**Webtek Lecture Prelim Notes**

Internet (originally “Internetworking”) 🡪 FNC (Federal Networking Council) 🡪 global information system

FNC- provided the formal definition of “Internet” (1995)

Nodes (e.g Computer)

IP address- used to identify every node in the network

Types of Interconnection Internet Protocol  
a) wired IPv4- 32 bit  
b) wireless IPv6- 64 bit

Protocols- communication standards W3C- World Wide Web Consortium

TCP/IP 1960’s- packet switched communication  
 - store and form  
ARPANET- (1969) 1972- e-mail  
WWW- Sir Tim Berners-Lee, CERN, Switzerland (1989)

WAIS (Wide Area Information Servers) - full text search engine for individual Internet Gopher prtotocols.  
 - client-server text search engines

Before browsers 🡪 Gopher protocol- distributing, searching, retrieving documents over the Internet.  
 - predecessor of the World Wide Web  
 🡪 Usenet- worldwide distributed discussion system available on computers.

Search Engines (Three standard finding tools)  
a) Archie- FTP archives  
b) Veronica- Gopher protocol  
c) Jughead- Gopher protocol

HTTP- Hypertext Transfer Protocol  
HTML- Hypertext Markup Language

HTTP 🡨🡪HTML 🡨🡪 URL Web server- web content, hosts  
 URL (Uniform Resource Locator)- web addressing

Web client application 🡪 Web browsers

HTTP- used to send requests and responses

Present: Semantic Web

HTTP  
 -Tim Berners-Lee, W3C, IETF (Internet Engineering Task Force)  
 -Application Layer Communication Protocol used to access resource on the www (Hypertext, Hypermedia)

Version History  
 -HTTP 0.9 (1991)  
 -HTTP 1.0 (RFC 1945, May 1996)  
 -HTTP 1.1 (RFC 2065 January 1997, RFC 2616 June 1999, RFC 7230-7235 June 2014)  
 -HTTP 2 (RFC 7540 May 2015) 🡪 patterned after SPDY by Google.  
 🡪 Transport Layer Protocol

2005 onwards 🡪 mobile and web space evolvement

HTTP Fundamentals  
 -Http runs on top of TCP/IP, using TCP port 80 by default or TCP port 443 for HTTPS (HTTP over SSL/TLS) .  
 -TCP- Transmission Control Protocol  
 -Packet- created when nodes want to communicate with other nodes.  
 -HTTP based on a client-server architecture  
 -client, a.k.a user agents (UA).  
 -Web browsers, web clients/spider, other end user tools and applications.

Port # 🡪 integer: 0-64k 🡪 more specific address

Creates default ports for well-known protocols.  
 -IANA (Internet Assigned Numbers Authority)  
 -ICANN (Internet Corporation for Assigned Names and Numbers)

IP Address + Port # = network socket 🡪 internal endpoint for sending or receiving data ata single node in a computer network.

Encryption Layer 🡪 used in https  
 🡪 it won’t make sense, encrypted data

TCP/IP- basic communication language or protocol of the Internet.

Servers  
 -origin servers  
 -proxy servers

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 communications protocol  
 -servers do not keep information about client in-between requests.

“polling” protocol - which nodes want to access the network  
“push” protocol (HTTP2)- server will push data

Provides support for other functionalities, such as:

* Cache control
* Content media type (MIME) specification
* Language and character set specification
* Content negotiation
* Client-server protocol negotiations
* Persistent connections (HTTP 1.1)
* Request pipelining (HTTP 1.1)

HTTP lets us control the cache

MIME- Multipurpose Internet Mail Extensions

URI- Uniform Resource Identifier 🡪 string of characters used to identify a resource.

HTTP resources are identifies using URI’s (RF 3986), or, more specifically, http URLs

* Scheme (http or https) \*FTP is also possible
* Authority

-user information or authentication credentials (deprecated)

* Host- resource locations  
   - domain name (resolved to and IP address using DNS) of the server whereee the resources resides (or will be created)
* Port number
* Path to resource- resolve relative to the document post on the server.  
   - may refer to a static or dynamic resources
* Query (start with ‘?’)  
   -Typically provided as key = value pars, with ampersand (&) separators between key/value pairs.

HTTP Request Message

* Request Line (CLRF-Terminated Line consisting of three space-separated values)  
  -Method  
  -Request URI  
  -HTTP Protocol Version
* Message Headers (general, request, and/or entity header)  
  -HTTP 1.1 requires at least the host request header to be provided
* Empty Line
* Message body and Payload (optional) @ HTTP Response Message
* Status line CRLF (Carriage Return Line Feed)- Terminated line consisting of 3 space-separated value status code: informational
* Reason phrase
* Empty Line (CRLF)
* Message Body (optional)

HTTP Request Methods

* Standard Methods

1. GET

-most commonly used HTTP methods  
-transfer a current selected representation of the resource identified by the request.

1. HEAD  
    -same as GET, except that the entity is not included.  
    \*Content-Length- how large the resource is.  
    \*Link rot- broken overtime  
    2xx- they are there  
    4xx- they are not there
2. POST  
    -perform processing of a specific data.
3. PUT  
    -putting info under that URI  
    -store the closed entity in the message boy under the specified request of the URI (i.e, the resource identified by the request URI is either created or replaced, using the enclosed entity).
4. DELETE  
    -remove the resource.  
    \*Extension Methods- WebDAV (RFC 4918) MKCOL  
    - PROFIND, PROPATCH, COPY, LOCK, UNLOCK  
    \*Safe Methods
5. OPTIONS  
    -what can I do?  
    -OPTIONS\*- general capabilities
6. TRACE  
    -request a loop-back of the request message (i.e, request the server to echo back to the client the received request message)  
    -typically used for testing/diagnostics of the request/response chain  
    -Chunk Encoding 🡪 Transfer-Encoding  
    -not sure how long the data is.
7. CONNECT  
    -request the establishment and a tunnel to the destination origin server, if successful, thereafter restrict its behaviour to blind-forwarding of packets, in both directions, until the tunnel is closed.  
    -commonly used to an end- to-end virtual connection through one or more.

* Safe methods, idempotent methods, cachable methods
* HTTP Message Headers (Describe Payload)
* General Header Fields
* Cache-control
* Connection
* Date
* Pragma
* Trailer
* Transfer-encoding
* Upgrade
* Via
* Warning
* Request Header Fields
* Accept
* Accept-Charset
* Accept-Encoding
* Accept-Language
* Authorization
* Expect
* From
* Host
* If-match
* If-modified since
* If-None-Match
* If-Range
* If-Unmodified-Since
* Max-forwards-forwards back
* Proxy-Authorization
* Range
* Referer
* TE- Trailer Encoding
* User-Agent

\*if no entity tag 🡪 If-modified-since  
\*Accept-Encoding 🡪 most known: Encrypt Encoding

* Response Header Fields
* Accept-Ranges
* Age
* E-tag
* Location
* Proxy authenticate
* Retry-after
* Server
* Vary
* Header Authenticate
* Entity Header Fields
* Allow
* Context-Encoding
* Content-language
* Content-Length
* Content-location
* Content-MD5
* Content-range
* Content-type
* Expires
* Last-Modified
* HTTP Status Codes
* Informational (1xx)

1. 100 continual
2. 101 Switching Protocols

* Success (2xx)

1. 200 ok
2. 201 Created
3. 202 Accepted
4. 203 Non-Authorative Information
5. 204 No Content
6. 205 Reset Content
7. 206 Partial Content

* Redirection (3xx)

1. 300 Multiple Choices
2. 301 Moved Permanently
3. 302 Found
4. 303 See other
5. 304 Not Modified
6. 305 Use Proxy
7. 306 (unused)
8. 307 Temporary Redirect

* Client Error (4xx)

1. 400 Bad Request
2. 401 Unauthorized
3. 402 Payment Required
4. 403 Forbidden
5. 404 Not Found
6. 405 Method Not Allowed
7. 406 Not Acceptable
8. 407 Proxy Authentication-required
9. 408 Request Time-out
10. 409 Conflict
11. 410 Gone
12. 411 Length Required
13. 412 Precondition Failed
14. 413 Request Entity too large
15. 414 Request URI too large
16. 415 Unsupported Media Type
17. 416 Requested Range Not Satisfiable
18. 417 Expectation Failed
19. 426 Upgrade Required (newly added, 2014 HTTP)

* Server Error (5xx)

1. 500 Internet Server Error
2. 501 Not Implemented
3. 502 Bad Gateway
4. 503 Service Unavailable
5. 504 Gateway Time-out
6. 505 HTTP version not supported