1. Internet

HOW DOES THE INTERNET WORK?

* All the world’s linked-up computers are called the internet
* Basic computer network
* Collection of standalone computers all loosely linked together (through copper cables, fiber-optic cables, wireless radio connections and satellite links)
* The internet moves computerized info (data) from one place to another (more like a postal service)

How does internet data move?

* Circuit Switching: direct (physical) connection between two parties alone (ie. Telephone line, or dialup connection to the Net)
* Packet Switching: data being sent is broken up into tiny pieces called “packets” and are allowed to travel separately in different routes but with the same destination. When reaching the destination, they are reassembled to the original data sent.

What are “clients” and “servers”?

* Servers are like electronic filing cabinets that simply store information and pass it on when requested
* Machines that hold ordinary documents are File servers; ones the hold people’s mail are Mail servers; the ones that hold Web pages are Web servers; etc.
* Clients are computers that gets information from the servers
* Routers are intermediate computers that make connections between different systems (ie. A Router connects multiple devices or computers to the internet)

TCP/IP and DNS

* Transmission Control Protocol/Internet Protocol is the internet’s fundamental ‘control system’ and it’s really two systems in one
* Internet’s Protocol or IP is simply the internet’s addressing system
* All the machines on the internet are identified by an Internet Protocol (IP) address that takes the form of a series of digits separated by dots or colons.
* Original version of IP is IPv4 consisted of four pairs of digits, such as 12.34.56.78 or 123.255.212.55
* The new IP system, IPv6, is much longer and looks something like this: 123a:b716:7291:0da2:912c:0321:0ffe:1da2.
* The other part of the control system, **Transmission Control Protocol (TCP)**, sorts out how packets of data move back and forth between one computer (in other words, one IP address) and another.
* The Domain Name System or DNS enables a computer to look up the IP address for any given website (ie. Google.com)

HTTP

What is HTTP?

* Hypertext Transfer Protocol is the foundation of the World Wide Web to load web pages using hypertext links
* It is an application layer protocol designed to transfer information between networked devices and runs on top of other layers of the network protocol stack
* A typical flow over HTTP involves a client machine making a request to a server, which then sends a response message.
* Communication between clients and servers is done by **requests** and **responses**:
  + A client (a browser) sends an **HTTP request** to the web
  + A web server receives the request
  + The server runs an application to process the request
  + The server returns an **HTTP response** (output) to the browser
  + The client (the browser) receives the response

What is in an HTTP request?

* It is the way internet communications platforms such as web browsers ask for the information they need to load a website.
* Each request carries with it a series of encoded data that carries different types of information. A typical HTTP request contains:
  + HTTP version type
  + URL
  + HTTP method
  + HTTP request headers
  + Optional HTTP body

HTTP Method or HTTP verb

* Indicates the action that the HTTP request expects from the queried server
* Example: two of the most common HTTP methods are ‘GET’ and ‘POST’; a ‘GET’ request expects information back in return (usually in the form of a website), while a ‘POST’ request typically indicates that the client is submitting information to the web server (such as form information, e.g. a submitted username and password).

HTTP request headers

* Contains text information in key-value pairs, and they are included in every HTTP request (and response, more on that later). These headers communicate core information, such as what browser the client is using what data is being requested.

What is in an HTTP response?

* is what web clients (often browsers) receive from an Internet server in answer to an HTTP request. These responses communicate valuable information based on what was asked for in the HTTP request.
* Contains HTTPS status code, response headers and optional HTTP body

HTTP status code

* are 3-digit codes most often used to indicate whether an HTTP request has been successfully completed.
* Status codes are broken into the following 5 blocks:
  + 1xx Information
  + 2xx Success
  + 3xx Redirection
  + 4xx Client error
  + 5xx Server error
* The “xx” refers to different numbers between 00 and 99.
* Status codes starting with the number ‘2’ indicate a success. For example, after a client requests a web page, the most commonly seen responses have a status code of ‘200 OK’, indicating that the request was properly completed.
* If the response starts with a ‘4’ or a ‘5’ that means there was an error and the webpage will not be displayed. A status code that begins with a ‘4’ indicates a client-side error (It’s very common to encounter a ‘404 NOT FOUND’ status code when making a typo in a URL).
* A status code beginning in ‘5’ means something went wrong on the server side. Status codes can also begin with a ‘1’ or a ‘3’, which indicate an informational response and a redirect, respectively.

HTTP Response Headers

* convey important information such as the language and format of the data being sent in the response body.

BROWSERS AND HOW THEY WORK

* It retrieves information from other parts of the web and displays it on your desktop or mobile device.
* Every webpage, image and video has its own unique [Uniform Resource Locator](https://wikipedia.org/wiki/URL) (URL), which is also known as a web address.
* Websites save information about you in files called [cookies](https://wikipedia.org/wiki/HTTP_cookie).
* is the client run on a computer that contacts the Web server and requests information.
* group of structured codes which together performs a series of tasks to display a web page on the screen.

Main components of a browser:

* + **User Interface**: the space where User interacts with the browser. It includes the address bar, back and next buttons, home button, refresh and stop, bookmark option, etc. Every other part, except the window where requested web page is displayed, comes under it.
  + **Browser Engine**: The browser engine works as a bridge between the User interface and the rendering engine. According to the inputs from various user interfaces, it queries and manipulates the rendering engine.
  + **Rendering Engine**: The rendering engine, as the name suggests is responsible for rendering the requested web page on the browser screen. The rendering engine interprets the HTML, XML documents and images that are formatted using CSS and generates the layout that is displayed in the User Interface. However, using plugins or extensions, it can display other types data also.
    - Different browsers use different rendering engines:  
      \*Internet Explorer: Trident

\*Firefox & other Mozilla browsers: Gecko

\*Chrome & Opera 15+: Blink

\*Chrome (iPhone) & Safari: Webkit

* + **Networking**: Component of the browser which retrieves the URLs using the common internet protocols of HTTP or FTP. The networking component handles all aspects of Internet communication and security. The network component may implement a cache of retrieved documents in order to reduce network traffic.
  + **JavaScript Interpreter:**It is the component of the browser which interprets and executes the javascript code embedded in a website. The interpreted results are sent to the rendering engine for display. If the script is external then first the resource is fetched from the network. Parser keeps on hold until the script is executed.
  + **UI Backend**: UI backend is used for drawing basic widgets like combo boxes and windows. This backend exposes a generic interface that is not platform specific. It underneath uses operating system user interface methods.
  + **Data Persistence/Storage:** This is a persistence layer. Browsers support storage mechanisms such as localStorage, IndexedDB, WebSQL and FileSystem. It is a small database created on the local drive of the computer where the browser is installed. It manages user data such as cache, cookies, bookmarks and preferences.

What browsers do

* + **DNS Resolution:** This process makes sure that once the user enters a URL, the browser knows which server it has to connect to. The browser contacts a DNS server to find that google.com translates to 216.58.207.110, an IP address the browser can connect to.
  + **HTTP Exchange:** a way for the browser to communicate with the server what it needs, and for the server to reply back. It involves the client (our browser) sending a **request**, and the server replying back with a **response**.
    - Example of request: GET / HTTP/1.1Host: google.comAccept: \*/\*
      * GET / HTTP/1.1: with this first line, the browser asks the server to retrieve the document at the location /, adding that the rest of the request will follow the HTTP/1.1 protocol (it could also use 1.0 or 2)
      * Host: google.com: this is **the only HTTP header mandatory in HTTP/1.1**. Since the server might serve multiple domains (google.com, google.co.uk, etc) the client here mentions that the request was for that specific host
      * Accept: \*/\*: an optional header, where the browser is telling the server that it will accept any kind of response back. The server could have a resource that’s available in JSON, XML or HTML formats, so it can pick whichever format it prefers.
    - Example of response: HTTP/1.1 200 OKCache-Control: private, max-age=0Content-Type: text/html; charset=ISO-8859-1Server: gwsX-XSS-Protection: 1; mode=blockX-Frame-Options: SAMEORIGINSet-Cookie: NID=1234; expires=Fri, 18-Jan-2019 18:25:04 GMT; path=/; domain=.google.com; HttpOnly