//KTR//

Chapter-01

Infrastracture:

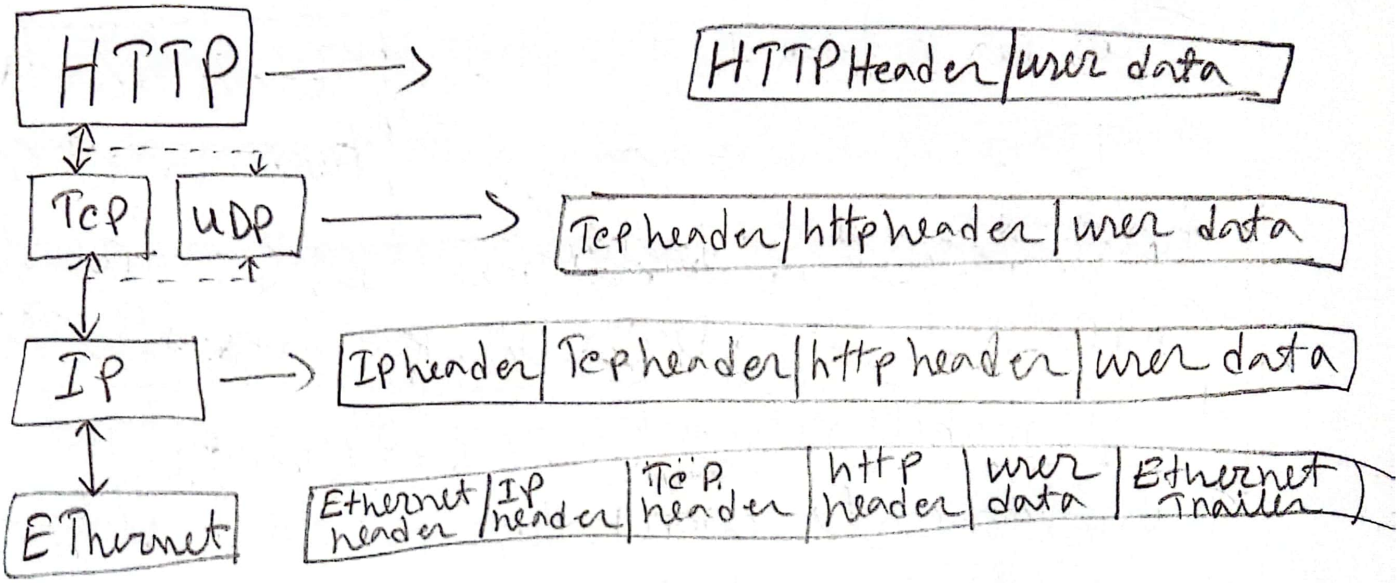
This is refers to the physical and virtual components that enables data transmission over the internet. It includes network infrastracture, protocols, ISPs, DNS, CDN, peering, cloud computing, mobile networks and IoT infrastructure. This elements work together to facilitate global connectivity and data exchange.

PDU (Protocol data unit):

It is the basic unit of exchange between entities that communicate using a specified networking protocol. There are several protocols uses PDU transfer in their protocol stack, such as TCP, udp, ip, Ethernet, HTTP, SMTP and DNS.

To transport a user data using layer specific protocol data units (PDU) with header, trailer and

transported service data units (SDU),



⊛ Format of a Tcp PDU / TCP Segment:

Source port	Destination port
Sequence Number	
Acknowledgement Number	
Data offset | Reserved | Control flag	Receiver window
Checksum	Urgent data pointer
Options (variable length)	
Application Data (e.g. Http, etc)	

⊛ IPv4 – packet format:

version (4 bit)	Head length (4 bit)	Type of service (8 bit)	Total length
Identifier (16 bit)		Flags	Fragment offset
Time to live (8 bit)		Protocol (8 bit) TCP, UDP, ICMP	Header checksum (16 bit)
Source IP Address (32 bit)			
Destination Ip address (32 bit)			
options			
Application data			

→ Purpose of header in IPv4 packet:

Header in an IPv4 packet, provides essential information for routing and delivering the packet. It helps to identify the packet, determine it's priority, specify it's size, set a maximum hop limit, encapsulated protocol and ensure header intrigrity.

components of web architecture in the internet:

① Client (web browser, mobile apps)

② web servers ③ Protocols (HTTP)

④ DNS ⑤ ISP ⑥ CDN ⑦ Database

⑧ web application frameworks and program-ming languages.

⑨ Security mechanism (SSL/TLS)

⑩ Standards and specifications defined by organizations like W3C (www consortium)

Basic components of a web architecture:

① clients: users device, like computer.

② Servers: stores and manages web resource and responds to requests.

③ Web browser: Displays web content to users

④ HTTP Protocol: Defines how data is transmit-ed over the web.

⑤ Webservers: Handels incoming requests and serves files.

⑥ Application / Database server

⑦ Font-end ⑧ Back-end

⑨ Caching: Stores frequently used data for quick retrival

⑩ load balancer: Distributes traffic among multiple servers

⑪ CDN (content delivery network): optimizes content delivery for speed.

HTTP protocol and its evolution:

✱ Basic notions:

-> WWW - world wide web
-> HTML - Hypertext markup language
-> HTTP - " transfer protocol
-> URL - uniform resource locator

#HTTP: Standard application protocol between web server and web client to data access and transport.

→ It applies Tcp as a reliable transport Protocol with port 80 at a server.

→ It is a readable ASCII protocol.

#URL: It is a uniform web address of each web page or objects in the web. URL denotes the logical location of an instance in a network context.

Ex: http://example.com/books/book-039

#URI:

uniform resource identifier, is a 7-bit encoded ASCII character string to identify objects in a uniform manner.

#URN: uniform resource name, an info

unit in the namespace having a designated identity.

Example: urn:isbn:987222156002 > URI
URN

#URC: uniform resource characteristics, it is a meta data of a resource.

Task and feature of HTTP:

⊛ Http uses tcp transport service to secure the data transportation. A client request initiates a Tcp connection and that will be closed after the data transfer in http 1.0 and kept open for further transfers in http 1.1.

⊛ HTTP is stateless. Because, state management is to expensive for a protocol. It will make the operation slow and complex

Persistent and non-persistent tcp connection:

Non-persistent:

-> Each request opens an own Tcp connection for the object data transport, of the replies.
-> The server closes the connections after a transfer.
-> It used in HTTP 1.0

Persistent:

-> An opened connection remains open.
-> All further object request use that open tcp connection.
-> used in HTTP 1.1 (in sequential order)
-> New request issued without waiting for a reply of previous one, in the same Tcp packet.
-> Better performance with small objects.

Limitations of Http 1.1, TLS and Tcp

-> Tcp connection in HTTP uses 3-way handshake and several slow starts for upto 4-8 parallel Tcp connections in the browser.

-> Head of line blocking due to a slow process of the first request.

-> High TLS complexity

Http 1.0 vs 2.1 vs 2.0 :

Every http protocol version is stateless.

* Http 1.0 :

-> Each http request establish a new connection to the server and is closed after receiving the response.

-> Connections are non-persistent.

-> Headers are often lengthy and inefficient.

-> Resource loading is slow because of multi-

-ple connection required.

→ Caching: Weak caching management.

⊗ HTTP (1.1):

→ Pnerist ent connection; which allows multi-ple HTTP request and response to be sent over a single TCP connection.

→ Host header; It allows multiple site to be hosted on a single ip address, which helps to adopt the virtual hosting feature.

→ Chunked Transfer-Encoding: It helps to transfer large amount of data efficiently.

→ Caching mechanisms were improved.

→ Compression; Added content compression to reduce data transfer size.

⊛ HTTP 2.0:

→ Multiplexing request: Http 2.0 can send multiple requests for data in parallel over a single TCP connection. Helps to faster loading times.

→ Header compression: Header fields are compressed and reduced the size to save bandwidth.

→ Server push: Here server can proactively push resources to client, like a server can push CSS or JavaScript files after the initial HTML request.

→ Stream prioritization: Requests can be prioritize by assigning priority level.

→ Binary protocol: uses binary protocol instead of text, which is efficient to data parsing.

→ Backward compitable: It designed to be compitable with HTTP 1.1, allowing older clients or server to work with it.

#QUIC protocol:

QUIC (Quick UDP Internet connection) is a transport protocol developed by google.

-> It is a generally applicable, secure, connection oriented session layer protocol with multiplexing functionality for data stream.

-> It uses UDP connection instade of Tcp in HTTP2.0. And combined with TLS 1.3.

-> Multiplexing of parallel uni- or bi-direct-ional data streams with or without reliab-ility and security requirements.

-> uses encrypted, authenticated packet header and payload elements for a data streams.

-> It prevents the HOL (head of line) blocking of data stream during the multiplexing along Tcp transport channel.

→ This proto-col can reconstruct lost data using a Redundant encoding of messages, e.g. by fountain codes.

Chapter: 1.2.2 / TCP/IP protocol stack

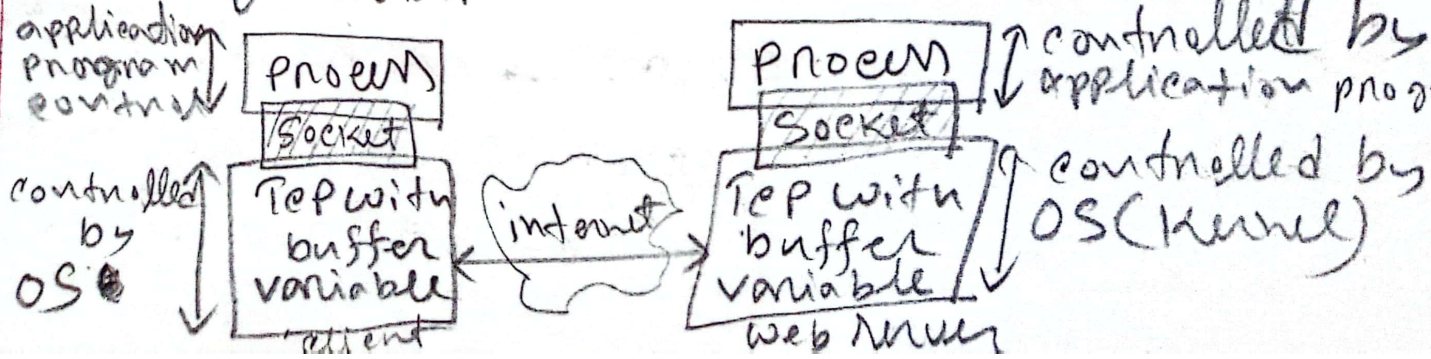
TCP/IP protocol suite:

-> DNS : Domain name system
-> HTTP: Hypertext transfer protocol
-> FTP : File Taansfer protocol
->SMTP: Simple mail transfer protocol
-> IMAP: Internet message access protocol
-> Telent: Remote call of programs

Transport connection setup by application program

-> Socket interface: It is a buffer and API for the data exchange between two application process in the user space and an end-toend transport protocol in the Kernel space.

-> Tcp transport service: This ensures a reliable transport of a byte stream from one process to another peering process on two interconnecting hosts.



Transport protocol concept:

① Link by link, perspective of the data link layer on a physical transmission segment.

② End to end, perspective of the transport layer on interconnected transmission segments called transmission path.

TCP features:

* 3-way handshake:

This is a fundamental process of TCP that establishes a reliable connection between a client and a server. It ensures both parties permission before data exchange. The 3 way handshake described below:

① Synchronize (SYN):

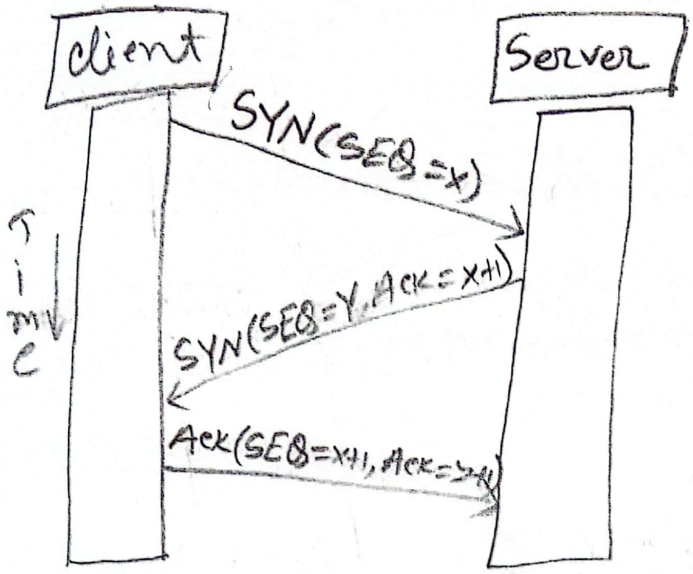
Here client initiates the connection and sends a Tcp segment with the SYN flag set to 1. Also a sequence number includes, which is an initial sequence number chosen

by the client. This SEQ number represents the first byte of data in the client message.

② Synchronize-Acknowledgement (SYN-ACK): Upon receiving the client's SYN segment, The server responds by sending a TCP segment with both SYN and ACK flags set to 1. This segment includes its own ISN (initial sequence number) and an acknowledgement number indicating the next expecting sequence number from the client.

③ Acknowledgement (ACK): Here client acknowledges the server's SYN-ACK segment by sending a TCP segment with the ACK flag set to 1. The ACK number is set to the server's ISN+1, indicating that the client is ready to start the data transmission.

TCP connection setup:

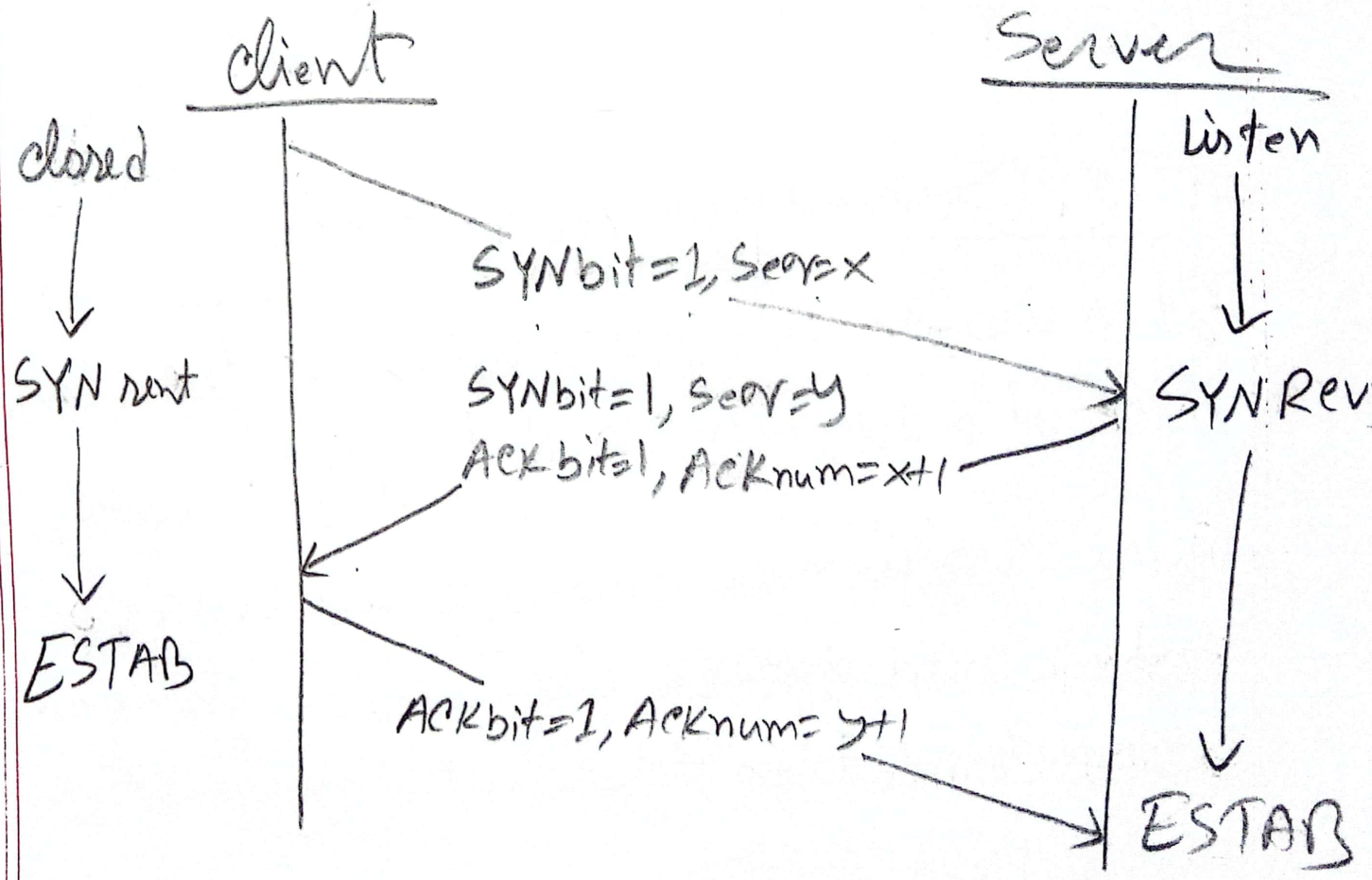


① Setting connection by a SYN message.

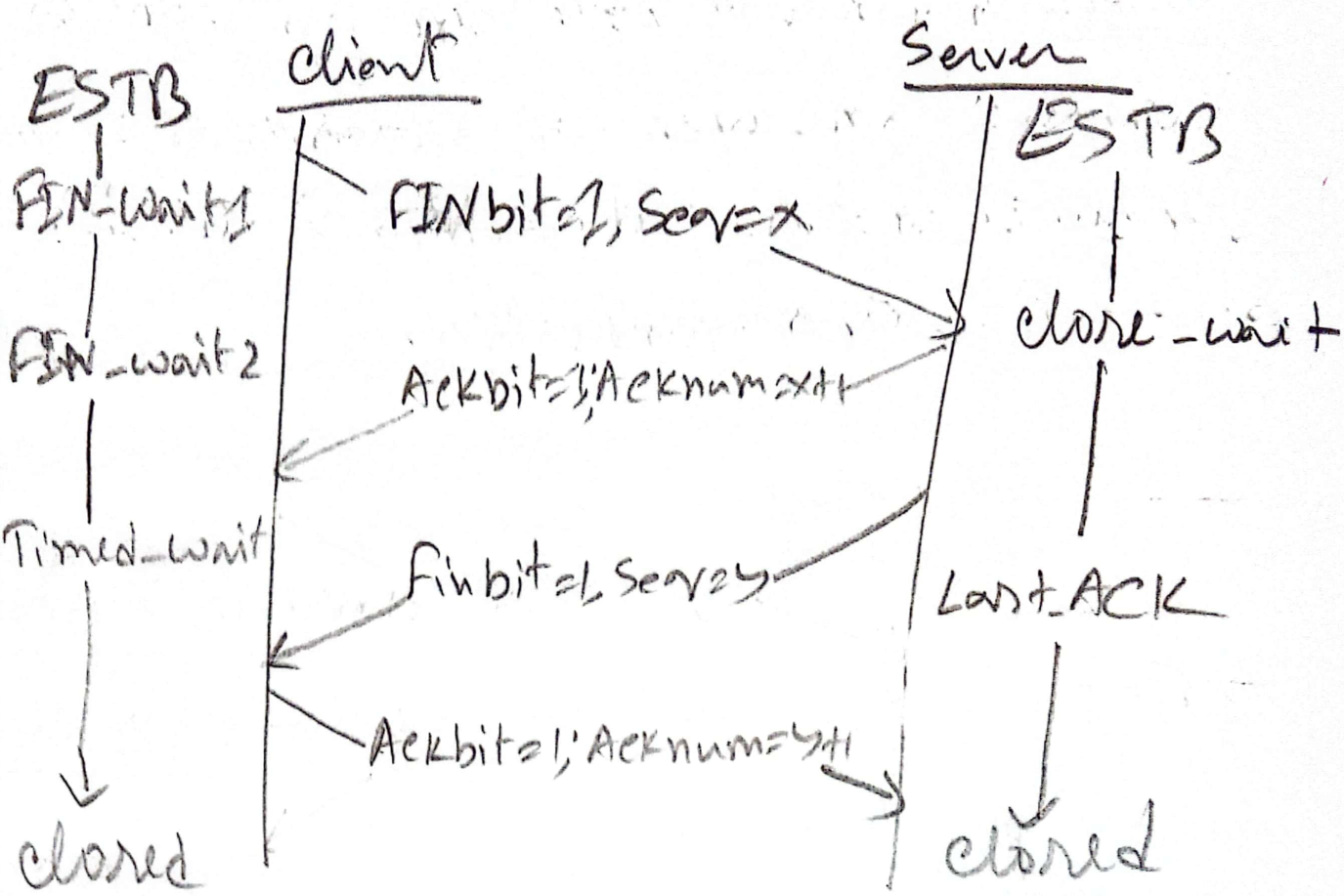
② At start numbering segments by a random sequence number.

③ The receiver for expected new segment by positive acknowledgement (cumulative Ack).

→ Tcp 3-way Handshake (Session establishment).



#TCP Session closing :-



#TCP Seqvance numbering :-

It is a byte stream number of the first byte of a segment's data part.

There seqvance numbers are used to track and manage data segments within a TCP connection to maintain data integri-ty and correctness.

#Acknowledgement Number: This is a seqnance number of the next byte expected from otherside. It ensures a reliable data transmission, maintain seqnance order and ~~control~~ supports flow control in Tcp connection.

#Flow control:

This is a mechanism that ensures, the sender does not overwhelm the receiver by adapting the rate of data transmission based on receivers capacity.

-> Receiver will "advertises" free buffer space by including a value of free buffer variable "rcwnd".

-> If the senders data transmission rate in too high or receiver's buffer is close to full, the receiver will reduce the advertised window size in Tcp acknowledgement. And will signal to slow down.

-> If the receiver have more space in buffer and can handle more data, it will increases

the advertised window size to allow sender for transmiting more data,
→ This dynamic window size adjustment ensures comfortable data transmission rate by a receiver, avoiding congestion and potential data loss.

#IP address class:

class	Range
A	1.0.0.0 to 126.0.0.0
B	128.0.0.0 to 191.255.0.0
C	192.0.0.0 to 223.255.255.0
D (multicast)	224.0.0.0 to 239.255.255.255
E (experimental)	240.0.0.0 to 255.255.255.255

#IPv4 address:

Logical name "wellington.cs.virginia.edu"
the IPv4 address is 128.143.137.144
IPv4 class B address: 128.143.0.0

host identification: 137.144
network mask : 255.255.0.0

→ Prefix notation : 128.143.137.144/16

→ network prefix of a class B network is 16 bits long.
→ Residual 16 bits defines the address portion for host identification.

Multiplexing/Demultiplexing:

→ Multiplexing (at sender):
It is a process of handling data from multiple sockets, add transport header of segment. which, later used for demultiplexing on receiver side.

→ De-multiplexing (at receiver):
It is process to deliver received segments to the correct socket's in an application process, using header information.

Connectionless Demultiplexing:

In this process, network protocols like UDP, direct incoming data packets to the correct application on the receiver side based on the destination port in the packet header. This allows multiple application to share network connection simultaneously without pre-established connection.

Port Numbers in TCP:

A port number identifies the endpoint of a TCP transport connection at the sending or receiving host of a traffic relationship. Some well known ports:

FTP - 21	Finger - 79	NNTP - 119
Telent - 23	HTTP - 80	
SMTP - 25	POP-3 - 110	
TFTP - 69		

Name Resolution:

→ HTTP Name resolution:

TCP does not operate with hostnames, that's why name must be translated into a IPv4 / IPv6 address.

→ DNS name resolution:

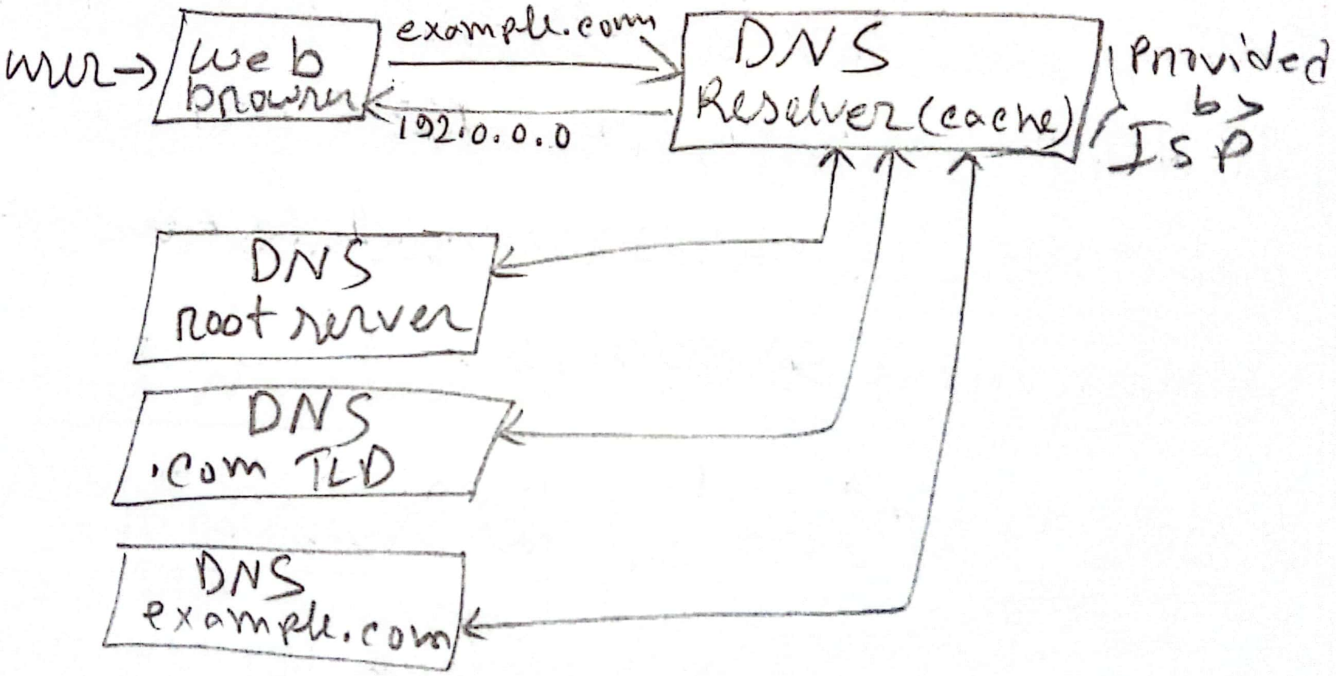
Name resolution of a logical host name to an IP address performed by sending a request to a distributed database system, which called Domain Name System.

DNS: It is a critical system that helps to translate human readable domain name to IP address to locate resources on the internet.

→ Domain:- A human readable name that represents a group of devices/resources on the internet. It is organized hierarchically with sub-domains branching of the main domain

→ DNS Zone: A collection of DNS records for a specific domain or subdomain in DNS namespace. Zones are used for administrative control and management of DNS settings.

DNS working mechanism:



* It is also the procedure of dns lookup.

DNS Types:-

A → IPv4 address record. (ID = 1)

AAAA → IPv6 " " . (Id = 28)

NS = Name Server (id=2)
CName = Canonical name (id=5)
DNSKEY = DNS Key resource Record (id=48)
MX = Mail exchanger record (id=15)
URI = uniform resource Identifier (id=256)

ARP (Address resolution protocol):

It is a network protocol that helps devices to finds the physical (MAC) address associated to the known IP address within the same network (local). It also allows devices to communicate on the same network's segment by mapping ipaddress to MAC addresses through ARP request and responses.

//Chapter: 2//

#Cross layer communication of a client Sever:

/Noted in Pc/ 2.1/slide-23

#IPv4 packet header:

Version (4 bit)	IHL (4bit)	Type of Servier (8 bit)	Total length (16 bit)
Identification (16 bit)		Flags (3 bit)	Fragment offset (13 bit)
Time to live (8 bit)	Protocol (8 bit)		Header checksum (16 bit)
Source Address (32 bit)			
Destination Address (32 bit)			
options (variable)			padding (variable)

Private ip addresses:

Private ip address are reserved for use within privat network and not routable on the public internet:

IPv4 private ip addresses:-

- → class A: 10.0.0.0 to 10.255.255.255
- → class B: 172.16.0.0 to 172.31.255.255
- → class C: 192.168.0.0 to 192.168.255.255

IPv6 private ip addaress Prifix:

- → fc00::/7
- → fe80::/10

#IPv6 Networking:

It is a 128 bit address system for large scale use. This address consist of 8 segment, every segment is 16 bit.

Ex: 4BF5:0000:0000:0000:BA5F:039A:000A:2176

Type prefix	Rest of address

→ IPv4 Embedding address:

8bits	88 bits	32 bits
00000000	All 0s	IPv4 address

→ Header format:

4 bit	8bit	20 bit
version	Priority	Flow lable
Payload length	next Header	Hop limit
Source address		
Destination address		

//KTR — Important Links for Lab //

vyos docker hub → unibaktr, for init.vbash basic config and lab config instruction.

vyos interface command → for init files

vyos next-hop ~~commands~~ interface → for static route in init.vbash

vyos dhcp command → for dhcp commands.