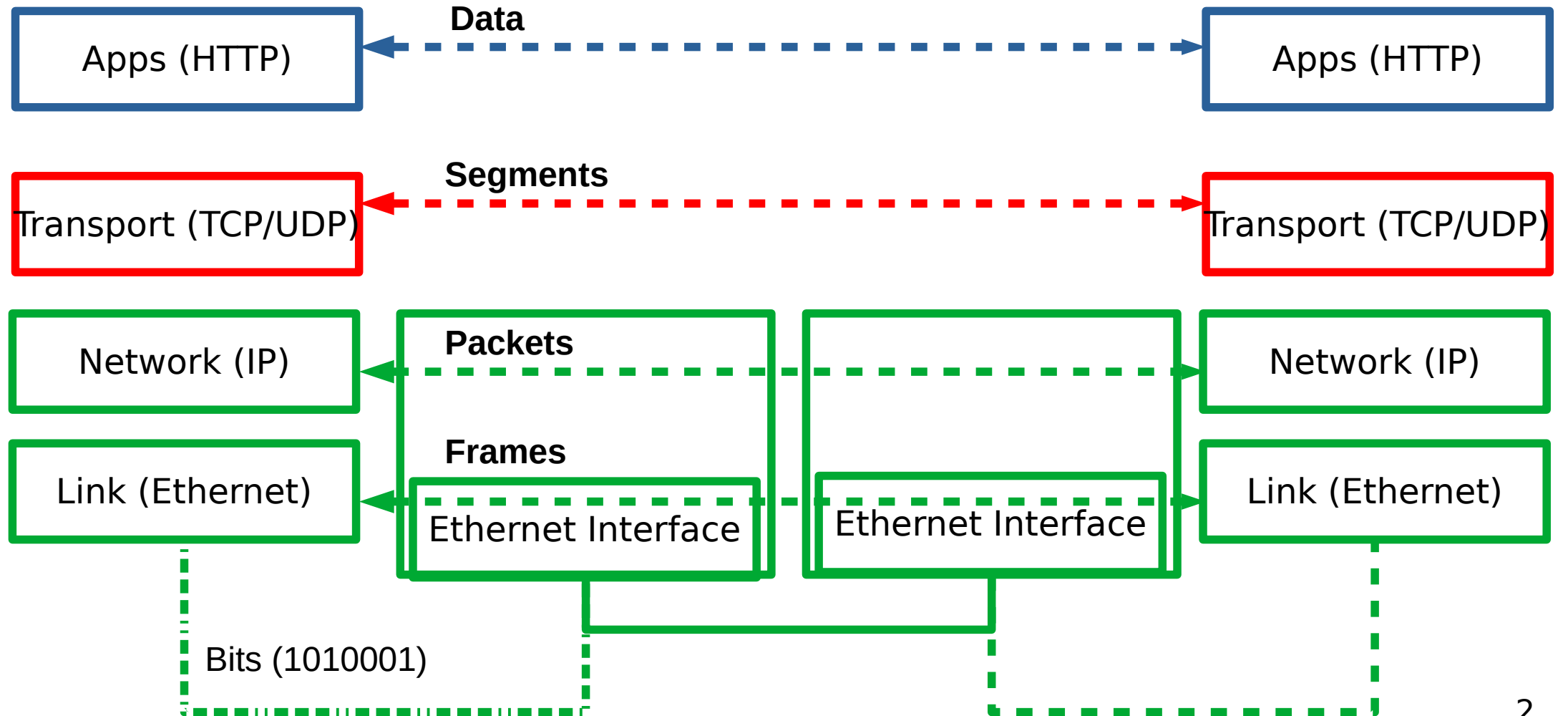


CSC4200/5200 – COMPUTER NETWORKING

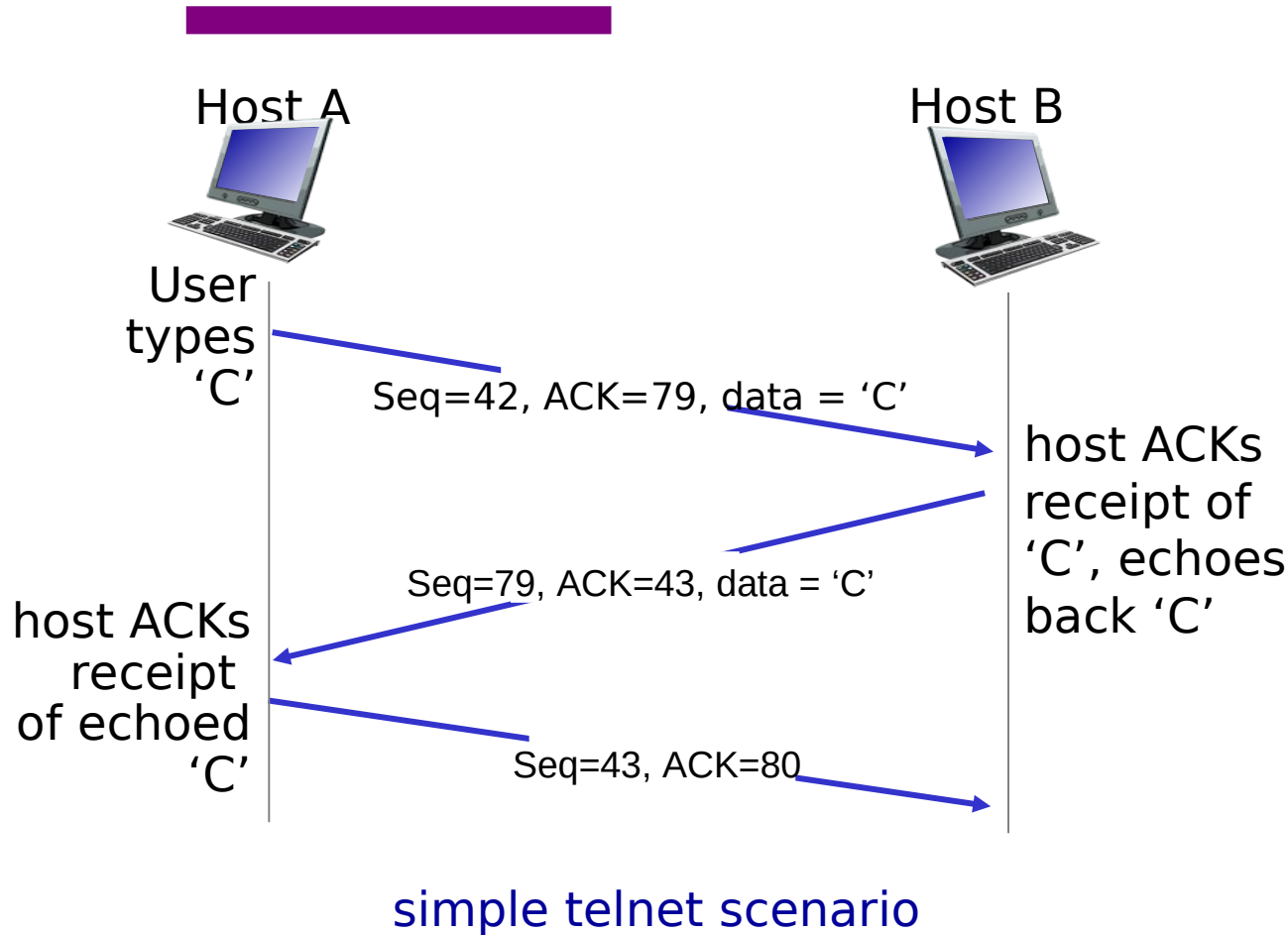
Instructor: Susmit Shannigrahi

CONGESTION CONTROL
sshannigrahi@tnitech.edu

GTA: dereddick42@students.tnitech.edu



TCP seq. numbers, ISNs

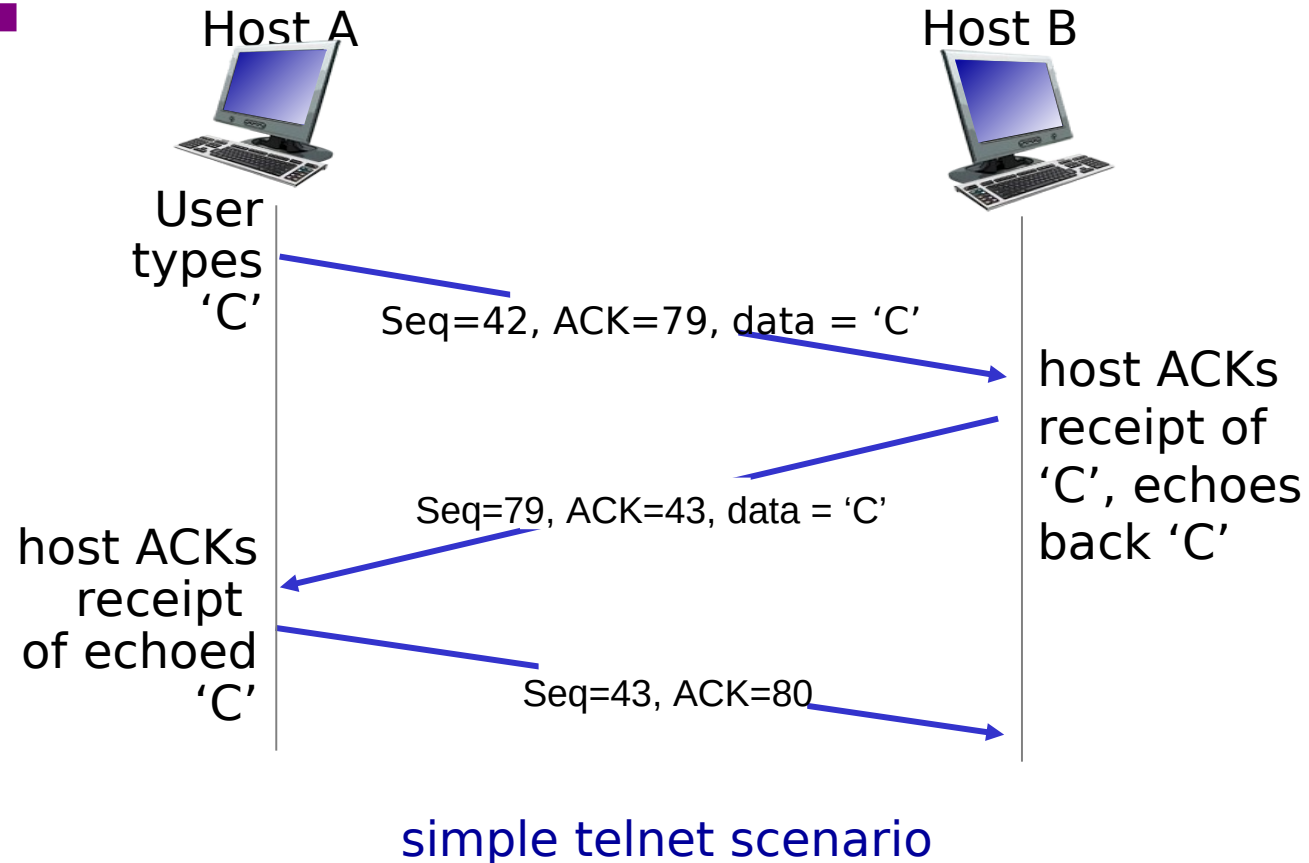


Sequence number for the first byte

Why not use 0 all the time?

- Security
- Port are reused, you might end up using someone else's previous connection
- Phone number analogy
- TCP ISNs are clock based
 - 32 bits, increments in 4 microseconds
 - 4.55 hours wrap around time

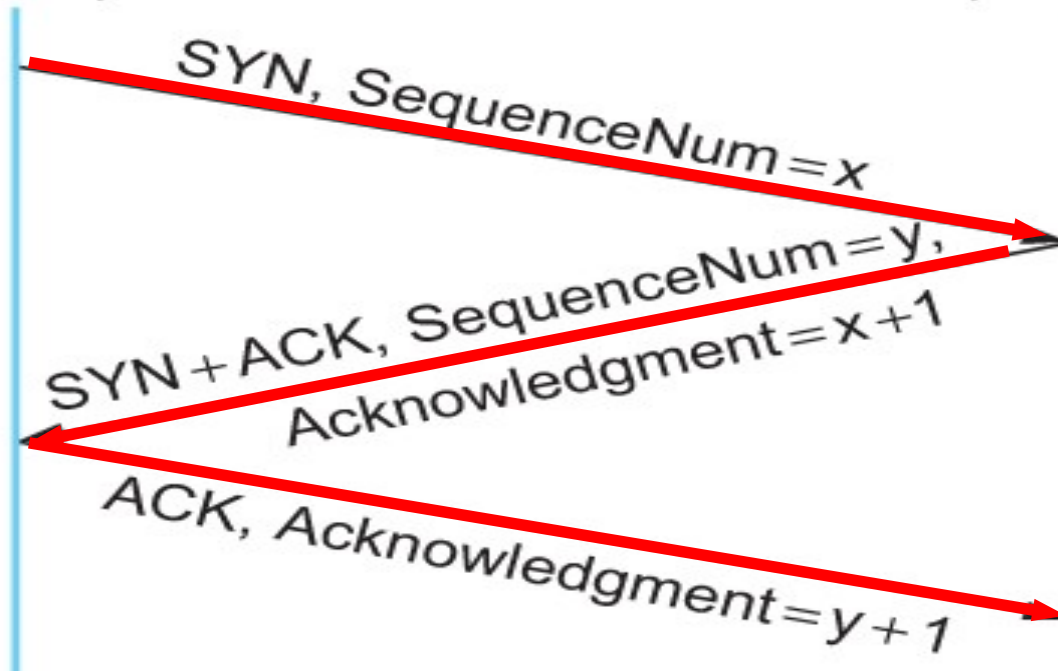
TCP seq. numbers, ACKs



TCP Three-way Handshake

Active participant
(client)

Passive participant
(server)



Timeline for three-way handshake algorithm

The idea is to tell each other
The ISNs

SYN → Client tells server that
it wants to open a connection,
Client's ISN = x

SYN+ ACK → Server tells
Client → Okay → Server's ISN
= y, ACK = CLSeq + 1

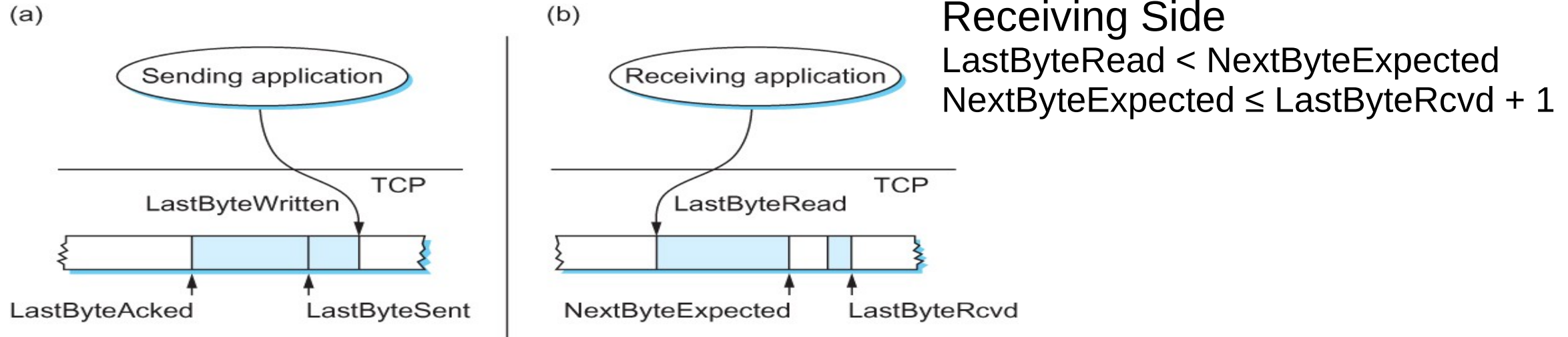
Why increment by 1?

Sliding Window Revisited

Sending Side

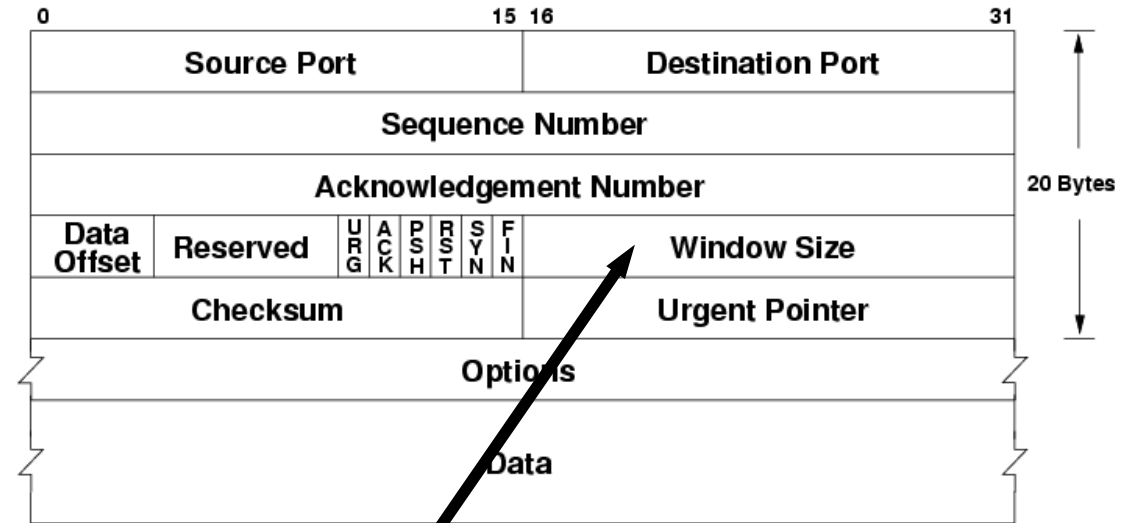
$\text{LastByteAcked} \leq \text{LastByteSent}$

$\text{LastByteSent} \leq \text{LastByteWritten}$



TCP flow control

- receiver “advertises” free buffer space in the header
- sender limits amount of unacked (“in-flight”) data to receiver’s **rwnd** value
- guarantees receive buffer will not overflow

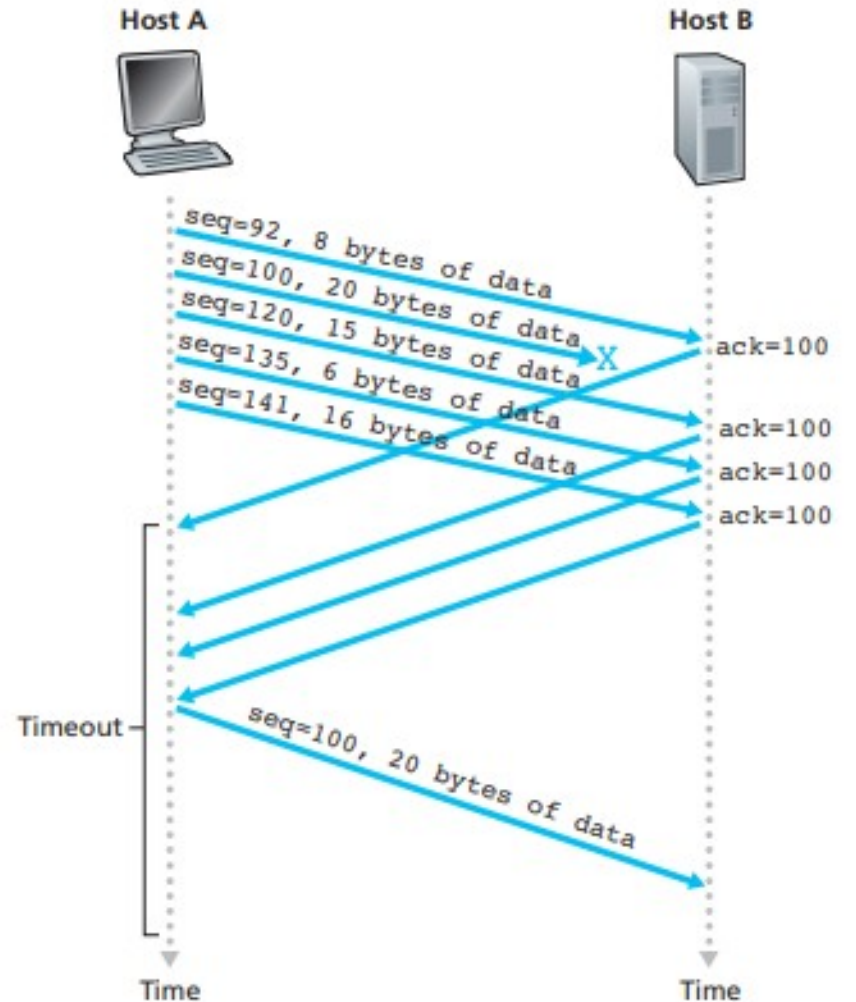


TCP Fast Retransmission

Timeouts are wasteful

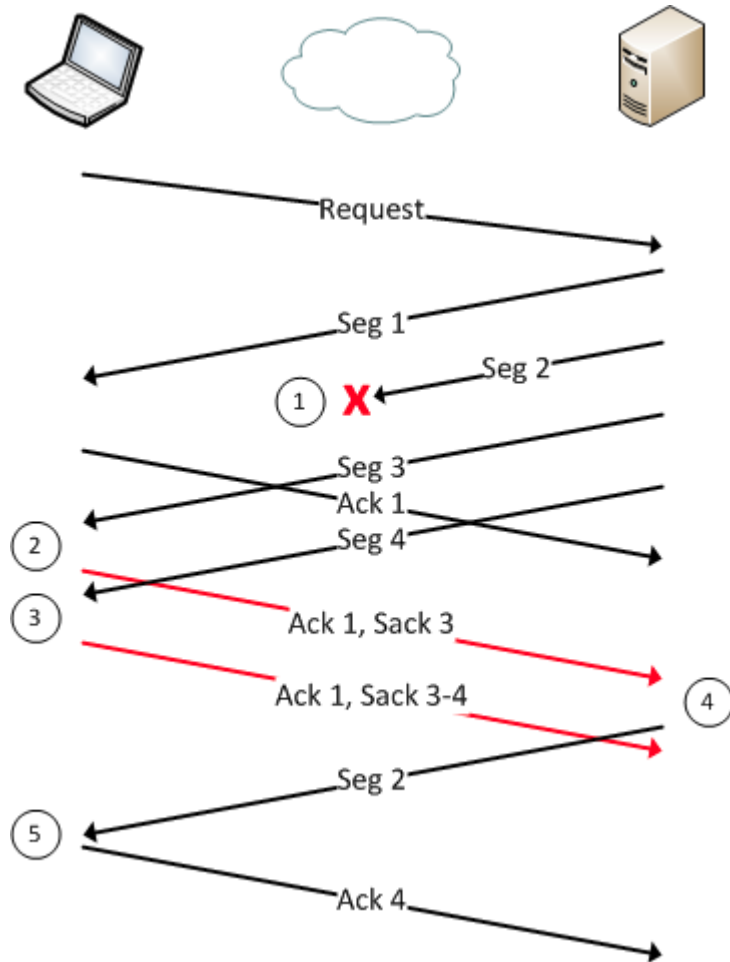
Triple duplicate ACKs

Retransmits before timeout



TCP Fast Retransmission - SACK

What if multiple segments are lost?



Very good explanation:

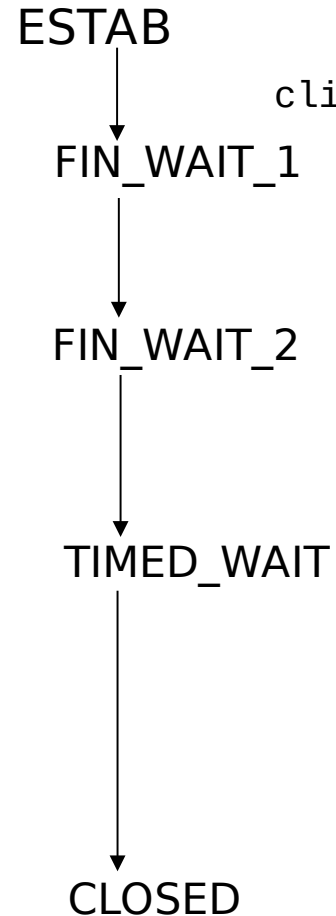
<https://packetlife.net/blog/2010/jun/17/tcp-selective-acknowledgments-sack/>

TCP: closing a connection

- client, server each close their side of connection
 - send TCP segment with FIN bit = 1
- respond to received FIN with ACK
 - on receiving FIN, ACK can be combined with own FIN
- simultaneous FIN exchanges can be handled

TCP: closing a connection

client state



`clientSocket.close()`

can no longer
send but can
receive data

wait for server
close

timed wait
for $2 \times \text{max}$
segment lifetime



FINbit=1, seq=x

ACKbit=1; ACKnum=x+1

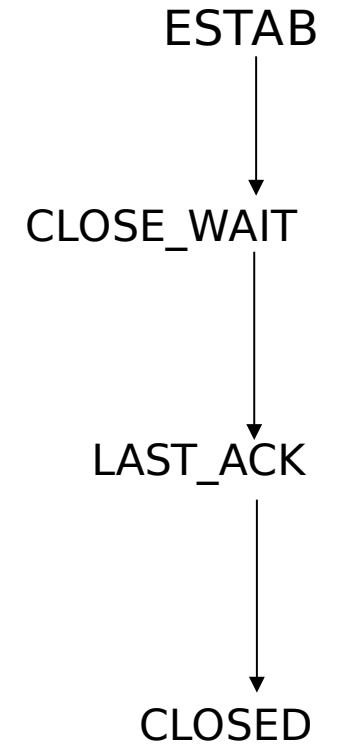
FINbit=1, seq=y

ACKbit=1; ACKnum=y+1

can still
send data

can no longer
send data

server state



Why do we need ack for closing?

- Data in-flight

Congestion Control



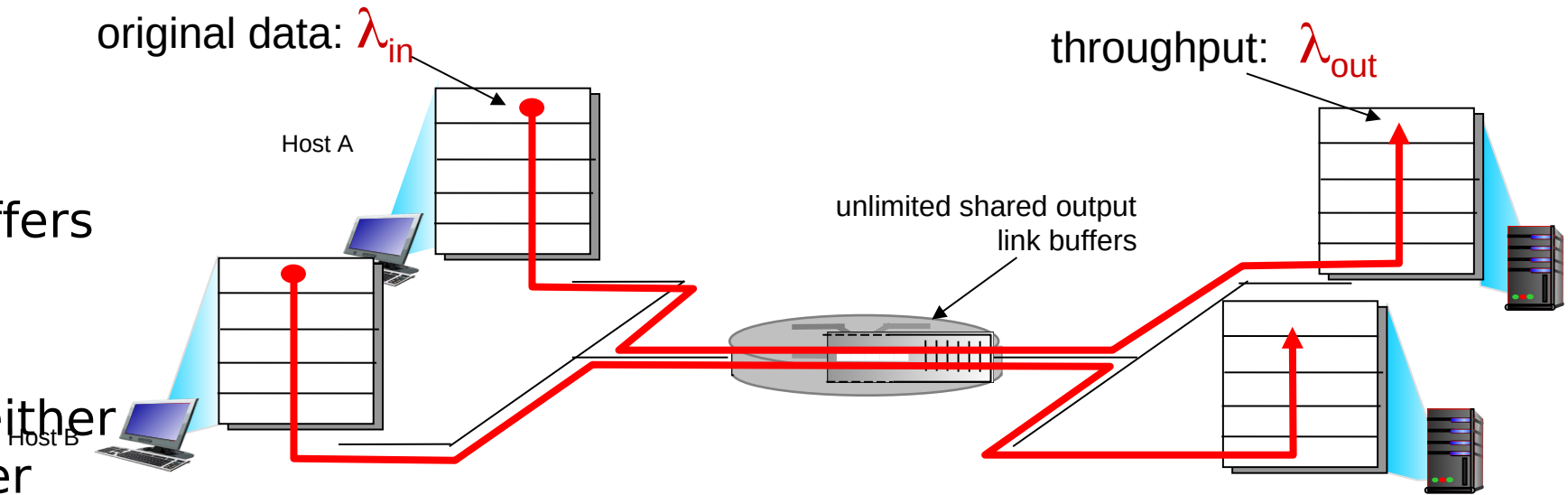
Principles of congestion control

congestion:

- informally: “too many sources sending too much data too fast for *network* to handle”
- different from flow control!
- manifestations:
 - lost packets (buffer overflow at routers)
 - long delays (queueing in router buffers)
- a top-10 problem!

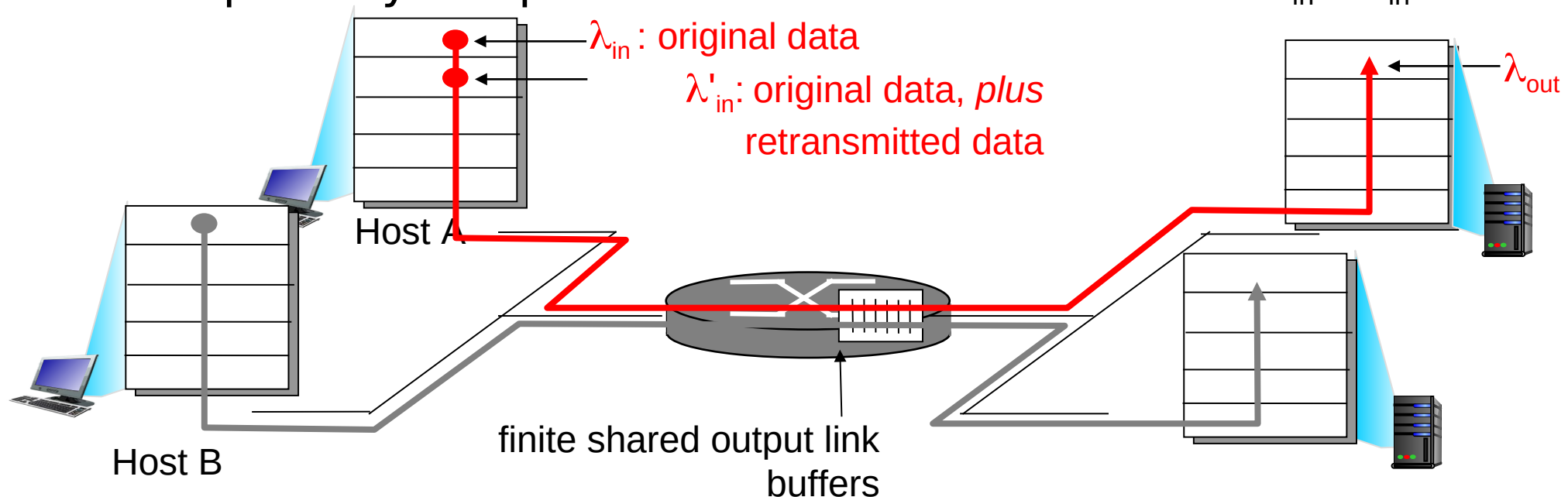
Congestion: scenario 1

- three senders, two receivers
- one router, infinite buffers
- output link capacity: R
- The router can only transmit one –... and either buffer or drop the other
- If many packets arrive,
- Buffer overflow



Causes/costs of congestion: scenario 2

- one router, *finite* buffers
- sender retransmission of timed-out packet
 - application-layer input = application-layer output: $\lambda_{in} = \lambda_{out}$
 - transport-layer input includes *retransmissions* : $\lambda_{in} > \lambda_{out}$



Metrics: Throughput vs Delay

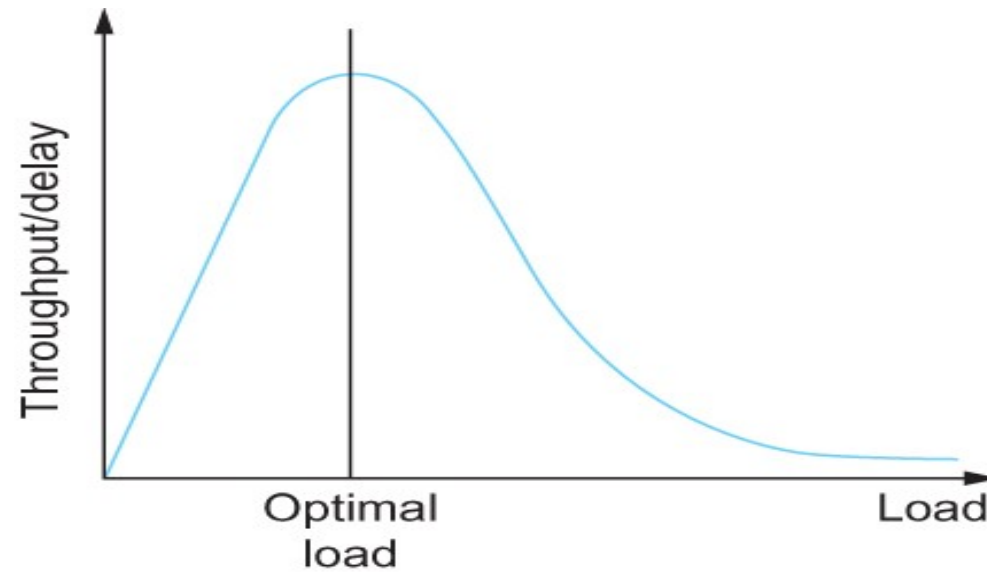
High throughput –

- Throughput: measured performance of a system –E.g., number of bits/second of data that get through
- Low delay –
- Delay: time required to deliver a packet or message –E.g., number of ms to deliver a packet •
- These two metrics are sometimes at odds –
 - More packets = more queuing

Issues in Resource Allocation

- Evaluation Criteria
 - Effective Resource Allocation

power of the network.
 $\text{Power} = \text{Throughput}/\text{Delay}$



Ratio of throughput to delay as a function of load

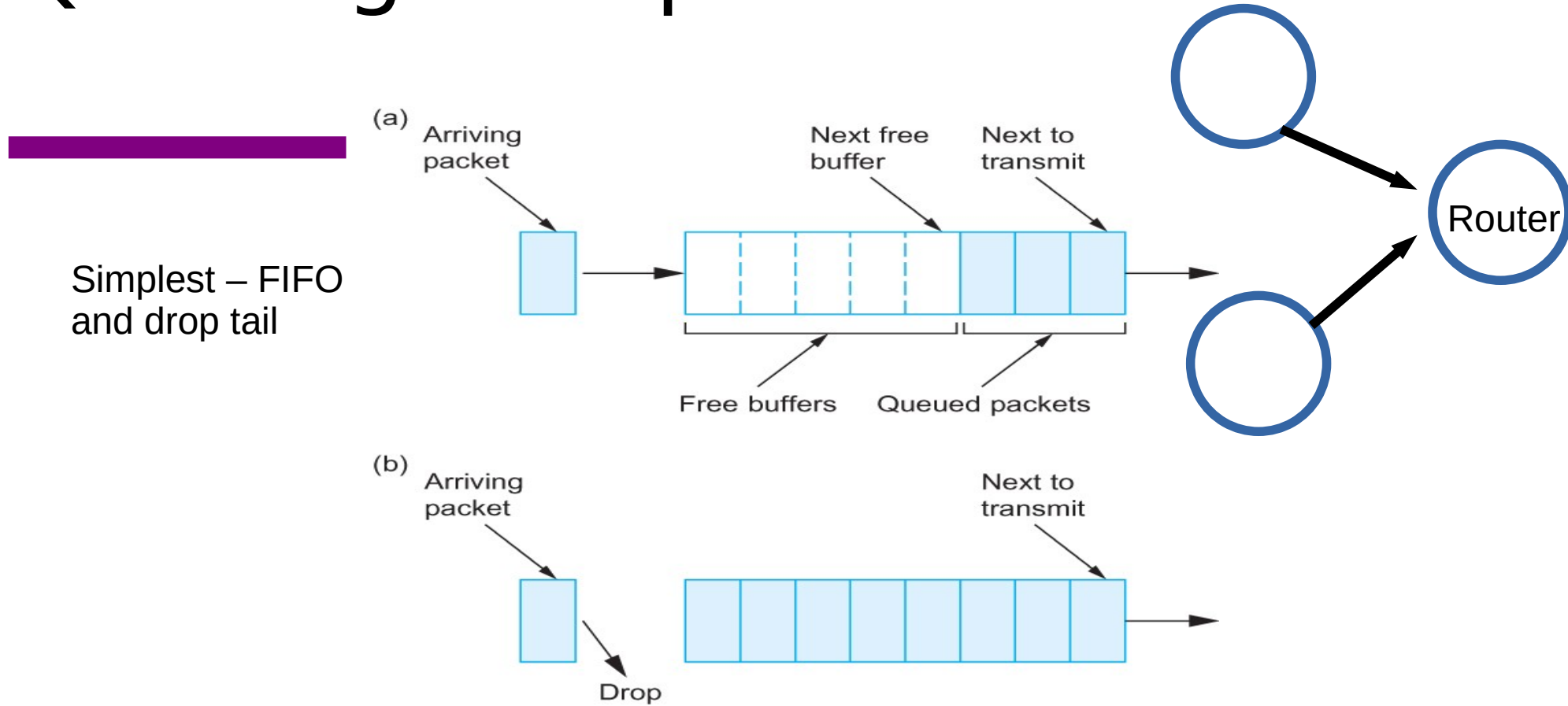
Issues in Resource Allocation

- Evaluation Criteria
 - Fair Resource Allocation
 - The effective utilization of network resources is not the only criterion for judging a resource allocation scheme.
 - We want to be “fair”
 - Equal share of bandwidth

But, what if the flows traverse different paths?

Open problem, often determined by economics

Queuing Disciplines



(a) FIFO queuing; (b) tail drop at a FIFO queue.

What are the problems?

Defining Fairness: Flows

“fair” to whom? – Should be Fair to a Flow

What is a flow?

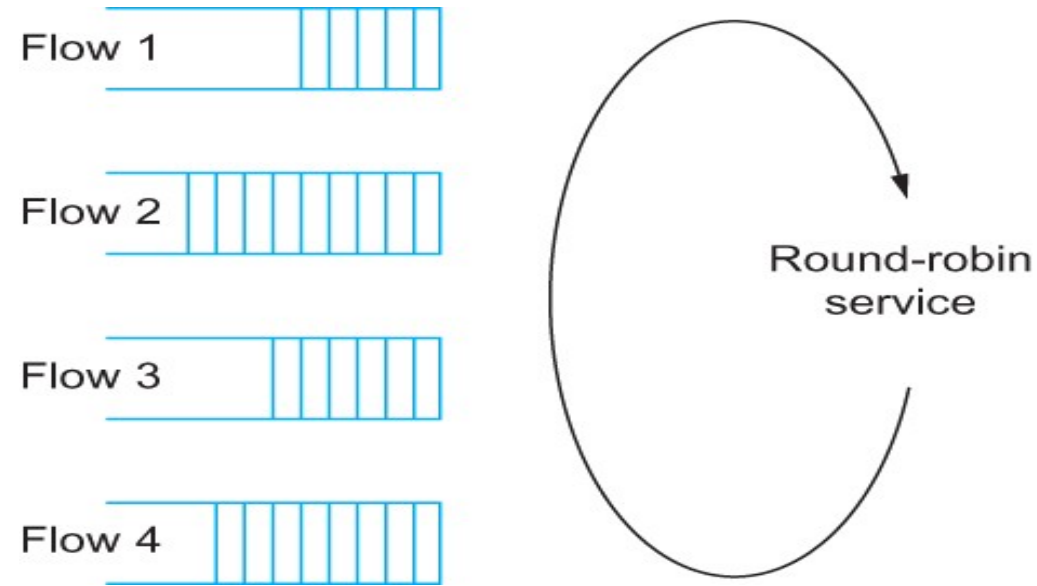
Combination of <Src IP, Src Port, Dst IP, Dst Port>

Fair Queuing

- Fair Queuing
 - FIFO does not discriminate between different traffic sources, or
 - it does not separate packets according to the flow to which they belong.
 - Fair queuing (FQ) maintains a separate queue for each flow

Queuing Disciplines

- Fair Queuing



Round-robin service of four flows at a router

Next steps



MaxMin algorithm and TCP Congestion control