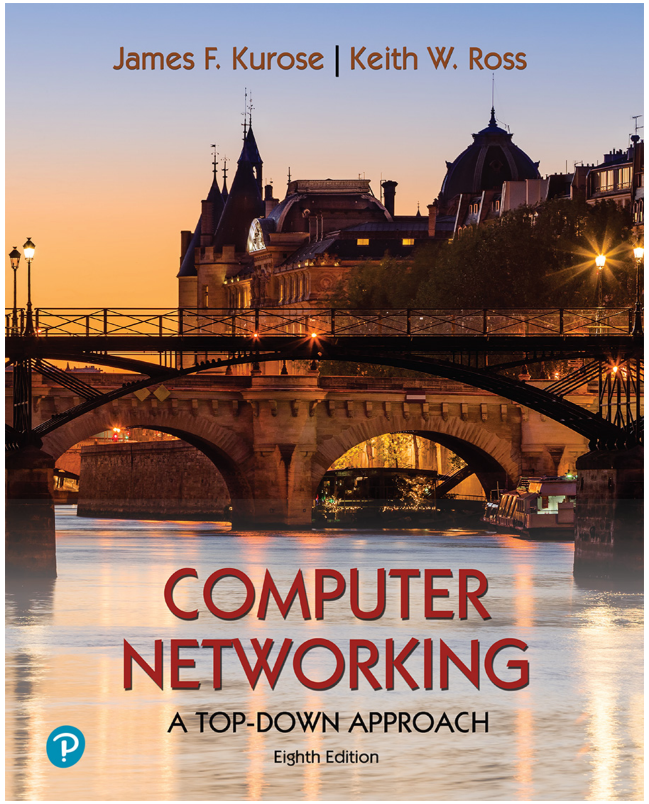
Chapter 3
Transport Layer



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

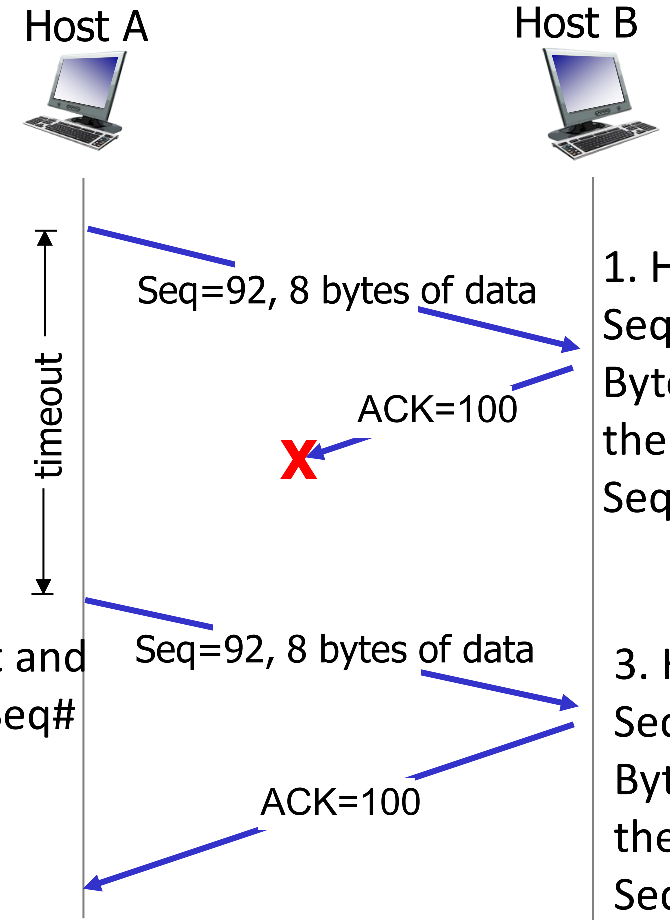
Acknowledgement: Based on the textbook's website:
https://gaia.cs.umass.edu/kurose_ross/index.php

Chapter 3: roadmap

- Transport-layer services
- Multiplexing and demultiplexing
- Connectionless transport: UDP
- Connection-oriented transport: TCP
  - segment structure
  - reliable data transfer
  - flow control
  - connection management
- Principles of congestion control
- TCP congestion control

IMPORTANT

TCP: retransmission scenarios

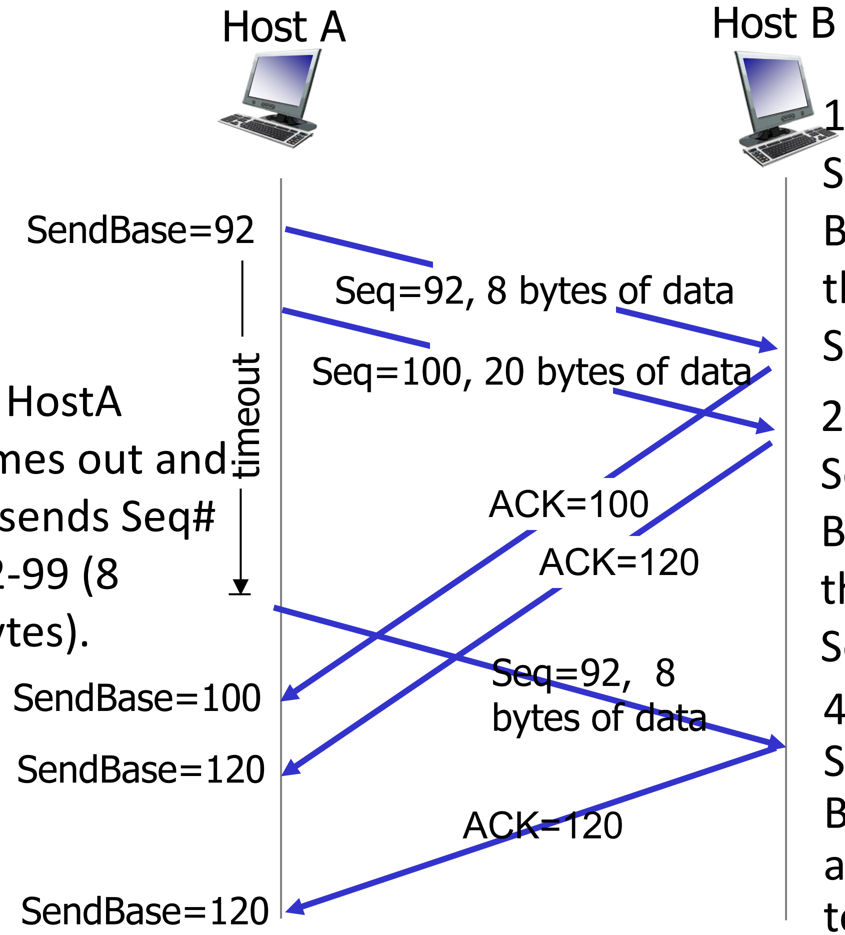


1. HostB receives Seq# 92-99 (8 Bytes), and expects the next received Seq# to be 100

2. HostA times out and resends Seq# 92-99 (8 Bytes).

3. HostB receives Seq# 92-99 (8 Bytes), and expects the next received Seq# to be 100

lost ACK scenario



1. HostB receives Seq# 92-99 (8 Bytes), and expects the next received Seq# to be 100

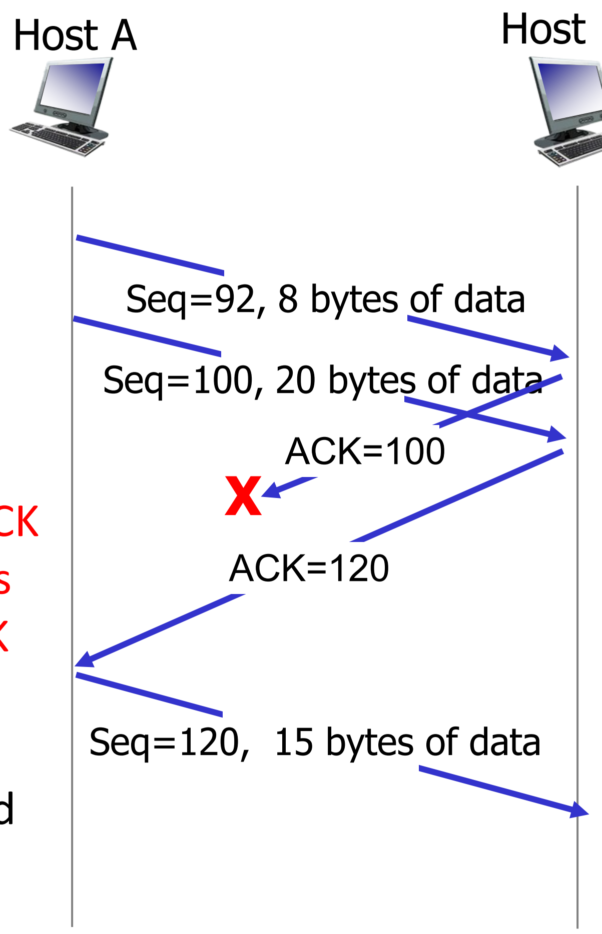
2. HostB receives Seq# 100-119 (20 Bytes), and expects the next received Seq# to be 120

3. HostA times out and resends Seq# 92-99 (8 Bytes).

4. HostB receives Seq# 92-99 (8 Bytes). But it has already received up to Seq# 119, so it sends cumulative ACK for Seq# 120

premature timeout

TCP: retransmission scenarios



1. HostB receives Seq# 92-99 (8 Bytes), and expects the next received Seq# to be 100, but the ACK is lost.
2. HostB receives Seq# 100-119 (20 Bytes), and expects the next received Seq# to be 120

3. HostA receives ACK for Seq# 120. This cumulative ACK of Seq# 120 covers for earlier lost ACK of Seq#100, so HostA knows that HostB has received all bytes up to Seq#119, so it can send the next 15 Bytes (Seq#120-134).

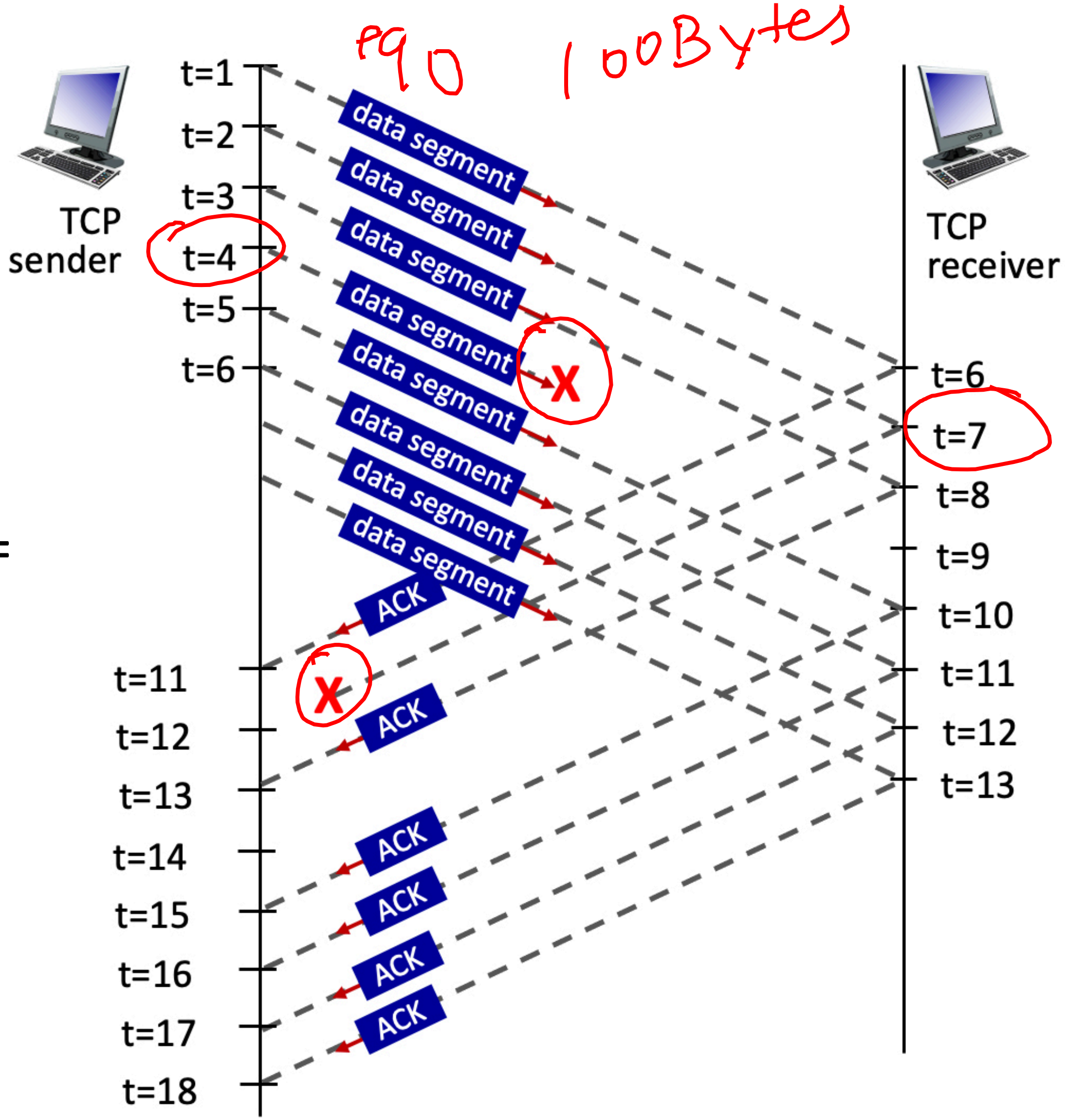
- Q: what happens if the segment with Seq=92, 8 bytes of data from Host A to Host B gets lost?
- A: Host B will NOT send ACK=120, since a cumulative ACK=120 implies that all previous segments with Seq < 120 have been received

3.5-2a. TCP sequence and ACK numbers.

- Consider the figure, where a TCP sender sends 8 TCP segments at t = 1, 2, 3, 4, 5, 6, 7, 8. Suppose the initial value of the sequence number is 0 and every segment sent to the receiver each contains 100 bytes. The delay between the sender and receiver is 5 time units, and so the first segment arrives at the receiver at t = 6. The ACKs sent by the receiver at t = 6, 7, 8, 10, 11, 12 are shown. The TCP segments (if any) sent by the sender at t = 11, 13, 15, 16, 17, 18 are not shown. The segment sent at t=4 is lost, as is the ACK segment sent at t=7.
- Q: What is the sequence number of the segment sent at t=2?
- A: 0+100=100.
- See notes for explanations

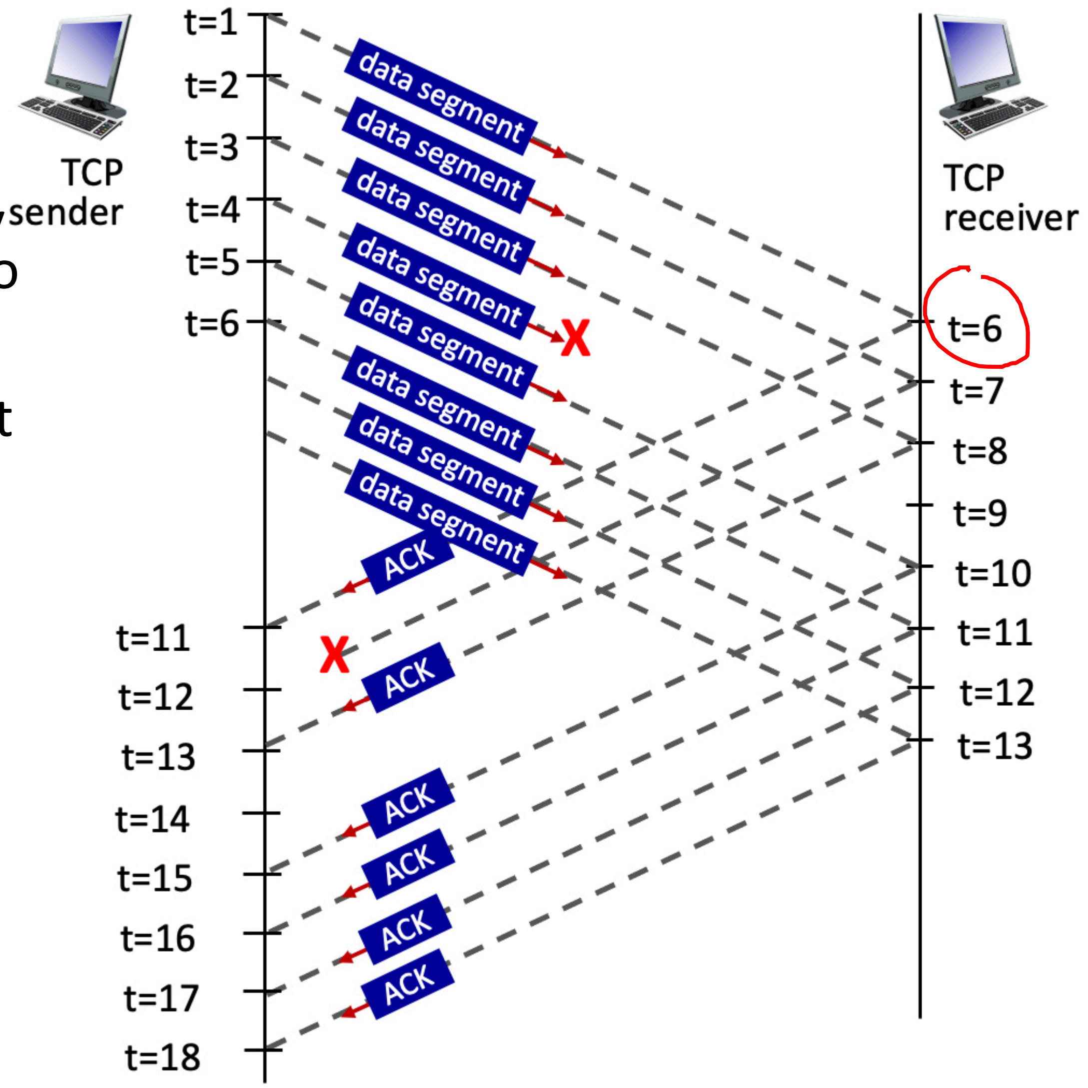
t=1 0-99

t=2 100-199



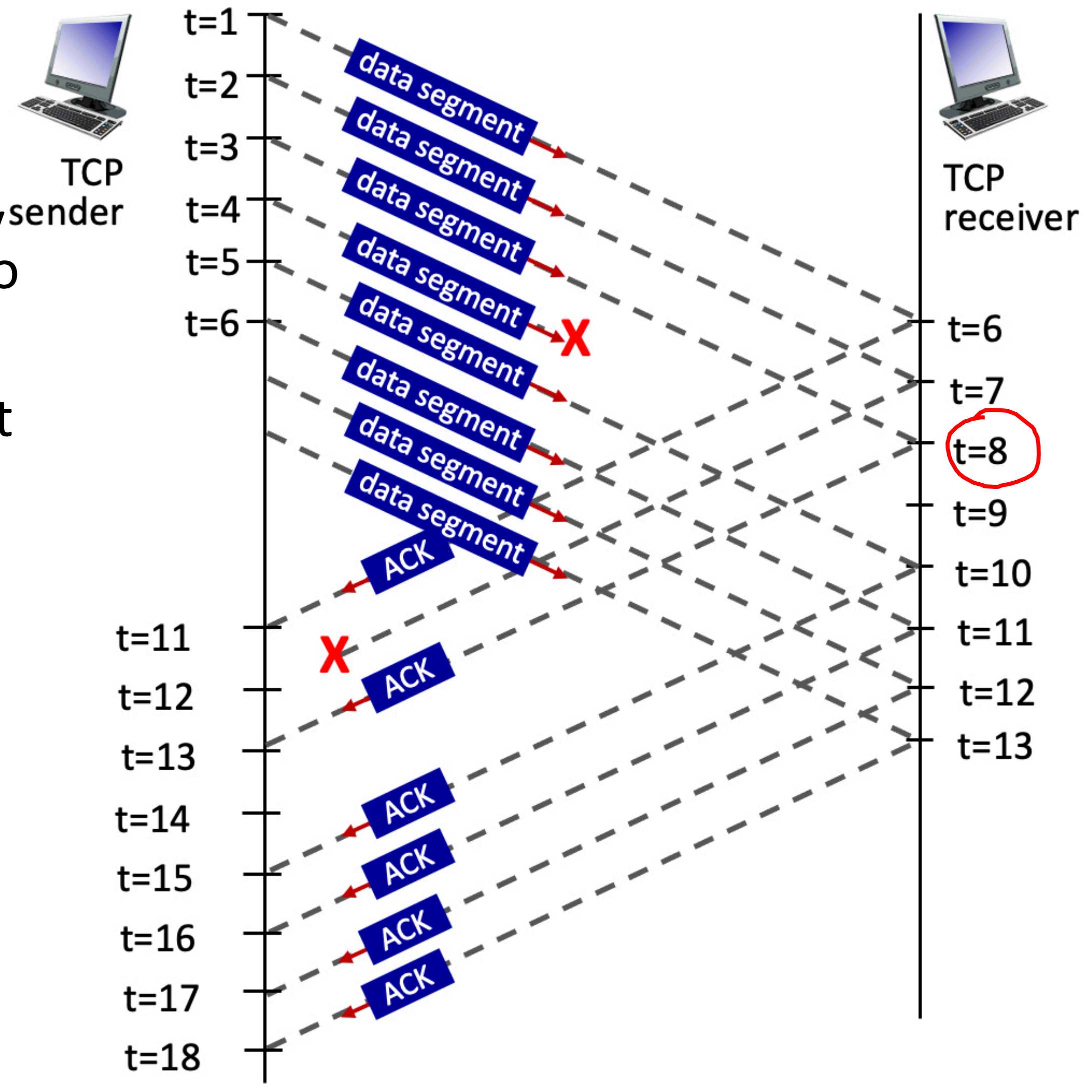
3.5-2b. TCP sequence and ACK numbers.

- Consider the figure, where a TCP sender sends 8 TCP segments at t = 1, 2, 3, 4, 5, 6, 7, 8. Suppose the initial value of the sequence number is 0 and every segment sent to the receiver each contains 100 bytes. The delay between the sender and receiver is 5 time units, and so the first segment arrives at the receiver at t = 6. The ACKs sent by the receiver at t = 6, 7, 8, 10, 11, 12 are shown. The TCP segments (if any) sent by the sender at t = 11, 13, 15, 16, 17, 18 are not shown. The segment sent at t=4 is lost, as is the ACK segment sent at t=7.
- Q: What is the ACK value carried in the receiver-to-sender ACK sent at t = 6?
- A: 0+100=100. By sending this ACK, the receiver is telling the sender: "I've received everything up to byte 99, and I'm now expecting byte 100."
- See notes for explanations



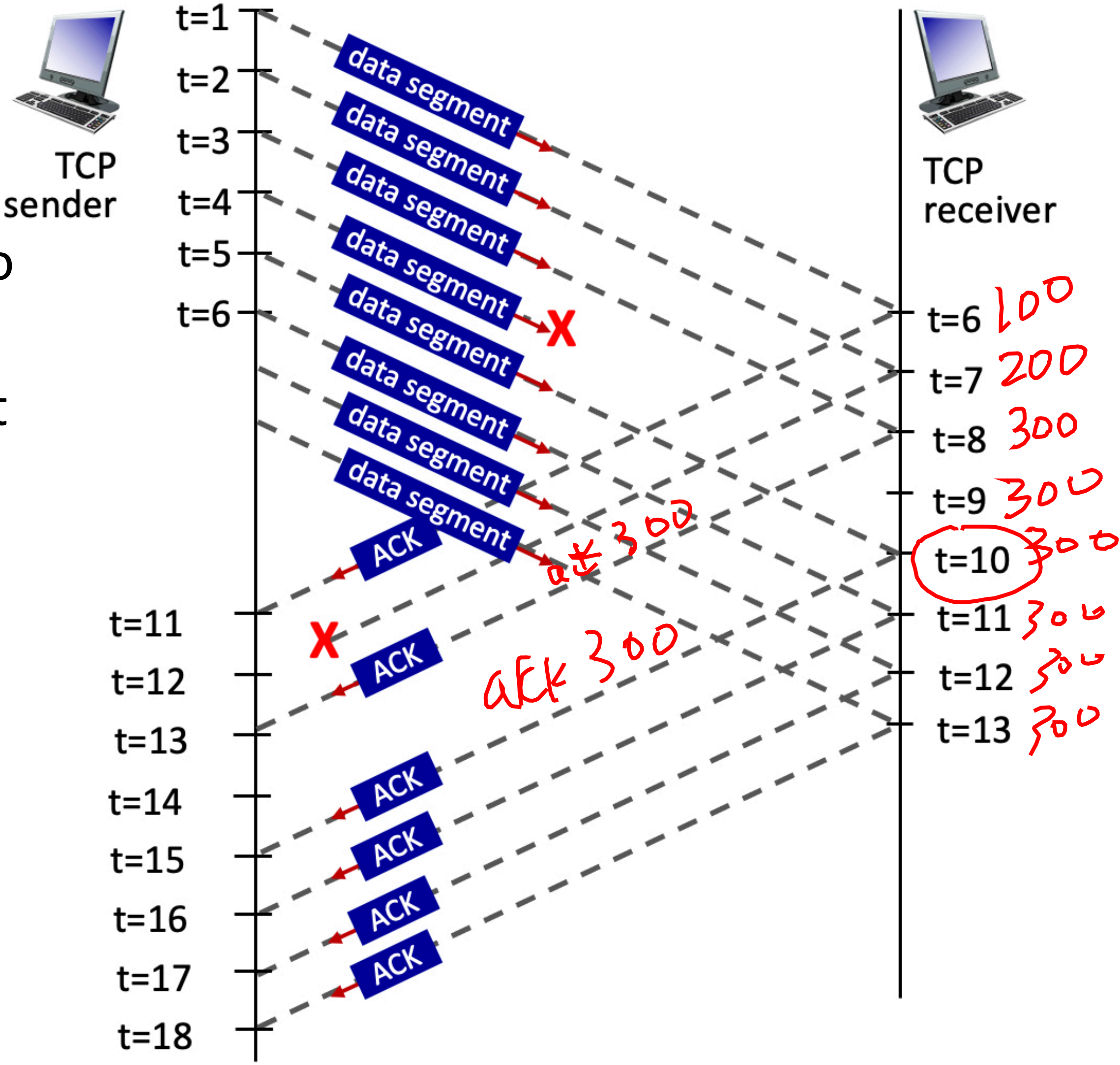
3.5-2c. TCP sequence and ACK numbers.

- Consider the figure, where a TCP sender sends 8 TCP segments at t = 1, 2, 3, 4, 5, 6, 7, 8. Suppose the initial value of the sequence number is 0 and every segment sent to the receiver each contains 100 bytes. The delay between the sender and receiver is 5 time units, and so the first segment arrives at the receiver at t = 6. The ACKs sent by the receiver at t = 6, 7, 8, 10, 11, 12 are shown. The TCP segments (if any) sent by the sender at t = 11, 13, 15, 16, 17, 18 are not shown. The segment sent at t=4 is lost, as is the ACK segment sent at t=7.
- Q: What is the ACK value carried in the receiver-to-sender ACK sent at t = 8?
- A: 300. By sending this ACK, the receiver is telling the sender: "I have received everything up to byte 299. Please send me the next byte, which should be 300."
- See notes for explanations

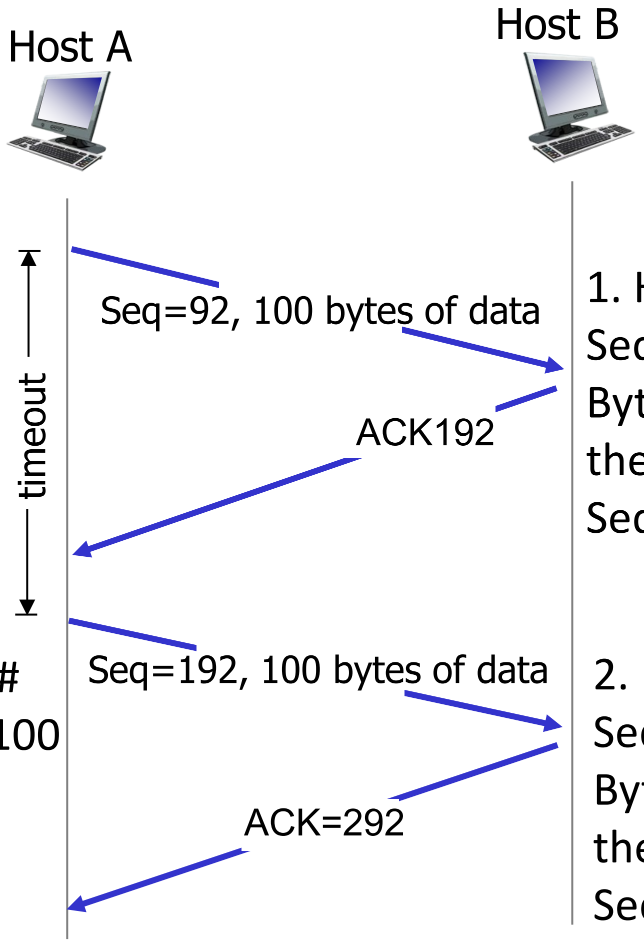


3.5-2d. TCP sequence and ACK numbers.

- Consider the figure, where a TCP sender sends 8 TCP segments at t = 1, 2, 3, 4, 5, 6, 7, 8. Suppose the initial value of the sequence number is 0 and every segment sent to the receiver each contains 100 bytes. The delay between the sender and receiver is 5 time units, and so the first segment arrives at the receiver at t = 6. The ACKs sent by the receiver at t = 6, 7, 8, 10, 11, 12 are shown. The TCP segments (if any) sent by the sender at t = 11, 13, 15, 16, 17, 18 are not shown. The segment sent at t=4 is lost, as is the ACK segment sent at t=7.
- Q: What is the ACK value carried in the receiver-to-sender ACK sent at t = 10?
- A: 300. By sending this ACK, the receiver is telling the sender: "I have received everything up to byte 299. Please send me the next byte, which should be 300."
- See notes for explanations



Quiz

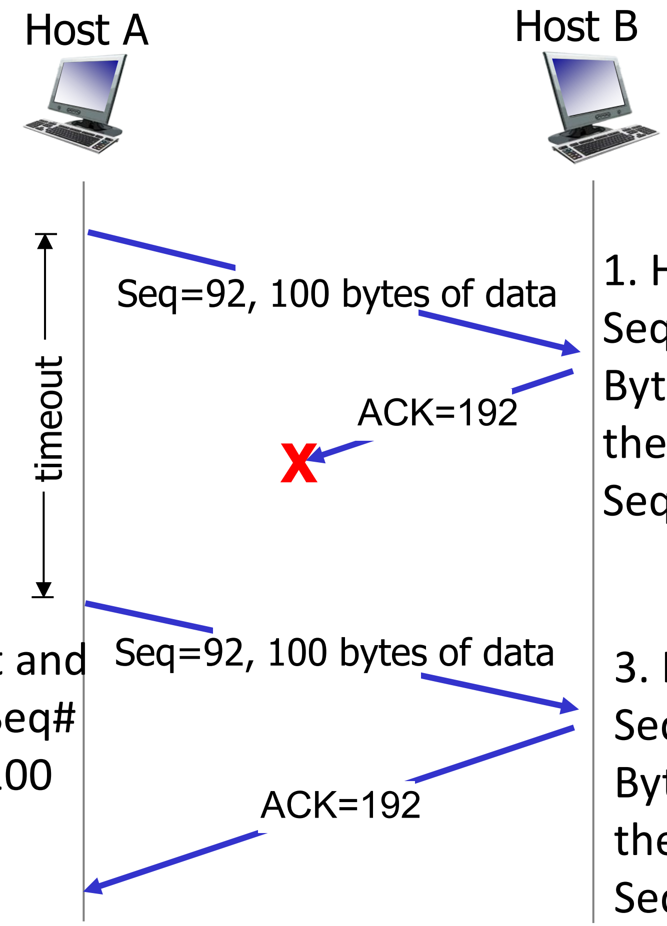


1. HostB receives Seq# 92-191 (100 Bytes), and expects the next received Seq# to be 192

2. HostA sends Seq# 192-291 (100 Bytes).

2. HostB receives Seq# 192-291 (100 Bytes), and expects the next received Seq# to be 292

Received ACK scenario



1. HostB receives Seq# 92-191 (100 Bytes), and expects the next received Seq# to be 192

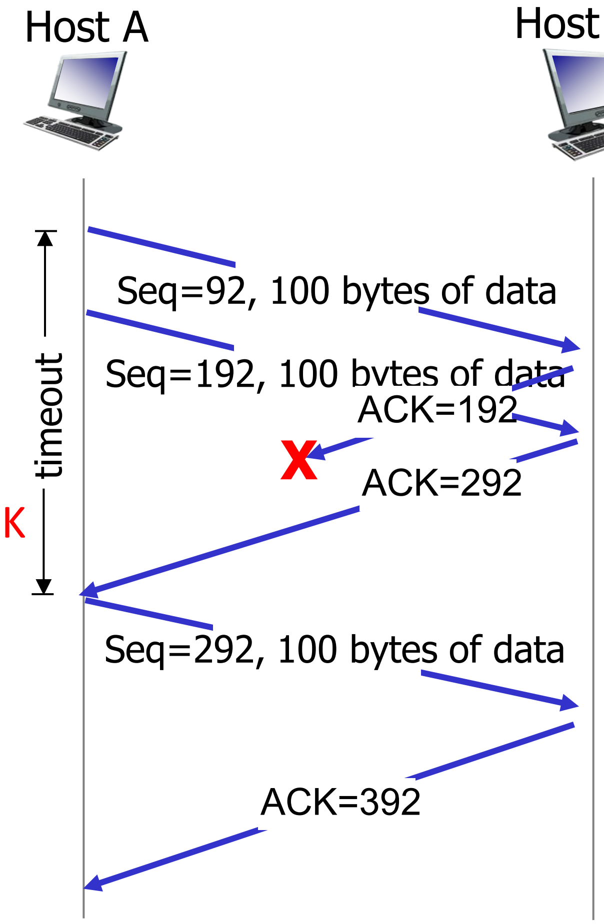
2. HostA times out and resends Seq# 92-191 (100 Bytes).

3. HostB receives Seq# 92-191 (100 Bytes), and expects the next received Seq# to be 192

lost ACK scenario

Quiz

3. HostA receives ACK for Seq# 292. This cumulative ACK of Seq# 292 covers for earlier lost ACK of Seq#192, so HostA knows that HostB has received all bytes up to Seq#291, so it can send the next 100 Bytes

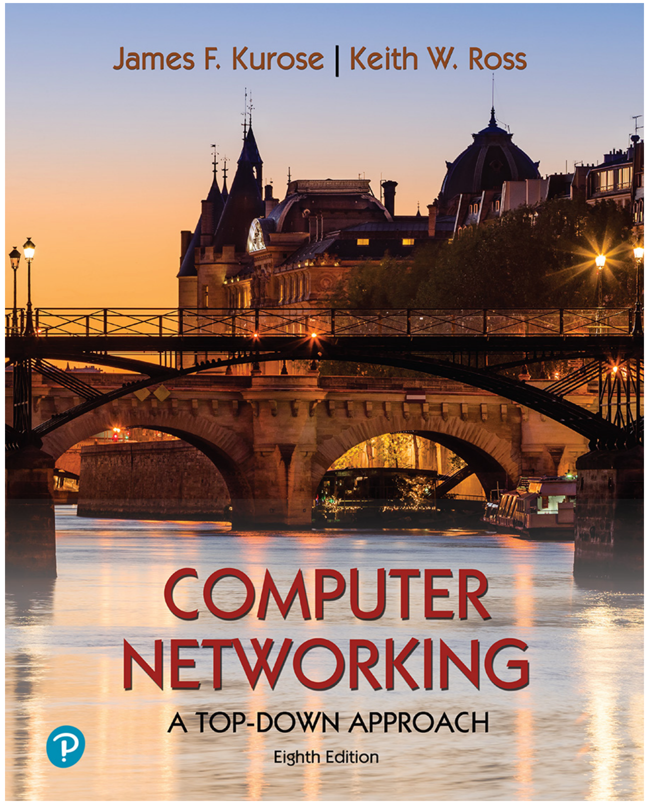


Cumulative ACK scenario

1. HostB receives Seq# 92-191 (100 Bytes), and expects the next received Seq# to be 192

2. HostB receives Seq# 192-291 (100 Bytes), and expects the next received Seq# to be 292

Chapter 4
Network Layer: Data Plane



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Acknowledgement: Based on the textbook's website:
https://gaia.cs.umass.edu/kurose_ross/index.php

Network layer: "data plane" roadmap

- Network layer: overview
  - data plane
  - control plane
- What's inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6

- Generalized Forwarding, SDN
  - Match+action
  - OpenFlow: match+action in action
- Middleboxes

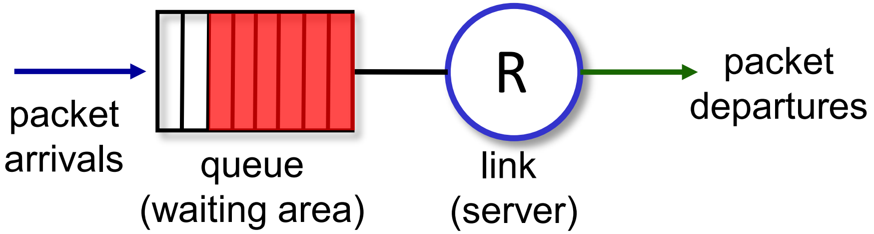
IMPORTANT

Packet Scheduling: FCFS

packet scheduling: deciding which packet to send next on link

- first come, first served
- priority
- round robin
- weighted fair queueing

Abstraction: queue



FCFS: packets transmitted in order of arrival to output port

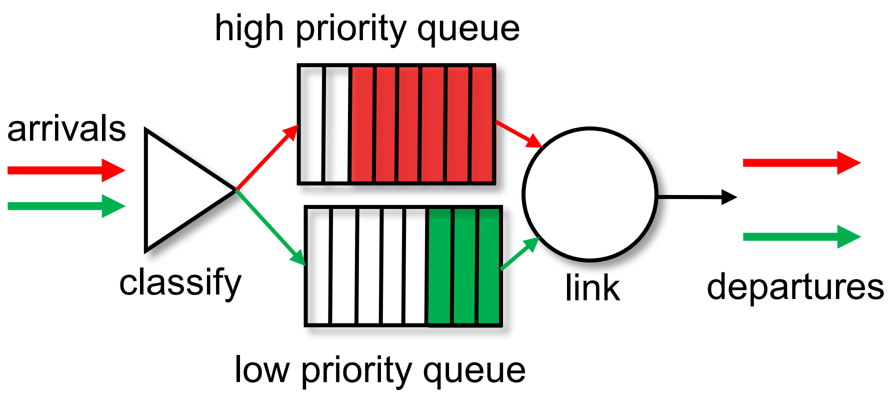
- also known as: First-in-first-out (FIFO)
- real world examples?

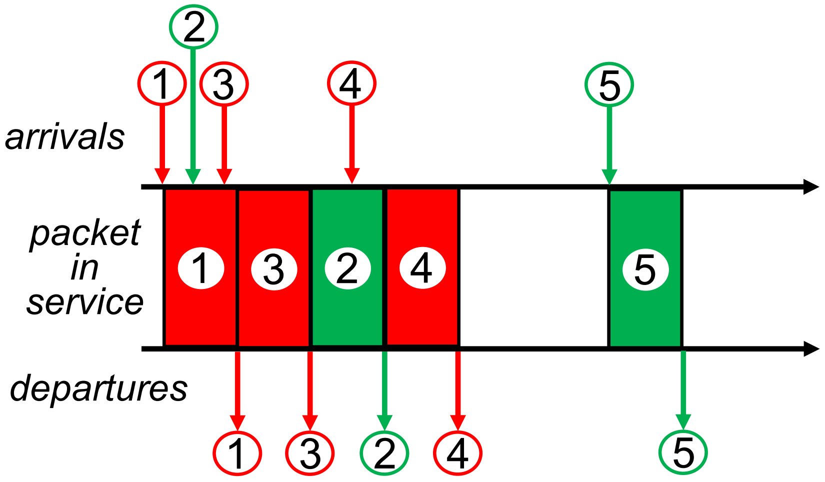
IMPORTANT

Scheduling policies: priority

Priority scheduling:

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- send packet from highest priority queue that has buffered packets
  - FCFS within priority class



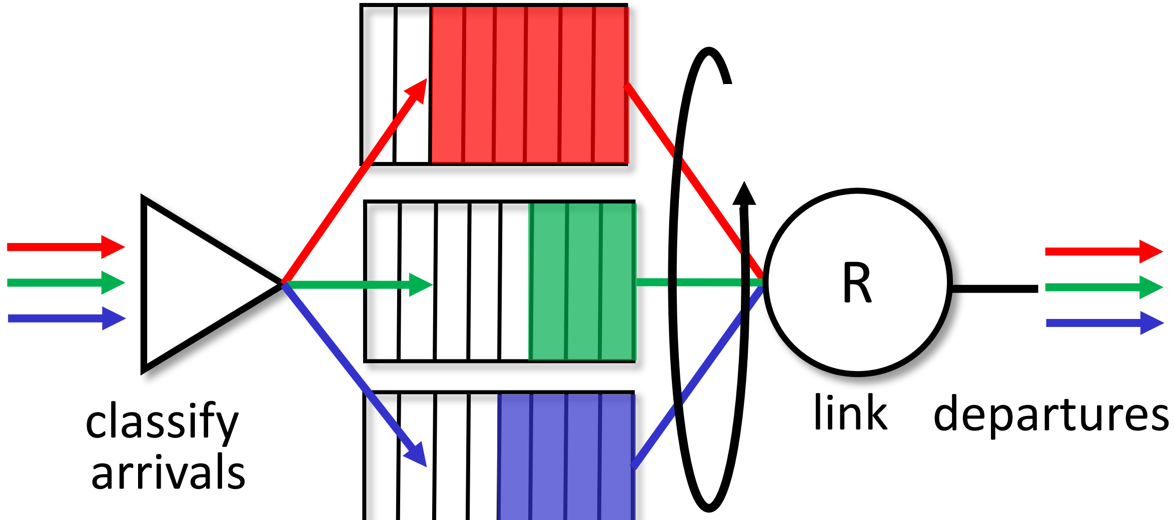


IMPORTANT

Scheduling policies: round robin

Round Robin (RR) scheduling:

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- server cyclically, repeatedly scans class queues, sending one complete packet from each class (if available) in turn



IMPORTANT

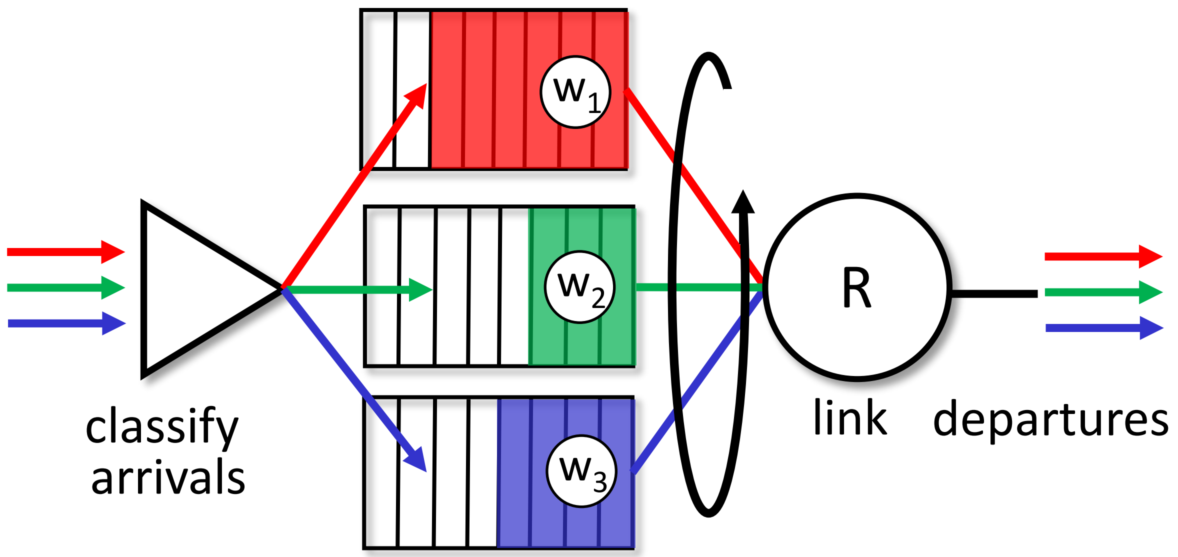
Scheduling policies: weighted fair queueing

Weighted Fair Queuing (WFQ):

- generalized Round Robin
- each class, *i*, has weight, w_i, and gets weighted amount of service in each cycle:

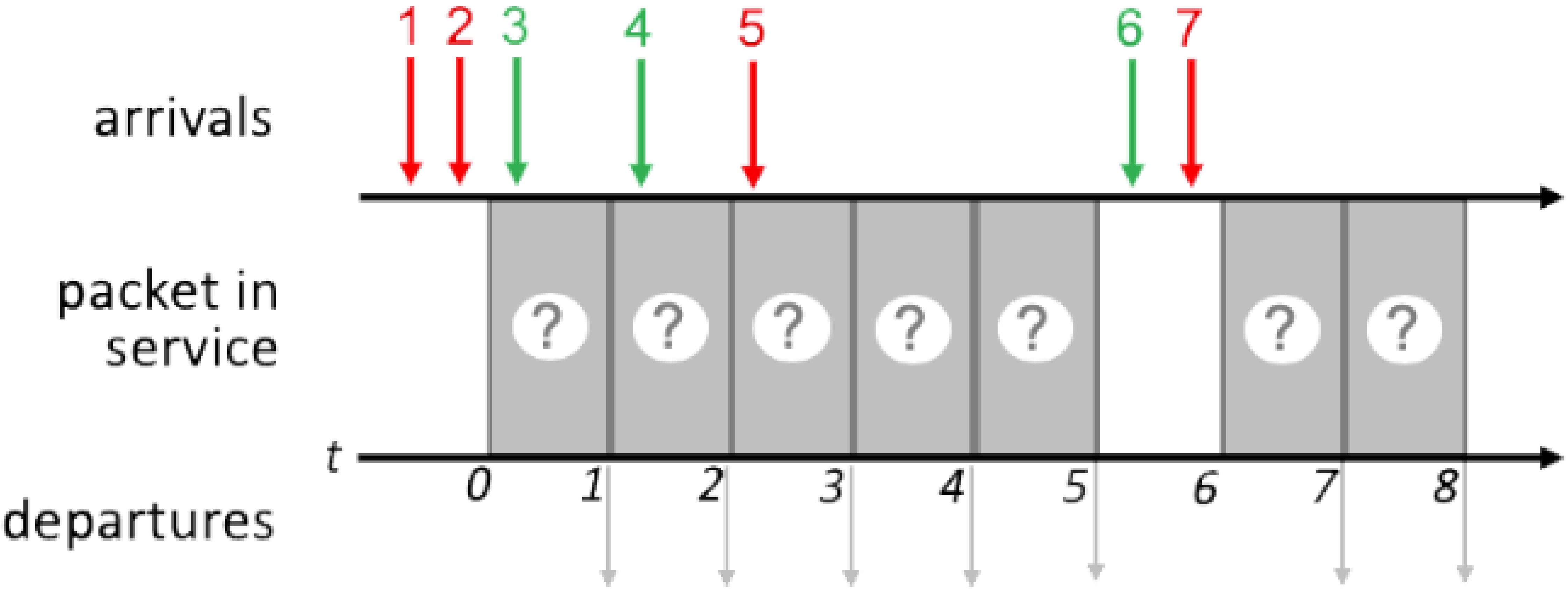
$$\frac{w_i}{\Sigma_j w_j}$$

- minimum bandwidth guarantee (per-traffic-class)

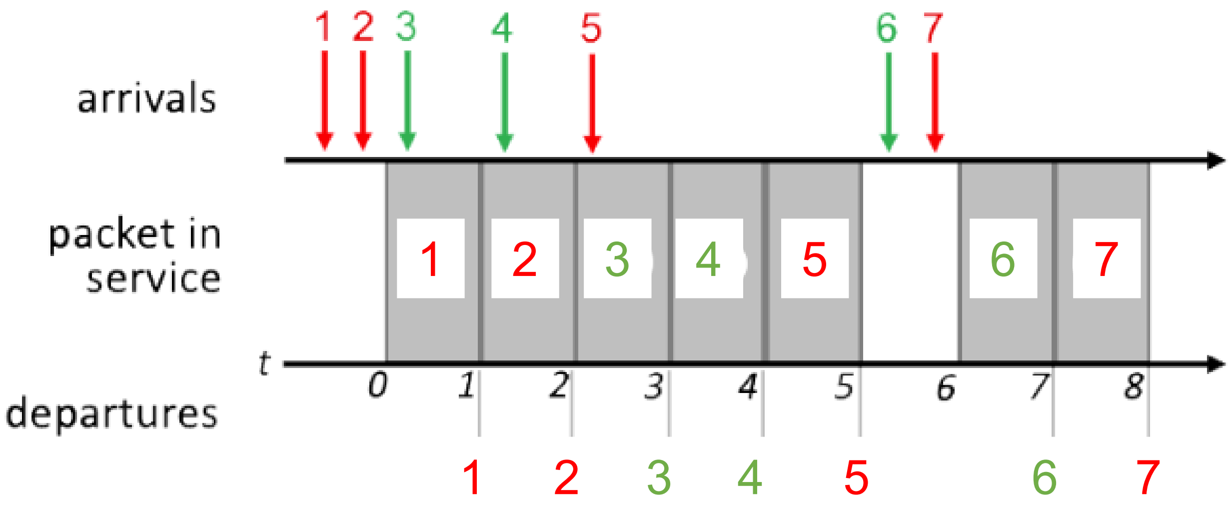


IMPORTANT

Quiz 1 4.2-7



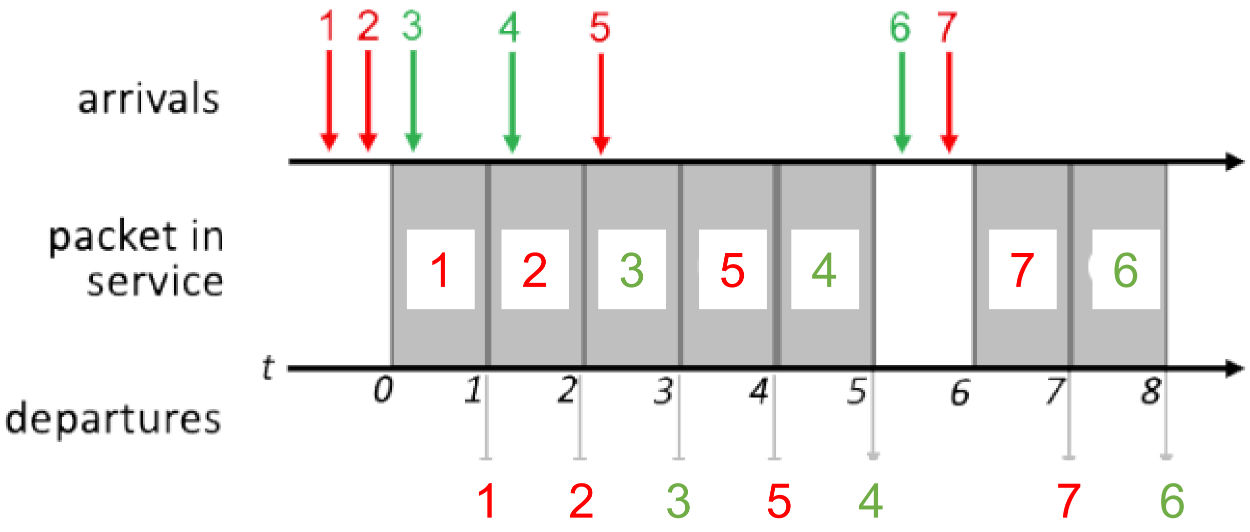
FCFS Scheduling



1 2 3 4 5 6 7

- Transmit order the same as packet arrival order of 1 2 3 4 5 6 7

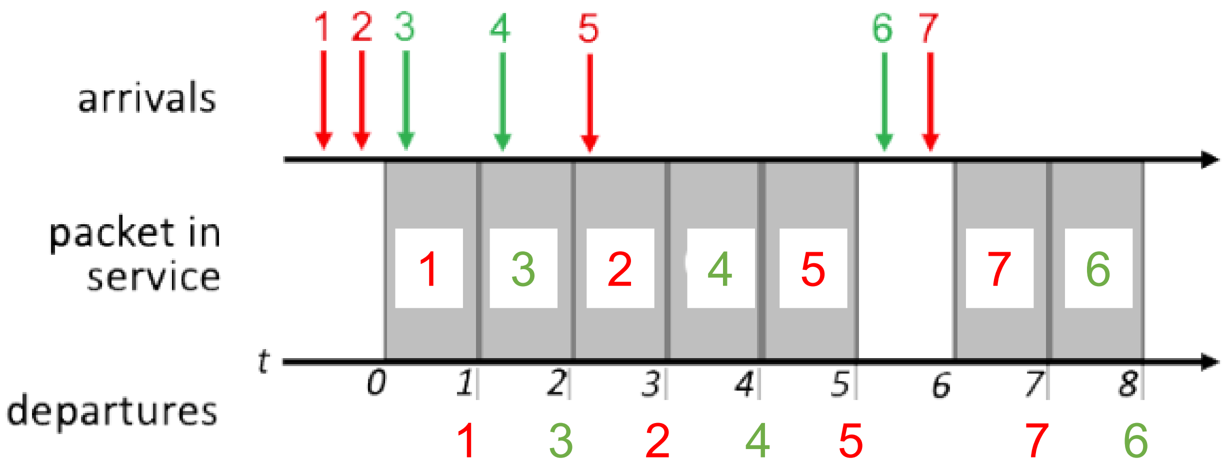
Priority Scheduling



1 2 3 5 4 7 6

- Time 0: 1, 2 in queue, transmit 1
  - FCFS within same priority
- Time 1: 2, 3 in queue, transmit 2
- Time 2: 3, 4 in queue, transmit 3
  - FCFS within same priority
- Time 3: 4, 5 in queue, transmit 5
- Time 4: 4 in queue, transmit 4
- Time 6: 6, 7 in queue, transmit 7
- Time 7: 6 in queue, transmit 6

Round Robin Scheduling

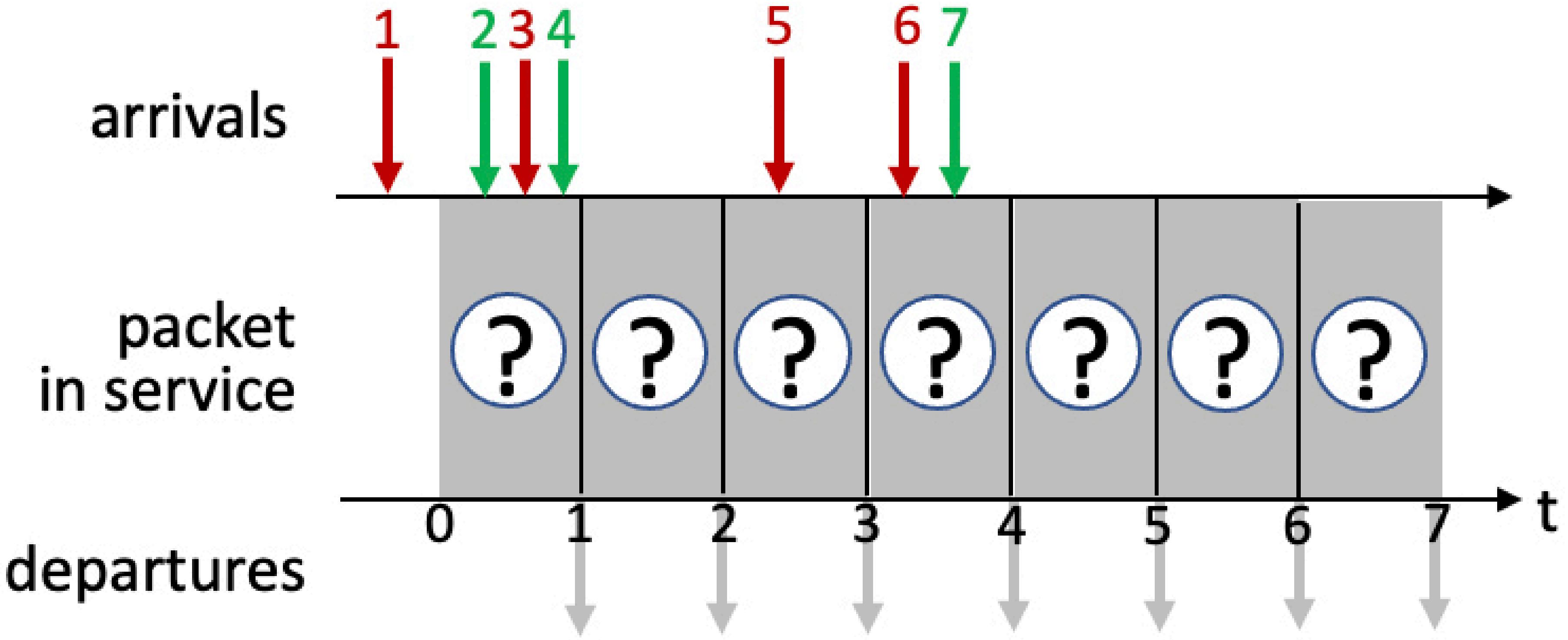


1 3 2 4 5 7 6

- Assume a round-robin scheduling cycle begins with red packets. i.e., (red, green) in each round.
- Time 0: 1, 2 in queue, transmit 1
  - 1^{st} round of (red, green)
- Time 1: 2, 3 in queue, transmit 3
  - 1^{st} round of (red, green)
- Time 2: 2. 4 in queue, transmit 2
  - 2^{nd} round of (red, green)
- Time 3: 4, 5 in queue, transmit 4
  - 2^{nd} round of (red, green)
- Time 4: 5 in queue, transmit 5
  - 3^{RD} round of (red, green). Since there is no green packet ready, this round is (red, null)
- Time 6: 6, 7 in queue, transmit 7
  - 4^{th} round of (red, green)
- Time 7: 6 in queue, transmit 6
  - 4^{th} round of (red, green)
- Summary:
- Times 0-1: 1^{st} round: (1, 3)
- Times 2-3: 2^{nd} round: (2, 4)
- Time 4: 3^{rd} round: (5, null)
  - No green packets ready
- Times 6-7: 4^{th} round: (7, 6)

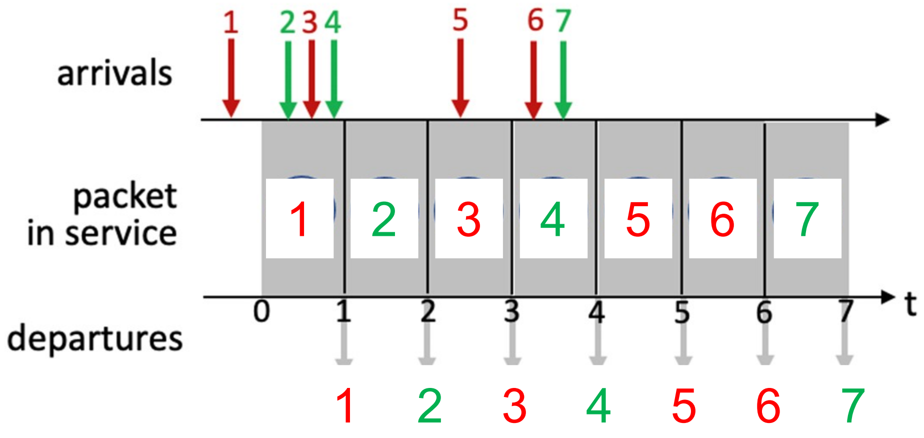
IMPORT

Quiz 2 4.2-3



IMPORTANT

FCFS Scheduling

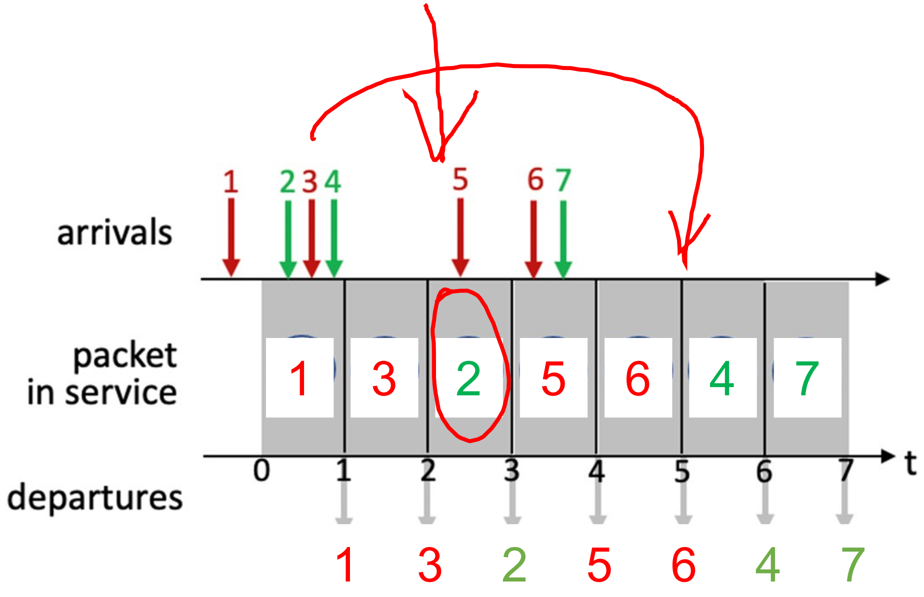


1 2 3 4 5 6 7

- Transmit order the same as packet arrival order of 1 2 3 4 5 6 7

IMPORTANT

Priority Scheduling

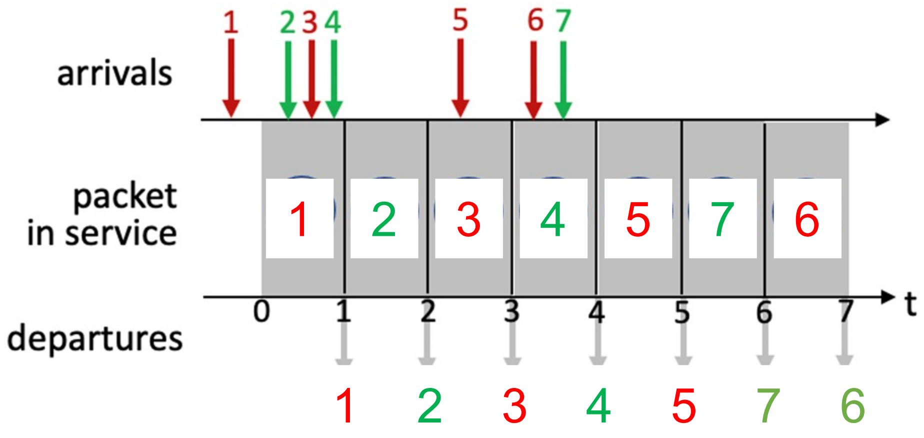


1 3 2 5 6 4 7

- Time 0: 1 in queue, transmit 1
- Time 1: 2, 3, 4 in queue, transmit 3
- Time 2: 2, 4 in queue, transmit 2
  - FCFS within same priority
- Time 3: 4, 5 in queue, transmit 5
- Time 4: 4, 6, 7 in queue, transmit 6
- Time 5: 4, 7 in queue, transmit 4
- Time 6: 7 in queue, transmit 7

IMPORTANT

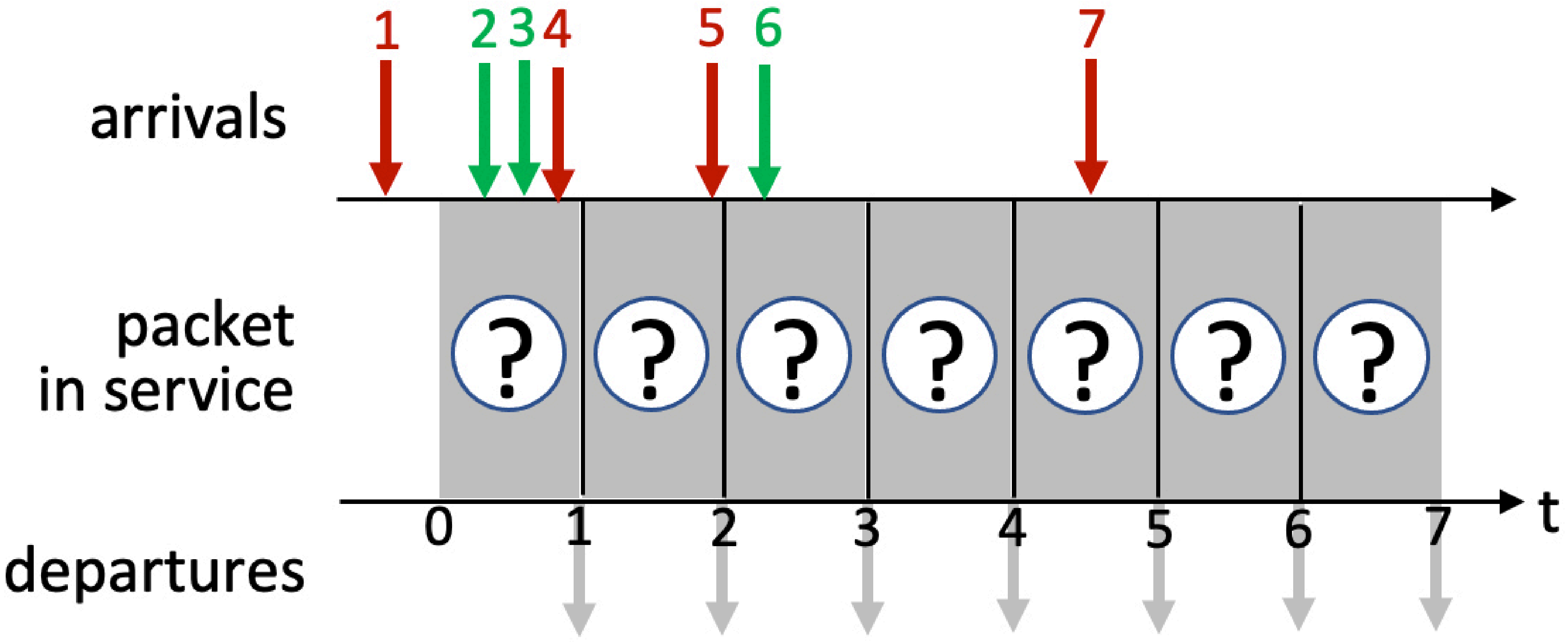
Round Robin Scheduling



1 2 3 4 5 7 6

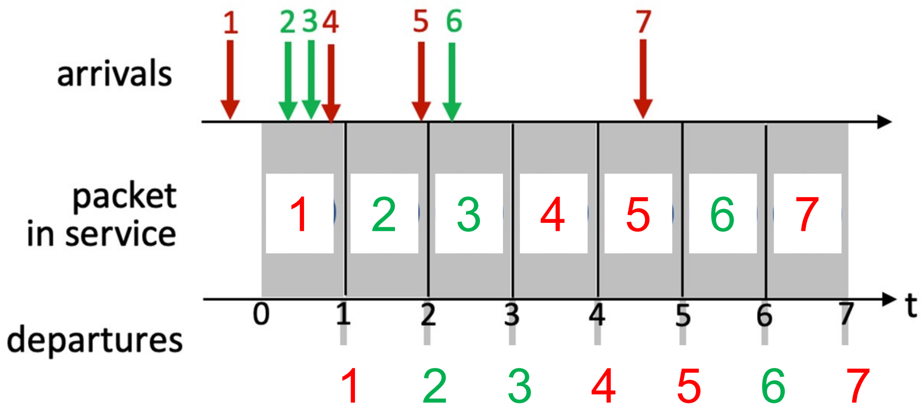
- Assume a round-robin scheduling cycle begins with red packets. i.e., (red, green) in each round.
- Summary:
  - Times 0-1: 1st round: (1, 2)
  - Times 2-3: 2nd round: (3, 4)
  - Times 4-5: 3rd round: (5, 7)
    - No green packets ready
  - Time 6: 4th round: (6, null)

Quiz 3 4.2-4



IMPORTANT

FCFS Scheduling

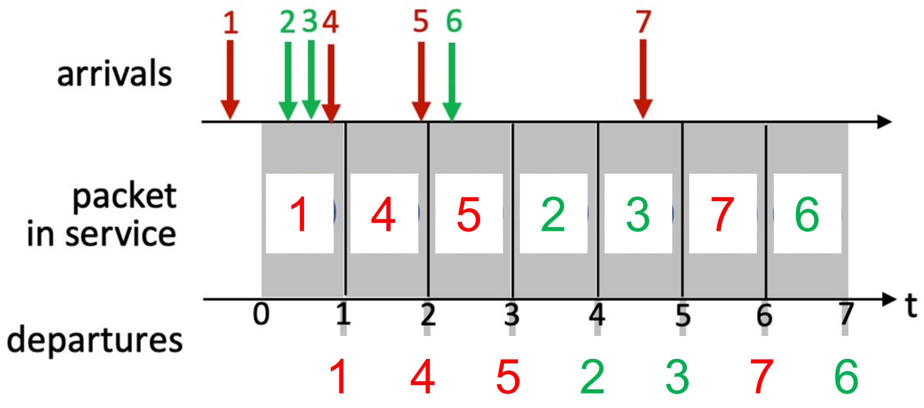


- Transmit order the same as packet arrival order of 1 2 3 4 5 6 7

1 2 3 4 5 6 7

IMPORTANT

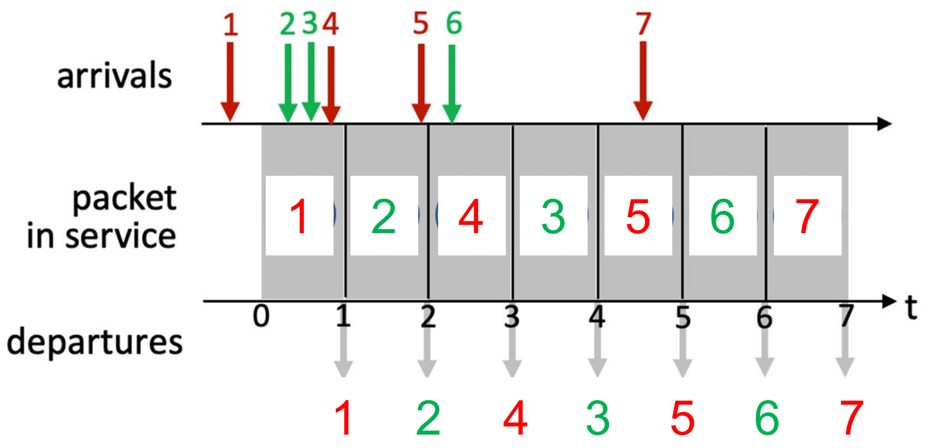
Priority Scheduling



1 4 5 2 3 7 6

- Time 0: 1 in queue, transmit 1
- Time 1: 2, 3, 4 in queue, transmit 4
- Time 2: 2, 3, 5 in queue, transmit 5
- Time 3: 2, 3, 6 in queue, transmit 2
- Time 4: 3, 6 in queue, transmit 3
- Time 5: 6, 7 in queue, transmit 7
- Time 6: 6 in queue, transmit 6

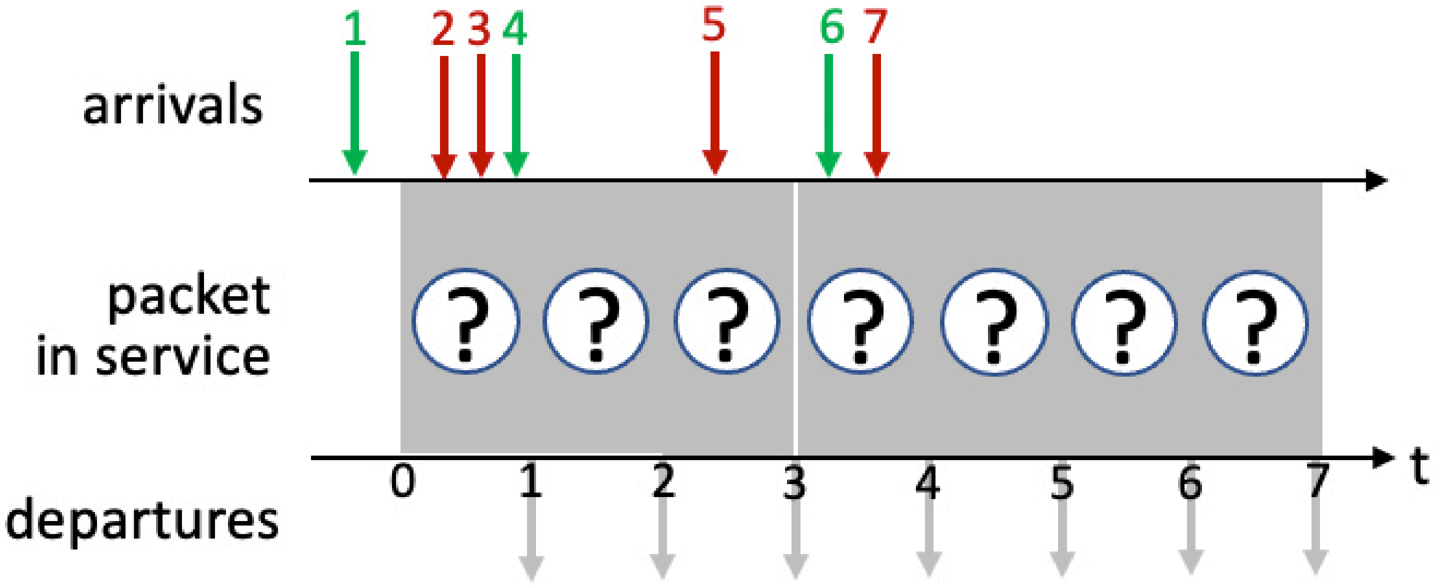
Round Robin Scheduling



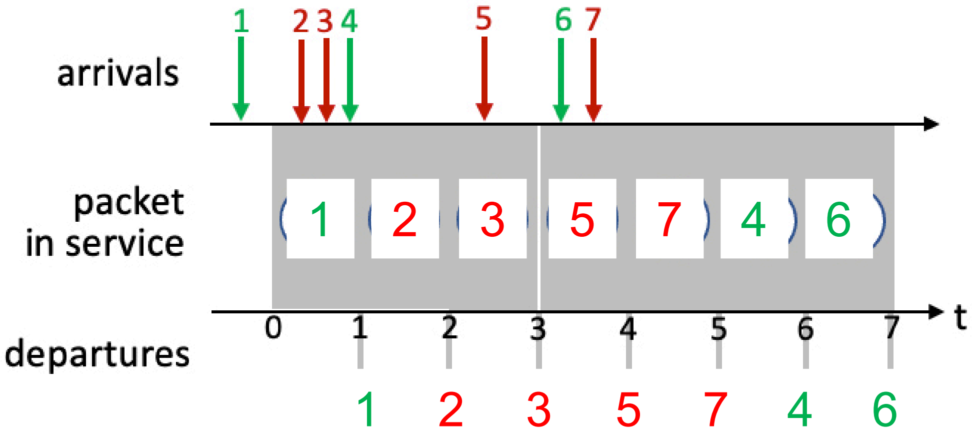
1 2 4 3 5 6 7

- Assume a round-robin scheduling cycle begins with red packets. i.e., (red, green) in each round.
- Summary:
  - Times 0-1: 1st round: (1, 2)
  - Times 2-3: 2nd round: (4, 3)
  - Times 4-5: 3rd round: (5, 6)
    - No green packets ready
  - Time 6: 4th round: (7, null)

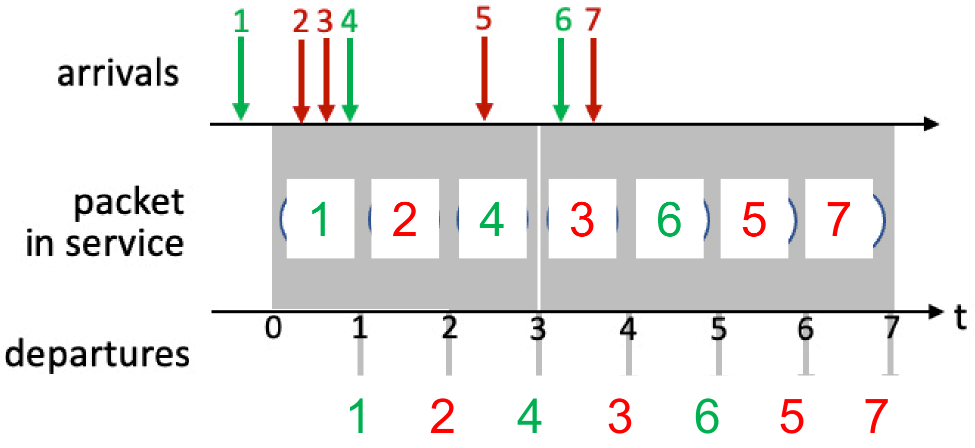
Quiz 4 4.2-1



Quiz 4 4.2-1 Priority

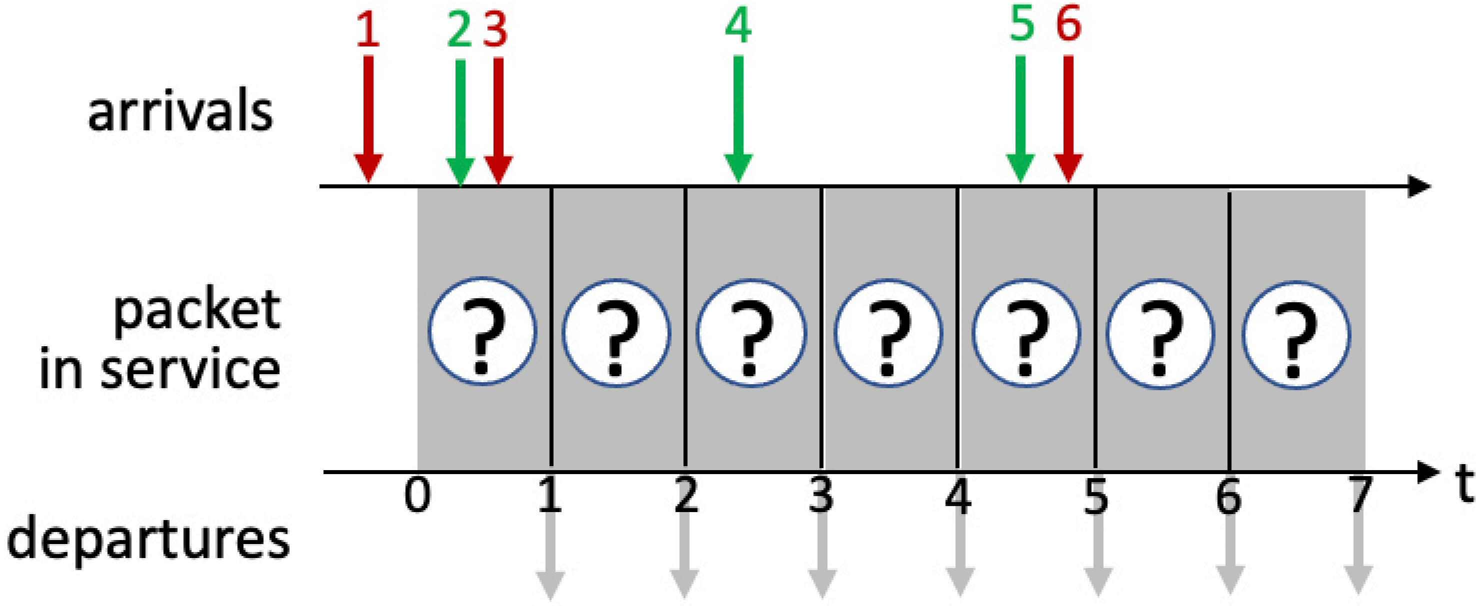


Quiz 4 4.2-1 Round Robin

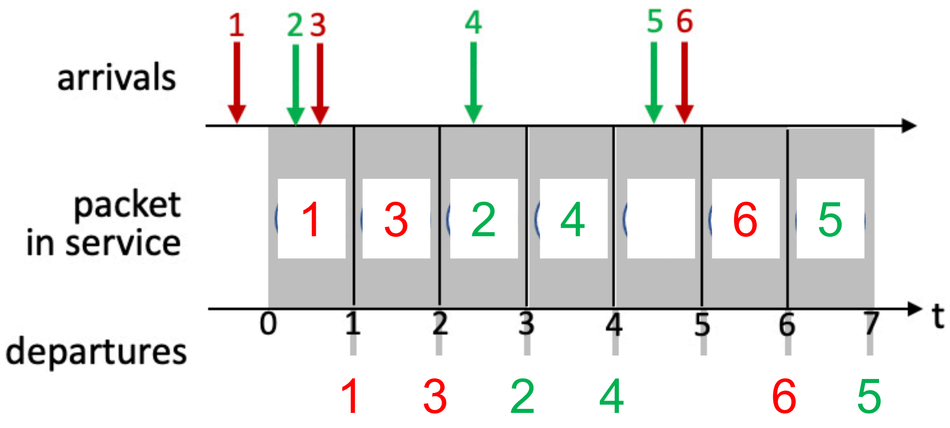


- Assume a round-robin scheduling cycle begins with red packets. i.e., (red, green) in each round.
- Summary:
  - Times 0: 1st round: (null, 1)
  - Times 1-2: 2nd round: (2, 4)
  - Times 3-4: 3rd round: (3, 6)
    - No green packets ready
  - Time 5: 4th round: (5, null)
  - Time 6: 5th round: (7, null)

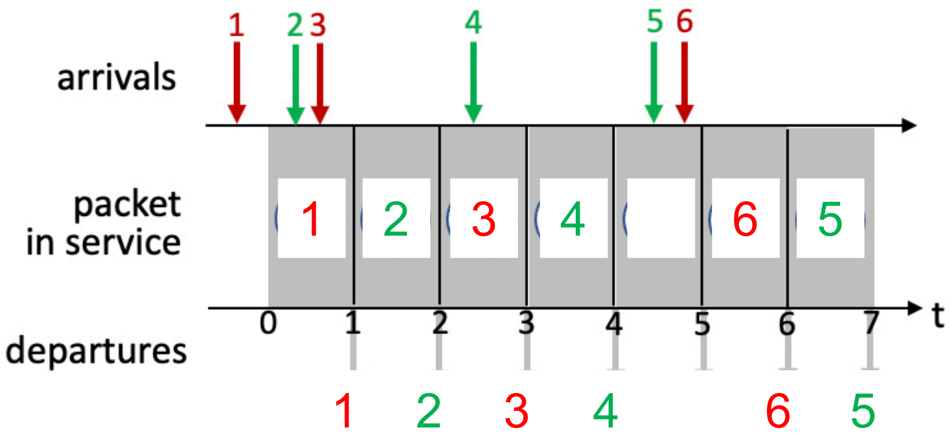
Quiz 5 4.2-2



Quiz 5 4.2-2 Priority



Quiz 5 4.2-2 Round Robin



- Assume a round-robin scheduling cycle begins with red packets. i.e., (red, green) in each round.
- Summary:
  - Times 0: 1st round: (1, 2)
  - Times 1-2: 2nd round: (3, 4)
  - Times 5-6: 3rd round: (6, 5)

Network layer: "data plane" roadmap

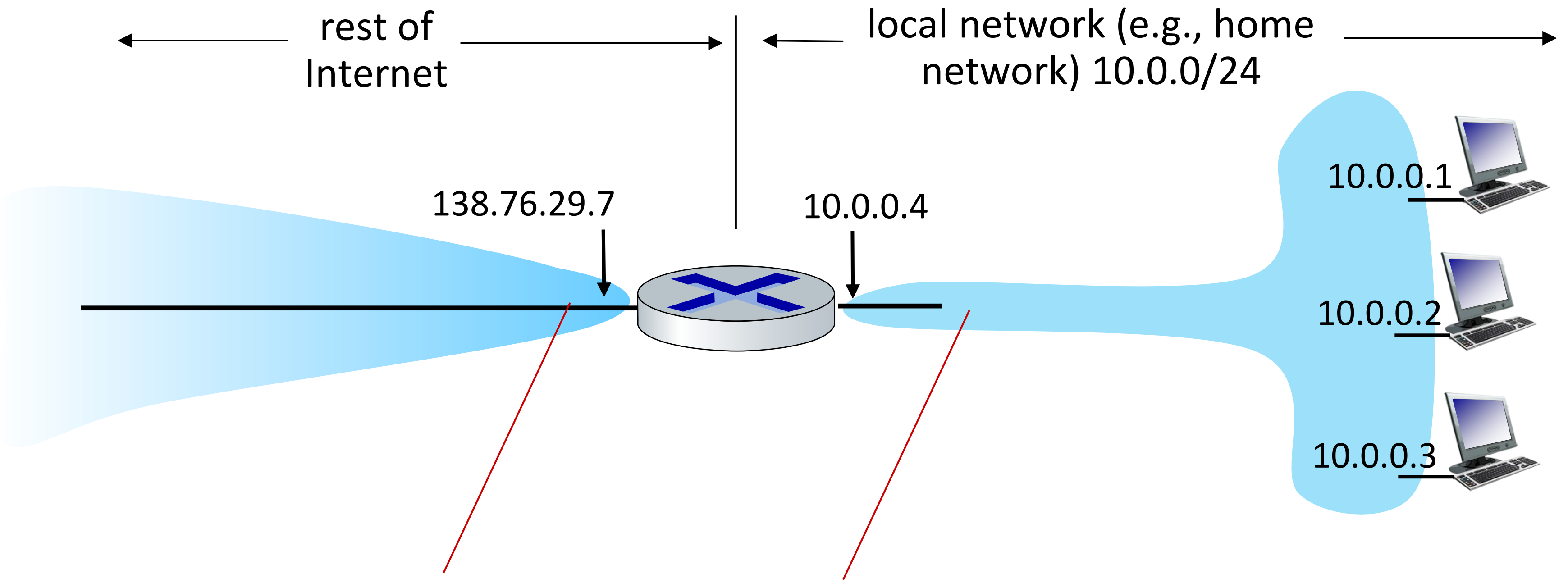
- Network layer: overview
  - data plane
  - control plane
- What's inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6

- Generalized Forwarding, SDN
  - match+action
  - OpenFlow: match+action in action
- Middleboxes

IMPORTANT

NAT: network address translation

NAT: all devices in local network share just one IPv4 address as far as outside world is concerned



all datagrams *leaving* local network have *same* source NAT IP address: 138.76.29.7, but *different* source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

IMPORTANT

NAT: network address translation

- all devices in local network have 32-bit addresses in a “private” IP address space (10/8, 172.16/12, 192.168/16 prefixes) that can only be used in local network
- advantages:
  - just one IP address needed from provider ISP for *all* devices
  - can change addresses of host in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - security: devices inside local net not directly addressable, visible by outside world

IMPORTANT

NAT: network address translation

implementation: NAT router must (transparently):

- outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  - remote clients/servers will respond using (NAT IP address, new port #) as destination address
- remember (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair
- incoming datagrams: replace (NAT IP address, new port #) in destination fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

IMPORTANT

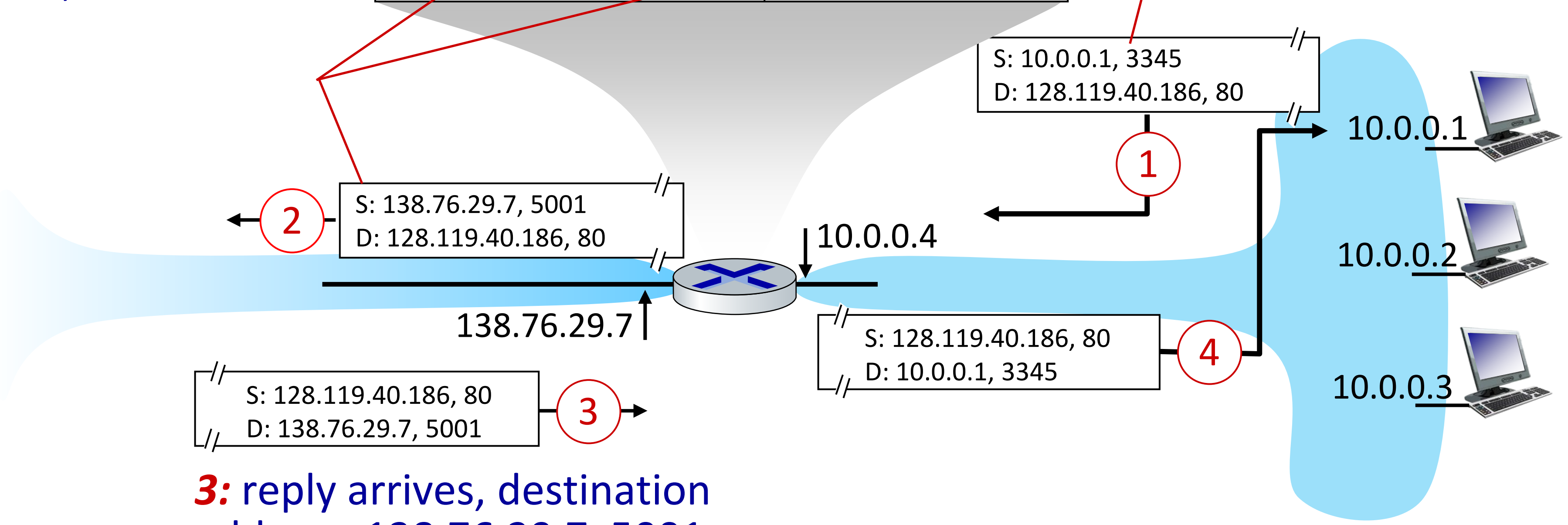
NAT: network address translation

2: NAT router changes datagram source address from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

NAT translation table

WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……

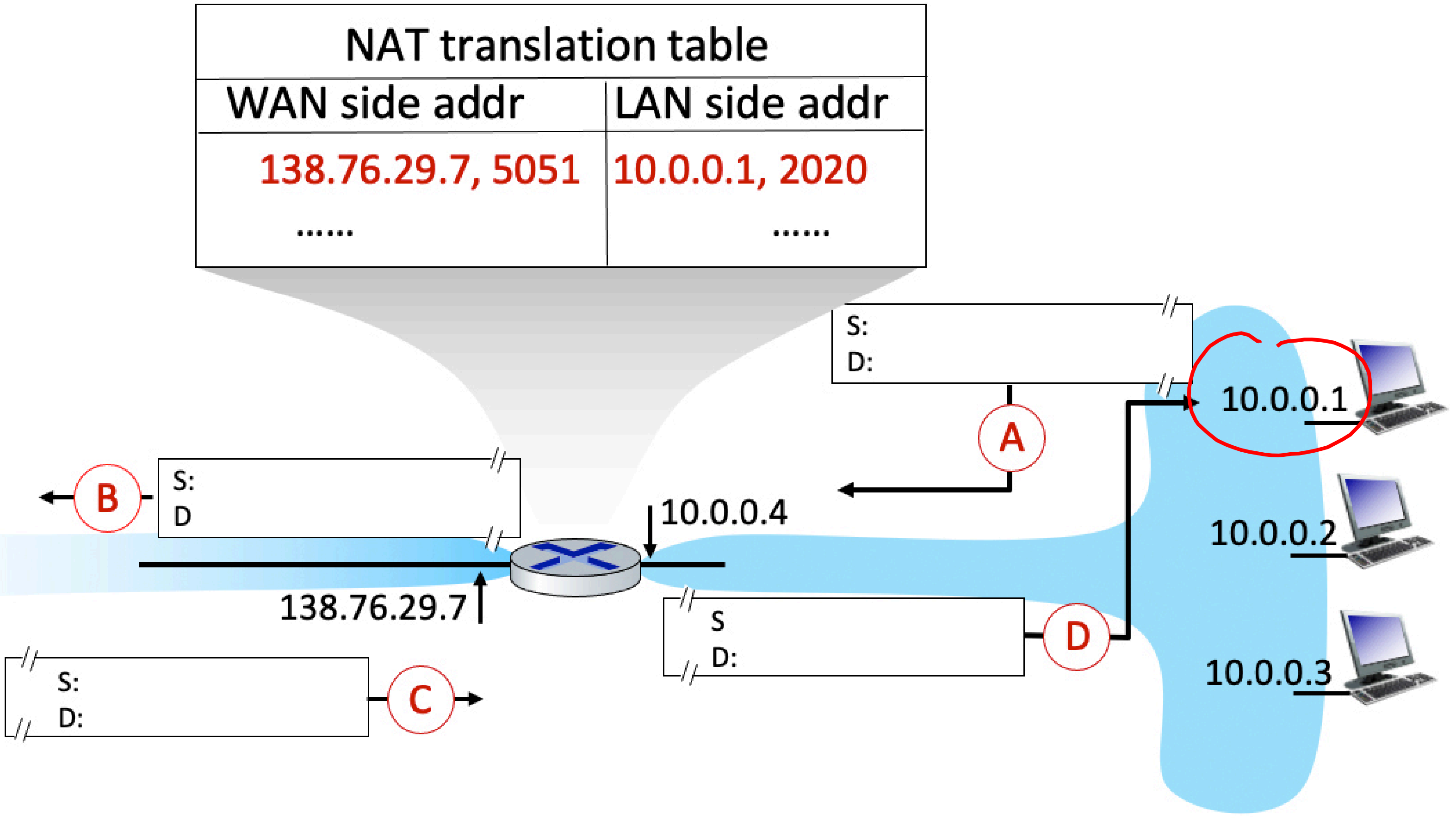
1: host 10.0.0.1 sends datagram to 128.119.40.186, 80



3: reply arrives, destination address: 138.76.29.7, 5001

Quiz

- 4.3-2a Network Address Translation (a). Consider the following scenario in which host 10.0.0.1 is communicating with an external web server at IP address 128.119.40.186, port 80. The NAT table shows the table entry associated with this TCP flow. What are the source and destination IP address and port numbers at points A B C D?



NAT translation table	
WAN side addr	LAN side addr
138.76.29.7, 5051	10.0.0.1, 2020
……	……