Super 4

ASSIGNMENT 1( How Internet Works?)

**1. How does the internet work?**

The internet is a global network that connects computers and devices using protocols like TCP/IP. Data is transferred in small packets through various networks, including ISPs (Internet Service Providers), which provide internet access. Routers direct these packets to their destination, while DNS (Domain Name System) translates domain names (like google.com) into IP addresses so computers can find websites.

**2. What is DNS and why is it important?**

DNS (Domain Name System) acts like a phonebook for the internet. It converts human-friendly domain names (e.g., facebook.com) into IP addresses (e.g., 192.168.1.1), which computers use to communicate.

**Types of DNS records:**

* **A Record:** Maps a domain to an IPv4 address.
* **AAAA Record:** Maps a domain to an IPv6 address.
* **CNAME Record:** Points a domain to another domain.
* **MX Record:** Handles email routing.

**3. Explain the client-server architecture.**

In a client-server model, the **client** (e.g., a web browser) requests data and the **server** responds with the requested information.

**Difference:**

* **Client:** Sends requests (e.g., Chrome, Firefox).
* **Server:** Processes requests and sends responses (e.g., Google’s web server).

**Real-world example:**  
When you visit YouTube, your browser (client) requests a video and YouTube’s server sends the video data back to your device.

**4. What are IP addresses and domain names?**

An **IP address** (e.g., 192.168.1.1) is a unique number assigned to each device on the internet. A **domain name** (e.g., amazon.com) is a human-friendly way to access websites instead of typing long IP addresses. DNS helps match domain names with their IP addresses, allowing users to easily visit websites.

**5. What are HTTP request methods?**

HTTP methods define actions that a client can perform on a server.

* **GET:** Retrieves data (e.g., loading a webpage).  
  *Example: Visiting example.com/products fetches a product list.*
* **POST:** Submits data (e.g., filling a form).  
  *Example: Submitting a login form sends user credentials.*
* **PUT:** Updates existing data.  
  *Example: Updating a profile picture.*
* **DELETE:** Removes data.  
  *Example: Deleting a post from a blog.*

**6. Compare TCP and UDP.**

| **Feature** | **TCP (Transmission Control Protocol)** | **UDP (User Datagram Protocol)** |
| --- | --- | --- |
| **Speed** | Slower (ensures data arrives correctly) | Faster (no verification of data arrival) |
| **Reliability** | Reliable (checks for errors, resends lost data) | Unreliable (no error checking, no resending) |
| **Use Cases** | Web browsing, email, file transfer (where accuracy matters) | Online gaming, live streaming, VoIP (where speed is key) |
| **Example Applications** | HTTP, HTTPS, FTP, SSH | DNS, video calls, multiplayer games |

**7. Describe the 3-way TCP handshake.**

TCP establishes a reliable connection using these three steps:

1. **SYN (Synchronize):** The client requests a connection.
2. **SYN-ACK (Synchronize-Acknowledge):** The server acknowledges the request.
3. **ACK (Acknowledge):** The client confirms and the connection starts.

🔹 This process ensures both devices are ready for data transfer.

**8. How does HTTPS secure communication?**

🔹 **HTTPS (HyperText Transfer Protocol Secure)** encrypts data to prevent hacking. It uses **SSL/TLS (Secure Sockets Layer/Transport Layer Security)** for encryption.

🔹 **HTTPS Handshake Process:**

1. **Client Hello:** Browser requests a secure connection.
2. **Server Hello:** Server sends its SSL/TLS certificate.
3. **Key Exchange:** Encryption keys are shared.
4. **Secure Connection Established:** All data is encrypted before transmission.

🔹 This prevents hackers from stealing sensitive data like passwords.

**9. Impact of DNS Caching on Website Performance**

🔹 **DNS caching** stores domain-to-IP lookups temporarily to speed up website access.

🔹 **Recursive Querying:**

* When a user enters a URL, the browser first checks local caches before contacting DNS servers.
* This reduces the time needed to load frequently visited websites.

🔹 **TTL (Time-to-Live):**

* Determines how long a DNS record stays in cache.
* A low TTL updates records faster but increases DNS queries.
* A high TTL reduces queries but might cause delays in updating changed IPs.

**10. Common HTTP Status Codes & Their Real-World Scenarios**

| **Code** | **Meaning** | **Real-World Example** | **Solution** |
| --- | --- | --- | --- |
| **200 OK** | Request successful | Loading a website correctly | No action needed |
| **301 Moved Permanently** | URL changed | Old webpage redirects to a new one | Set up a proper redirect |
| **403 Forbidden** | No permission | Trying to access a restricted page | Check user permissions |
| **404 Not Found** | Page missing | Typing an incorrect URL | Fix broken links or create a custom error page |
| **500 Internal Server Error** | Server problem | Website crashes | Check server logs and fix errors |

🔹 **Handling 404 & 500 Errors:**

* **404:** Set up a custom error page to guide users.
* **500:** Restart the server and debug issues.