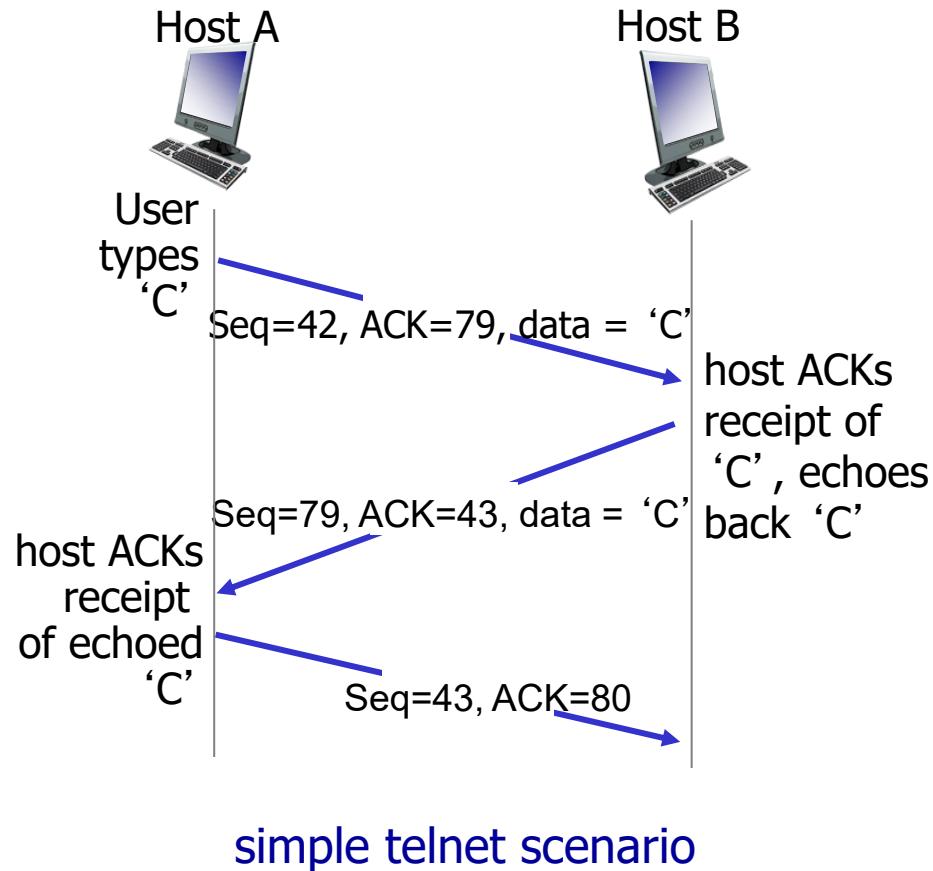


## Problem 1

- a) false
  - For all received data from Host A, Host B needs to send back ACK as feedback for reliable transfer in TCP.
- b) false
  - The sequence numbers for TCP segments are counted in bytes in the data stream.
- c) false
  - ACK in the sent segment is to acknowledge the data received in previous rounds of transmission.

# TCP seq. numbers, ACKs

- ❖ E.g.: Telnet application:  
Host A (client) types a character and Host B (server) echoes back
- ❖ suppose the starting sequence numbers are 42 and 79 for the client and server, respectively
- ❖ Segment I:
  - ❖ Client is waiting for byte 79



# Problem 1

- d) false
  - The rwnd is a variable kept track by receiver, who also inform its value to sender for flow control.
- e) true
- f) true

## Problem 1

- g) false

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$



↑  
estimated RTT      ↑  
"safety margin"

$$\text{EstimatedRTT} = (1 - x) * \text{EstimatedRTT} + x * \text{SampleRTT}$$

$$\text{DevRTT} = (1 - y) * \text{DevRTT} + y * |\text{SampleRTT} - \text{EstimatedRTT}|$$

# IP fragmentation

- IPv4, the Maximum Transmission Unit (MTU) defines the largest size of a single packet (datagram) that a network link or interface can transmit without it being divided into smaller pieces. Each link type (e.g., Ethernet) has its own MTU limit—common values range from 576 bytes (conservative WAN default) to 9,000+ bytes (jumbo frames on high-speed LANs).
  - A low MTU like 500 bytes might occur in specialized scenarios, such as certain VPN tunnels
  - Recall TCP's MSS, MSS occurs on the transport layer while MTU occurs on the network layer
  - Like TCP header, a standard IPv4 header has 20 bytes
  - **Non-final fragments** must carry payloads that are multiples of 8 bytes

# IP fragmentation, reassembly

*example:*

- ❖ 4000 byte datagram
- ❖ MTU = 1500 bytes

1480 bytes in  
data field

offset =  
 $1480/8$

	length =4000	ID =x	fragflag =0	offset =0	
--	-----------------	----------	----------------	--------------	--

*one large datagram becomes  
several smaller datagrams*

	length =1500	ID =x	fragflag =1	offset =0	
--	-----------------	----------	----------------	--------------	--

	length =1500	ID =x	fragflag =1	offset =185	
--	-----------------	----------	----------------	----------------	--

	length =1040	ID =x	fragflag =0	offset =370	
--	-----------------	----------	----------------	----------------	--

# Problem 2

- Total Payload to Fragment: 1,580 bytes.
- Payload per Non-Final Fragment: 480 bytes (max possible).
  - Number of Full Fragments Needed:  $\text{floor}(1,580 / 480) = 3$  (covering  $3 \times 480 = 1,440$  bytes).
  - Remaining Payload:  $1,580 - 1,440 = 140$  bytes (for the 4th fragment).
  - Total Fragments Generated: 4 (3 full + 1 partial).
- Each fragment will have Identification number 291 (x). Each fragment except the last one will be of size 500 bytes (including IP header). The last datagram will be of size 160 bytes (including IP header). The offsets of the 4 fragments will be 0, 60, 120, 180. Each of the first 3 fragments will have flag=1; the last fragment will have flag=0.

# Problem 2

Fragment #	Total Length (bytes)	Payload Length (bytes)	Frag Flag (MF)	Offset (8-byte units)	Actual Byte Offset	Explanation
1	500	480	1	0	0	First fragment; starts at beginning of original payload. Full max size.
2	500	480	1	60	480	$\text{Offset} = 480 / 8 = 60$ . Continues from Frag 1.
3	500	480	1	120	960	$\text{Offset} = (480 \times 2) / 8 = 120$ . Continues from Frag 2.
4	160	140	0	180	1,440	$\text{Offset} = (480 \times 3) / 8 = 180$ . Last fragment (MF=0); remainder fits easily under MTU.

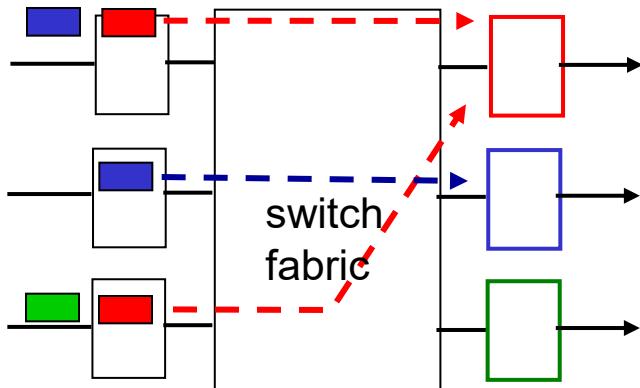
# Problem 3

Head-of-Line blocking: In packet-switched networks, such as routers or switches, a packet that is first in line at an input port queue must wait because there is no available buffer space at the output port to which it wants to be forwarded.

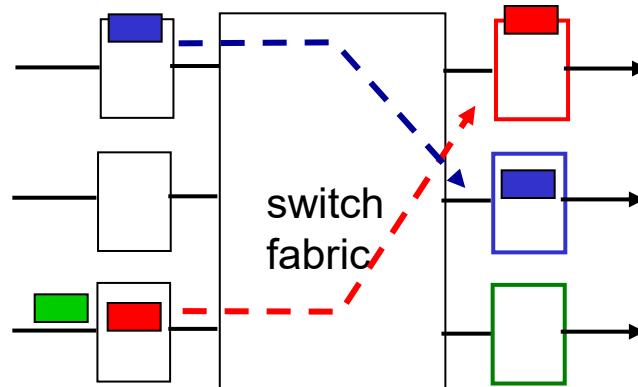
HOL blocking occurs at the input port. Packets arrive at input ports and are queued there before crossing the switching fabric to output ports.

# Input port queuing

- Switch fabric often slower than input ports combined -> queueing may occur at input queues
  - *Result: Queuing at input ports → Delays (packets wait longer) and Losses (if input buffers overflow, packets are dropped).*
- Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward



output port contention (busy):  
only one red datagram can be transferred.  
*lower red packet is blocked*



one packet time  
**later:** green packet  
still experiences HOL  
blocking

# Problem 4

## Forwarding table

<u>Destination Address Range</u>	<u>Link Interface</u>
11100000 00000000 00000000 00000000 through 11100000 11111111 11111111 11111111	0
11100001 00000000 00000000 00000000 through 11100001 00000000 11111111 11111111	1
11100001 00000001 00000000 00000000 through 11100001 11111111 11111111 11111111	2
otherwise	3

# Problem 4a

<b>Prefix Match</b>	<b>Link Interface</b>
• 11100000	0
• 11100001 00000000	1
• 11100001	2
• otherwise	3

# Problem 4b

11001000 100010001 01010001 01010101

Prefix match for first address is 4<sup>th</sup> entry

-> link interface 3

11100001 00000000 11000011 00111100

Prefix match for second address is 2<sup>nd</sup> entry

-> link interface 1

11100001 10000000 00010001 01110111

Prefix match for first address is 3<sup>rd</sup> entry

-> link interface 2