**Internetwork**

-outside/ crossing boundaries.

-connecting devices together.

-global information system that is locally linked together by a globally unique address space based in the Internet Protocol (IP)

**ARPANET (1969)**

- Network operated by DARPA  
- connected 4 computers and let them communicate with each other.

**Internet**

- Global information system

- logically linked together by globally by unique address space based on the IP

- able to support communications using the TCP/IP

- provides, uses or makes accessible, either publicly or privately, high-level services layered on the communications and related infrastructure.

**1972**

- “email” – first killer application that was deployed.

**World Wide Web (WWW)**

* 1989 (Sir Tim Berners-Lee) CERN
* HTTP, HTML, URL, 1st web server and 1st web client

HTTP – communication of web server and client

URL – addressing of the pages

Web Client – fetches document from web server

**WAIS (Wide Area Information System)**

- Internet system in which specialized subject databases are created at multiple server locations.

**Hypertext Transfer Protocol (HTTP)**

* Application layer, accessing web resources
* High-level protocol
* Developed by w3c and IETF

Connection

Request

Response

Disconnection – terminate

**HTTP Fundamentals**

* runs on top of the TCP/IP (port 80 –default)
* based on client-server architecture

client – user-agent

- web browser, web crawlers/ spiders, end users

server – provides service that is consumed by clients

- origin servers(Apache), proxy server(Authentication), gateways, tunnels

* HTTPS – takes HTTP message and then encrypts it.
* uses request-response protocol (pull protocol)

- polling – periodically check information

- push – contacts server that there’s something new.

* Stateless communication protocol

- server serves you a request and then forget it.

* Support functionalities

- cache control – local storage wherein you can store files that you may use again in the future.

- content media type(Multipurpose Internet Mail Extension) Spec – text/html, img/jpg

- language and character specs – ASCII, Unicode

- content/transfer codings – compression

- content negotiation

- client-server protocol negotiation – can be downgrades/upgrades

- undertaken via the detail of HTTP

- persistent connections IO (connect, request, response, close)

- connection via socket and wait for the response

- HTTP 1.1 were established

* Request pipelining

- One active request only

- connect and wait for the response before you request another.

- HTTP 1.1 – streaming request

- HTTP 2 – server push request something and will return your request and another informations related to it (package)

* Authentication/ authorization

**HTTP Resource Addressing**

* Uniform Resource Identifier (URI)

**-** Two subtypes: URN and URL

**-** Uniform Resource Name can identify but don’t tell where the resource is (ISBN)

**-** Uniform Resource Locator identifies a resource and identify where it resides.

* Uniform Resource Locator

- scheme (http or https) ftp:// , file://

- authority (server and port)

Host(name of the machine where the server is running)

-registered domain name

-Domain Name System (DNS)

- path – unix file system /\_/\_/

- document root /-from the root

- static or dynamic resources

- static served as is while dynamic usually scripts and query

- query - ? products.php?id=1245&color=red&size=XL

- key=value pairs with & separators between key/value pairs

- may be URL-encoded(stud.jsp?name=John+Jones

-fragment identifier - #

**HTTP Request Message**

Request –separated by single space

* Method
* Request URI – identify for the method being target
* HTTP Protocol Version

**HTTP Message Headers**

* General Header Fields

- Connection, data, pragma

* Request Header Fields

**-** client, service agent

* Response Header Fields

**-** response message/ client

* Entity Header Fields

**-** describe payload – type of entity

Upgrade – Insecure – Requests

* Fetching something in HTTPS but there’s something link to a HTTP resource.

**HTTP Response Message**

Status Line: HTTP protocol version

Status Code: 3 digit code that indicates the stats

**HTTP Status Codes**

* Informational (1xx)
* Success (2xx)
* Redirection (3xx)
* Client Error (4xx)
* Server Error (5xx)