**Name: Claire Njeri**

**TOPIC: What happens when one clicks** [**www.google.com**](http://www.google.com) **?**

There are layers to what usually occurs when one clicks a URL link in the browser that we will delve into.This is an overview of client-server architecture which involves things to do with, DNS ,TCP/IP and SERVERS.

**CLIENT SERVER ARCHITECTURE**

Client/server architecture is a model where clients request information and the server handles the request by providing responses. There are tons of servers in the entire world and each service held within the browser has its own server which has its own IP address. The web browser allows one to access to all the websites and content within the internet. Clients can be anything a computer, Mobile phone or even an I-Pad as long as it has internet connectivity.

**DNS**

This stands for domain name system, which works on the principle of translation from human readable text to an IP address. When we click [www.google.com](http://www.google.com) the DNS translates it to 108.177.122.139 which represents the google servers ensuring that content can be located and accessed. Take the DNS server as a logbook that contains IP address for all servers. Before contacting the DNS server, the web browser checks its cache to see if it already knows the IP address.

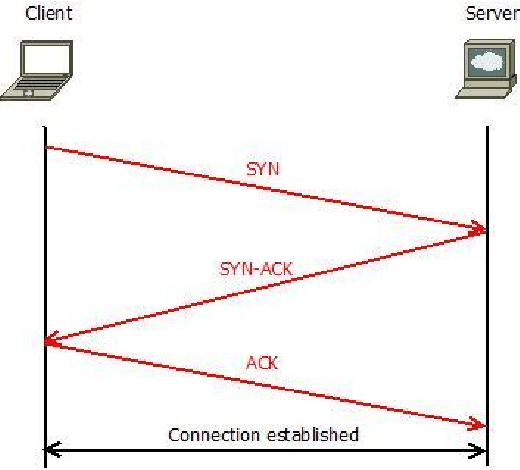
**TCP/IP MODEL**

**TCP/IP** is a model of transmission that means transmission control protocol and internet protocol that defines how data can be transmitted over a network. This establishes a connection between the client and the server.

The TCP/IP model has 4 layers which include:

* Application layer: where applications browser communicates directly and contains different protocols like HTTP(website), FTP( viewing )and many more each with their distinct port.
* Transport layer: consist of TCP, UDP(user datagram protocol) This is where data is divided into packets for transmission purposes with a TCP header containing the destination, source ports declaring where its coming from and going to.
* Internet layer: This consist of IP and its where the packets now contain the IP header which has the details of the IP addresses of both the destination and source ensuring that packets are not lost.
* Data link layer: This ensures data is send across the network either by wifi or Ethernet.

In order for a session to be created there is a term called 3-way handshake that ensures that both client and server can establish a connection for the data to be transmitted.

SYN (synchronize) -The client sends a SYN packet to the server to establish connection

SYN-ACK(synchronize- acknowledge) -The server responds with a SYN-ACK packet that accepts the client request and acknowledges using a sequence number

ACK- (acknowledge)-The client sends an ACK packet that confirms the connection.

**HTTPS request and response**

The https means that http is a secure protocol and there is a layer of encryption that whatever happens to user data it is protected using the SSL or secure socket layer.

**1. HTTP Request**

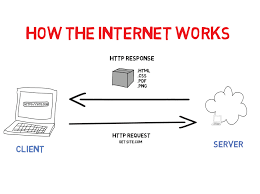
An [HTTP request](https://www.geeksforgeeks.org/different-kinds-of-http-requests/) is how your browser asks the server for something. It includes:

**HTTP Version**; The version of HTTP being used (http/1.1),**URL**: The specific address of the resource ( https://www.google.com),**HTTP Method**: The type of action being requested (either the GET to retrieve information or POST to send data).

**2. HTTP Response**

An [HTTP response](https://www.geeksforgeeks.org/state-the-core-components-of-an-http-response/) is the server’s answer to your request. It includes:

**HTTP Status Code**: A number that tells you if the request was successful or not (404 Not Found means the requested page doesn’t exist). **Response Headers**: Information about the response, like what kind of data is being sent (Content-Type: text/html means it’s an HTML page).**Response Body**: The content that the server sends back (HTML code that the browser will use to display the webpage).



**Rendering**

This is the process of converting code into viewable, interactive web content for the client to see. This approach allows for dynamic and interactive user experiences, as content can be loaded and changed without requiring a full page reload.