\*\*Dito natin lagay mga notes natin :)\*\*

**Internet** - global network of networks(google)

- hierarcy, infrasturcture

INTER NETWORKING

**inter** - outside

**intra** - within the border

**Network** - interconnected devices ('nodes')

- share data

characterized by

1**.Hardware**

a. nodes - the actual devices i.e computers

IoT - Internet of Things

b.Interconnected Technologies - Wired - utp, fiber; Wireless - infrared,bluetooth

2.**software**

a.protocols

b.device drivers -software that run active (something missing here)

computer to computer - serial communication

- one cable, send data with 1s and 0s

Problems of wired and wireless - susceptible to noise

**LAN** - Local Area Network

interconnection of networks with the use of ISPs

different ISPs can connect with each other

connection between countries - underwater(preferred)/ satellite

**1969 - modern internet was born**

circuit switch network -- used by old telephones

store and forward - cellphones (texting)

**IPv4 - 32-bit**

**IPv6 - 64-bit**

**Early 1960's -->** DARPA (Defense Advanced Research Project Agency) --> packet switch

circuit switch - relaying and keeping the connection

packet switching

send and forward --> send-->store-->send-->...until it reaches the destination

**ARPANET** (Advanced Research Project Agency NETwork) --> 4 computers communicate

**First popular app --> email - 1972**

**World Wide Web** - information system that uses Hypertext links

- information system on the internet

- allows documents to be connected to other documents by hyperlinks

- was created for researchers and scientist to share their findings and research

- back then it is more textual

**Wide Area Information Service (WAIS)** - multiple server locations

- early information service

**Gopher (protocol)** - application layer protocol

- similar to WAIS

- hierarchical in nature

- distibuting, seraching, retrieving

**USETNET** - worldwide distributed discussion system

- similar to discussion groups

**1989 (Sir Tim Bernerns-Lee) --> HTTP, HTML, URL**

HTML -

URL –

**HTTP FUNDAMENTALS**

HTTP - jointly developed by the 23c and the IETF

- standard way of communicating

Version history

HTTP 0.9 (1991)

HTTP 1.0 (RFC 1945, edited may 1996)

HTTP 1.1 (RFC 2068 Rewritten specifications on Jan 1997)

HTTP 2 (RFC 7540 May 2015)

* Patterned by SPDY
* Backwards compatible with HTTP 1.1

-HTTP runs on top of TCP/IP, port 80 by default. Port 443 for HTTPS (HTTP over SSL/TLS)

-Socket is the combination of IP address and a port

-Port numbers range from 0-1024

-IANA

-HTTP is based on client-server architecture

* Clients AKA user agents

-web browsers, web crawlers/spiders (google bot), other end user tools

* Servers

-engine servers, proxy servers, gateways, tunnels

-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
* Pull push poll

-HTTP is a stateless communications protocol

* Servers do not keep info about clients in between requests

-HTTP provides support for other functionalities such as

* Cache control
* Content media type (MIME – multipurpose internet mail extension) specification
* Language and character specification
* Context/transfer codings
* Content negotiation
* Client server protocol negotiations
* Persistent connection
* Request pipelining
* Authentication/autorization

web Server

Web Client

Semantic web - next step

- uses natural language

**HTTP RESOURCE ADDRESSING**

Http resources are identified doing URI’s (RFC 3986) or more specifically HTTP URIs

* Scheme (http/https)
* Authority
  + 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
* Path – path to resource (resolved relative to the document root on the server)
  + May refer to a static/dynamic resource
* Query
  + Typically provided as key value pair with (&) separators between key/value pairs
  + May be URL – encoded.
* Fragment identifier (bookmark) - #

**HTTP REQUEST MESSAGE**

* Request time
  + Method
  + Request URI
  + HTTP Protocol Version
* Message Headers (general, request, and/or entity headers)
  + HTTP 1.1 requres at least the host request header to be provided
* Empty line (CRLF)
* Message Body

**HTTP RESPONSE MESSAGE**

* Status Line
  + HTTP Protocol Version
  + Status Code
  + Reason Phrase
* Message Headers (general, request, and/or entity headers)
* Empty Line
* Message Body

**HTTP REQUEST METHODS**

* Put
  + Store the enclosed entity in the message body under the specified request URI
* Delete
  + Delete a data in a server
* Options
  + To know what the option request type that is allowed to the client
* Trace
  + Request a loopback of the request (request the server to echo back to the client the received request message)
* Connect
  + Request the establishment of a tunnel
* Safe methods
  + Not affecting any data
* Idempoted methods
  + Repeatedf execution, same result
* Cacheable methods
* Extension Methods
  + WEBDAV RFC 4918
  + Propfind, proppicthc,mkcol,copy,move,lock,unlock.

**HTTP Message Headers**

**Response Header Fields**

* + - Used by servers
    - Response messages
* Accept Ranges
* Age
* Etag
* Location
  + used for redirection
* Proxy Authenticate
* Retry-after
* Server
* Vary
* WWW Authenticate

**Entity Header Fields**

* Allow
* Content Encoding
* Content Language
* Content Length
* Content Location
* Content-MD5
* Content Range
* Content Type
* Expires
* Last Modifier