Internet Applications

Introduction

Lecture # 01

What is the Internet?

- A global heterogeneous network that connects a collection of computers all over the world using transmission media (copper, fiber, wireless, etc.), special purpose devices (routers, switches, etc.), network operating systems (NOS) and applications software (email, Internet browsers, etc).
- Each computer on the Internet is called the host computer.
- Computer which provides services for other computers is called server.
- The goal is to provide connectivity between machines and between users to
  - Share resources
  - Collaboration
  - Access remote information

Internet Applications

- An Internet application is a client/server application that uses standard Internet protocols for connecting the client to the server.
- Most Internet applications follow a client/ server model of initiation communication:
  - Server waits for client to initiate communication
  - Client initiates communication
  - Ones the communication is initiated, data can flow in both directions(client to server and server to client)

Common Internet Applications

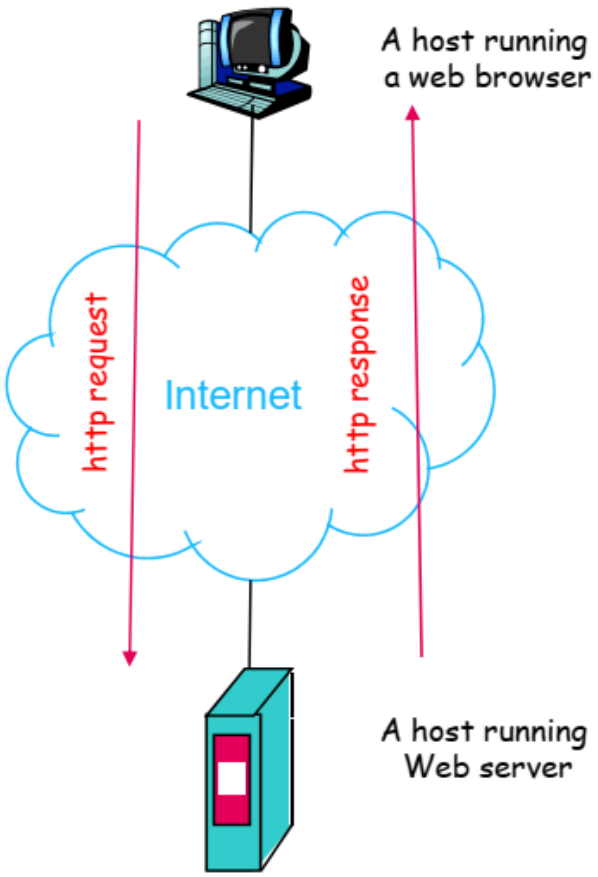
- The World-Wide Web (WWW).
- Electronic mail (email)
- Remote login (e.g. telnet)
- Streaming multimedia (e.g. Internet feeds of live audio and video, video on demand (VoD))
- Internet telephony (VoIP)
- Chatting
- Search Engines

The World-Wide Web

- The World Wide Web (WWW), or simply Web, is a way of accessing information over the medium of the Internet.
- A collection of web documents (web pages) and other web resources uniquely identified (using URLs), can be accessed via the Internet, and are linked to each other.
- Web documents can have different types of information (multimedia): text, images, audio and video
- Main features of the Web: Scalability, User friendly.

Web Architecture and Operation

- Web documents are hosted (stored) in machines running special software called Web servers
- Web documents can be accessed and viewed using special programs called browsers.
- Browsers and Web servers are often running on different machines.
- Browsers use HTTP protocol to communicate with the Web servers.



The web is a client-server Internet application

Web Architecture and Operation

- Web documents (or web pages).
- Web browser (also called user agent, web client or HTTP client).

  -Application program that represents the user interface to the Web
  -Bring information from Web server and displays it to the user

  Ex: (Google Chrome, Mozilla, Firefox, Opera, Safari, many others)
- Web server (or HTTP server)
  - Stores a set of Web documents (web pages)
  -Responding to requests from the browser by sending a copy of the document

- Web standards
  - Transfer protocol
    -HyperTextTransfer Protocol (HTTP)
  - Hypermedia links
    - Uniform Resource Locator (URL) to identify web resources
  - Document representation
    -HyperText Markup Language (HTML)

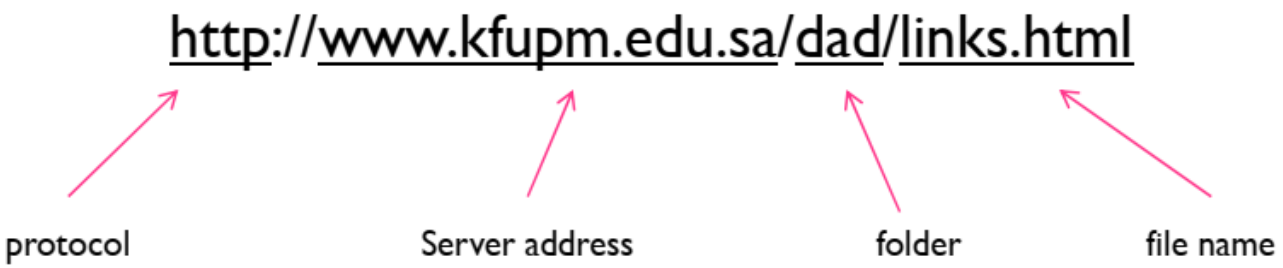
Web Architecture and Operation

- A user starts a browser on his computer and request a web document by specifying its URL
  - E.g. http://www.sabratha.edu.ly
- The browser resolve the URL to get the server IP address using the DNS (Domain Name Service) server.
- Then, the browser sends a message to the server requesting the required document
- The server finds the document in its file system and sends it back to the browser.
- The browser interprets the content of the document

URL Structure

- Uniform Resource Locator (URL)
  - Represents the address of documents and other resources on the Web
- URL defines:
  - Protocol used to access/transfer the document (such as HTTP or FTP; the default is HTTP)
  - Server that hosts the document and its domain name
  - Protocol port number of the server (optional; the default is 80)
  - Path and document name (the default is index.html)
- General form of URL

  protocol: / /server.domain_name:port/item_name
- Example



URL Structure

- The browser broke the URL into three parts:

1. The protocol ("http")
2. The server name
3. The file name

- The browser communicated with a name server to translate the server name into an IP Address, which it uses to connect to the server machine.
- The browser then formed a connection to the server at that IP address.

Web Cookies

- Web cookies (also called HTTP cookies, Internet cookies, browser cookies, or simply cookies)
- A mechanism that enables to store a small amount of text information (up to 4KB) on the client and is sent back from the client to the server with each HTTP request.
  - Enables a Web server distinguish between clients
  - Used to track user browsing and to customize/personalize pages based on user preferences
- A cookie is associated with a specific web site
- A cookie is sent in HTTP header

Web Documents

- Web documents or Web pages are two types
    1. Static web page
    2. Dynamic webpage
- **Static web page** where there is no specific interaction with the client.
- **Dynamic web page** which is having interactions with client and as well as validations can be added.
- A simple HTML page is called a static web page. However, simply adding JavaScript to an HTML page does not automatically make it dynamic.
- Dynamic pages require server-side processing (e.g., PHP, Python, databases) to change content based on user input.
- JavaScript enables client-side interactivity (e.g., form validation, animations), but the page remains static if it doesn’t communicate with a server to fetch/update data.

Dynamic web page

- **Examples of dynamic effects**:
  - Server-Side Dynamic Examples:
    - User-specific dashboard content (different data for each user)
    - E-commerce product inventory showing real-time stock levels
    - Social media news feed personalized to each user
    - Search results generated from database queries
    - Shopping cart that persists across sessions
  - CLIENT-SIDE (Browser):
    - Interactive form validation
    - Image galleries and sliders
    - Hover effects and animations
    - Real-time calculators/games
    - Dynamic styling changes

Scripting languages

- Scripting languages are two kinds one is client-side other one is servers-side scripting.
- In general client-side scripting is used for verifying simple validation at client side, server-side scripting is used for database verifications.
- JavaScript is example for client-side scripting.
- ASP.Net, php, Node.js, python etc. are examples of server-side scripting

Web Development Tools & Technologies

- Front-End Frameworks
  - React
  - Angular

- Back-End Frameworks
  - Node.js - Server-side JavaScript
  - Python (Django, Flask)
  - PHP (Laravel)

Web Development Tools & Technologies

- Developer Tools
  - Code Editors: VS Code, Sublime Text
  - Version Control: Git, GitHub, GitLab

- Servers & Databases
  - Web Servers: Apache, Nginx, IIS
  - Databases:
    - Relational: MySQL, PostgreSQL
    - NoSQL: MongoDB, Firebase