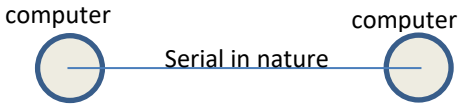
Web System and Technology Reviewer (PRELIMS)

Internet

- comes from: internetworking; inter (outside the boundary); network (connecting devices together); intra (inside the boundaries)

- global network of computer to communicate



*stand alone are not useful without network

- nodes – devices that are connected to the network

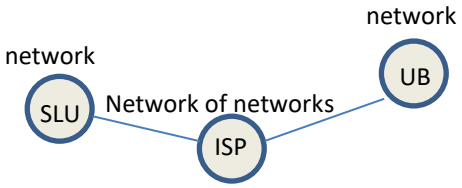
- Internet of Things (IoT) – way of equipping common household with connectivity

    - access it remotely

- Interconnection: 2 options:

    - Wired – Unshielded Twisted Pair (UTP) , Fiber : physical connection

    - Wireless – electromagnetic signaling : infrared, Bluetooth, wireless fidelity (WiFi)



- Device Drivers

    - network protocols

- History

    - October 24 1995

        - Internet – global information system that is logically linked together by a globally unique address space based on the IP (internet protocol)

        - 192.168 – ipv4 32-bit addressing; ipv6 – 64 – bit addressing

        - TCP (transmission control protocol) – sending information over the net

            - provides high level services publicly or privately

            - a platform that create higher level services for users

    - 1969 (established)

        - 4 computers (that talks to one another) connected to the ARPANET (advance research projects agency network) by DARPA (defense advanced research projects agency) – pocket switch network (research)

    - 1972

        - first killer application was deploy on the web (electronic mail – e-mail)

World Wide Web (WWW) (1989)

- Inventor: Sir Tim Berners-Lee

    -Worked at CERN

    - Made the HTTP, HTML, URL, Web Server, Web Client, Hyperlinking of documents

    - Scientist and Mathematicians

-Wide Area Information Servers (WAIS) - periodically queries databases and get the result and prints what you need

- Gopher(protocol) – goes to indexes and to your document

*there is no link of one document from another

- ip address – identify the node in the network

- **WWW Technologies**

Hypertext Transfer Protocol (HTTP)

- application layer communication protocol used to access

    - standard that is used by application that communicate with one another

- invented by Tim Berners-Lee (Timothy)

- jointly developed by the W3C (world wide web consortium) and the IETF (internet engineering task force)

- version history

    - HTTP 0.9 (1991)

        - original HTTP as defined in 1991

        - phases: connection -> request -> response -> disconnection (4 steps)

    - HTTP 1.0 (RFC 1945, May 1996)

    - HTTP 1.1 (RFC 2068, Jan 1997; RFC 2616, Jun 1999) (RFC 7230-7235 Jun 2014)

    - HTTP 2.0 (RFC 7540, May 2015)

        - application on the client/server side

-SPDY – accessing from the browser

- HTTP(unsecure) Fundamentals

    - runs on top of TCP/IP using TCP port 80 by default, or TCP port 443 for HTTPS (secured) (HTTP over SSL/TLS (secure socket layer/transport layer security)

    - 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 (eg. 200 ok)

    - is a stateless communications protocol

    - server do not keep information from client in-between

    - server send a requests and forget about it

    - based on a client-server architecture

        + clients a.k.a user agents (ua)

            - web browser, web crawlers/spiders, other end user tools and applications

        + server

            - origin server

            - proxy server (eg. Authentication server), gateways, tunnels

            - access by the client

    -IANA – internet assigned numbers authority

    -ICANN – Internet Corporation for assigned names and numbers

    - push protocol – server can volunteer information

    - polling – periodically check for the information

    -provides support for other functionalities, such as:

        - cache control – stored of CSS

        - content media type (MIME – multipurpose internet mail extensions) specification

        - language and character set specification

        - content/transfer codings

        - content negotiation

- client-server protocol negotiations
    - 3 versions:
        - HTTP/ HTTPS 1.0, 1.1, 2 – for the upgrades and downgrades according to the server.
- persistent connections – http 1.1
- request pipelining – one connection
- authentication/authorization (eg. Screen readers – to read the content for you) content and transfer coding

- HTTP resource addressing

– identifies using URI (uniform resource identifier) (RFC 3986) or more specifically, HTTP URIs

- 2 subtypes of URI:
    - URN – uniform resource name (eg ISBN(international standard book number), UPC(universal product code))
    - URL – uniform resource locator (eg. Google.com)
- authority
    - user information/aunthentication 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
- path to resource (resolved relative to the document root on the server)

*absolute authority – http://usr:pwd@server.org:81/ in/prof.php – path ; <img src: “a.jpg”>

- may refer to a static or dynamic resource

*relative

- query
    - typically provided as key = value pairs, with (& - ampersand) separators between key/value pairs.
    - maybe URL-encoded
- fragment identifier (eg. #adr) – starts with a hashtag #

- HTTPS – get HTTP message and encrypts (transformational that is math based)
    - letencrypt.org – digital certificates for free
- **HTTP Request Message**
    - Request line (CRLF (carriage return line feed) – terminated line consisting)
    - Method
        - Standard Method

            Safe Methods: get (retrieve), head, options, trace, connect

            Modify server/ not safe: post, put, delete

            Idempotent: get, head, put, delete, options, trace
        - Extension Method

            - WebDAV (RFC 4918) - PROPFIND, PROPPATCH, MKCOL (make a collection/create folder), COPY, MOVE, LOCK, UNLOCK

            - Options – query what the server can do

            - Trace – request a loop-back of the request message; typically used for testing/diagnostic of the request/response chain; to know where you committed error (eg. 34 – length of the section (hex 1 bytes)

                Trace /index.html HTTP/1.1

                Host: www.webtek.org (actual data))

            - Connect – establishment of a tunnel to the destination origin servers and if successful, thereafter restrict its behavior to blind-forwarding of pockets in both directions, until the tunnel is closed

            *cacheable method

            *safe method

            *idempotent method
    - Request URI

        *RFC 2616 – 4236 Update Required

- **HTTP Protocol Version**
    - Message Headers (general, request and/or entity header)
        - fieldname: value
        - general (request and response, cache-control), request, respond, entity (payload to your message, allow) , header fields
        - HTTP 1.1 requires at least 1 header (most request)
        - empty line (CRLF) – part of the structure; to end
        - message body aka payload (optional)
        - General (use by user agent) , Request (server), Response (server) , Entity (payment or actual body) HEADER FIELDS
            - **General Header Fields**
                - cache – control – request/response; local storage to put resources (eg. Browser, gateways); no cache, max-age = 0
                - connection – establish persistent connections; (eg. keep-alive: timeout=5, max=100; close)
                - data – indicate when your transfer the resource
                - pragma – generic directive
                - transfer encoding
                - trailer – list of header
                - upgrade – switching to a different
                    - h2 – http v.2; h2c – https v.2
                - via – walkthrough
                - warning – errors
            - **Request Header Fields**
                - (eg. Accept, accept-charset, accept-encoding, accept-language, host)
                * */*; lowest priority
                *html, application
                * q = priority value; highest :1(default); lowest:0
                *image/webp (negotiation) = higher priority

- Accept language – en-US, en;q=0.8, es;q = 0.6
    - setting -> advance setting -> language and input setting
    - language information by the agent
- Accept encoding – concept length = 1893 bytes resource
    - none, compressed html (decompressed to be actual html)
    - gzip, deflate, sdch (can handle)
    - vary – response header
- Authorization – authorize user can access; www.authentic; if not authorized: 401
- Proxy – authentication
- Expect – 2 face operation
    - content length (send) -> expect: 100 continue -> 100 continue/ 417 expectation failed
- From – contact address of the user generating the request
- Host – required from HTTP 1.1; dedicated hosting: 1.1; shared virtual hosting: 1:m

- **Response Header Field**
    - Accept –Ranges
    - Age – age of the response; cache resource to particular age
    -ETag – entity tag; soft/weak identifier for entity; conditional request
    - Location – redirection; (eg other methods)
    - proxy-authenticate – enable authorization/ authenthication with HTML
    - WWW -authenticate
    - retry-after – can't serve now, try again later on
    - server – use indicate information about the server

- vary

- **Entity Header Fields**

- allow

- content – encoding

- content – language – request content; identify language

- content – length – bytes

- content – location – located the exact file

- content – MDS – deprecated; hashing algorithm; change the content; hash value

- content-range – bytes 0-100; content range = 100

- content-type – text/html (kind of resource)

-expires – retrieve to another content; when to end

- last-modified – controlling the cache

- Conditional Header

- if-match

- request – if-non-match: “764-567cad0823” (hex bytes long)(date/time) entity tag (identifier for entity)

- condition – if-modified.since: sun, 05 feb 17 19:04:51 GMT; conditional retrieval

- If-ranges(part of copy that you have = send the copy) and accept-ranges (support: partial-retrieval if)

- MAX – Forward – use for tracing

- referrer page: reference of other sites/page

- Trailer encoding (TE) – end of the data; chunk or compressed

- User-Agent – identity of the user and client; responsive deignl have 2 different content (mobile and web (full desktop) website)

- Extended Headers

- upgrade – insecure requests

- DNT (do not track)

- Entity Header – content- length and content-type

- get – request to transfer; supported by all compliant server

- head – entity not included as a response; retrieve metadata without transferring the entity itself

    - code rot and link rot – don't maintain your sites; link checker – read HTML File -> parse the content (web resources (eg. <link href>) -> fetch it (get: it means that exist/alive)

- post – resource-specific entity; submit data -> get/post

- put – store the enclosed entity in the message body under the specified request URI

- delete – remove the request URI

- **HTTP Response Message**

    - status line (special line)

        - HTTP protocol version

        - reason phrase – descriptive

        - status code – case of failure (numeric)

    - status code

        - **1xx (informational)**

            - 100 continue

            - 101 switching protocols – request to switch using upgrade; protocol negotiation

        - **2xx (success)**

            - 200 ok

            - 201 created

            - 202 accepted (not process but accepted)

            -203 non authoritative information

            - 204 no content

            - 205 reset content (submit data via POST)

            - 206 partial content

- **3xx (redirection) (location:)**
    - 300 multiple choices – request with several version
    - 301 moved permanently – no longer there
    - 302 found (redirection)
    - 303 see other (redirection)
    - 304 not modified – conditional but has missing
    - 305 use proxy – don’t fetch it directly
    - 306 (unused)
    - 307 temporary redirect
- **4xx (client error)**
    - 400 bad request
    - 401 unauthorized (protected)
    - 402 payment required (commercial application)
    - 403 forbidden (not allowed to access)
    - 404 not found (not correct request)
    - 405 method not allowed
    - 406 not acceptable
    - 407 proxy authentication required
    - 408 request timeout
    - 409 conflict
    - 410 gone (none)
    - 411 length required
    - 412 precondition failed
    - 413 request entity too large
    - 414 request URI too large
    - 415 unsupported media type
    - 416 requested range not satisfiable
    - 417 expectation failed

        - 426 upgrade required (3000 characters)
    - **5xx (server error)**
        - 500 internal server error
        - 501 not implemented
        - 502 bad gateway (cot configure)
        - 503 service unavailable
        - 504 gateway time-out
        - 505 http version not supported

HTTP – extensible protocol

- basic ways (communicate to new base protocol functionality):
    - new request method
    - new message header
    - new status code

**NOTES ©