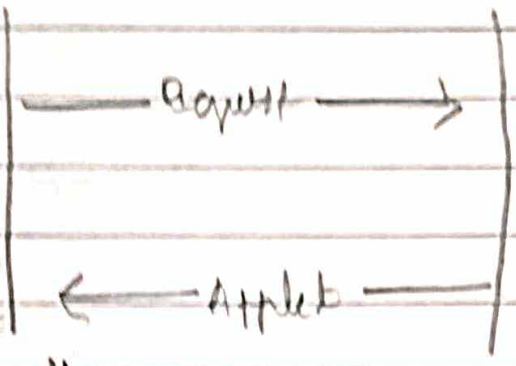
Active (also known as client-side dynamic document

client server



Run the applet to get result

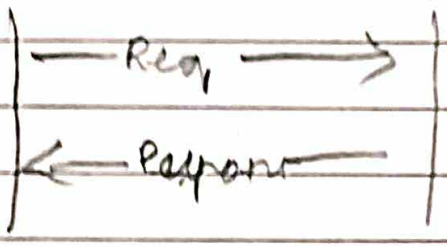
Result

⇒ HTTP

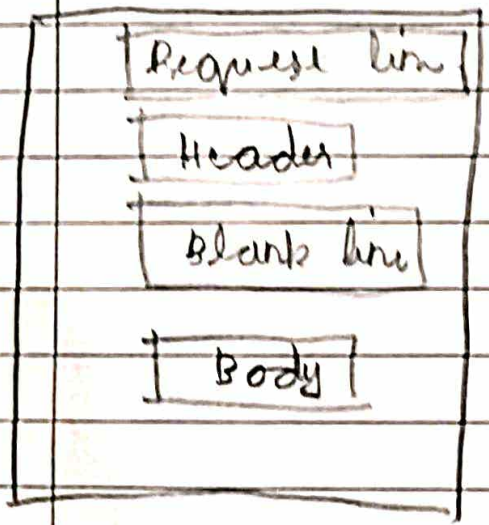
is a protocol used mainly to access data on WWW. combination of FTP and SMTP

• Port 80

client server



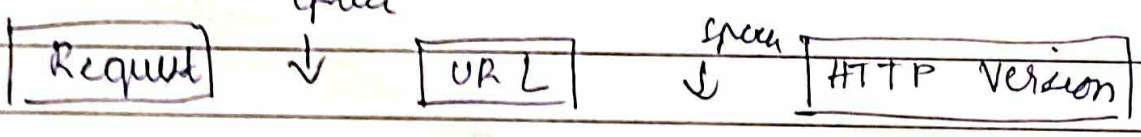
Request and response message



status line
Header
Blank line
Body

Request and static lines

Request



static

HTTP version

Status code

status phrase

Methods

GET - Request a document from the server
HEAD - " info about a doc but not the document itself
POST - Send some info from the client to the server
PUT
TRACE
CONNECT

Status Codes

100 - continue } informational
101 - switching }

200 - OK } SUCCESS
201 - Created
202 - Accepted
204 - No content

301 - Moved permanently } Redirection
302 - " temporarily
304 - Not modified

client error
400 - Bad req
401 - unauthorized
403 - Forbidden
404 - Not found
405 - Method not allowed
406 - Not acceptable

500 - internal server error } Server Error
501 - Not implemented
503 - service unavailable

Header format

Header name : Space Header value

General headers
HTTP headers that detects browser i.e. ~~specific info about caching~~ caching behaviour

Cache Control -

Connection - shows whether the port is closed or not
Date - shows current date __/__/____
MIME - Version -
upgrade - Specifies the preferred communication protocal.

Request headers

Accept - charset, encoding, language
Authorization
From
Host
if-modified-since, if-match, if-range, if-unmodified-since
Referrer
User agent

Response headers

Accept-range
Age - (timeline)
Public
Retry-after
Server

* HTTP Version 1.1 specifies a ~~con~~ persistent connection by default

12/23 Thursday

Browser ~~will~~ saves certain resources such as images and website data in a store known as cache.

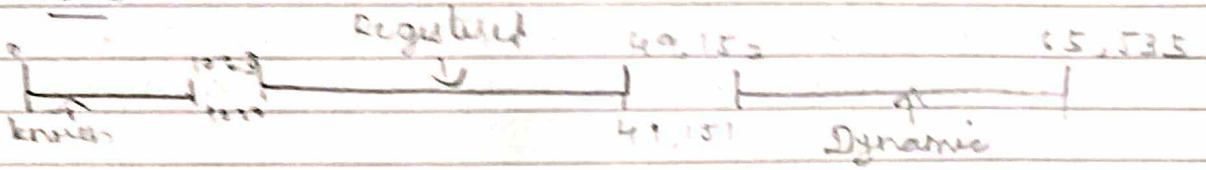
cache control

Browser will only store this resource for certain specified time period known as time to leave (TTL)

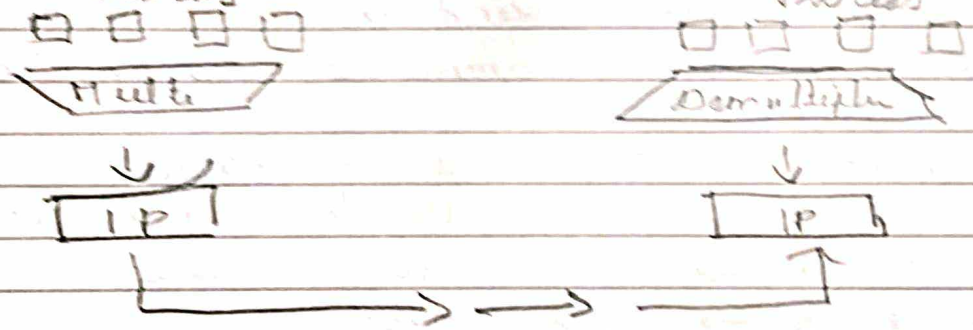
TCP

- used for process-to-process communication delivery
- communicate in client-server
  - → Port no select the process
  - → IP address select the host

Ports



Multiplexer - combinational circuit, which takes multiple input and gives a single o/p.



Demultiplexer - single i/p to multiple o/p.

Error control

- TCP is a connection-oriented protocol it creates a virtual connection b/w two TCP to send data.
- uses flow and error control.

stream - data is send byte by byte.

Sending and receiving buffers (used for flow control)

- Buffer considered as a circular array
- Buffer - temporary storage

Grey, white, Pink ~~Turkey~~ - data sent but not acknowledged
White → empty (ready to write)
Pink - data is available but it is not sent yet

⇒ * - The bytes of data being transferred in each connections are numbered by TCP. The numbering starts with a randomly generated no:

① → Suppose a TCP connection is transferring a file of 5000 bytes and the 1st byte is 10001 what are the sequence no: for each segment if data are sent in 5 segments.

Ans:- Segment 1 → Sequence no: 10,001 (range: 10,001 to 11000)

Segment 2 → Seq no: 11001 (range: 11001 to 12000)

" 3 → " 12001 (range: 12001 to 13000)

" 4 → " 13001 (range: 13001 to 14000)

" 5 → " 14001 (range: 14001 to 15000)

⇒ * The value in the sequence number field of a segment defines the no: of the 1st data byte contained in that segment.

⇒ * The value of acknowledgement number field in a segment defines the no: of the next byte a party expect to receive. The acknowledgement number is cumulative

TCP segment format

Sending and receiving buffers (used for flow control)

- Buffer considered as a circular array
- Buffer - temporary storage

Grey, white, Pink ~~stands~~ Grey - data sent but not acked, Pink - data is available but it is not send yet, White → empty (ready to write)

⇒ * - The bytes of data being transferred in each connections are numbered by TCP. The numbering starts with a randomly generated no:

① → Suppose a TCP connection is transferring a file of 5000 bytes and the 1st byte is 10001 what are the sequence no: for each segment if data are sent in 5 segments.

Ans: Segment 1 → Sequence no: 10,001 (range: 10,001 to 11000)

Segment 2 → Seq no: 11001 (range: 11001 to 12000)

" 3 → " 12006 (range: 12001 to 13000)

" 4 → " 13001 (range: 13001 to 14000)

" 5 → " 14001 (" 14001 to 15000)

⇒ * The value in the sequence number field of a segment defines the no: of the 1st data byte contained in that segment.

⇒ * The value of acknowledgement ~~number~~ field in a segment defines the no: of the next byte a party expects to receive. The acknowledgement number is cumulative

TCP segment format

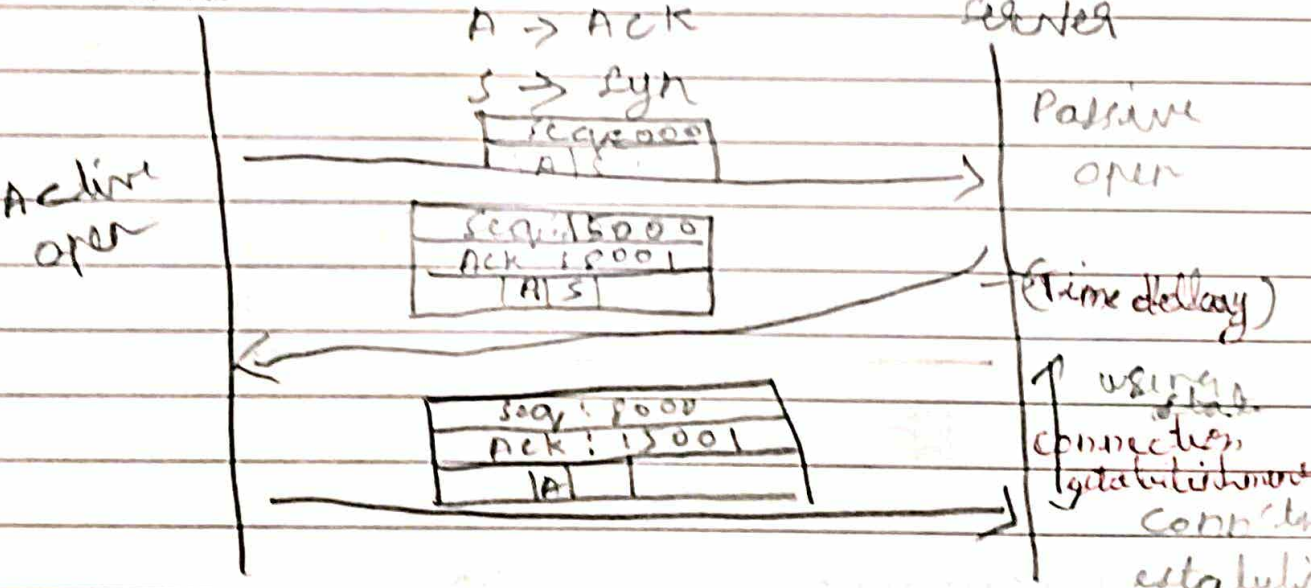
Control field (flags)

URG	- Urgent pointer is valid
ACK	- Acknowledgment "
PSH	- Request for push
RST	- Reset the connection
SYN	- Synchronize the sequence number.
FIN	- Terminate the connection

13/2/23 Monday

3 - way handshake

Passive - ready to accept request
Active " send request
client



Flooding attack - denial of service
↓
Blocking of legitimate users.

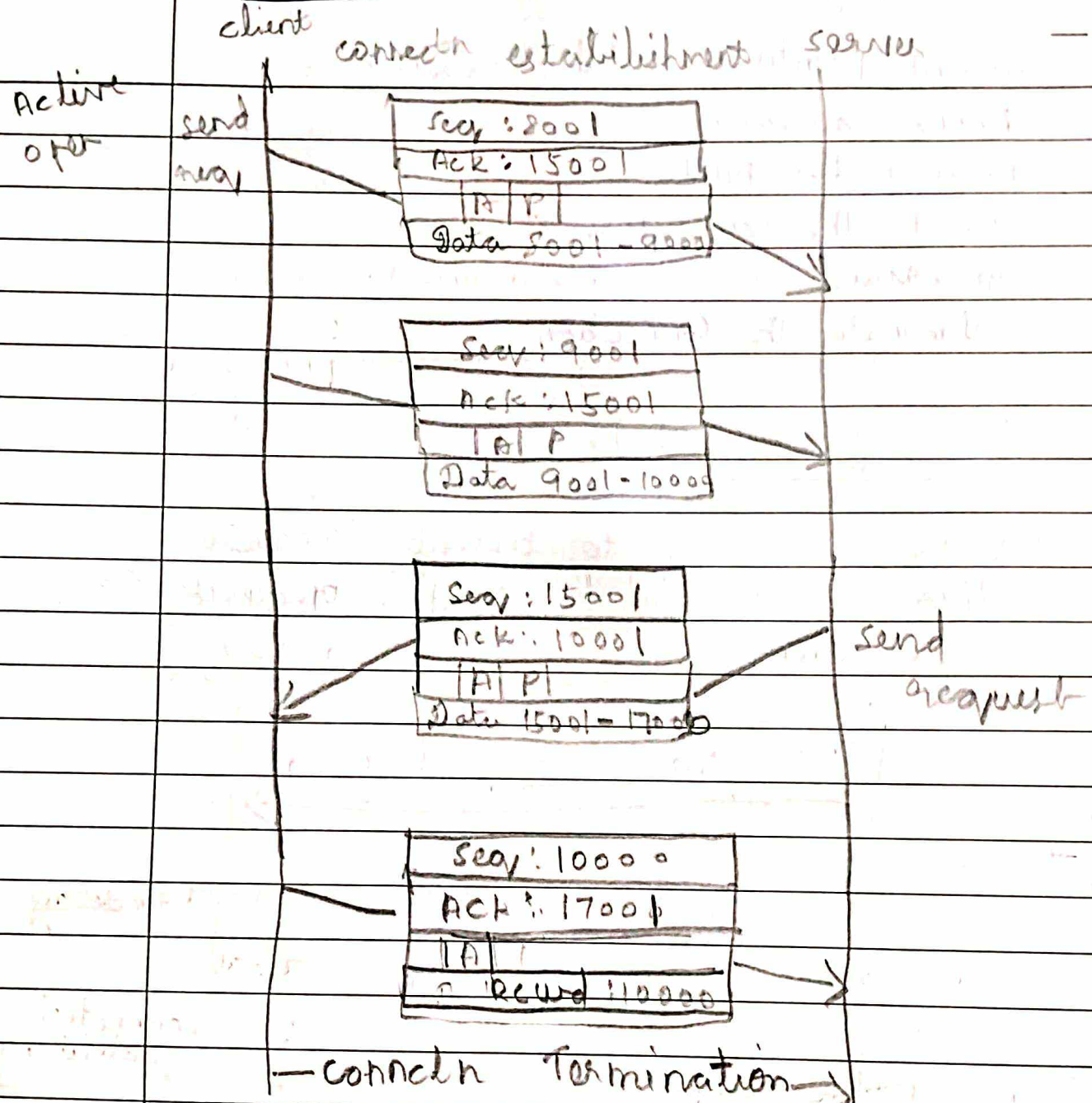
Syn flooding attack

~~Data~~

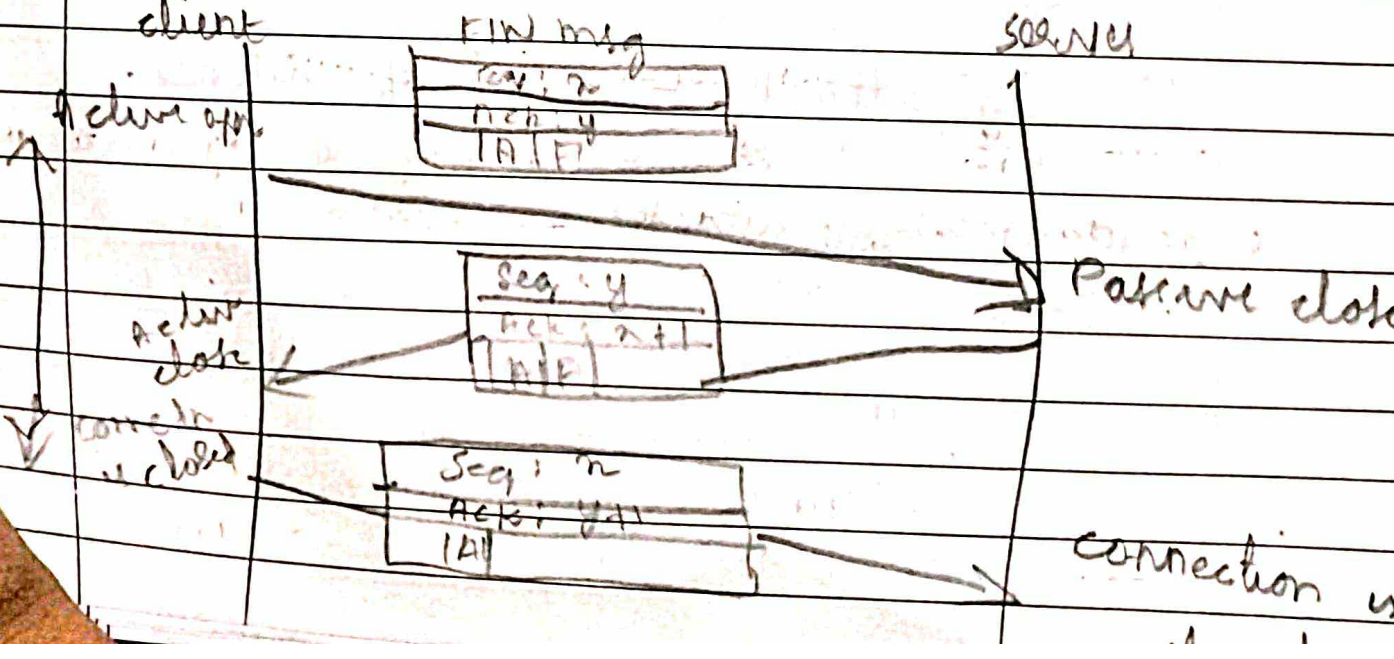
Data transfer

P = Push

__/__/____



connection Termination

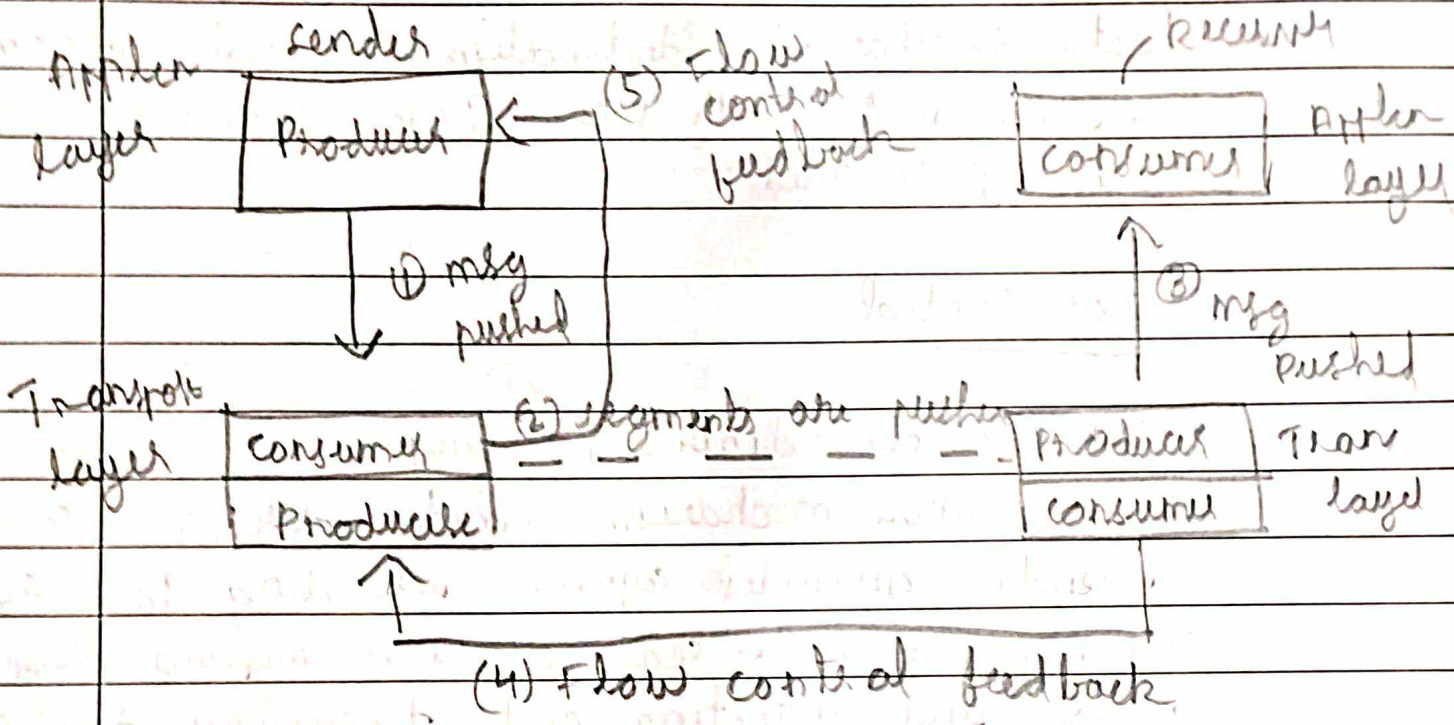


Flow control (sliding window)

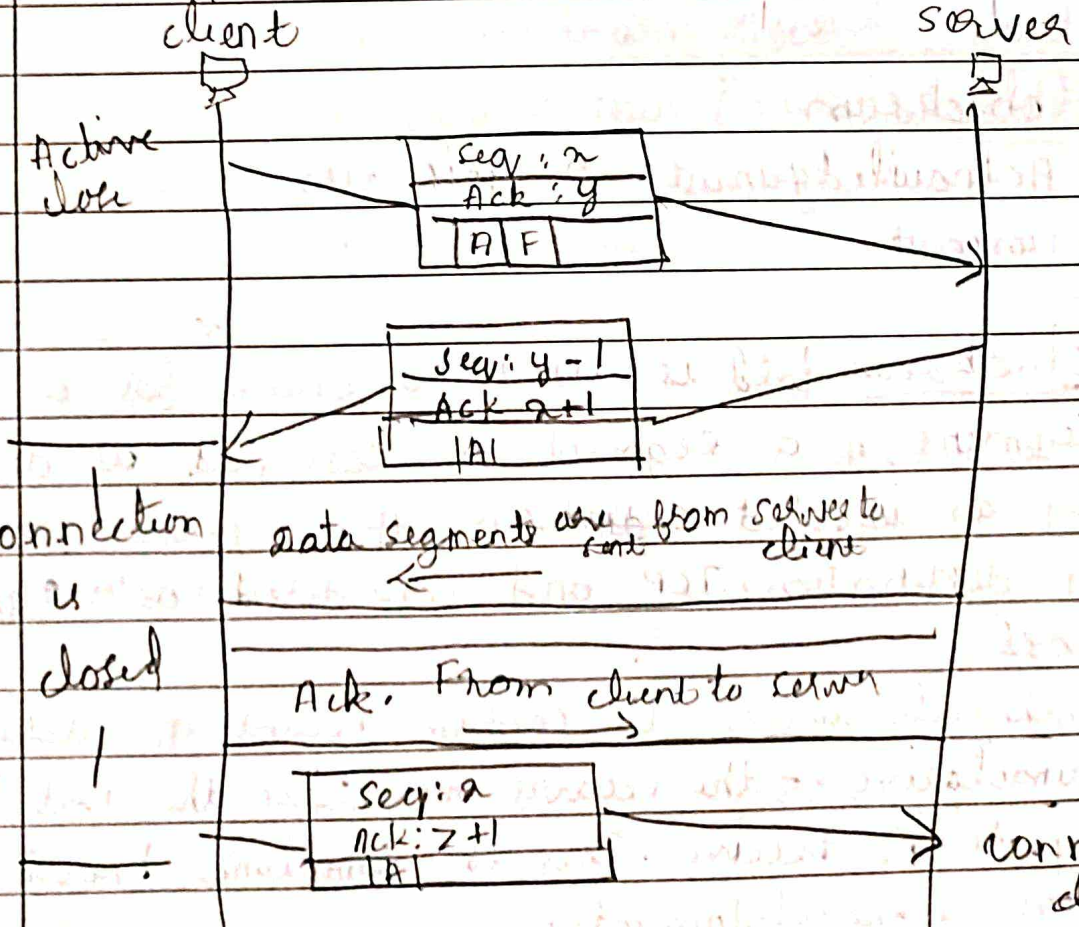
(sender)

⇒ • it balances the rate of producer creates the data with rate consumer can use the data

(receiver)



Half close



⇒ • TCP at one end may deny connection request, may avoid an existing connection or may terminate an idle connection using reset (RST) flag.

⇒ • Sliding window is used to make transmission more efficient as well as to control the flow of data so that the destination does not become overwhelmed with data. TCP sliding windows are byte-oriented.

Error Control

⇒ • TCP provides reliability using error control. Error control mechanism includes detecting and resending corrupted segments, resending lost segments, storing out of order segments until missing segment arrive and detecting and discarding duplicate segments.

⇒ • Error cntrl in TCP is achieved using 3 simple tools
1) checksum
2) Acknowledgement – 2 types cumulative, selective
3) Timeout

Checksum field is used to check for a corrupted segment, if a segment is corrupted as deleted by an invalid ~~segment~~ checksum, the segment is discarded by destination TCP and considered as segment lost.

TCP uses Acknowledgements to confirm receipt of data segment
Cumulative – the receiver advertises the next byte it expects to receive. This is sometimes known as +ve acknowledgements.

Selective acknowledgement reports additional info to the sender

The heart of error control mechanism is segment retransmission. when the retransmission timer expires or when the sender receives 3 duplicate acknowledgements for the previous segment in the queue then particular segment is retransmitted.

Congestion control

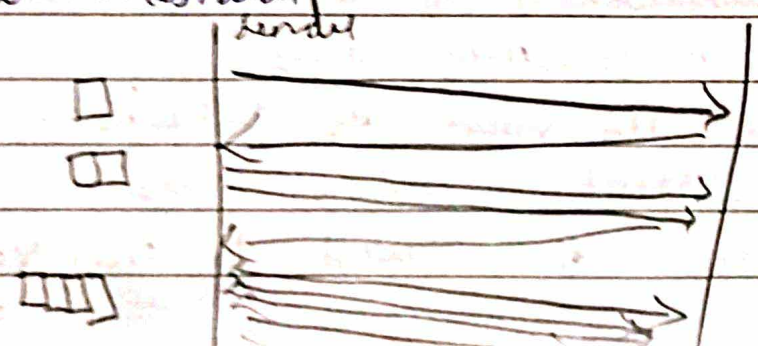
⇒ • ~~If the n/w~~ TCP uses congestion window and policy to avoid congestion

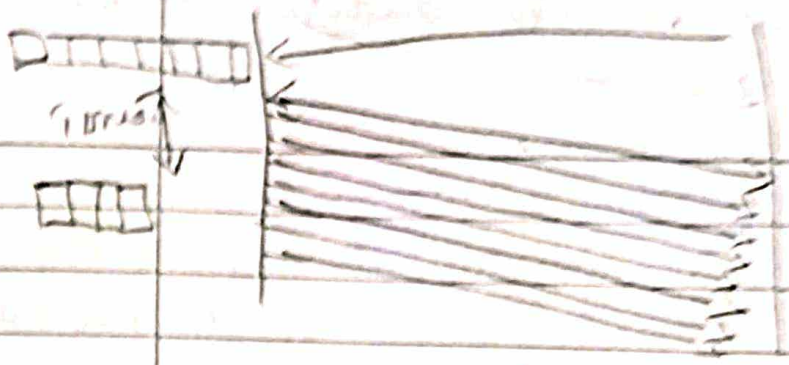
if the n/w cannot deliver the data as fast as it is created by the sender it must tells the sender to slow down. From the receive point of view n/w determines the size of sender window. The sender has 2 pieces of info 1st one is received (rwnd) advertised window size and the 2nd one is congestion (cwnd) window sized.

Actual window = min (rwnd, cwnd)

⇒ • TCP handles conjunction based on 3 phases slow start, congestion avoidance, congestion detection

In slow start algorithm, the size of the con. window increases exponentially until it reaches to threshold





⇒ • In congestion avoidance algorithm, the size of cwn window increases additively until cong is detected.

$i = 1$

cwnd i

" $i+1$

" $i+2$

" $i+3$

:

until threshold is detected

⇒ • If congestion occurs, the cwn window size must be decreased

i) If timeout occurs, there is a stronger possibility of congestion (a segment has probably been dropped in the n/w. Now TCP congestion detection mechanism reacts to this in following ways

a) it sets the value of threshold to half of the current window size b) it reduces cwnd

c) it starts slow start phase again.

if 3 duplicate acks are received there is vigour possibility of a congestion. Here now TCP reacts in following way

i) it states the value of threshold into half of the current window size.

ii) it states the cwnd to the value of threshold.

iii) it starts congestion avoidance phase.

15/2/23 wednesday

__/__/____

→ Data Encoding techniques

* Analog and digital transmissions

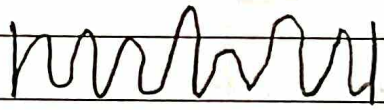
① → Digital data, Analog signals [modem]
② → " Digital " [wired LAN]
③ → Analog data digital signal [codec]

①

⇒ Basis for analog signaling is a continuous constant - frequency signal known as the carrier frequency.
Digital data is encoded by 3 characteristics
amplitude, frequency, phase


→ binary signal


- Amplitude modulation


- frequency modulation


phase modulation

phase changes when transition occurs.

⇒ Modem (modulator & demodulator)

no: of signals transmitted per second is called baud.

Modem ~~Modem~~ acts

→ Digital data Digital signals.

Digital signal - is a sequence of discrete, discontinuous voltage pulses.

Bit duration - the time it takes for the transmitter to emit the bit.

Issues are: Bit timing, recovery from signal noise immunity

⇒ NRZ (Non-Return-to-zero) codes

Voltage levels are to represent bits

NRZ-L is used for short distances b/w terminal and modem or terminal & comp

NRZ-I (invert on ones)

Voltage is constant during the bit interval.

It is a differential encoding scheme

Manchester encoding

1 <→ low-to-high transition

0 <→ high-to-low transition.

- There is always a mid-bit transition
- The direction of the mid-bit transition represents the digital data.

Differential manchester encoding

- mid bit transition is only for clocking

1 <→ absence of transition at the beginning of bit interval

0 <→ presence of transition at the beginning of the bit interval.

→ Analog Data - Digital signals

__/__/__

Eg: → PCM (Pulse code modulation.

· Multiplexing - · Analog signal is sampled

0 0000 → ——————

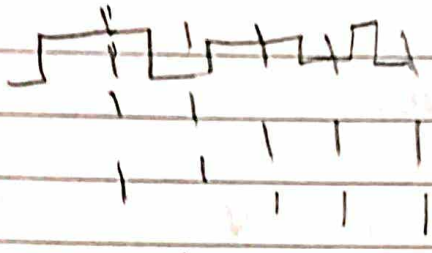
1 0001 ——⎍——

2 0011 ——⎍⎍——

16/2/23 Thursday

Diff b/w HTTP & HTTPS

1 | 0 | 1 | 1 | 0 | 1 | 0 | 1



machechester encoding acco'ding to IEEE standard

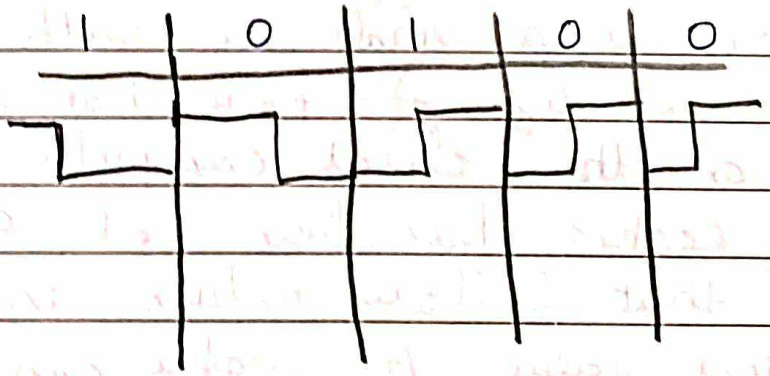
⎍ = 1

⎌ = 0

Differential manchester encoding :-

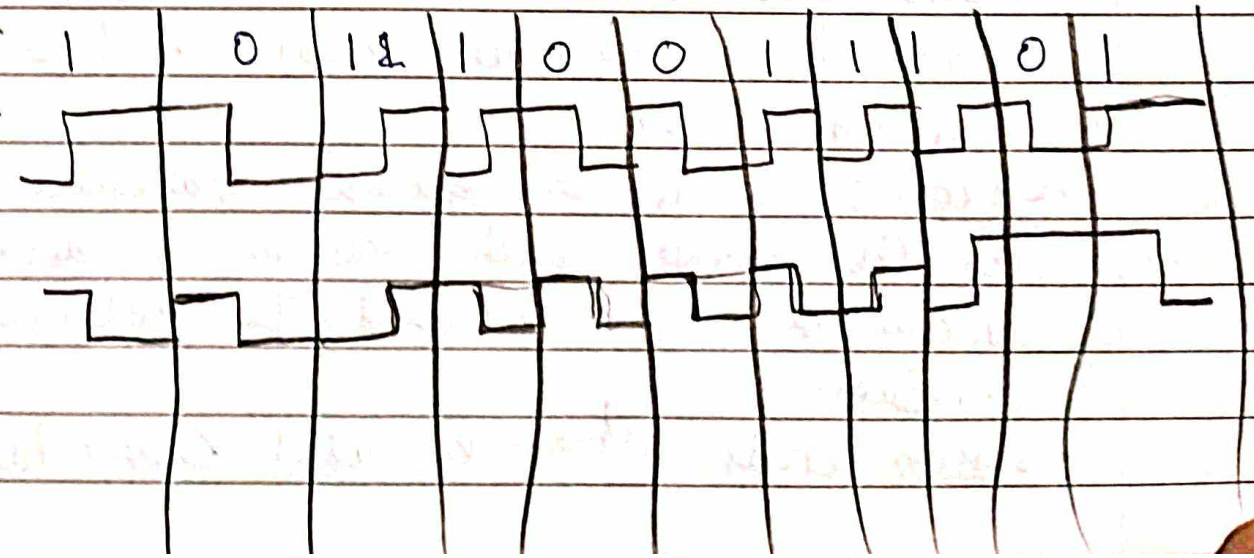
1) 0 ⇒ ⎍⎌ ⎌⎍

2) 1 ⇒ ⎌⎍ ⎍⎌



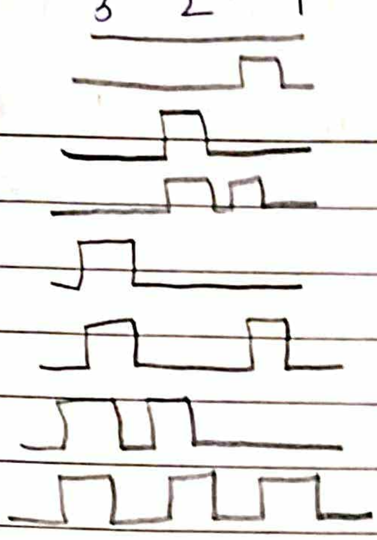
manchester →

differential



W3Techs.com

0	0	0	3 2 1
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



Representation of 3-bit PCM form in 1st bit

bit by bit Representation

20/2/23 Monday

Fingerprint web server

① ⇒ whatweb google.com 80 -V

gws - google web server

② ⇒ nc

nc 192.168.1.3 80

Head / HTTP/1.0

③ Head > Requesting transaction
Tool - HTTP print

④ W3Tech

⑤ builtwith.com

→ cookies ends depending on the lifetime you set for it.

* Diff b/w cookies and sessions

→ server side
Session - server
cookie - stored in browser
→ Personalization
→ client side

cookie : it is a small file with the max. size of 4KB that the web server on the client computer.

- once a cookie has been set, all page requests that follow return the cookie name and value. A cookie can only be read from domain that it has been issued from.

Session : It is a global variable stored on the server. Each session is assigned a unique id which is used to retrieve stored values.

- session ends when a user closes his browser.

5 well known webservers websites hosted