417

- TCP

Transmission Control Protocol

reliable

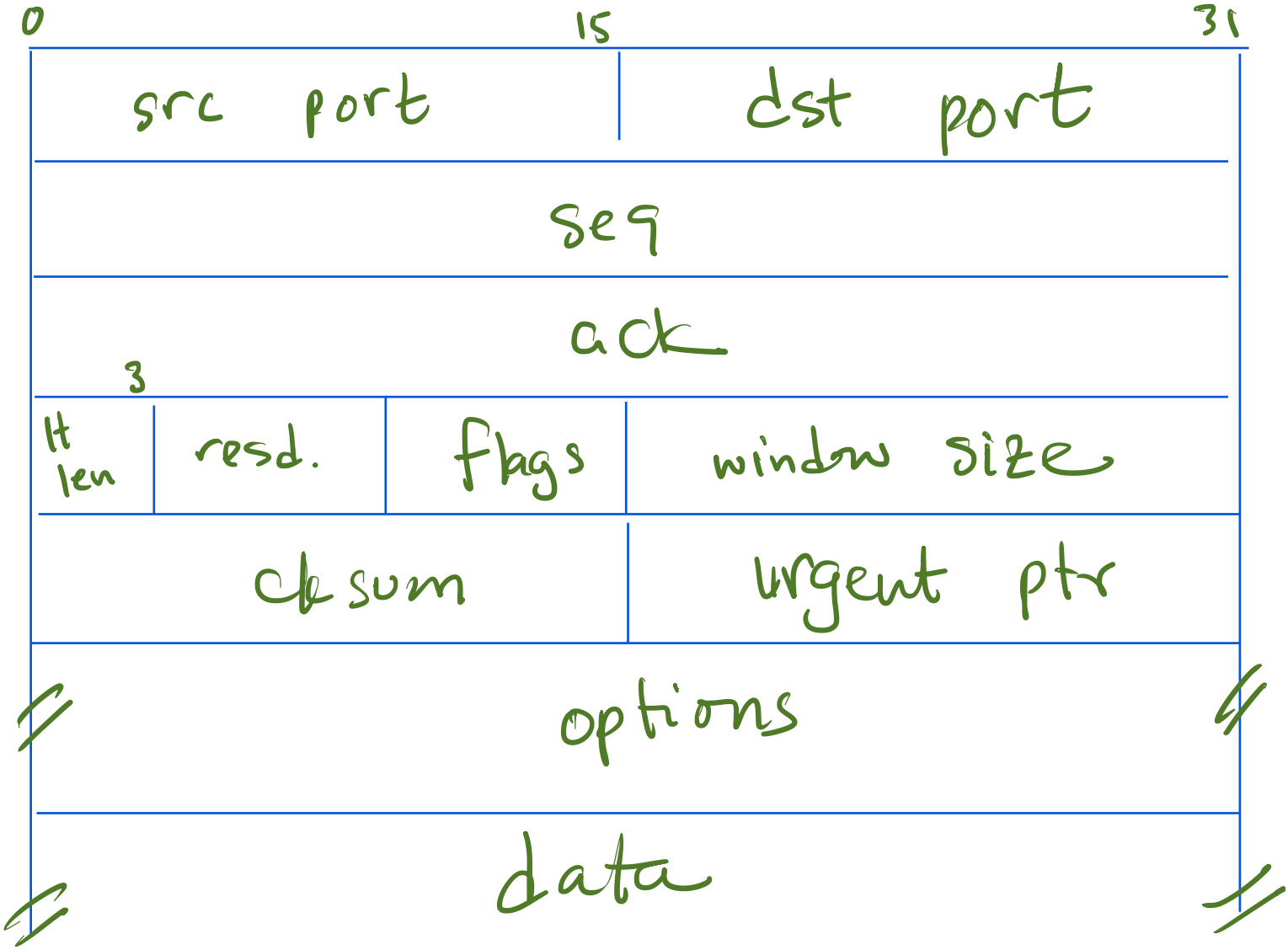
full-duplex

connection-oriented

byte-stream

service

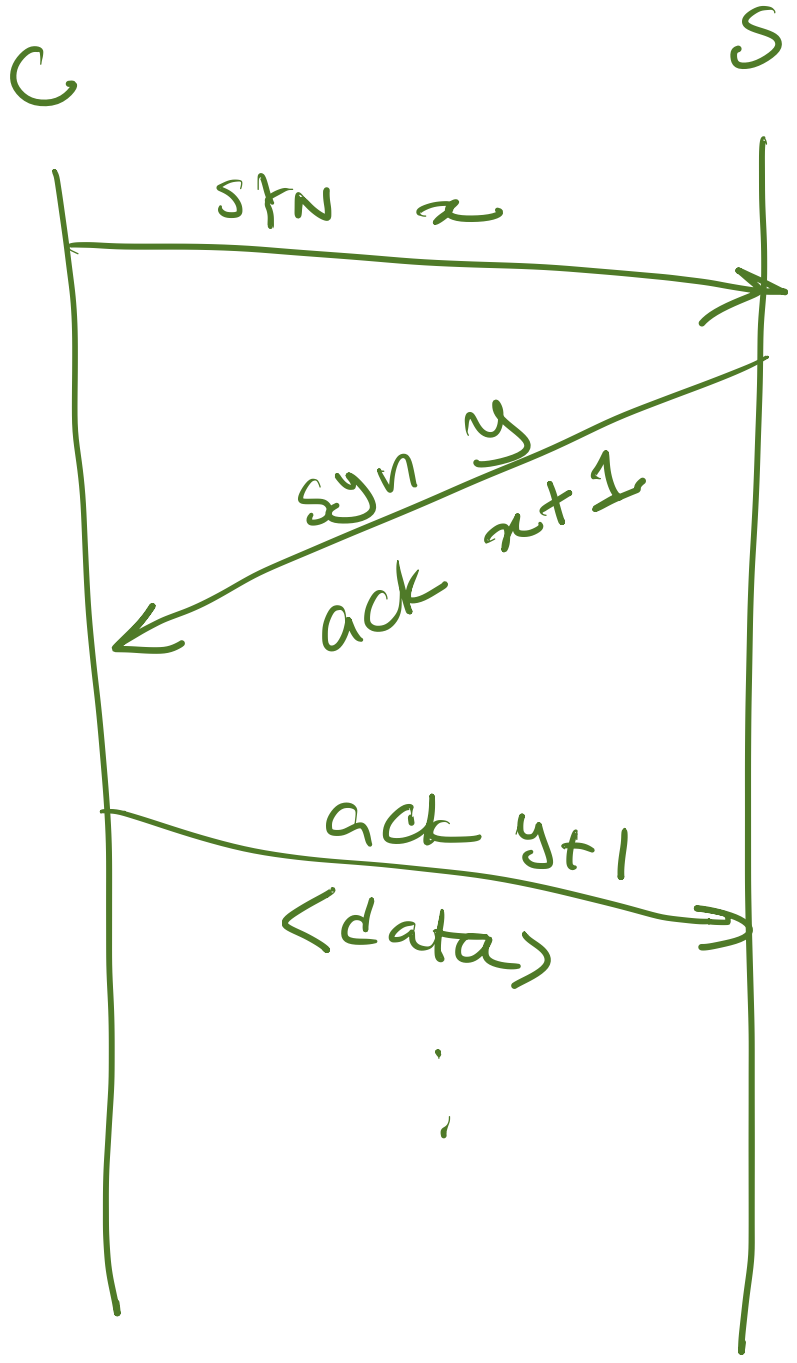
TCP Header

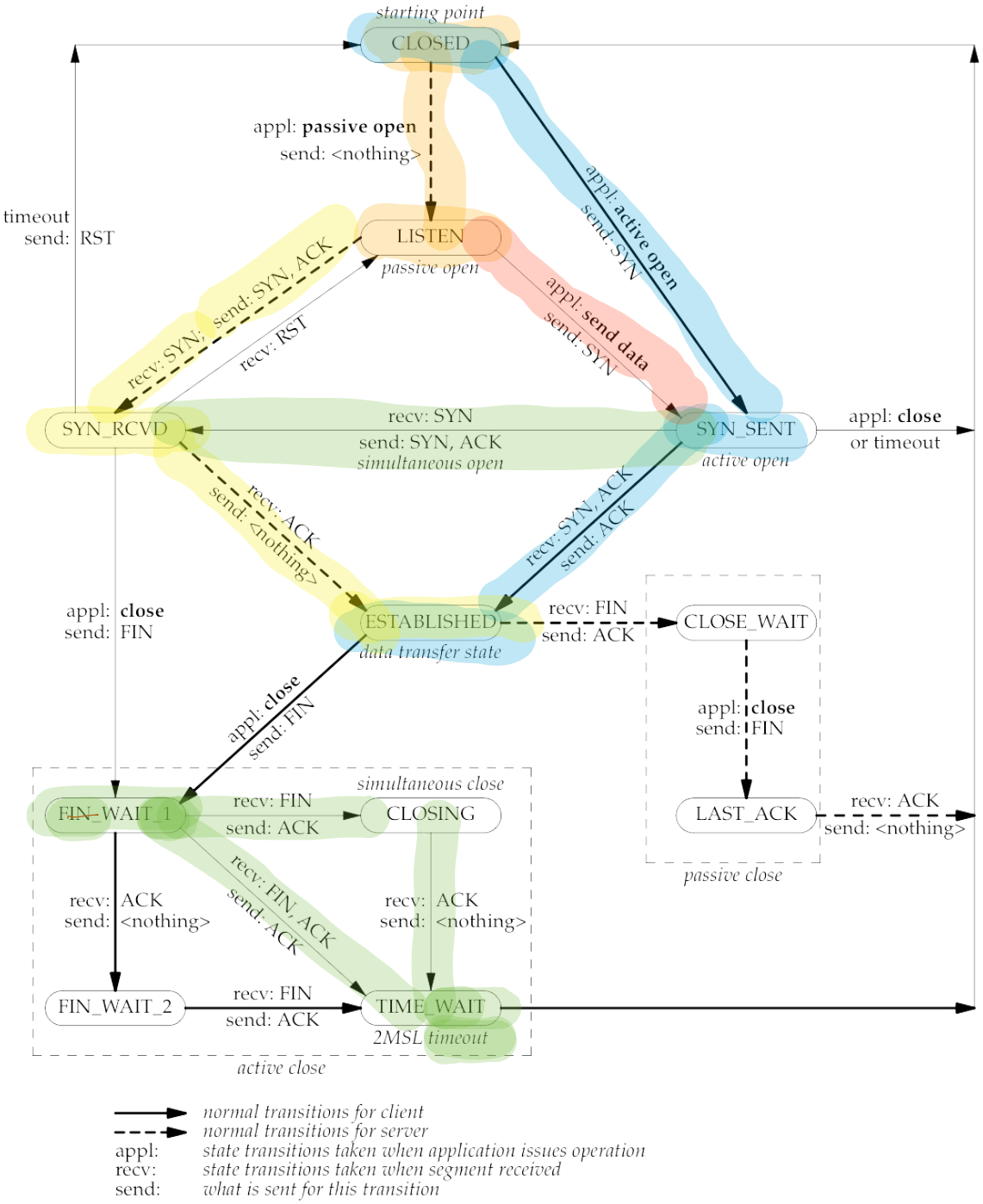


Flags

U	A	P	R	S	F
R	C	S	S	Y	I
G	K	H	T	N	N

TCP 3-way handshake (opening a conn.)





TCP state transition diagram.

example transitions

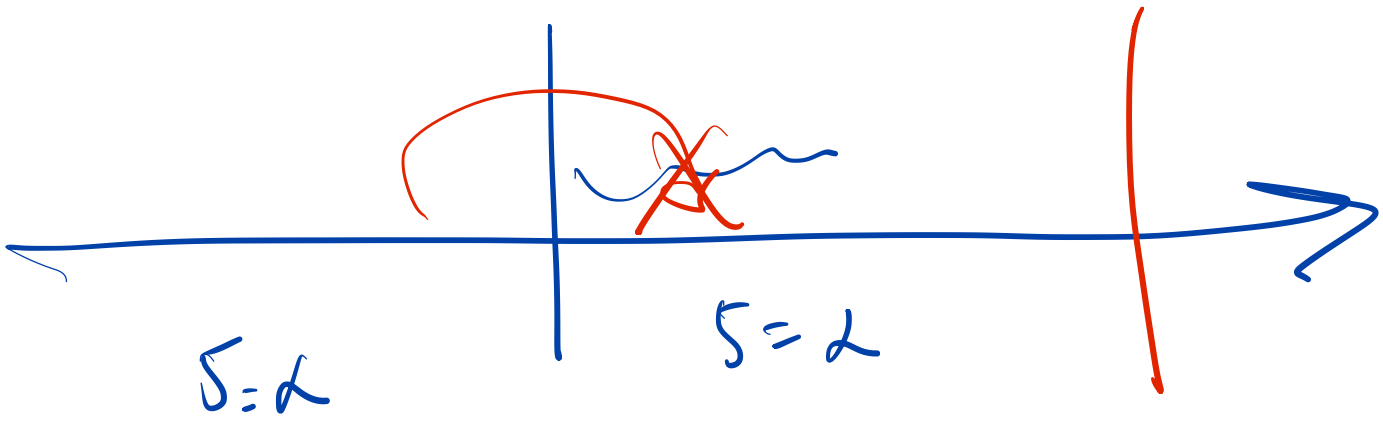
- simultaneous open
- time-wait
- connection close

(TCP) 5-tuple

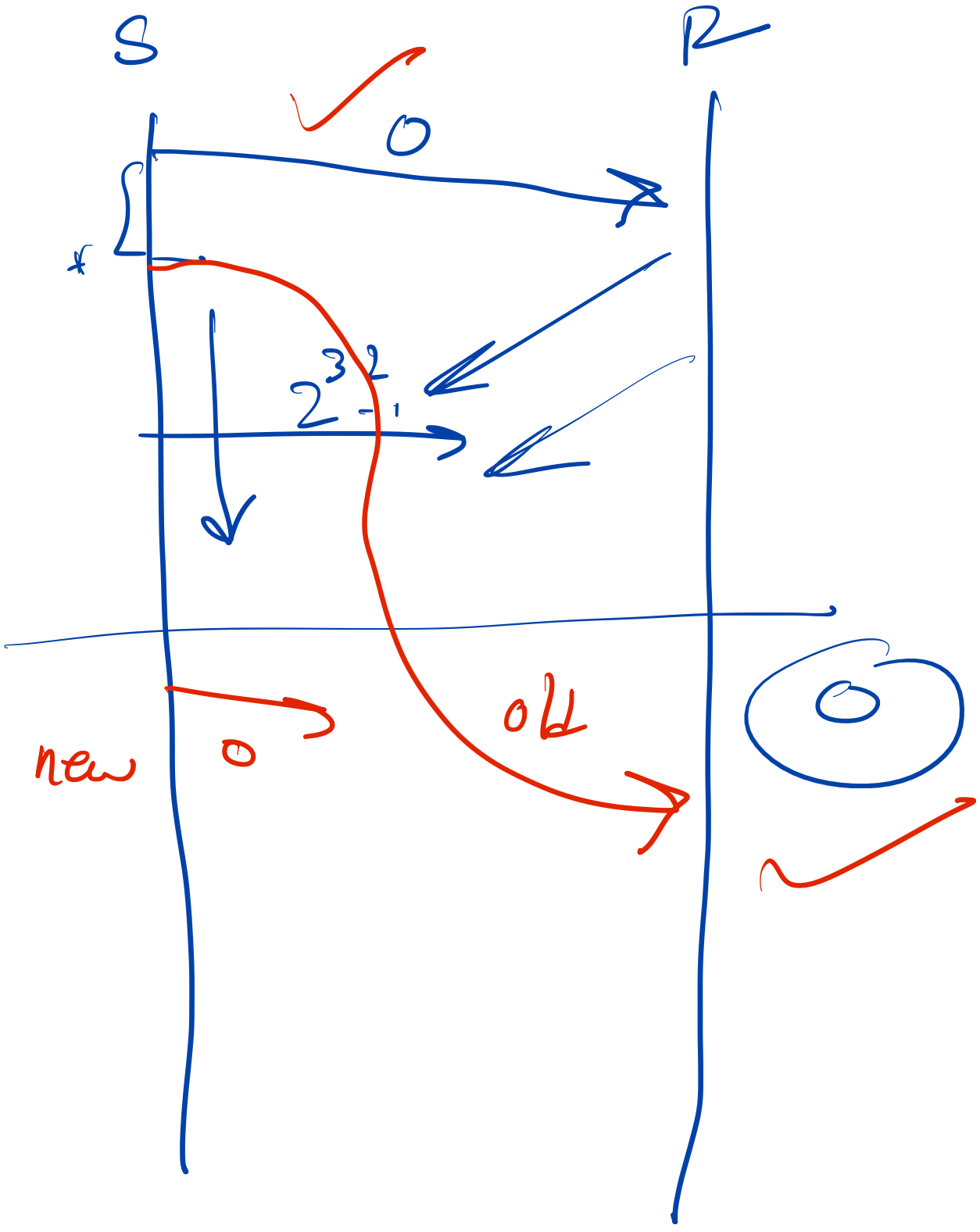
src IP addr , src port

dst IP addr , dst port

IP proto



example



TCP Flow Control

- advertised window

(what happens if adv. window is shut?)

Implicat^n's:

- SWS $< 2^{32}/2$

if NO reordering

- Seq # space should not wrap in 2 MSL

T1 (1.5 Mbps) : 6 hours

OC-48 (2.5 Gbps) : 14 seconds

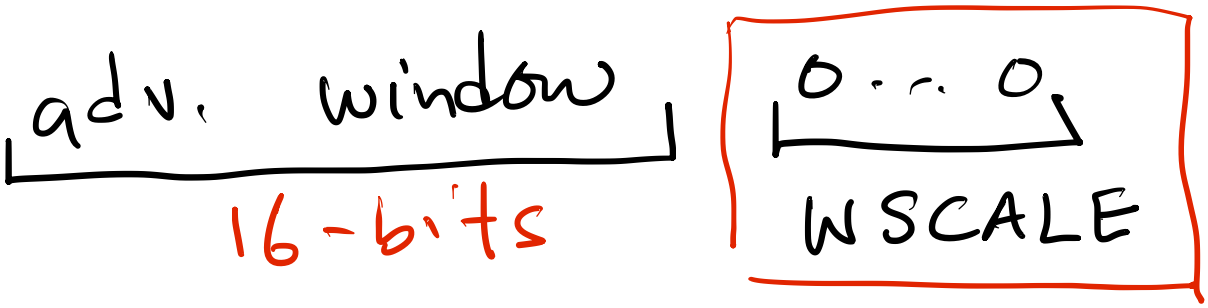
- 16 bit adv window

- worse!

PAWS :

Protection Against Wraparound Seq #s.

- window scaling



- changes 'unit' of adv. window

max WSCALE ?

TS (timestamp) option

- Each segment has a 32 bit ts

- TS must increase at least once per window

'virtual 64 bit seq space'

WSCALE = 14 TS example

time	bytes	sent seq#	ts	rcvd?
A	0 - 1G	0 - 1G	1	✓
B	1 - 2G	1 - 2G	2	ReX once
C	2G - 3G	2 - 3G	3	✓
D	3 - 4G	3 - 4G	4	✓
E	4G - 5G	0 - 1G	5	✓
F	5 - 6G	1 - 2G	6	?

ReX of B : 1 - 2G, ts = 2

F : 1 - 2G, ts = 6

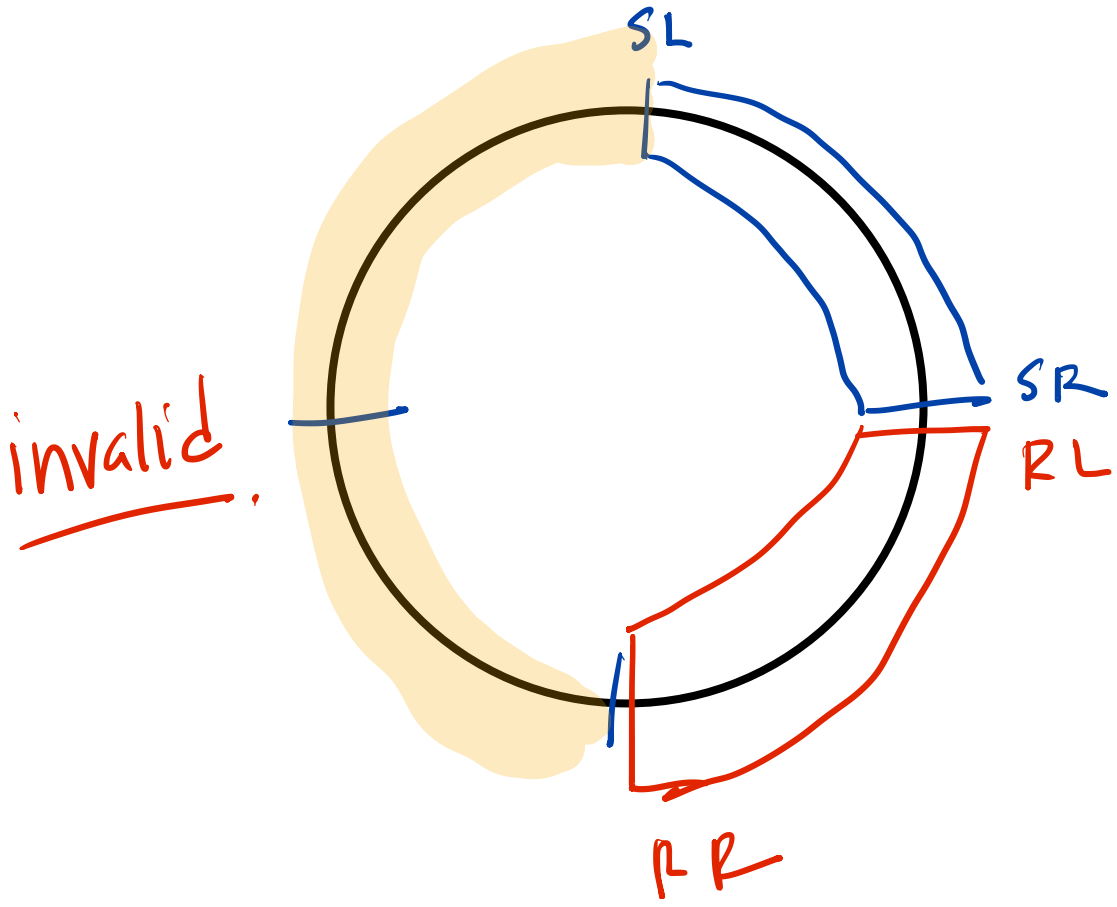
seq ts

TCP Sender Algorithms

- silly window syndrome
- Nagle's Alg
- RFC 791 ReX

TS/ wscale .

RFC 7323 p10 Sec. 5.5



TCP sender

- receiver advertises small window
- sender aggressively xmits

→ stop & go

"silly window syndrome"

after advertising 0 window rcevr must advertise at least 1 MSS

Nagle's Alg (self-clocking)

when to xmit?

if app. data &
adv. window ≥ MSS

→ send full segment

else if unacked data in flight

→ buffer until ack arrives

else

send now

TCP_NODELAY

When to ReX?

RFC 791

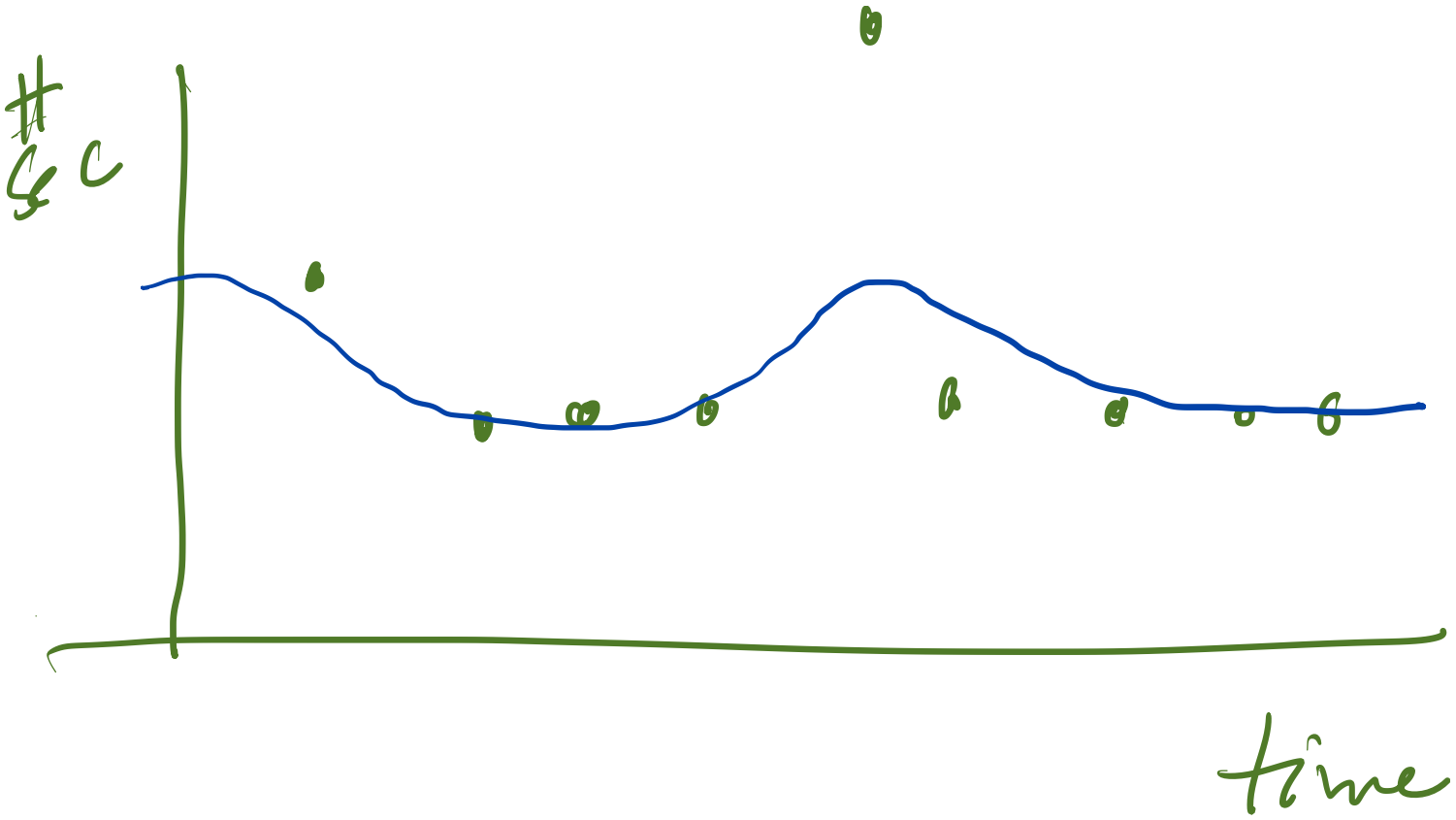
- Keep running avg. of RTT
- Compute timeout based on avg. RTT

$$\text{Est RTT} \leftarrow \alpha * \text{Est RTT} + (1-\alpha) * \text{sample RTT}$$

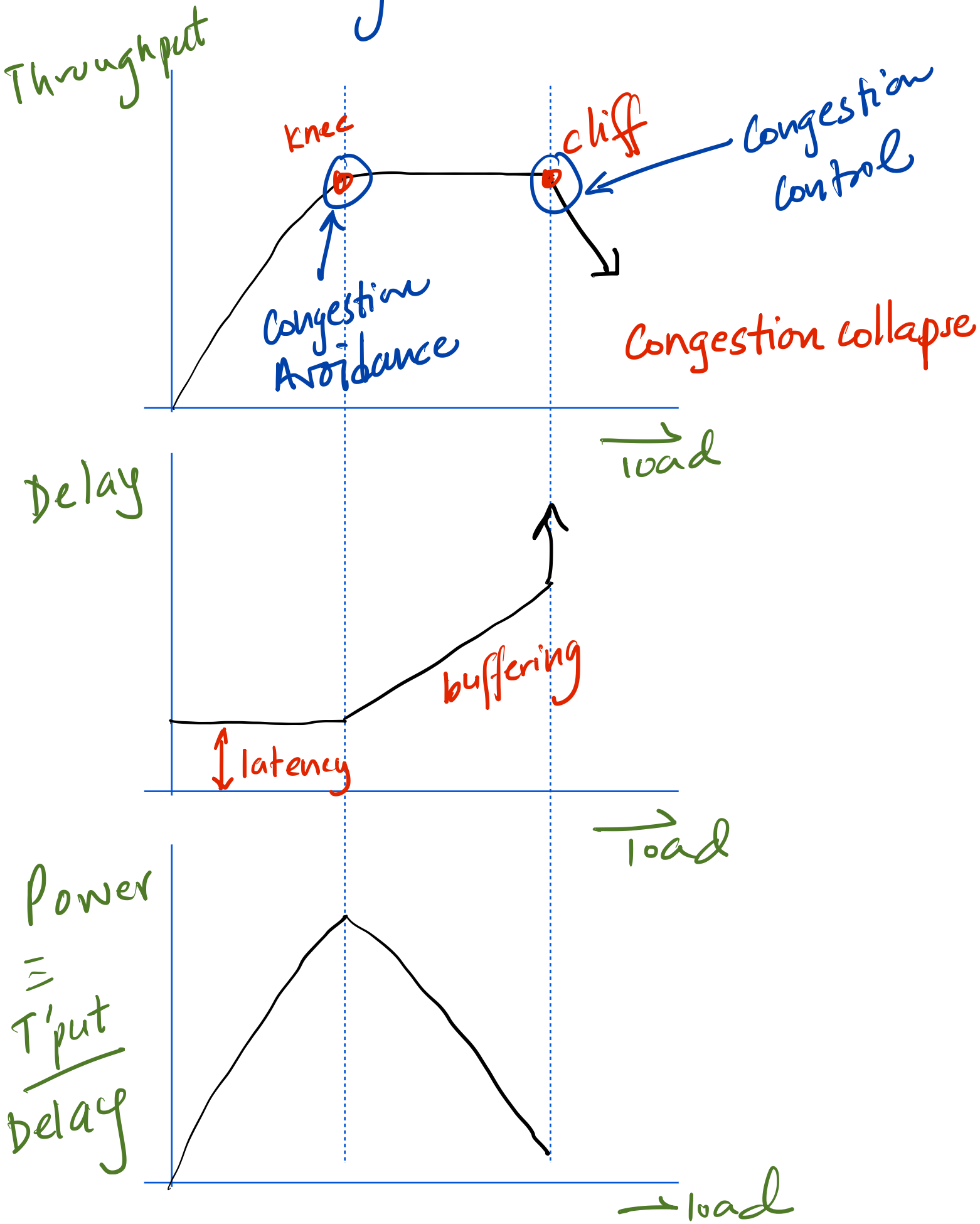
$$\text{Timeout} \leftarrow 2 * \text{Est RTT}$$

$$\alpha \in (0.8, 0.9)$$

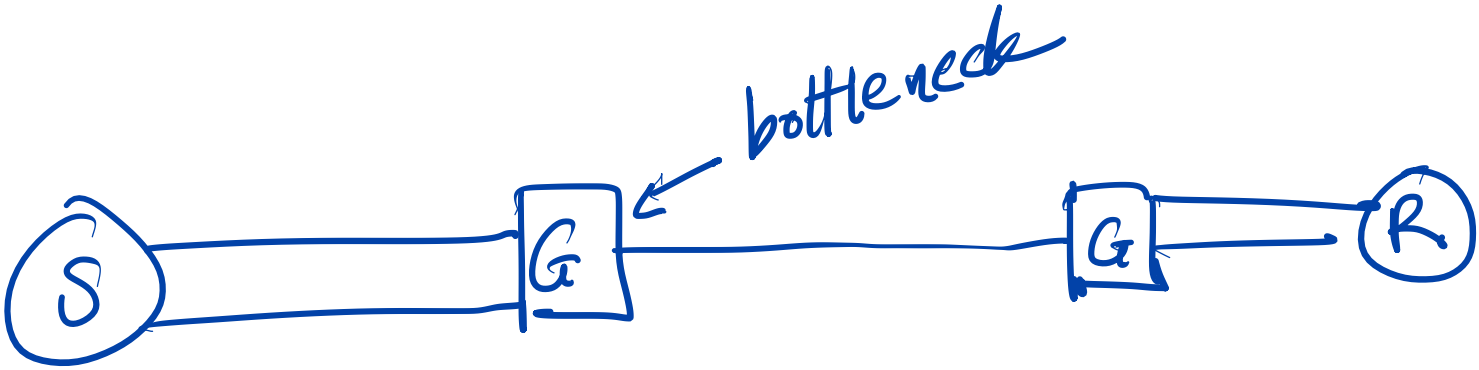
Low pass filter



Congestion Control



TCP Classic

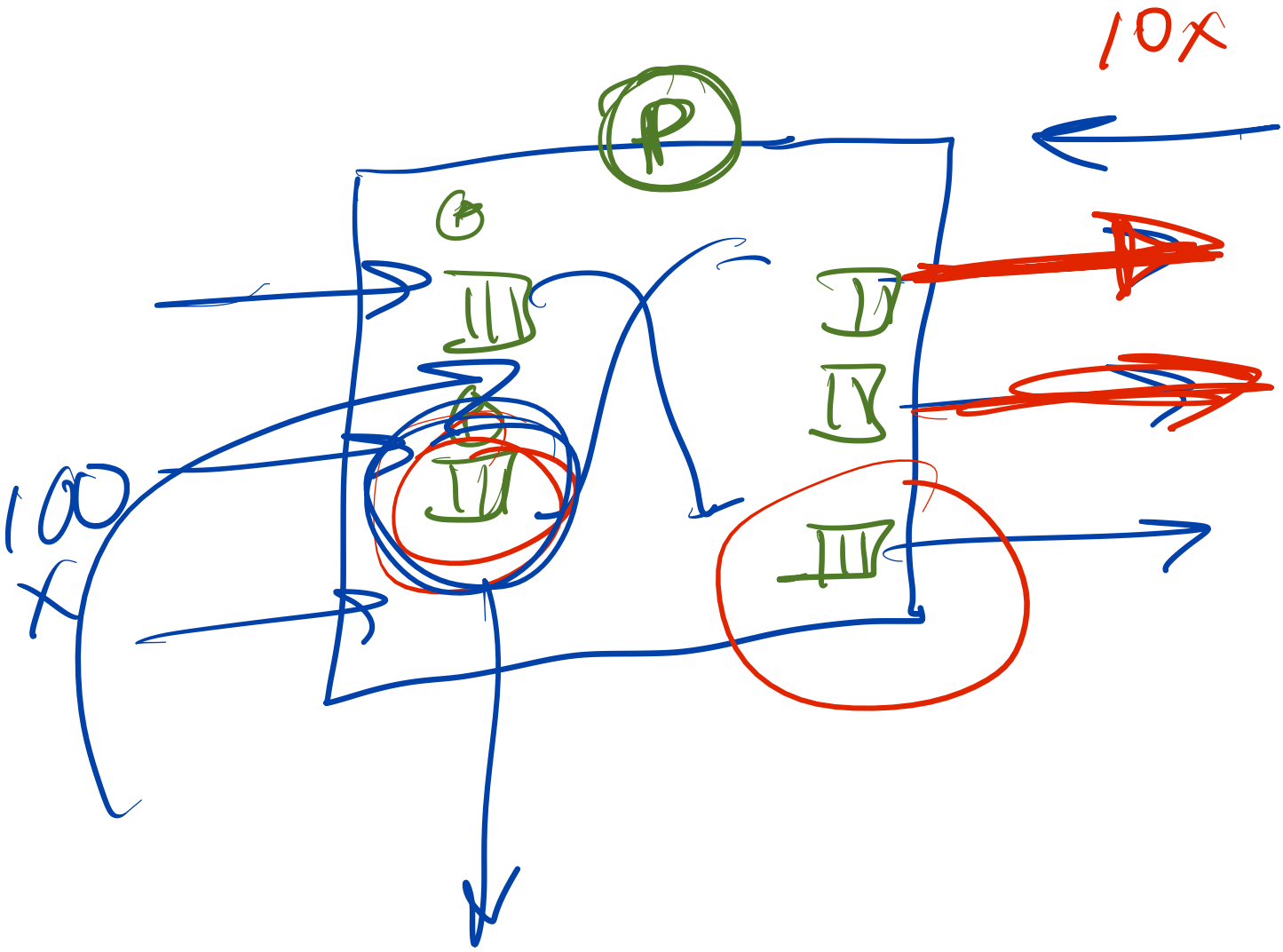


- Receiver advertises full window

- Sender sends window amount of data

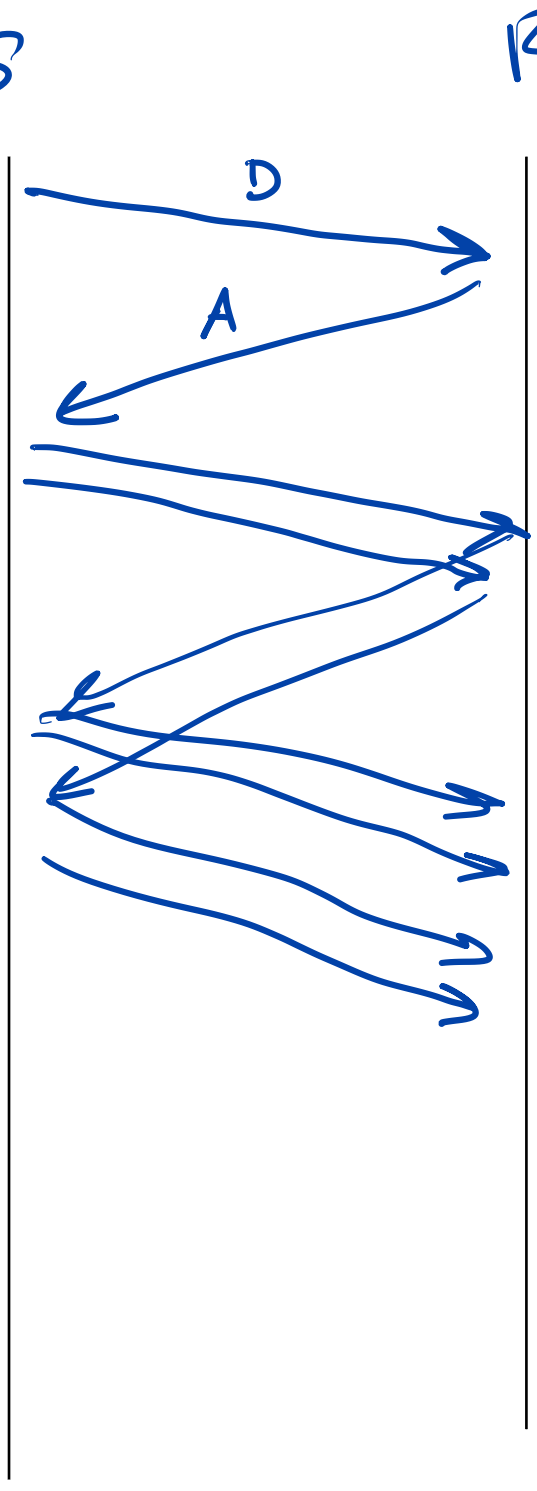
→ Congestion collapse!

Switch Schematic



VJCC

- self clocking
- Starting the ACK Clock



"Slow Start"

new variables in TCP sender

cwnd : congestion window

ss_thresh : slow-start threshold

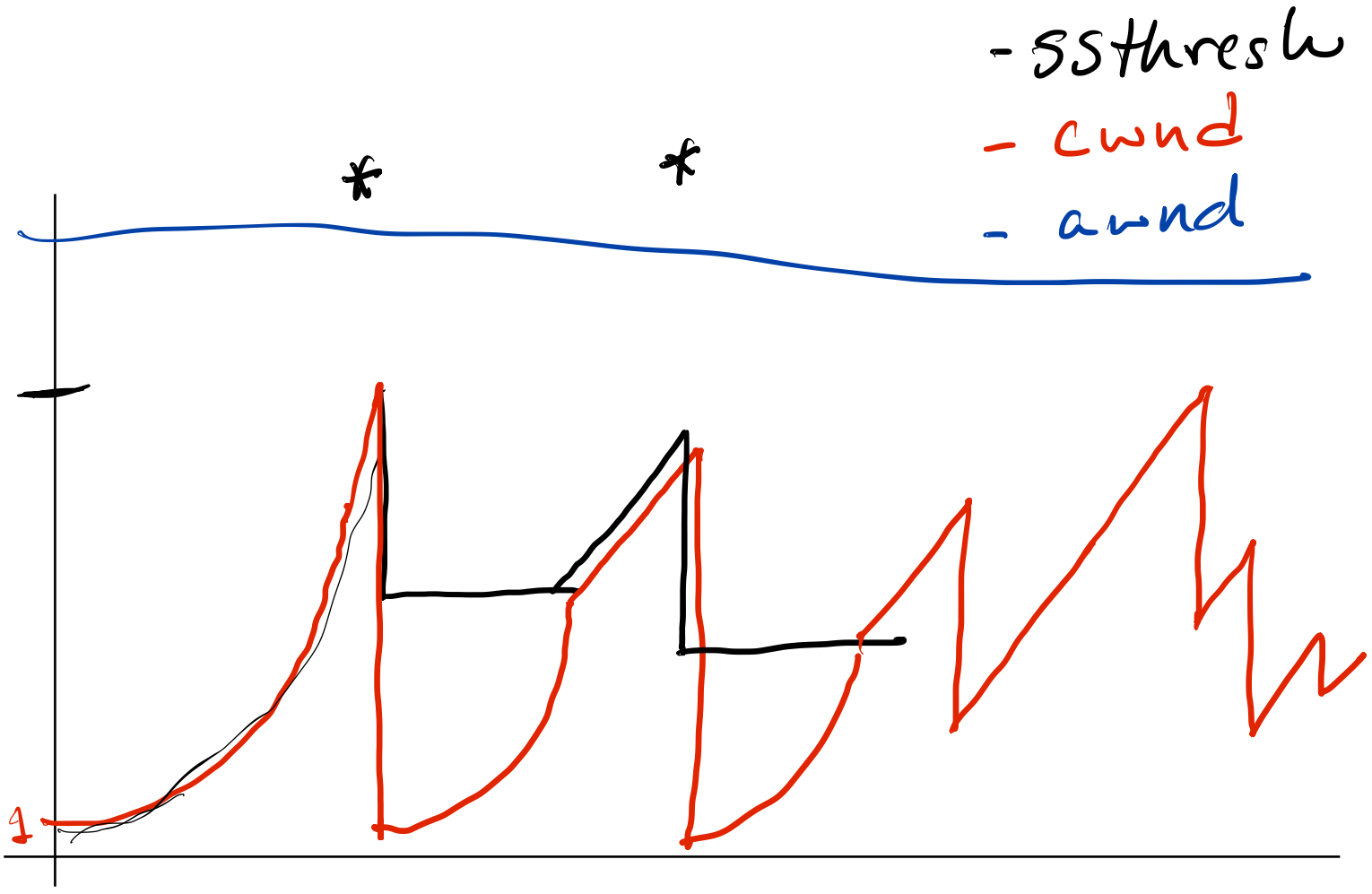
send min (cwnd, adv-window)

cwnd: CC

adv-window: FC

Congestion Avoidance

AIMD : Arithmetic Increase
Multiplicative Decrease



TCP Vegas

Fast ReX

Fast Recovery

RFC 2001