Java Web Application Development Technology
N13020008

Web concepts

Tamás Tompa, PhD

assistant professor
Department of Information Technology
University of Miskolc

2024.

What is WEB?

- **The Web is an Internet-based distributed information system**
- Anyone with a computer connected to the Internet can easily retrieve information by giving a Web address
- Great way to disseminate information and making it available 24/7
- Web a powerful tool for mass communication, e-business and e-commerce
- **WEB: network of networks based on the TCP/IP** communications protocol.
- Community of people who use and develop those networks
- Services: email, telnet, FTP, WWW, etc.

Web components

- The **Web consists of these major components**:
  - **Networks**: The local-area and wide-area networks connecting computers world-wide forming the Internet

  - **Servers**: Constantly running programs that serve up information to the Web

  - **Clients**: Web browsers that enable end-users to access the Web

  - **Documetns**: Web pages, mostly coded in HTML, that supply information on the Web

  - **Protcols**: The Hyper Text Transfer Protocol HTTP that Web clients and servers use to talk to one another and the TCP/IP (Transmission Control Protocol) on which HTTP depends
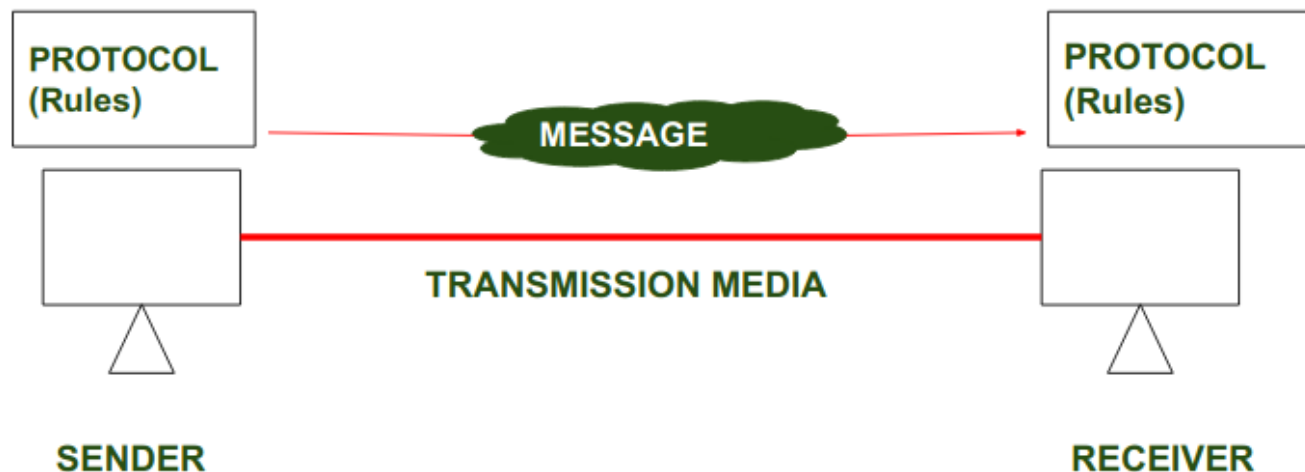
Computer network
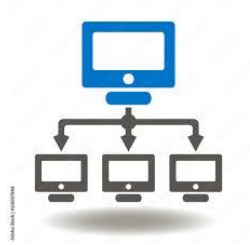
- **The WEB is based on the computer network**
  - The computer network is a **high-speed communications medium** connecting many, possibly dissimilar, computers or hosts
  - A network is a **combination of computer and telecommunication hardware and software**
  - **In order for programs and computers from different vendors to communicate on a network**, a detailed **set of rules** and conventions **must be established** for all parties to follow. Such rules are known as **networking protocols:**
    - address format of hosts and processes
    - data format
    - manner of data transmission
    - sequencing and addressing of messages
    - initiating and terminating connections
    - establishing remote services
    - accessing remote services
   - network security

Computer network protocols

- **In order for a process on one host to communicate with another process on a different host, both processes must follow the same protocol, e.g.:**
  - **Internet Protocol (IP)** is the most widely used, the basic protocol for the Internet
  - The Web is a service that uses **HTTP (Hyper Text Transfer Protocol)**, which is based on Internet protocols
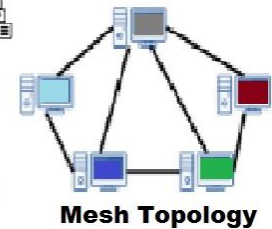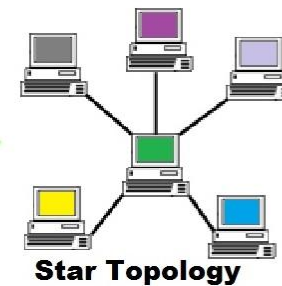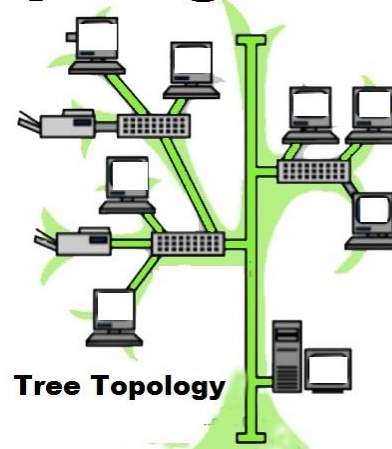
Computer network types
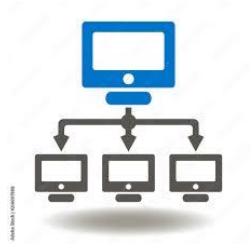
- **Network topology refers to the arrangement of elements** (such as links, nodes, devices) within a communication network
- Used to define or describe **how different components of a network are interconnected**
  - hosts can be connected via cables, wireless connections, or network interfaces

- **Physical topology** refers to the actual physical layout of the devices and cables
- **Logical topology** describes how data flows within the network (regardless of its physical design)

Network Topologies

Hybrid Topology

Ring Topology

Tree Topology

Star Topology

Mesh Topology

BUS Topology (BNC Terminator Connector

Rana Mehtab Alam
www.lunarcomputercollege.com
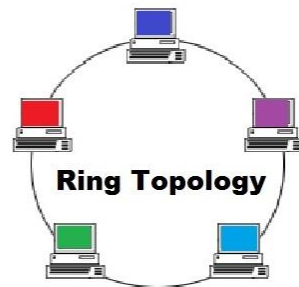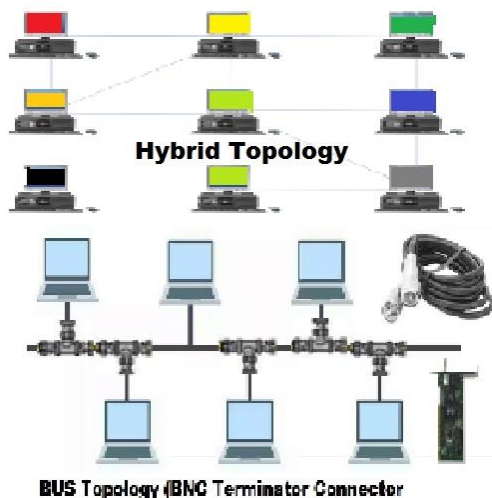
Computer network addresses
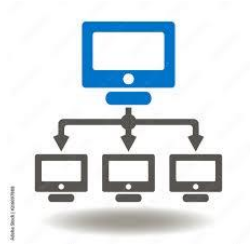
○ **Every host** on the Internet **has a unique network address** that identifies the host for communication purposes

○ The addressing scheme is an important part of a network and its protocol

○ Each host has a unique **IPv4 address**, which **is represented by 4 bytes** (32-bit), written in dotted decimal format (e.g., 192.168.1.1), or an **IPv6 address**, which **uses 16 bytes** (128-bit) and is written in colon-hexadecimal format (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334)

  ● **IPv6** was introduced to address the exhaustion of IPv4 addresses, offering a vastly larger address space

  ● **IPv4** addresses are typically divided into two parts: the network portion and the host portion, depending on the subnet mask

○ For example (IPv4 address): 131.123.35.92

1 0 0 0 0 0 1 1	0 1 1 1 1 0 1 1	0 0 1 0 0 0 1 1	0 1 0 1 1 1 0 0
131	123	35	92
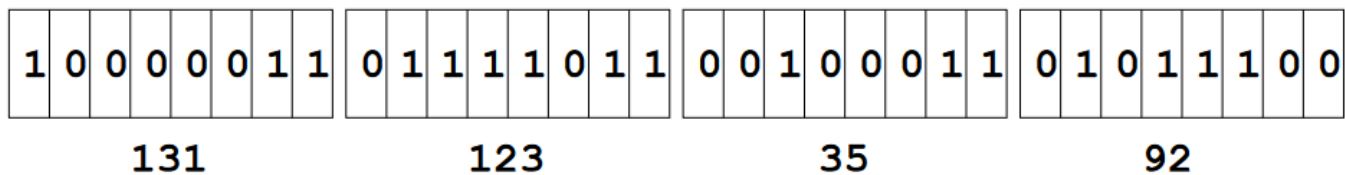
Computer network addresses
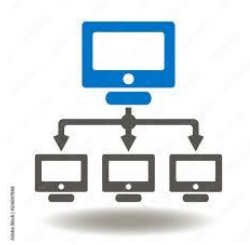
- **IPv4 subnet mask**
  - **divide an IP address into two parts: the network portion and the host portion**
  - subnet mask is used to determine the network and host portions
  - the process used to identify the network and host portions is called ANDing
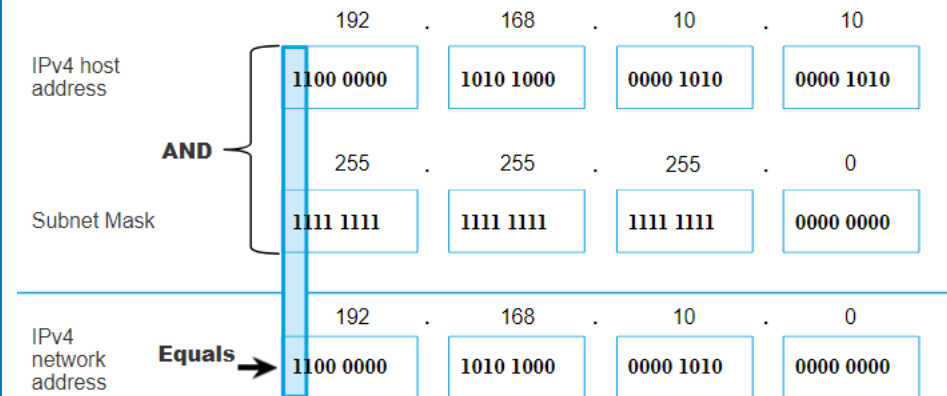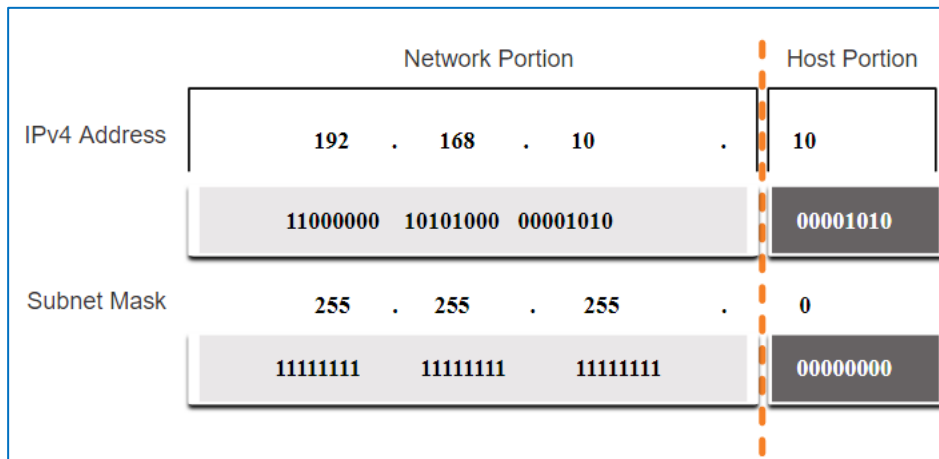
What is WWW?

- **WWW stands for World Wide Web**
  - **Invented by Sir Tim Berners-Lee in 1989** while working at CERN, and became publicly accessible in 1991
  - The **World Wide Web is the universe of network-accessible information**, an embodiment of human knowledge
  - **Web browsers** like Chrome, Firefox, and Safari are used to **access and navigate the World Wide Web**
  - **URLs** (Uniform Resource Locators) are **used to identify and locate resources on the Web**
  - All the resources and users on the Internet that are **using the Hypertext Transfer Protocol (HTTP)**
    - The HTTP (Hypertext Transfer Protocol) is the foundation of communication over the Web, allowing for the transfer of web pages and resources
  - Using **Client-Server architecture**

The client-server architecture
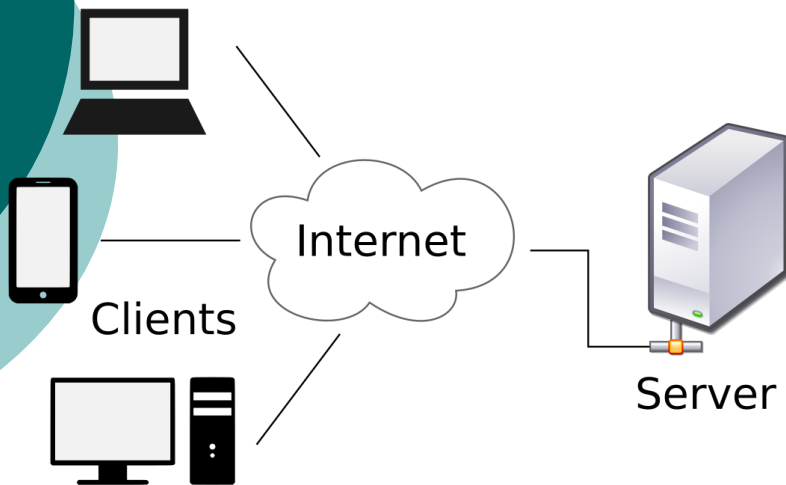
- The **server** is a powerful central system that **provides services to multiple clients**
- **Clients** are devices that **request services from the server**
- The **communication** between client and server happens over a network, **using protocols** like HTTP or TCP/IP
- **The server is responsible for processing requests** and managing resources, while **clients** handle the user interface and **interact with the server for data**
- This architecture allows for centralized control, scalability, and security, as servers manage most of the data and logic
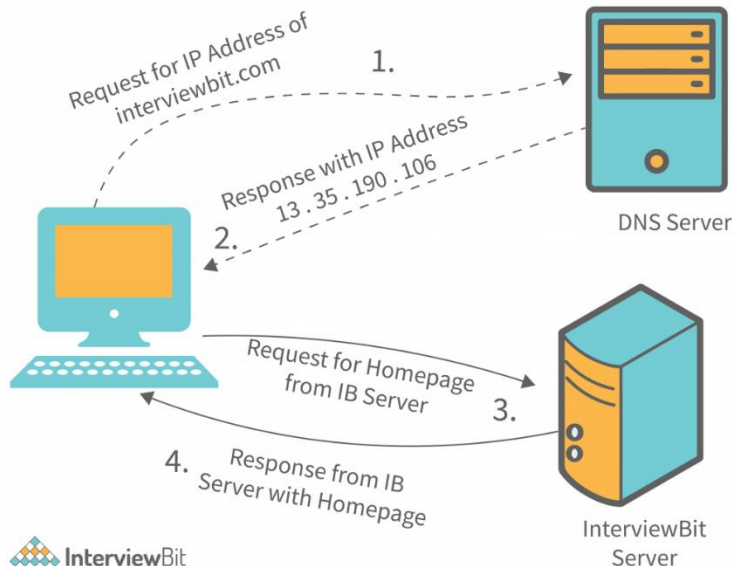
The Domain Name System
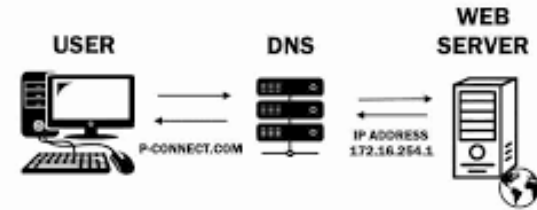
- **DNS: Domain Name System**
  - **hierarchical and distributed system used to translate human-readable domain names (e.g., www.example.com) into IP addresses (e.g., 192.168.1.1)**
  - every host on the Internet has a unique IP address
  - network name space is the set of all host names
  - a network client (such as a web browser) typically uses DNS to resolve a domain name into an IP address before it can establish a connection to a server
  - DNS operates as a distributed database service spread across many servers globally, which ensures scalability and fault tolerance
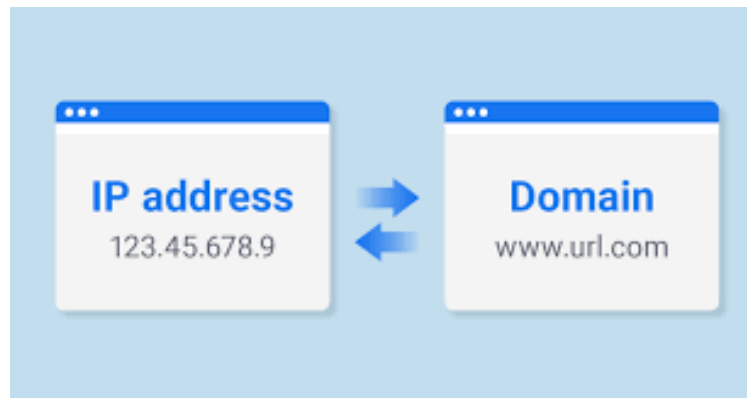
The Domain Name System
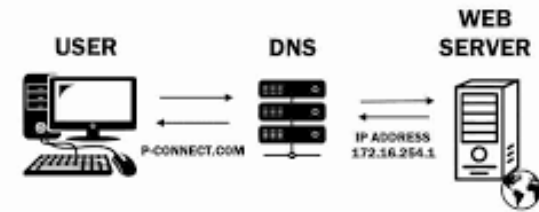
- **DNS: Domain Name System**
  - **hierarchical and distributed system used to translate human-readable domain names (e.g., www.google.com) into IP addresses (e.g., 208.117.229.214)**
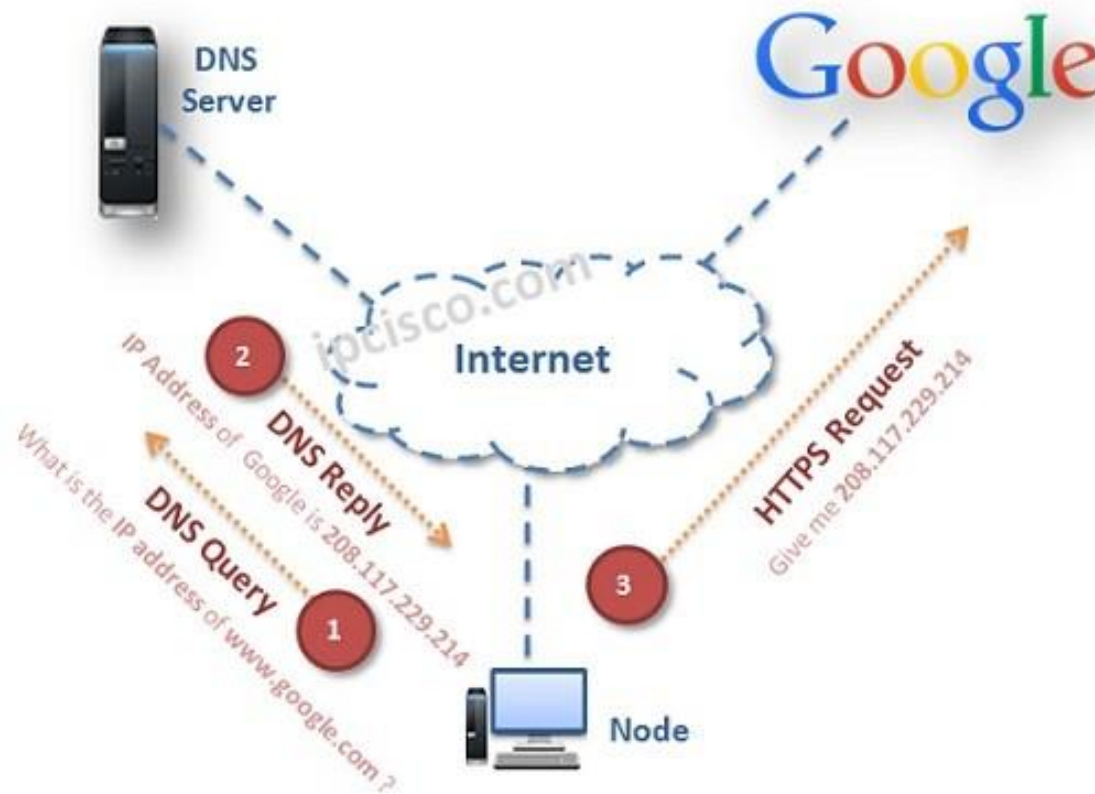
What is URL?

- **Uniform Resource Locators (URLs) to identify (locate) resources (files and services) available on the Internet**
  - They provide the specific location of a resource and the method to retrieve it, forming the cornerstone of web navigation
- URL may identify a host, a server port, and the target file stored on that host

  scheme://server:port/pathname?search=query#section

- **Scheme**: This part specifies the protocol to be used for communication with the resource (e.g., HTTP, HTTPS, FTP). The scheme tells the browser or client software what type of service is being requested and how to access it
- **Host/Server**: The domain name (e.g., www.example.com) or IP address of the server where the resource is hosted
- **Port**: Optional part that specifies the network port used by the service on the host. If omitted, default ports are used (e.g., port 80 for HTTP or 443 for HTTPS)

What is URL?

- **Uniform Resource Locators (URLs) to identify (locate) resources (files and services) available on the Internet**
  - They provide the specific location of a resource and the method to retrieve it, forming the cornerstone of web navigation
- URL may identify a host, a server port, and the target file stored on that host

   scheme://server:port/pathname?search=query#section

- **Pathname**: Indicates the specific file or resource on the server (e.g., /images/photo.jpg). The path helps locate the file in the directory structure of the server.
- **Query**: Optional parameters for more specific requests, typically following a ? (e.g., ?search=example). These parameters are often used in dynamic web pages to pass additional data to the server.
- **Fragment**: A part of the URL following a #, used to direct users to a specific section within a web page (e.g., #section2).

URL example

https://www.example.com:8080/articles/news?search=technology#latest

- **https**: The protocol (scheme), indicating the use of HTTPS, which is a secure version of HTTP.
- **www.example.com**: The server (host), which represents the domain name or IP address of the web server
- **:8080**: The port number, in this case, 8080. If no port is specified, the default for HTTPS is 443
- **/articles/news**: The path on the server, pointing to a specific directory or file within the server's structure
- **?search=technology**: The query string or searching parameter, providing additional data (in this case, a search term) for the server to process
- **#latest**: The fragment, which links to a specific part or section of the web page, identified by "latest" in this example.

What is HTTP?

- **HTTP: Hypertext Transfer Protocol**
  - On the Web, **browser-server communication follows the HTTP protocol**
  - Defines **how messages are formatted and transmitted between clients (such as web browsers) and servers**
  - When a client requests a web page, HTTP allows the client to fetch the necessary resources (e.g., HTML, CSS, JavaScript, images) from the server
  - HTTP is stateless, meaning each request is independent of other
  - HTTPS is a secure version of HTTP that encrypts data to protect it during transmission
  - Based on TCP/IP (Transmission Control Protocol/Internet Protocol)

What are TCP/IP and UDP?

- **HTTP: Hypertext Transfer Protocol**
  - Based on **TCP/IP (Transmission Control Protocol/Internet Protocol)**
    - defines how data is packaged into packets, addressed, transmitted, routed, and received between devices across network boundaries. TCP/IP enables reliable and orderly communication between computers by handling data transmission (TCP) and directing packets to their destinations (IP)
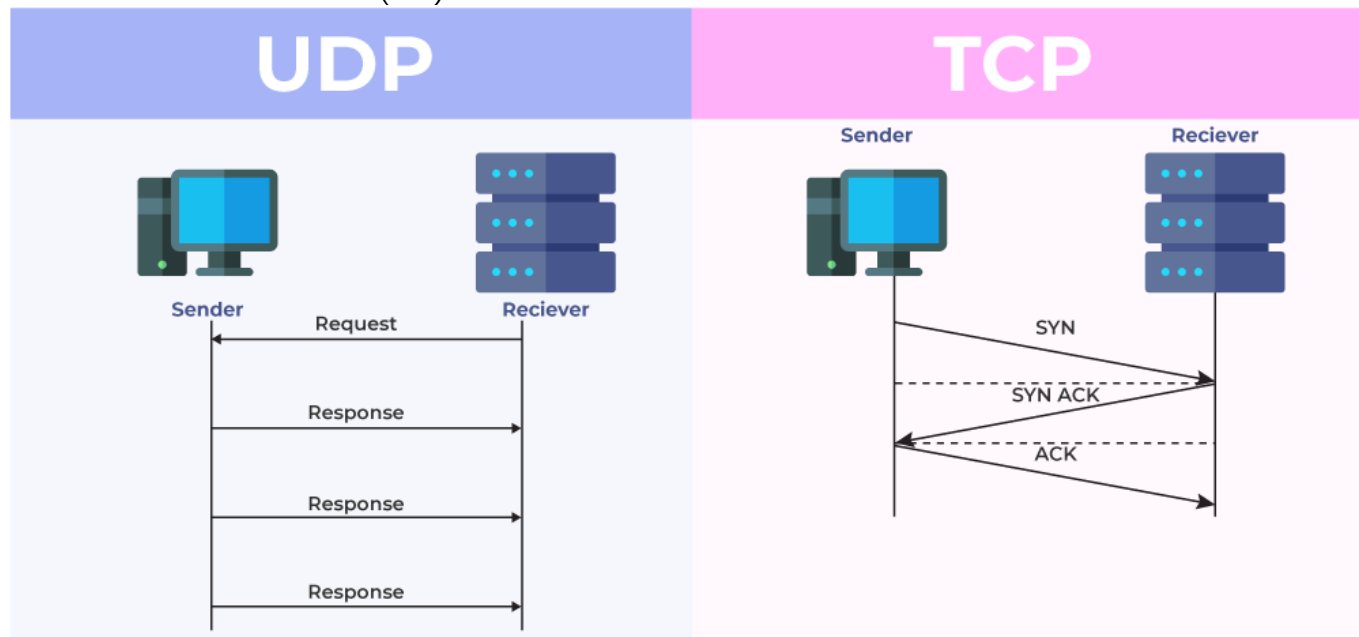
TCP and UDP

Differences between TCP and UDP

The main difference between TCP (transmission control protocol) and UDP (user datagram protocol) is that **TCP is a connection-based** protocol and **UDP is connectionless**

- While TCP is more reliable, it transfers data more slowly. UDP is less reliable but works more quickly

Protocol	TCP	UDP
Connection	connection-oriented	connectionless
Usage	high reliability, critical-less trans-mission time	fast, efficient transm-ission, small queries, huge numbers of clients
Ordering of data packets	rearranges packets in order	no inherent order
Reliability	yes	no
Streaming of data	read as a byte stream	sent and read indivi-dually
Error checking	error checking and recovery	simply error checking, no error recovery
Acknowle-dgement	acknowledgement segments	no acknowledgment

TCP and UDP

- **Differences between TCP and UDP**

TCP vs UDP Communication

TCP

Client — SYN → Server
Client ← SYN-ACK — Server
Client — ACK → Server

UDP

Client ← Request — Server
Client — Response → Server
Client — Response → Server
Client — Response → Server

Unicast
(One-to-One)

Broadcast
(One-to-All)

Multicast
(One-to-Several)
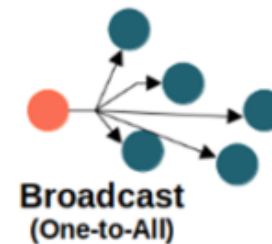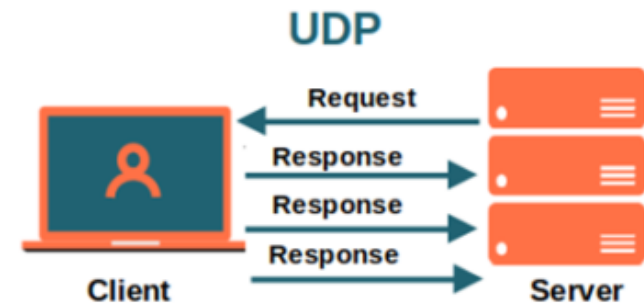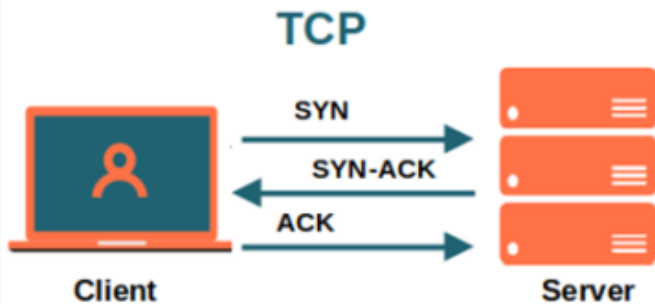
HTTP transaction

HTTP

○ **Hypertext Transfer Protocol transaction**

- **1. Connection**: A browser (client) opens a connection to a server
- **2. Query**: The client requests a resource controlled by the server
- **3. Processing**: The server receives and processes the request
- **4. Response**: The server sends the requested resource back to the client.
- **5. Termination**: The transaction is done and the connection is closed unless another transaction will take place immediately between the client and server

HTTP Client

HTTP Server

HTTP Request

HTTP Reply
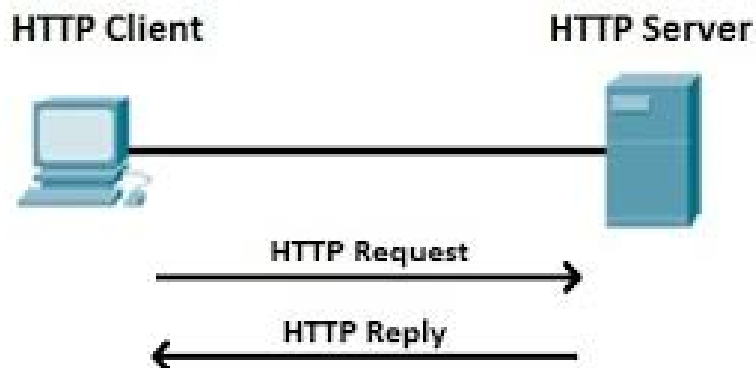
Hypertext Transport Protocol (HTTP)

Connection

Request

Response

Close

Client

Mega Server Inc.

Server

Stages in an HTTP Transaction

Gregory S. Aist
April 28, 1995

HTTP query

- **Hypertext Transfer Protocol query**
  - A **query line has three parts**, separated by spaces: a query method name, a local path of the requested resource, and an HTTP version number. For example:

    GET /path/to/file/index.html HTTP/1.1

  - **GET**: This is the HTTP method. In this case, GET is used to request data from the server
  - **/path/to/file/index.html**: This is the path to the resource on the server. It indicates the location of the file or resource that the client is requesting
  - **HTTP/1.1**: This specifies the HTTP version being used. HTTP/1.1 is a version of the HTTP protocol that supports persistent connections, meaning multiple requests and responses can be sent over the same connection, which improves performance

HTTP methods

- **HTTP methods**
  - enable various types of interactions between clients and servers, facilitating the dynamic and flexible nature of web applications

  - **GET**:
    - Retrieves data from a specified resource
    - It is used to request data without modifying it
    - Data can be sent in the URL (query string), but it is not secure for sensitive information
  - **POST**:
    - Sends data to the server to create or update a resource
    - It is often used when submitting form data or uploading files
    - Data is included in the body of the request, making it more secure for sensitive information compared to GET
  - **PUT**:
    - Updates a specified resource with the data provided in the request body
    - It is idempotent, meaning multiple identical requests will have the same effect as a single request

HTTP methods

- **HTTP methods**
  - enable various types of interactions between clients and servers, facilitating the dynamic and flexible nature of web applications

  - **DELETE**:
    - Removes a specified resource from the server.
    - Like PUT, it is also idempotent, so sending multiple DELETE requests for the same resource will result in the same state.
  - **HEAD**:
    - Similar to GET, but it only retrieves the headers of the resource, not the actual content.
    - It is useful for checking if a resource exists or to get metadata without downloading the resource.
  - **OPTIONS**:
    - Describes the communication options for the target resource.
    - It is used to determine which HTTP methods are supported by the server for a specific resource.

HTTP methods

- **HTTP methods**
  - enable various types of interactions between clients and servers, facilitating the dynamic and flexible nature of web applications

  - **PATCH**:
    - Partially updates a resource with the data provided in the request body.
    - Unlike PUT, which replaces the entire resource, PATCH allows for partial modifications.

HTTP methods

- HTTP Request Methods

GET — Retriving a sinlge or multiple resources

POST — Creating a new resource

PATCH — Updating an existing resource

PUT — Updating Or Creating new resource

DELETE — Destroying a resource

Mahmoud217TR

mahmoud-mahmoud-ab8708183

HTTP versions

HTTP

Feature	HTTP/1.0	HTTP/1.1	HTTP/2	HTTP/3
Connection Management	New connection for each request	Supports persistent connections (keep-alive)	Multiplexing over a single connection	Uses QUIC over UDP
Caching	Basic support	Advanced caching with cache-control headers	Efficient header compression	Built-in encryption
Data Format	Text-based	Text-based	Binary protocol	Binary protocol
Request/ Response	Limited methods and status codes	More methods (e.g., OPTIONS, PATCH)	More efficient with multiplexing	Improved multiplexing capabilities
Header Management	No specific requirements	Requires `Host` header	Uses HPACK compression	Improved header management
Server Push	Not available	Not available	Supports server push	Supports server push
Prioritization	Not available	Limited prioritization	Supports prioritization	Enhanced prioritization

HTTP versions

HTTP

HTTP/1.1 Baseline

HTTP/2 Multiplexing

Time

HTTP 1.1 Vs. HTTP 2.0
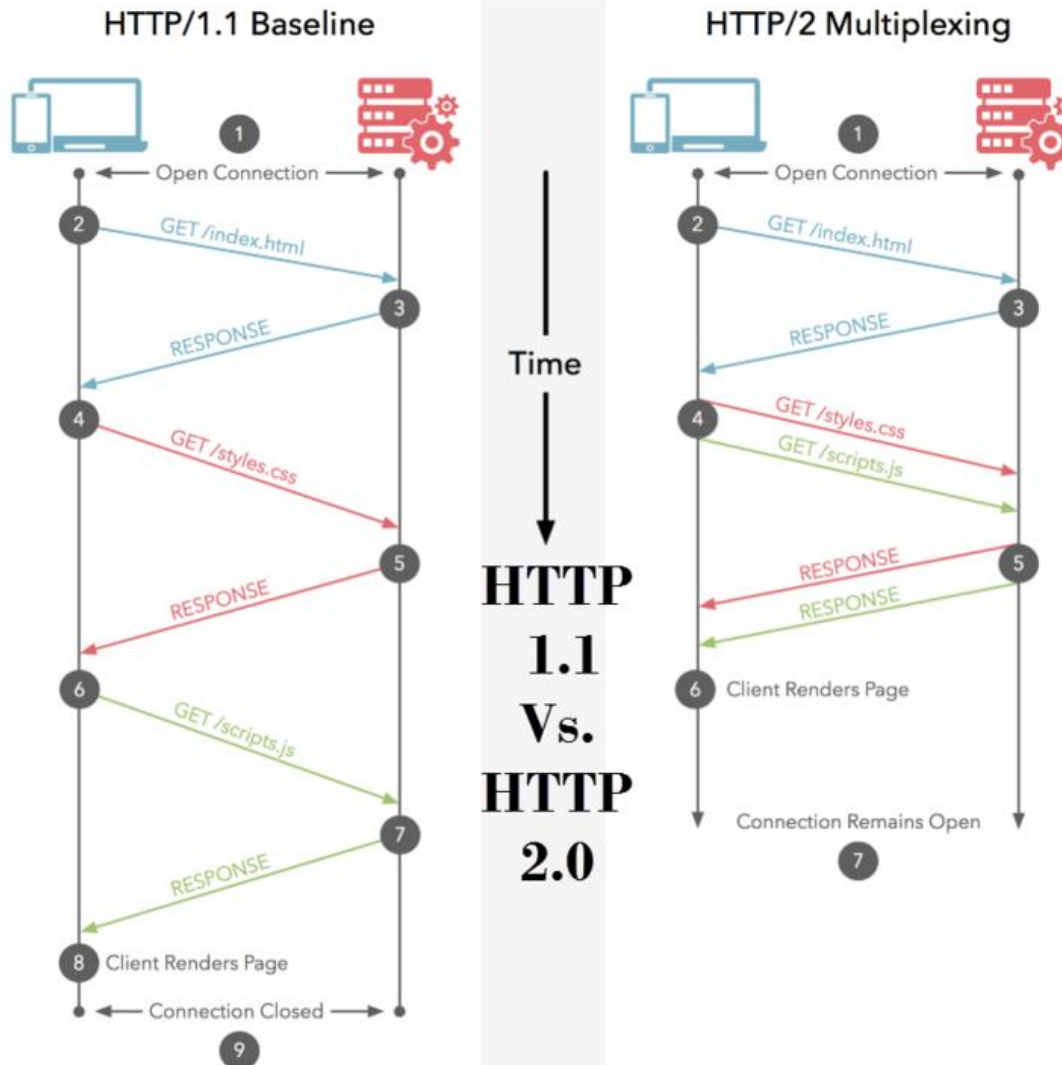
HTTP status codes

- **HTTP status codes**
  - HTTP status codes provide valuable information about the outcome of a request and help clients understand how to proceed based on the server's response

  - Types:
    - **1xx (Informational)**
    - **2xx (Successful)**
    - **3xx (Redirection)**
    - **4xx (Client Error)**
    - **5xx (Server Error)**

HTTP status codes

- **HTTP status codes**

  - 1xx (Informational):
    - 100 Continue: Indicates that the initial part of a request has been received and the client can continue with the rest of the request.
    - 101 Switching Protocols: Indicates that the server is switching to a different protocol as requested by the client.

  - 2xx (Successful):
    - 200 OK: The request was successful, and the server has returned the requested data.
    - 201 Created: The request was successful, and a new resource has been created (usually in response to a POST request).
    - 204 No Content: The request was successful, but there is no content to send in the response.

HTTP status codes

- **HTTP status codes**

  - 3xx (Redirection):
    - 301 Moved Permanently: The requested resource has been permanently moved to a new URL.
    - 302 Found: The requested resource is temporarily located at a different URL. This code is commonly used for temporary redirects.
    - 304 Not Modified: The resource has not been modified since the last request. The client can use its cached version.

  - 4xx (Client Error):
    - 400 Bad Request: The server cannot process the request due to a client error (e.g., malformed request syntax).
    - 401 Unauthorized: The request requires authentication, and the client has not provided valid credentials.
    - 403 Forbidden: The server understands the request but refuses to authorize it.
    - 404 Not Found: The requested resource could not be found on the server.

HTTP status codes

- **HTTP status codes**

  - 5xx (Server Error):
    - 500 Internal Server Error: The server encountered an unexpected condition that prevented it from fulfilling the request.
    - 502 Bad Gateway: The server, while acting as a gateway or proxy, received an invalid response from the upstream server.
    - 503 Service Unavailable: The server is currently unable to handle the request due to temporary overload or maintenance.

HTTP status codes

○ **HTTP status codes**

HTTP Status Codes

1xx Informational

100	Continue
101	Switching protocols
102	Processing
103	Early Hints

3xx Redirection

301	Moved Permanently
302	Found
303	See Other
304	Not Modified

2xx Succesful

200	OK
201	Created
202	Accepted
203	Non-Authoritative
204	No Content
205	Reset Content
206	Partial Content
207	Multi-Status
208	Already Reported

4xx Client Error

400	Bad Request
401	Unauthorized
402	Payment required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
409	Conflict
413	Payload Too Large
429	Too Many Requests

5xx Server Error

500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	HTTP Version Not Supported
506	Variant Also Negotiates
507	Insufficient Storage
508	Loop Detected
510	Not Extended
511	Network Authentication Required

Web technology

- **Web Technology** refers to the **various tools and techniques** that are **utilized in the process of communication** between different types of devices **over the Internet**

  - **WWW**: Web browsers, Hypertext Markup Language (HTML), and Hypertext Transfer Protocol (HTTP)

  - **Web browser**: interface between the server and the client and requests to the server for web documents and services

  - **Web server**: processes the network requests of the users and serves them with files that create web pages

  - **Web pages**: digital document that is linked to the World Wide Web

  - **Web development**: building, creating, and maintaining of websites

Web development

- **Frontend Development:**
  - The part of a website that the user interacts directly is termed as front end
  - It is also referred to as the **'client side' of the application**

  - HTML, CSS, JS, ES6, TypeScript, JS Framework, Package manager, Angular JS, Vue.js, JQuery, React JS, …

Frontend Development

HTML	CSS	JavaScript

CSS Frameworks			CSS Preprocessors
Bootstrap	Tailwind CSS	Bulma	SASS
Foundation CSS	Primer CSS	Spectre CSS	LESS
Materialize CSS	Onsen UI	Semantic UI	
Blaze UI	Pure CSS	Tachyons	

ES6	TypeScript	Package Manager	JS Frameworks	JS Libraries
		npm		
		yarn		

Angular JS
Angular ngx Bootstrap
Angular PrimeNG
Angular Material UI

Vue.js
Nuxt.js

jQuery
jQuery UI
jQuery Mobile
jQWidgets
jQuery EasyUI

React JS
Next.js
Ant Design
React Desktop
React Rebass
React Bootstrap
React Reactstrap
Blueprint.js
React Suite
React Evergreen

Lodash
Underscore.js
Collect.js
Moment.js
Tensorflowjs
p5.js
D3.js
Fabric.js

Web development

Backend Development:

Backend is the **server side** of a website

It is the part of the website that **users cannot see** and interact

- It is the portion of software that does **not come in direct contact with the users**

- It is used to **store and arrange data**

- Ruby, Python, PHP, NodeJS, Spring, .NET, npm, pip, …

Thank you for your attention!