Internet – (internetwork), a global information system that is logically linked together by a globally unique address space based on IP.

* Began in 1969
* ARPANET – (Advanced Research Projects Agency Network) an early packet switching network and the first network to implement the protocol suite TCP/IP.(foundation of the internet)
* DARPA – (Defense Advanced Research Projects Agency) research program to allow networked computers to communicate transparently across multiple, linked packet networks.
* 1972 – electronic mail/email

Inter – outside boundary

Intra – within a boundary

Network – connecting devices together

Nodes – different devices connected to the network.

IOT – internet of things

Network Protocol – standardized, a formal set of rules that governs how computers and other network devices exchange information over a network.

WWW – a system made up of internet servers, all the interlinked HTML pages that can be accessed over the Internet. (1989) by Tim Berners Lee www was designed in (1991) while he was at CERN.

CERN – center for European nuclear research

W3C – (World Wide Web Consortium) Web standards.

RFC - (Request for Comments) is a formal document from the (IETF) that is the result of committee drafting and subsequent review by interested parties.

IETF – (Internet Engineering Task Force) defines standard Internet operating protocols such as TCP/IP.

WAIS – (Wide Area Information Server) do thing document by document.

Gopher (protocol) – almost similar with WAIS

Usenet – forum ex. Stack overflow

Web servers– web program that host web resources

Client application – ex. browsers

Semantic Web – a more intelligent web

HTTP – application layer communication protocol used to access resources (hypertext/hypermedia) on the WWW.

* Invented by Tim Berners-Lee
* Jointly developed by the W3C and IETF
* Version history:
* HTTP 0.9 (1991)
* HTTP 1.0 (RFC 1945, May 1996)
* HTTP 1.1 (RFC 2068 Jan. 1997, RFC 2616 June 1999) RFC 7230-7235 June 2014
* HTTP 2 (RFC 7540 May 2015)

SPDY – protocol of Google

* HTTP Fundamentals – http runs on the top of TCP/IP using TCP port 80 by default or TCP port 443 for HTTPS (HTTP over SSL/TLS)

SSL – (Secure Sockets Layer) cryptographic protocol, encrypt link between a server and a client

TLS – (Transport Layer Security) cryptographic protocol that provide communications security over a computer network

* http is based on a client server architecture
  + Client aka User Agents (UA):
    - Web browsers, web crawlers/spider, other end user tools and applications
* Servers:
* Origin servers – the server on which a given resource is to be created.
* Proxy servers – acts as an intermediary between a server and a client.

IANA – (Internet Assigned Numbers Authority) is a department of ICANN, oversees global IP address allocation, autonomous system number allocation, root zone management in the Domain Name System (DNS), media types, and other Internet Protocol-related symbols and Internet

ICANN – (Internet Corporation for Assigned Names and Numbers) is responsible for coordinating the maintenance and procedures of several databases related to the namespaces of the Internet, ensuring the network's stable and secure operation.

DNS – (Domain Name System) system for converting alphabetic names into numeric IP addresses.

HTTP – uses a request – response standard protocol

* The client sends an HTTP request message to the server
* The server processes the request and replies with an HTTP response message

HTTP – is a stateless communication protocol

* Push – server can now initiate, server contact the client
* Servers do not keep information about clients in-between request (only keeps access log).
* Pull –
* Polling – periodically checks the new information manually or uses script and it is not very efficient.

HTTP – provides support for other functionalities such as:

* Cache control – local storage
* Content media type (MIME) specification
* Language and character set specification
* Content or transfer codings
* Content negotiation
* Client server protocol negotiations
* Persistent connections
* Request pipelining
* Authentication or authorization

MIME – (Multipurpose Internet Mail Extensions) ex. Text, img

Google analytics – search engine, know the traffic of your website

Additional protocol:

* Server push – request something the server response and also request something (server volunteer to request associated resources)
* HTTP Resource Addressing

HTTP Resource – are identified using URI’s or more specifically HTTP URL’s.

URN – (Uniform Resource Name) name ex. ISBN an international book number or serial number, UPC a universal product code

URL – (Uniform Resource Locator) locator ex. Links

* Scheme – http/https
* Authority – information about the server, port, authentication
  + - User information or authentication credentials (deprecated)
    - Host – domain name (resolved to an IP address using DNS) of the server where the resource resides or will be created.
    - Port number – optional
* Path to resource (resolved relative to the document root on the server)

Relative URL – without http://.....:.....@.......:81

Absolute URL – complete may refer to a static or dynamic resource (ex. /info/profile.php?)

* + Query – typically provided as key = value pairs, with ampersand separators between key or value pairs. – may be URL encoded
  + Fragment identifier – start with hash tag or pound sign ex. Page1.1#xyz…
* HTTP Request Message
* Request Line
  + Methods – HEAD, GET, POST, PUT, DELETE, TRACE, CONNECT, OPTION
    - WebDAV – (WWW Distibuted Authoring and Versioning) extension to HTTP collaborative editing and file management
  + Request URI – target
  + HTTP Protocol Version
* Message Headers (general, request, response, entity) ex. fieldname:value
  + General – used by either client or server
  + Request – client
  + Response – server
  + Entity – content

HTTP 1.1 – requires at least the Host request header to be provided

* Empty Line (CRLF)
* Message Body (payload) – optional
  + DNT – do not track, an extension
  + Upgrade – Insecure – Request – also an extension
  + Blank line after the Accept- Language – means terminate
* HTTP Response Message
* Status Line (CRLF terminated line consisting of 3 space separated values)
  + HTTP protocol version
  + Status Code – always the same
  + Reason Phrase – can change
  + Status Code
    - Information(1XX)
    - Success(2XX)
    - Redirection(3XX)
    - Client Error(4XX)
    - Server Error(5XX)
    - Message Headers (general, request and entity header)
    - Empty Line – terminate
    - Message Body – (payload) optional
  + Request header – ex. HOST, user-agents
  + General header – ex. connection both request and response
  + HTML, Script, CSS – text file
* HTTP Request Methods
* Standard Methods
  + GET – transfer of a current selected representation of the target resource.
  + HEAD – same as GET, but only transfer the status line and header section.
    - Composer – you can compose a request and modify GET to HEAD
    - Code rot - slow deterioration of software performance over time eventually lead to software becoming unusable
    - Link rot – hyperlinks that have become permanently unavailable.
      * Link checker – look for the value and try fetching.

Fiddler – a transparent proxy

* POST – perform resource specific processing on the request payload.
* PUT – replace all current representations of the target resource with the request payload.
* DELETE – removes all current representations of the target resource.

Dav- APACHE WAMP

* Extension Methods
* WebDAV (RFC 4918)
  + - PROPFIND, PROPATCH, MKCOL
      * PROPFIND – retrieves properties
      * PROPPATCH – processes instructions
      * MKCOL – creates a new collection resource
  + OPTION - special syntax is to add \* - means asking server a general information
  + TRACE – request a loop back of the request message (ei. Request the server to echo back to the client the received request message) – typically used for testing or diagnostics of the request or response chain.
    - Transfer-Encoding: chunked – section of the data
  + CONNECT – request the establishment of a (encryption) tunnel to the destination origin server and if successful there after restrict it behavior to blind forwarding of packets, in both directions, until the tunnel is closed.

Fiddler – man in the middle – intercept

* Categorize – safe method, not safe, idempotent method, cacheable method
* Safe Method – no changes has done – GET, HEAD, OPTION, TRACE
* Not safe – POST, PUT, DELETE
* Idempotent – GET, HEAD, OPTION, TRACE, DELETE, PUT – no changes even if you retrieve several times, same result – POST is not idempotent because POST can make changes
* Cacheable methods
* HTTP Message Headers

4 Categories

* General Header Fields
* Request Header Fields
* Response Header Fields
* Entity Header Fields
* General Header Fields
  + **Cache control** – (ex. Cache-control: max-age=0)
  + **Connection** – (ex. Connection: keep-alive) allows the sender to specify options that are desired for that particular connection and MUST NOT be communicated by proxies over further connections.
  + **Date** – date at which the message we generated.
  + **Pragma** – used to include implementation- specific directives that might apply to any recipient along the request/response chain.
  + **Trailer** - header is at the end of the data.
  + **Transfer encoding** - what type of transformation has been applied to the message body in order to safely transfer it between the sender and the recipient.
  + **Upgrade** – 101 switching protocols
  + **Via** – where message past through used for tracking message
  + **Warning** – that error is not on my hand
* Request Header Fields
  + **Accept -** specify certain media types which are acceptable for the response.
  + **Accept Charset -** character sets are acceptable for the response.
  + **Accept Encoding -** similar to Accept, but restricts the content-codings that are acceptable in the response.
  + **Accept Language –** similar to Accept, but restricts the set of natural languages that are preferred as a response to the request.
  + **Authorization –** a user agent wishes to authenticate itself with a server.
  + **Expect –**indicate that particular server behaviors are required by the client.
  + **From -** should contain an Internet e-mail address for the human user who controls the requesting user agent.
  + **Host -** specifies the Internet host and port number of the resource being requested(generally an HTTP URL)
  + **If – Match -** conditional
  + **If – Modified – Since -** conditional
  + **If – None – Match -** conditional
  + **If – Range –** complete partial content
  + **If – Unmodified – Since -** conditional
  + **Max – Forwards -** provides a mechanism with the TRACE
  + **Proxy – Authorization -** allows the client to identify itself to a proxy which requires authentication.
  + **Range –** transfer of only one or more sub ranges of the selected representation data rather than the entire data.
  + **Referer –** specify the address (URI) of the resource from which the Request-URI was obtained  (Traffic)
  + **TE -** transfer- encoding.
  + **User Agent -** contains information about the user agent originating the request.

Dedicated hosting – single computer in a network reserved for serving the needs of the network. (expensive)

Shared virtual hosting – hosting multiple domain names (cheaper)

Stale – under If – None – Match

Etag – can be disabled

Precondition Failed – 412 – operation failed since it is already created.

* Response Header Fields
  + **Accept – Ranges –** allows the server to indicate its acceptance of range request for a resource.
  + **Age –** estimate the amount of time since the response was generated at the origin server.
  + **ETag -** provides the current value of the entity tag for the requested variant.
  + **Location -** to redirect the recipient to a location other than the Request-URI for completion of the request
  + **Proxy –** part of 407 (Proxy Authentication Required)
  + **Retry-After -** used with a 503 (Service Unavailable) response to indicate how long the service is expected to be unavailable to the requesting client.
  + **Server -** contains information about the software used by the origin server to handle the request.
  + **Vary -** to reply to a subsequent request without revalidation.
  + **WWW – Authenticate -** MUST be included in 401 (Unauthorized) response messages.
* Entity Header Fields
  + **Allow –** lists the set of methods supported by the resource identified by the Request URI.
  + **Content Encoding -** used as a modifier to the media-type.
  + **Content Language -** describes the natural language(s) of the intended audience for the enclosed entity.
  + **Content Length -** indicates the size of the entity-body.
  + **Content Location -** used to supply the resource location for the entity.
  + **Content MD5 -** deprecated
  + **Content Range -** sent with a partial entity-body to specify where in the full entity-body the partial body should be applied.
  + **Content Type -** indicates the media type of the entity-body
  + **Expires -** gives the date/time after which the response is considered stale.
  + **Last Modified -** indicates the date and time at which the origin server believes the variant was last modified.
* HTTP Status Code
* **Information (1XX)** - indicates an interim response for communicating connection status or request progress prior to completing the requested action and sending a final response.

**100 – continue** - request has been received and has not yet been rejected by the server.

**101 – switching protocols** - server understands and is willing to comply with the client's request, via the Upgrade header field for a change in the application protocol being used on this connection.

* **Success (2XX)** - indicates that the client's

request was successfully received, understood, and accepted.

**200 – ok** - request has succeeded.

**201 – created** - request has been fulfilled and has resulted in one or more new resources being created.

**202 – accepted** - the request has been accepted for processing, but the processing has not been completed.

**203 – non authoritative information -** the request was successful but the enclosed payload has been modified from that of the origin server's 200 (OK) response by a transforming proxy.

**204 – no content -** the server has successfully fulfilled the request and that there is no additional content to send in the response payload body.

**205 – reset content -** user agent reset the "document view".

**206 – partial content -** the server has fulfilled the partial GET request for the resource.

* **Redirection (3XX)** - indicates that further action needs to be taken by the user agent in order to fulfill the request.

**300 – multiple choices -** the target resource has more than one representation, each with its own more specific identifier, user can select a preferred representation.

**301 – moved permanently -** the target resource has been assigned a new permanent URI.

**302 – found -** the target resource resides temporarily under a different URI.

**303 – see other -** the server is redirecting the user agent to a different resource.

**304 – not modified -** the client has performed a conditional GET request and access is allowed, but the document has not been modified.

**305 – use proxy -** now deprecated

**306 – (unused) -** no longer used

**307 – temporary redirect -** the target resource resides temporarily under a different URI and performs an automatic redirection to that URI.

**Client Error (4XX)** - indicates that the client

seems to have erred.

**400 – bad request -** the server cannot or will not process the request due to something that is perceived to be a client error

**401 – unauthorized -** The request requires user authentication.

**402 – payment required -** is reserved for future use.

**403 – forbidden -** the server understood the request but refuses to authorize it.

**404 – not found -** the origin server did not find a current representation for the target resource .

**405 – method not allowed -** the method received in the request-line is known by the origin server but not supported by the target resource.

**406 – not acceptable -** the target resource does not have a current representation that would be acceptable to the user agent.

**407 – proxy authentication required -** similar to 401 (Unauthorized), but indicates that the client must first authenticate itself with the proxy.

**408 – request timeout -** the server did not receive a complete request message within the time that it was prepared to wait.

**409 – conflict -** the request could not be completed due to a conflict with the current state of the target resource.

**410 – gone -** access to the target resource is no longer available at the origin server and that this condition is likely to be permanent.

**411 – length required -** the server refuses to accept the request without a defined Content-Length

**412 – precondition failed -** the precondition given in one or more of the request-header fields evaluated to false when it was tested on the server.

**413 – request entity too large -** the server is refusing to process a request because the request payload is larger than the server is willing or able to process.

**414 – request URI too large -** the server is refusing to service the request because the request-target is longer than the server is willing to interpret.

**415 – unsupported media type -** the

origin server is refuse to service the request because the format is not supported.

**416 – requested range not satisfiable -** if a request included a Range request-header field and none of the range-specifier values in this field overlap the current extent of the selected resource.

**417 – expectation failed –** the expectation given in the request's Expect header field could not be met.

**426 – upgrade required (new addition using a feature which will require to upgrade) -** the server refuses to perform the request using the current protocol but might be willing to do so after the client upgrades to a different protocol.

**Server Error (5XX)** - indicates that the server

is aware that it has erred or is incapable of performing the requested method.

**500 – internal server error -** the server encountered an unexpected condition that prevented it from fulfilling the request.

**501 – not implemented -** the server does not support the functionality required to fulfill the request.

**502 – bad gateway -** the server, while acting as a gateway or proxy, received an invalid response from an inbound server it accessed while attempting to fulfill the request.

**503 – service unavailable -** the server is currently unable to handle the request due to a temporary overload or scheduled maintenance.

**504 – gateway timeout -** the server, while acting as a gateway or proxy, did not receive a timely response from an upstream server it needed to access in order to complete the request.

**505 – http version not supported –** the server does not support, or refuses to support, the major version of HTTP that was used in the request message.

HTTP Extensible -protocols