of interconnected devices or systems that can communicate and share resources with each other. Networks typically consist of **computers, servers, routers, and other devices that are connected to each other, either physically (using cables) or wirelessly (using Wi-Fi or other wireless technologies).**

The purpose of a network is to enable **communication and the exchange of data between devices** - by connecting devices together, users can share information with each other, whether it's within a **local area (such as a home or office network)** or over a **wide geographical area (such as the internet)**. The **internet is a global network of networks,** connecting millions of individual networks worldwide. It is the largest and most comprehensive network system, allowing devices across the globe to exchange information with each other.

The internet links various smaller networks, such as those within homes, businesses, and institutions, and then **routes that data,** sending it to where it's directed. This is done via a **complex system of routers and servers that direct data from one network to another** until it reaches its destination.

A computer is a crucial component within a network.
When a PC connects to a network, it becomes part of that network, enabling it to send and receive data to and from other devices on the same network and grants access to networked services.

Networks are the foundation of our digital interactions.
They support real-time interaction, whether accessing local files or streaming global content. As technology evolves, these networks will expand, reshaping how we work, learn, and communicate.

NETWORK CONNECTIONS

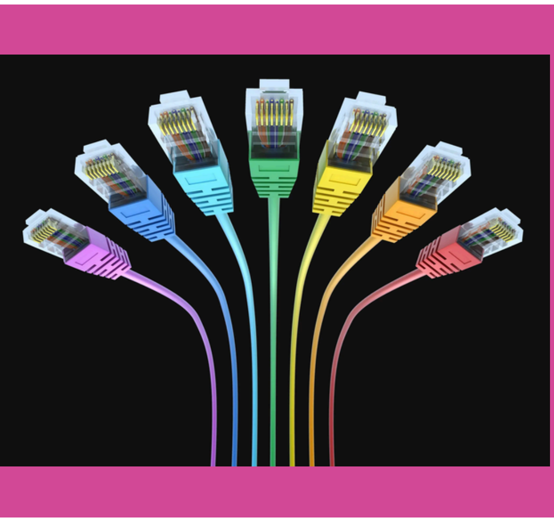
HOW NETWORKS CONNECT

Ethernet and Wi-Fi are two different methods for connecting devices to a network, each with its own characteristics and use cases.

Ethernet offers a stable and high-speed wired connection ideal for stationary devices and environments with high data demands, while Wi-Fi provides flexible, wireless connectivity suitable for mobile devices and areas where running cables is impractical.

ETHERNET

Ethernet is a wired network technology that uses physical cables to connect devices to a network.



It generally provides faster speeds and more reliable connections than Wi-Fi, **providing reliable and predictable data transfer speeds and network performance. This means that the quality of the connection remains stable even when multiple devices are using the network.**

Unlike Wi-Fi, which uses radio waves and can be disrupted by physical obstacles (like walls) and other electronic devices (such as microwaves or cordless phones), **Ethernet cables transmit data through physical wires.** These cables are less susceptible to external interference because they are **shielded from electromagnetic signals and physical barriers.**

Additionally, Ethernet typically has **lower latency.** Latency refers to the time delay between initiating a request and receiving a response in a network or computing system.
This makes an Ethernet connection ideal for activities that require real-time responses, such as gaming or video conferencing.

While setting up an Ethernet network **involves running physical cables between devices and network equipment** – which can be cumbersome in some settings –it ensures stable and secure connections.

WI-FI

Wi-Fi is a wireless network technology that uses radio waves to connect devices to a network, allowing for mobility and flexibility without the need for physical cables.

It provides a wireless connection through a wireless router or access point, which communicates with devices using radio signals.

Wi-Fi transmits data through a series of steps.

First, a device **converts the data into a digital format of binary bits (0s and 1s).**
To send this binary data over radio waves, it **needs to be converted into a form that can be transmitted wirelessly.** This is where modulation comes in.

Imagine a steady radio wave (like a constant hum) that will carry our data. This wave is called the **carrier signal.** This is a steady, continuous radio wave with a specific frequency serves as the base wave that carries the data.

We **encode binary data into radio signals by adjusting the carrier signal** in several key ways, which embeds the binary data into the signal.
Essentially, **the data is hidden within the signal's properties, such as its frequency**, so that it can be sent over a communication channel (like radio waves) and recovered at the other end.

The carrier signal is adjusted in a few key ways.
Amplitude modulation alters the strength of the wave, varying its intensity to represent the data.
Frequency modulation involves changing how rapidly the wave oscillates, and adjusting the speed of the wave's cycles.
Phase modulation involves timing adjustments by shifting the wave's peaks and troughs.

These changes encode the binary data into the carrier signal, allowing it to be transmitted wirelessly.

The **modulated data is sent as radio signals through antennas on the Wi-Fi router or access point.**

On the **receiving end, the router's antennas capture these radio signals sent by the computer's wireless network adapter.**

The router then **demodulates the signals**, decoding them and converting them back into their original digital format. Once the original data is retrieved, the router processes it to **route it to its intended destination,** whether that's another device on the same network or an external server on the internet.

While modern Wi-Fi standards, such as Wi-Fi 6, offer high speeds, they can be less consistent than wired Ethernet connections due to varying factors such as **the range of the signal, distance from the router and physical obstructions.**

Wi-Fi is also s**usceptible to interference from other wireless devices, networks, and environmental factors,** which can further impact connection stability and performance.

Latency in Wi-Fi networks is often higher compared to Ethernet, which can impact the performance of applications that require low delay.

Setting up a Wi-Fi network is relatively simple and flexible, as it eliminates the need for physical cables and allows for easy expansion and reconfiguration of the network.

NETWORK TYPES

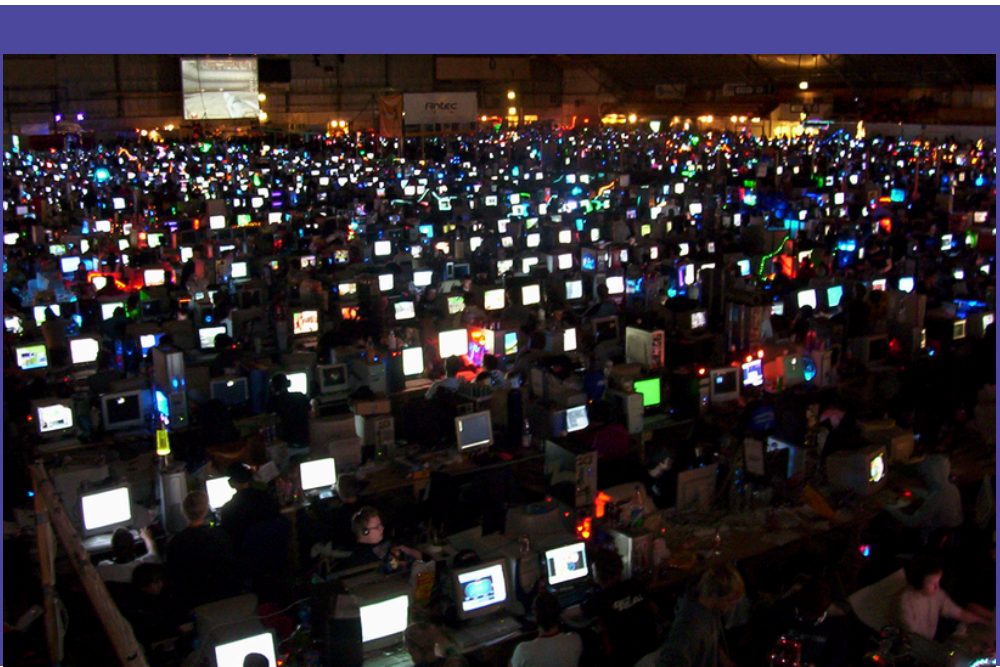
**There are several types of networks, each designed for specific purposes and scales. Each network type is engineered to meet specific needs, whether it's for limited,
high-speed communication or extensive, long-distance connectivity**

LAN, WAN AND MORE

A **LAN is a network that connects devices within a limited geographical area**, such as a home, office, or school, typically using technologies like **Ethernet and Wi-Fi** for local communication.

LANs connect to the internet through a gateway device, usually a router or a modem. When a device within the LAN, such as a computer or smartphone, requests data from the Internet, the router or modem sends the request through an Internet Service Provider (ISP).

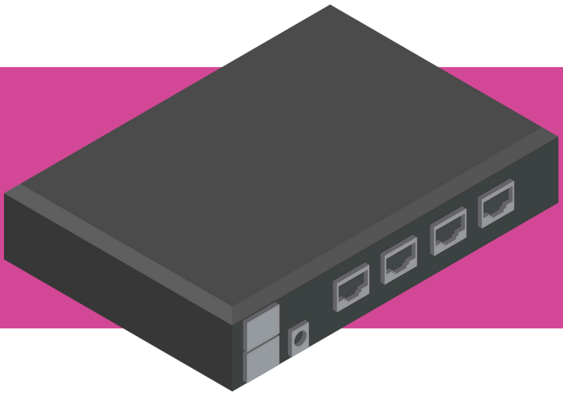
When data returns, the router or modem receives it from the ISP and **directs it to the appropriate device within the LAN.** This setup allows devices within the LAN to access the internet.



A switch is a network device—a hardware component essential for communication and data exchange—**that connects multiple devices within a LAN.** It manages and directs data traffic **specifically within the LAN,** providing efficient communication and resource sharing among the connected devices.

Unlike **older network hubs, which broadcast data to all connected devices,** a **switch intelligently directs data only to the specific device intended to receive them.**

Many modern routers include **built-in switches,** which allow them to **connect multiple devices to the same network through wired Ethernet connections.** This is common in home and small office routers, where a single device can manage both routing functions (directing traffic between the local network and the internet) and switching functions (connecting local devices).



In contrast, **a WAN (Wide Area Network) spans larger distances and connects multiple LANs together,** often using public infrastructure like the Internet. WANs frequently use the existing global network of servers, routers, and data links provided by ISPs to link LANs and extend their reach beyond local boundaries.

Metropolitan Area Networks (MANs) span a city or large campus, providing high-speed connectivity within a more localized region.

Additionally, **personal area networks (PANs) connect devices in close proximity, such as through Bluetooth,** while the global network known as the Internet ties together networks of all sizes across the world.

ROUTING & DATA TRANSMISSION

Routers are devices that connect multiple networks and determine the best path for data transmission by routing - the process of **directing network traffic efficiently and forwarding data** between different networks. Using routing tables, routers decide how to direct data from their source to their destination, optimizing network performance and reliability.

ROUTING & ROUTING TABLES

Routers are devices that connect multiple networks and determine the best path for data transmission by routing - the process of **directing network traffic efficiently, and forwarding data** between different networks.

Using **routing tables - databases stored in routers - that contain information and provide details on available network paths.**
These tables help the routers make routing decisions to direct and forward data, finding the most efficient path from their source to their destination.

When data is transmitted over the internet, **each router it encounters examines the destination IP address of the data** - the address where the data should be delivered - and then looks up this address in its routing table.

It then **forwards the data** along the best route **toward the next stop**, known as the "next hop," on its way to the final destination.

The **"next hop" refers to the next device—such as another router, switch, or network device**—along the path.

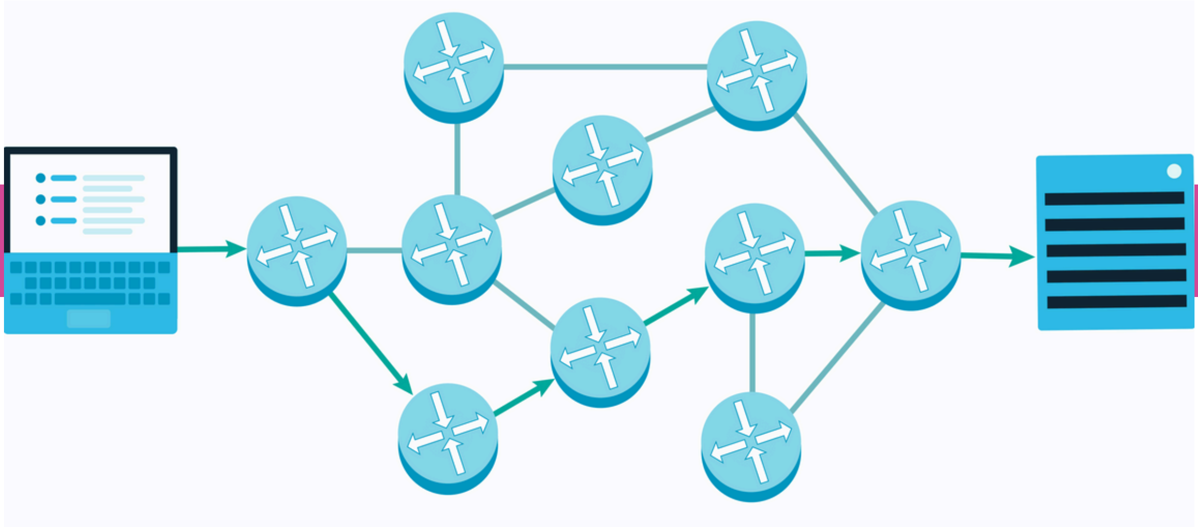
Each router consults its routing table, which specifies the next hop and includes details on available routes and their metrics. The data continues to be forwarded from hop to hop until it reaches its endpoint.

The tables are dynamically updated using protocols that exchange information with other routers about network changes, ensuring that routes remain accurate.

NETWORK PATHS

Network paths refer to the various routes that data can take through interconnected networks.

In a network, data does not always follow a single fixed route.



Instead, it may **travel through various intermediate routers, switches, and network links.** These paths can vary in length and complexity, depending on the network's architecture and current conditions.

Routers **continuously evaluate network paths using cost metrics** associated with each route to assess and **compare different options** for data transmission. These metrics include factors such as bandwidth, latency, network congestion, and error rate. This process helps in directing data along the most suitable path and optimizing network efficiency for various types of data.

DATA PACKETS

When data is transmitted over a network, it is broken down and taken apart into smaller, manageable pieces known as data packets. This packetization helps in efficient transmission and routing of data between networks.

Before transmission, large data files are divided into smaller units called packets. This is the format in which the data is routed by routers, as it hops between network devices along network paths.
Once the data is segmented into packets, **each packet is sent through the network** - packets first travel through the local network, and are then forwarded to the router at the edge of the local network.

Routers guide packets destined for external networks by forwarding them through a series of interconnected networks and routers to the next hop on their path toward the final destination.

The **data is too large to be sent all at once, so breaking it into packets ensures it can be efficiently transmitted** to it's the destination without overwhelming the network. Each packet contains a portion of the original data, along with additional information needed for routing and reassembly, such as metadata containing its source and destination IP address.

PACKET - EMAIL EXAMPLE		
HEADER	PAYLOAD	TRAILER
SENDERS IP ADDRESS RECIEVERS IP ADDRESS PROTOCOL PACKET NUMBER	ACTUAL DATA	DATA TO SHOW END OF PACKET ERROR CORRECTION
96 BITS	896 BITS	32 BITS

When packets reach their destination, they are reassembled into the original data file. Reassembly is the process by which these smaller data packets, which were originally part of a larger file or message, are put back together in their correct order at the destination.

Each packet may take a different route to reach the destination, and they might not arrive in the order they were sent.
The receiving device collects all the packets and uses the metadata within them, such as sequence numbers, **to reassemble them in the correct order, recreating the original file or message.** It also checks for errors during this process.

Once the data is fully reassembled, it is presented to the user or application, ensuring that the data received is exactly the same as the data that was sent.

ADDRESSES IN NETWORKING - IP, NAT, DNS, & URLS

NAT

Routers implement Network Address Translation (NAT) to convert private IP addresses used within a local network into a public IP address assigned by an Internet Service Provider (ISP).

An **IP address is a unique numerical identifier** assigned to each device connected to a network, enabling it to communicate with other devices by specifying its location.

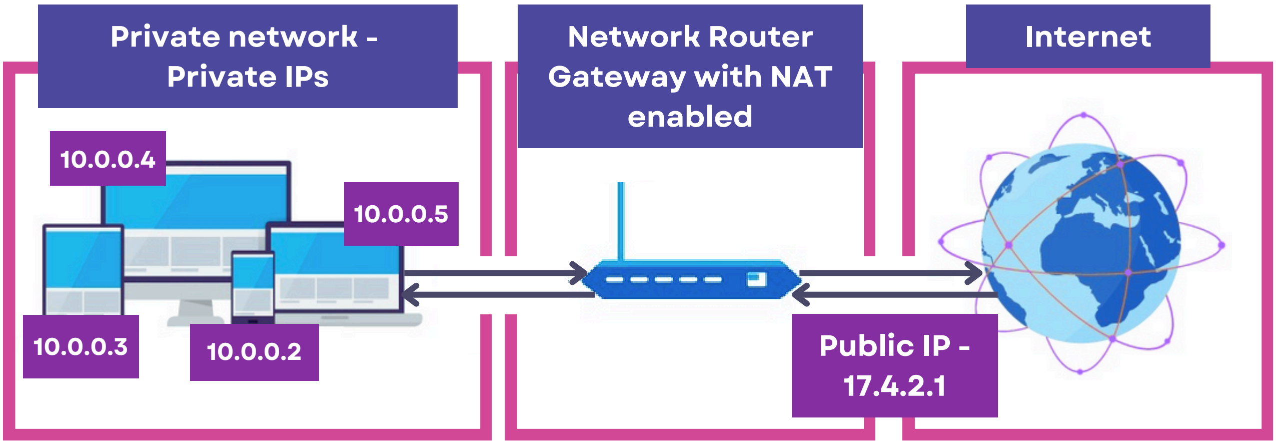
Private IP addresses are used within internal networks and are not routable over the internet, allowing multiple devices within a local network to share a single public IP address. **Public IP addresses**, on the other hand, are assigned by an ISP and are used to **identify and communicate with devices over the Internet.**

When a device on your local network wants to communicate with a device on the internet, it sends data to the router. **The router uses NAT to replace the private IP address in the outgoing data with its public IP address.**

When the router receives responses from the internet, it translates the public IP address back to the corresponding private IP address and forwards the data to the correct device within the local network.

For example, when **Device A with a private IP address of 10.0.0.4** wants to access a website on the internet, it sends the request to the router. T**he router replaces Device A's private IP address with its public IP address (e.g., 17.4.2.1)** and sends the request to the website server.

The website server **sends the response back to the router's public IP address. The router then uses NAT to map the response back to Device A's private IP address** and forwards the data to Device A.


Private network - Private IPs
10.0.0.4
10.0.0.5
10.0.0.3
10.0.0.2
Network Router Gateway with NAT enabled
Internet
Public IP - 17.4.2.1

NAT allows multiple devices on a local network to share a single public IP address, conserving IP addresses.

Additionally, it enhances security by hiding the internal network structure and IP addresses from the outside world, making it harder for external entities to directly access internal devices.

IP ADDRESSES IN DEPTH

An IP address is a **unique identifier** assigned to **each device on a network**, allowing devices to locate and communicate with each other.

An IP address is divided into two main parts: the network portion and the host portion. The **network portion identifies the specific network** to which the device belongs, while the **host portion identifies the individual device** within that network.

There are two types of IP addresses: private and public.

Private IP addresses are used within a **local network, such as a home or office network**, and are not unique on the global internet. **Multiple networks can have the same private IP addresses.** Examples of private IP addresses include ranges like 192.168.x.x, 10.x.x.x, and 172.16.x.x to 172.31.x.x.

Devices like computers, smartphones, and printers within the same local network use private IP addresses to communicate with each other.

Public IP addresses, on the other hand, are used to identify devices on the global internet and are unique across the entire internet.

Your ISP assigns a **public IP address to your router, which then represents your entire local network to the outside world.**

NETWORKING PROTOCOLS

Networking protocols are standardized rules and procedures that define how data is transmitted and received over networks. They govern various aspects of data exchange, including formatting, sequencing, error handling, and data integrity, enabling seamless and interoperable communication between different systems and applications.

They provide a **standard method of communication** to ensure compatibility and reliable data transfer, providing the foundation for data transmission and routing.

TCP/IP PROTOCOL

TCP (Transmission Control Protocol) is a key networking protocol that guarantees reliable, ordered, and error-checked delivery of data between devices, establishing a connection before transmitting data and ensuring reliable and ordered delivery of data packets.

TCP segments the data into manageable packets into smaller packets for transmission, and is responsible for ensuring that data packets are transmitted reliably between devices. **It guarantees that packets arrive intact, in the correct order, and without errors, even if they arrive out of sequence.**

Each packet includes a sequence number, which helps in reassembling the data in the correct order. **TCP handles reordering and error correction**, ensuring that the data is correctly assembled and delivered to the receiving application.

Before data transfer begins, **TCP establishes a connection between the sender and receiver**, involving exchanging signals to confirm that both ends are ready for communication. This is called the ”TCP handshake”.

The TCP handshake is a three-step process used to establish a connection between a client and a server. It involves the exchange of synchronization (SYN) and acknowledgment (ACK) messages: **the client sends a SYN message, the server responds with a SYN-ACK message, and the client completes the handshake with an ACK message**, establishing a reliable connection for data transfer.

IP (Internet Protocol) is a core networking protocol, and is responsible for addressing and routing data packets across networks, specifying the destination for each packet to ensure it reaches the correct device.

The IP protocol directs packets across networks by attaching destination IP addresses to each packet. As packets travel through the internet, each router uses this destination IP address to look up its routing table.

The TCP/IP model forms the basis of the internet and enables the World Wide Web to function.

HTTPS/HTTP

HTTP (Hypertext Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure) operate on top of the TCP/IP suite.

They rely on TCP/IP protocols to handle the actual data transmission and provide the basic communication and routing mechanisms.
HTTP and HTTPS are both protocols that define the rules for how web data should be formatted, retrieved, and sent between web browsers and servers. They also define how servers should respond to these requests, including the format of the data (e.g., HTML, JSON)

HTTPS builds on HTTP by adding a layer of security through encryption, ensuring that data transmitted between the browser and server is secure and protected from eavesdropping or tampering. This encryption is achieved using protocols like SSL/TLS, which encrypt the data being sent to protect it from unauthorized access.

OTHER PROTOCOLS

In addition to HTTP, there are various other protocols that govern different aspects of network communication. Each protocol operates on top of the TCP/IP stack, serving distinct purposes in the realm of network communication.

FTP (File Transfer Protocol) is used for transferring files between devices, **SMTP (Simple Mail Transfer Protocol)** for email transmission, and **SSH (Secure Shell)** for secure remote access to systems.

WebSocket is a network protocol that is designed to provide interactive communication between a client (such as a web browser) and a server over a single, long-lived connection.

WebSocket provides a way to maintain a **persistent connection for real-time data exchange**, which is especially useful for applications requiring constant updates, such as **chat applications, live streaming, and online gaming.**

WebSocket begins by establishing a HTTP connection, and once established, it switches to the WebSocket protocol. Once the server agrees to upgrade the connection, the actual WebSocket communication is established over a TCP connection, relying on the TCP handshake process.

SSH (Secure Shell) lets you securely connect to another computer over the internet or a network. It's like having a secure, private line to access and control a remote computer, such as a server, from your own computer.
It ensures that the data sent over the network is encrypted and that the communication between the your local machine and the the remote system or computer is secure.

SSH allows you to safely upload or download files from the remote computer. **Additionally, after connecting you can run commands on the remote computer just as if you were sitting in front of it**, such as check files, run programs, or make changes.
SSH is both a protocol and a tool that leverages that protocol to enable secure remote access and management of computers.

BROWSERS, WEB SERVERS, URLS AND DNS

BROWSERS

A web browser is a software application that enables users to view websites on the internet. It acts as a client that displays data it receives from the server when requesting resources.
This communication between the client and server is possible thanks to web servers **using protocols like HTTP and HTTPS.**

The HTTP handshake is used to establish a connection between a client and server before data can be exchanged.

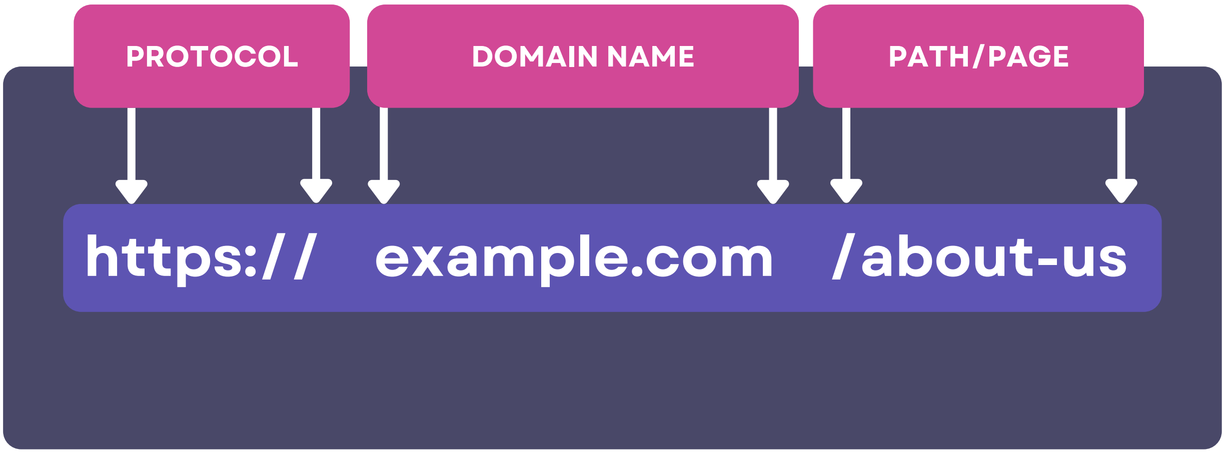
The HTTP protocol can be upgraded to HTTPS to ensure secure communication between a client and server. This process begins with the client sending an HTTP request to the server. In response, the server indicates that it supports HTTPS and advises the client to switch to a secure connection. **The client then redirects to the HTTPS version of the URL.**

Once the secure connection is established, data is exchanged using HTTPS, ensuring that it remains encrypted and protected from interception and tampering, and guaranteeing that data transmitted between the client and server remains secure.

URLS

A URL (uniform resource locator) is the address used to locate a resource - such as webpage or file (such as video, image, GIF, etc.) - on the internet. When you enter a URL into a web browser, the browser sends a request to the corresponding web server to retrieve the web page associated with that URL.

A URL consists of several key components: **the protocol (HTTP or HTTPS) specifies how data is transferred; the domain name (like www.example.com) identifies the server hosting the resource; and the path (e.g., /page1.html) points to the specific resource on the server.**



PROTOCOL

DOMAIN NAME

PATH/PAGE

https:// example.com /about-us

Additionally, URLs can include **query parameters to pass extra data**, or a **hash used to direct the browser to a specific part of a web page.**

URLs can also have subdomains (e.g., blog.example.com) to organize different sections of a website, and special characters in URLs are encoded (e.g., spaces are shown as %20) to ensure proper transmission.

URL - Protocol, Domain Name & Path

https://example.com/about-us

URL - Protocol, Domain Name, Subdomain & Path

https://blog.example.com/posts/2024

Example URL with a Query String

https://example.com/search?q=laptops&sort=price

Example URL with a Hash

https://example.com/products#reviews

A **query string in a URL is used to pass parameters to the server**, providing additional information to refine or modify the content being requested, and tailor the response according to the user's request. The query string starts after the question mark (?).
In the example https://example.com/search?q=laptops&sort=price,
the query string *?q=laptops&sort=price* includes two parameters -

q=laptops: This parameter specifies the search query, in this case, looking for "laptops."
sort=price: This parameter indicates how the results should be sorted, in this case, by price.

In https://example.com/products#reviews, **the hash #reviews refers to a section within the /products page**, typically a specific part or element like a review section.

The hash does not affect the server-side request; it only **affects how the browser displays the page** to the user, usually **by scrolling** to or highlighting the designated section.

DNS

The URL is the human-readable address of a website. When you enter a URL, your browser needs to translate the human-readable domain name into an IP address, which is done by the DNS (Domain Name System).

DNS is a system that translates domain names (e.g., www.example.com) into numerical IP addresses, such as "192.168.1.1", which computers use to identify each other on the network. It serves as the "phone book" of the internet, **mapping human-readable domain names to IP addresses.**

This allows users to access websites via easy-to-remember names rather than complex numerical addresses.

When a user types a domain name into a web browser or clicks on a link, **the browser sends a DNS query to resolve - convert- the domain name into an IP address.**

If the browser has recently visited the site, it may have the IP address stored in its cache and can skip the DNS query step. If it doesn’t, your computer needs to find out the exact location of the website on the internet.

When your browser doesn’t already have the IP address for a domain name (e.g., example.com) stored in its local cache, it **sends out a request called a "DNS query" to a DNS resolver.** The DNS resolver is typically provided by your Internet Service Provider (ISP) or configured by your network settings.

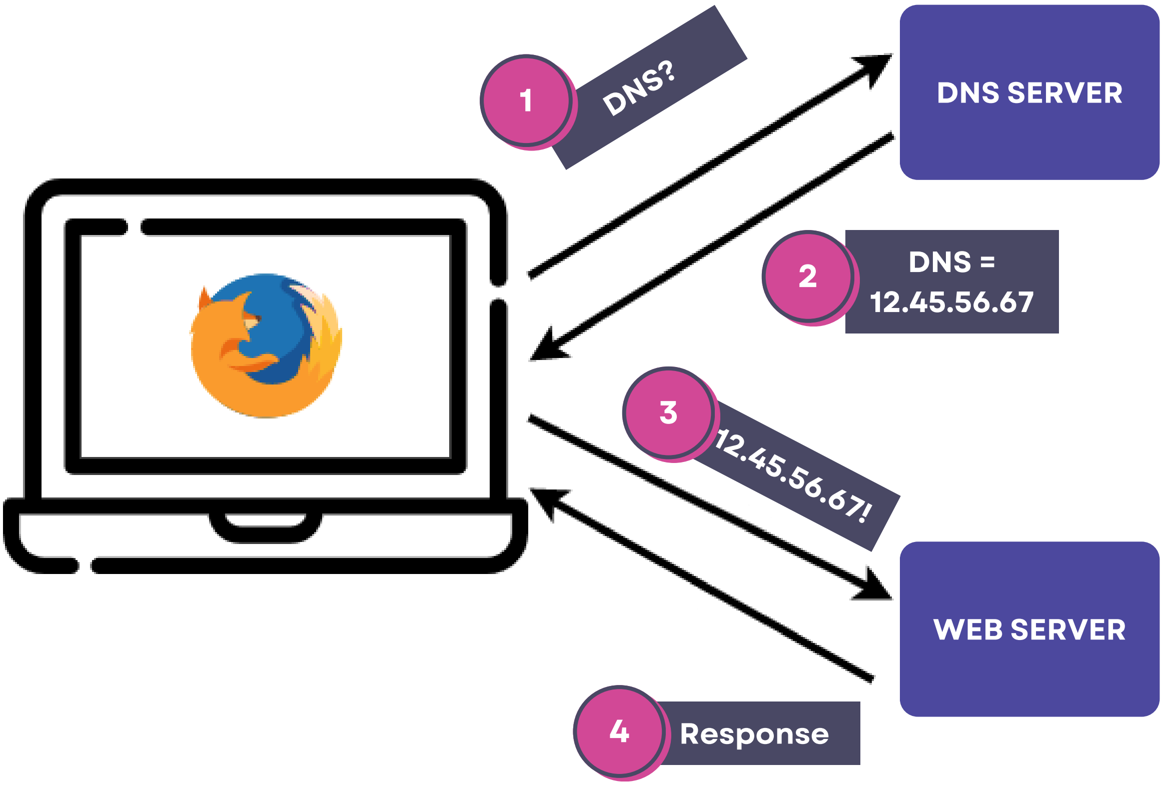
The resolver checks its own cache first. If the IP address isn’t in its cache, the **resolver then queries other DNS servers**, to find the correct IP address for the domain name.
The DNS Resolver query DNS servers if needed, while DNS servers respond to queries from the DNS resolver to provide the necessary IP address.

The DNS server then responds with the IP address, and it is returned to the browser.

Once the browser has the IP address, it uses the TCP/IP protocols to establish a connection to the web server. After establishing the connection, the browser communicates with the web server using either the HTTP or HTTPS protocol to request the web page or resource.

The DNS resolution process happens in milliseconds, enabling fast and seamless access to websites.

The DNS system is decentralized, with multiple levels of DNS servers working together to ensure that domain name queries are resolved efficiently.


1
DNS?
DNS SERVER
2
DNS =
12.45.56.67
3
12.45.56.67!
WEB SERVER
4
Response