

Date: ...13/02/2024...

CSE421 - Computer Networks - SFD

Lecture 07 - TCP Math

TCP - Transport layer

↳ Port Address

↳ 20-60 bytes

basic rest 40 for options

Sequence Number - a random number as a start point 2000-2099 → 100 bytes of data

Acknowledgement Number - a number to expect next after receiving → 2100

Initial Sequence Number (ISN) will be given, suppose ISN = 2000

Control Bits / Control Flag

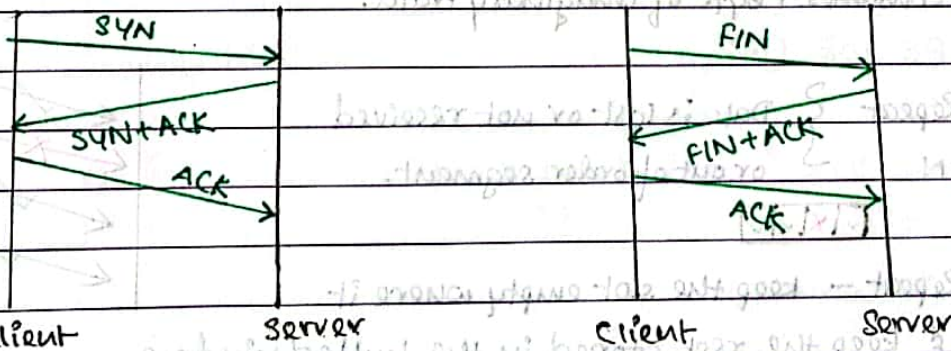
SYN (sync)

ACK (acknowledge)

FIN (finish)

To establish connection (3-way handshake)

Terminate



Connection Establishment

Termination

stores data

Window size (rwnd)

200 bytes

rwnd = 5000 bytes

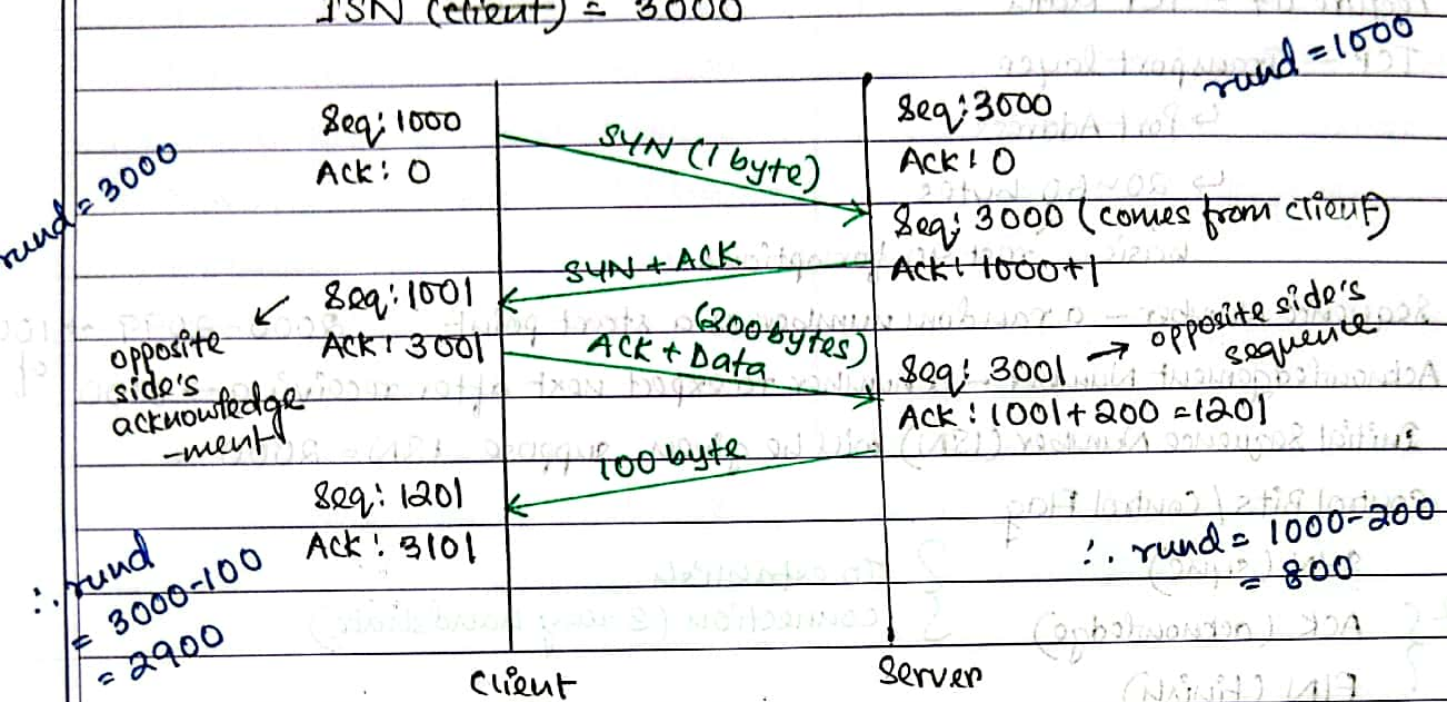
- 200 bytes

→ 4800 bytes

current window size

rwnd = 2000 bytes

Example - ISN (client) = 1000
 ISN (server) = 3000



Syn flag consumes 1 byte of imaginary data.

Selective Repeat } Data is lost or not received
 Go-Back-N } or out of order segment.

1 x 1 1 1



Selective Repeat - keep the slot empty where it was lost & keep the rest stored in the buffer/window.

Go-Back-N - discards the rest after point of loss.

Ex: Go-Back-N TCP connection

ISN (client) = 2045

ISN (server) = 8935

HTTP Req 1 \rightarrow 320 bytes

HTTP Req 2 \rightarrow 378 bytes

RWND (client) = 3020

RWND (server) = 6450

(*) Image does not show TCP part; starts w/ HTTP. So make sure to start w/ TCP.

client

server

S: 2045
A: 0

SYN

S: 8935
A: 0

S: 2046
A: 8936

SYN+ACK

S: 8935
A: 2046

ACK+320

S: 8936
A: 2366

S: 2366
A: 9888

DS1 (952)

DS2 (378)

S: 2366
A: 10266

DS3 (455)

S: 2366
A: 10266

(1)

(I) correction: client's

(II) RWND = 3020

(go-back-n)

SYN

(III) seq: 2045+1+320
+389 = 2755

client

(whatever you have
sent from client)

SYN

DS3
was not
received.

Ack: 8935+1+952
+378 = 10266

DS-4 & DS-5 also
not counted

(III) seq: 8935+1+952+378

+455 = 10721

server

(whatever server has
sent)

Ack: 2045+1+320+389
= 2755

(whatever client has sent)

CSE421 - Computer Networks - SFQ

Lecture 09 - Transport layer (TCP)

* Functions of TCP - (I) Stream delivery

(II) Segmenting + Reassembling

(III) Multiplexing

(IV) Full Duplex Service

(V) Identifying + tracking the segments of different appn

(VI) Connection oriented service

(VII) Reliable service

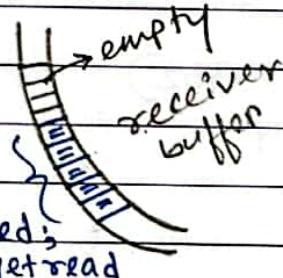
Stream Delivery - sends and receives data as a stream of bytes

Buffer - sending + receiving processes may not read/write at the same rate so TCP stores the data in a place (empty) before being processed; storage space → buffer



- (I) error control (lost + corrupted data); can resend
- (II) stored till receiver assures that it has received data; ACK (waiting)

(II) next in line for being sent in queue



- (I) received; not yet read
- (II) stop ed the data from sender's to which layer (appn) will it be sent from transport layer; data not yet pulled by Appn layer

Network layer - sends data in packets.

PDU - Segments (TCP groups a number of bytes) into a packet

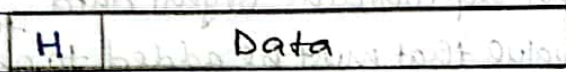
These segments are sent to network layer whose protocol is IP -

IP headers are then added to segments. [segment - any size]

TCP takes help of IP and sends its packets from one process/device to another.

- * Different segment — different routes → out of order
↳ Receiver end — it is reassembled
- * Full Duplex — Exchange of data betw two entities at the same time
- * Multiplexing — Sender
Demultiplexing — Receiver
- * Identifying + tracking the segments
↳ to differentiate seg/datagrams betw appⁿ — TCP uses port numbers.

④ TCP Segment Header —



20~60 bytes

Min^m Header size — 20 bytesExtra information/padding — 40 bytes → Max^m Header size — 60 bytesHeader — Source Port (16 bits) Destⁿ Port (16 bits)

Seq. Number (32 bits) Ack. Number (32 bits)

Header Length, HLEN (4 bits) → to tell the receiving process from where the data/seg. starts as header is of variable length. So we specify the Header length is HLEN.

Reserved bits (6 bits)

Header size = Multiple of 4

(TCP) = $4 \times (5 \sim 15) = 20 \sim 60$ bytes

HLEN → number of 4 byte words

Control Bit/Flags — URG

PSH → Push

RST → Reset

ACK

SYN

FIN

off
related
to seq.
+ ack
number

Urgent Pointer (16 bits)

window size (16 bits)

checksum (16 bits) → Mandatory field

Optional/Padding (40 bytes) → MSS (Max^m seg. size)PSH = 1 → segment is not stored + sent to the appⁿ directly (from S) (in buffer) → bypassing buffer

RST = 1 → close/Reset TCP connection

UDP Header
~ 8 bytes
(fixed)

20 bytes

Faster
transmission

56 bytes of Header \rightarrow 20 bytes base header information/size
 $56 - 20 = 36$ bytes \rightarrow header padding/options

Byte number - bytes of data being transferred in each connection are numbered by TCP $\rightarrow 0 \sim 2^{32}-1$

First byte - 1067 & 3000 data is sent (bytes)

Last byte - 4066

Urgent Pointer - only valid if URG = 1.
 (16)

- used when seq. contains urgent data

- holds a value that must be added to obtain the seq. number the number of the last urgent byte in the data section of the segment.

Seq - 1000

UP - 50

$1000 + 50 = 1050 \rightarrow$ byte number

MSS - largest block of data that a sender using TCP will send to a receiver.