Course: Communication Networks
Date: 10/23
Topics Covered: HW #4

- MATTHEW KRUEGER -

1. Hosts X and Y, with receive windows of size 512 and 2048 bytes, are exchanging messages. At time t_0 host Y sends X a packet with

E_0
$$t = t_0 : Y \to X \text{ [Seq\# = 100, Ack\# = 2000, Win = 2048, No Data]}.$$

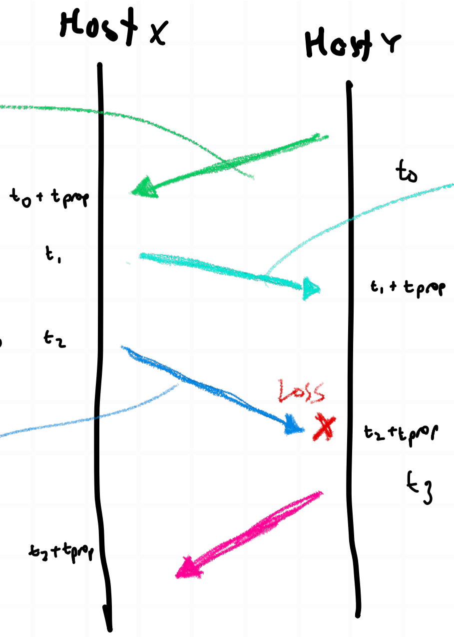
In a diagram like Fig. 3.36, sketch this exchange as well as the exchanges arising from the following events given that data is sent as soon as possible. You can assume that the link's one way delay is t_{prop}–and that host X's receive buffer is empty.

- E_1 At $t_1 > t_0 + t_{prop}$: 1024 bytes of data arrive (from the application layer) at X and are sent to Y.
- E_2 At $t_2 > t_1$: 1024 bytes of data arrive at X and are sent to Y; however, the packet is lost en route.
- E_3 At $t_3 > t_2 + t_{prop}$: Y removes 512 bytes of data from its receive buffer; 128 bytes of data arrive at Y and are sent to X.

In each case, please note the time, direction of transfer, Seq. #, Ack. #, Win, and amount of data carried in the segment. Fig. 3.33's sender logic may be helpful!

Event 0: Y → X
- Seq #: 100
- Ack #: 2000
- Win: 2048
- bytes: None

* Host Y sends a segment to X Acknowledgment only package



Event 1: $t_1 > t_0 + t_{prop}$; X → Y
- Seq #: 2000
- Ack #: 100
- Win: 512
- bytes: 1024

* X sends a segment to Y that carries 1024 bytes of data

Event 2: $t_2 > t_1$; X → Y
- Seq #: 3024
- Ack #: 100
- Win: 512
- bytes: 1024

* X sends data to Y, but packet is lost en route.

Event 3: $t_3 > t_2 + t_{prop}$; Y → X
- Seq #: 100
- Ack #: 3024
- Win: 1536
- bytes: 128

* Y removes 512 bytes from its receive buff
128 arrive at Y from application layer

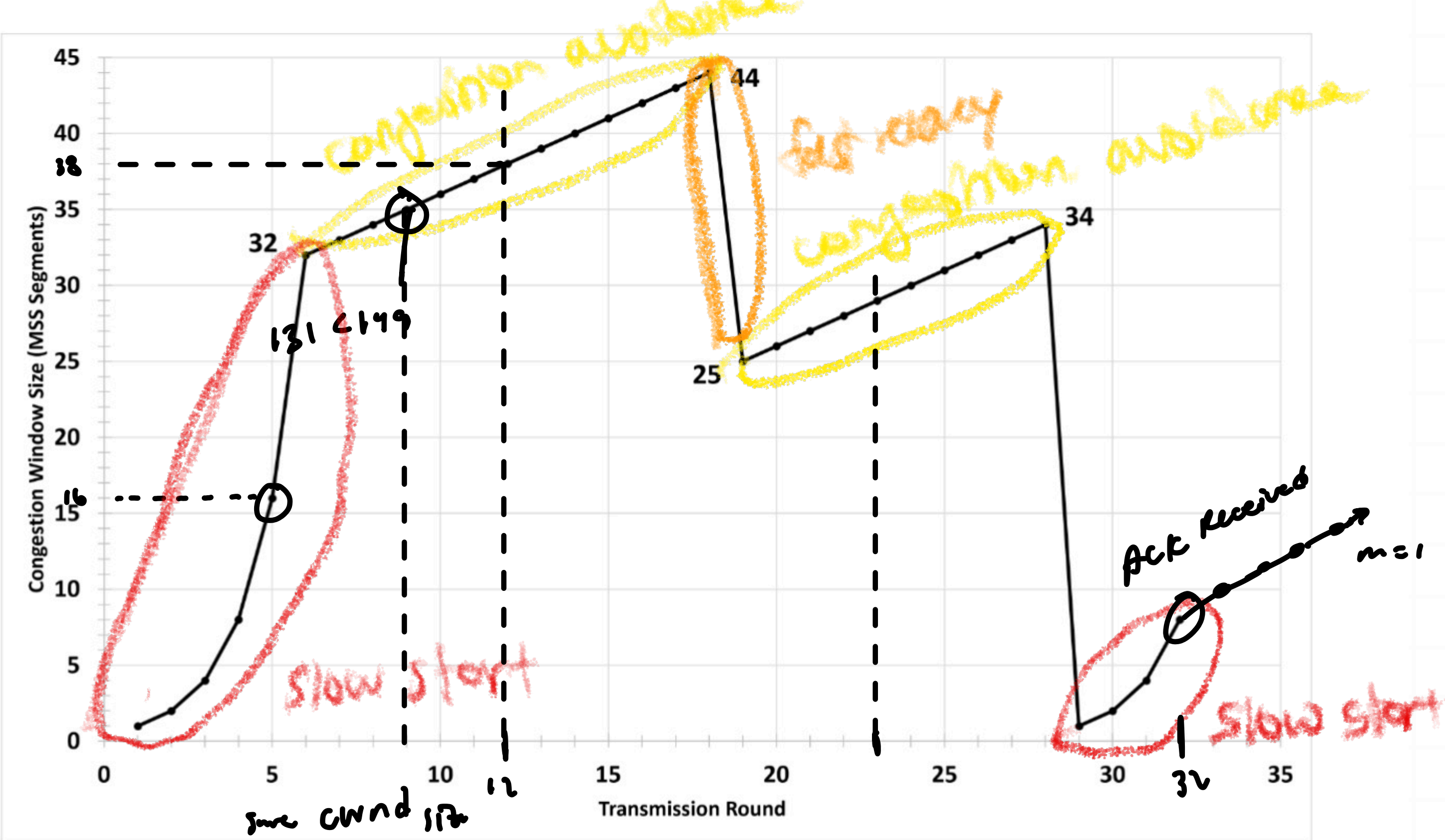
Course: Communication Networks

Date: 10/23

Topics Covered: HW #4

- MATTHEW KRUEGER -

2. The congestion window size, in MSS segments, of the TCP Reno protocol is plotted below vs. the transmission rounds for a sequence of transmissions. Using text Fig. 3.51 and Fig. 3.52:



(a) Identify the transmission round interval(s) over which TCP is in **slow start**.

(b) Identify the transmission round interval(s) over which TCP is in **fast recovery**.

(c) Identify the transmission round interval(s) over which TCP is in **congestion avoidance**.

(d) Identify the cause of the congestion window size drop after the 18th transmission round.

(e) Identify the cause of the congestion window size drop after the 28th transmission round.

(f) Determine the value of the congestion window threshold (ssthresh) during Round 12.

(g) Determine the value of the congestion window threshold (ssthresh) during Round 23.

(h) During what transmission round is the 131st segment sent?

(i) Given that a new ACK is received after completion of the 32nd round, determine the congestion window threshold and congestion window size of the 33rd round.

d) : Most Probable

Triple duplicate Acks
- packet loss detected by receipt of three duplicate Acks
- congestion window reduced by half

e) Timeout event

Timeout
- no Ack received within retransmission timeout
- congestion window reset to 1 max segment size

f) 32 MS ; TCP initializes ssthresh to 1 upon connecting

g) 22 MSS ; 44/2 = 22

h) transmission round 9

i) cwnd 16 ssthresh 17

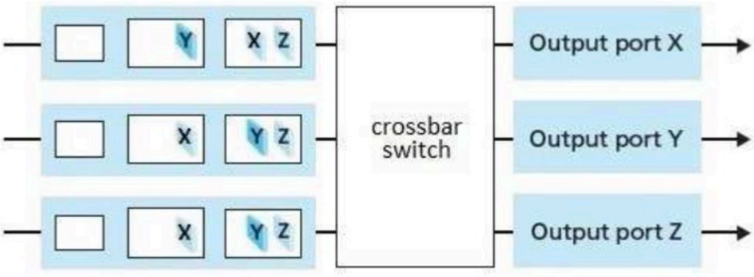
Ack received therefore slow start → congestion avoidance (m=1) ∴ 8·2 = 16

last cwnd/2 → 34/2 ∴ 17

Course: Communication Networks
Date: 10/23
Topics Covered: HW #4

- MATTHEW KRUEGER -

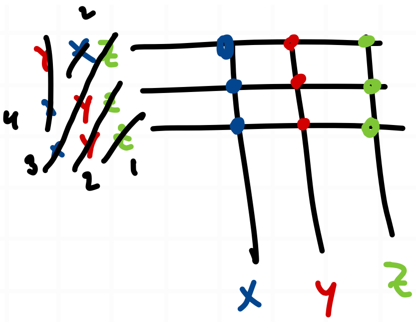
3. This question illustrates the impact that a switch's queuing architecture can have on switch performance. Suppose that fixed length packets headed for output ports X, Y, and Z of a **crossbar switch** are queued as shown below. As explained in Sec. 4.2, due to the switch's construction, at most one packet can be transferred to a given output port in a single time slot, but transfers to different output ports may occur simultaneously.



(a) **Assuming that in each slot only datagrams at the head of each queue can be transferred**, determine the minimum number of time slots needed to transfer the packets shown from the input ports to the output ports (assuming optimal scheduling) and list, in order, the transfers you used to achieve this minimum.

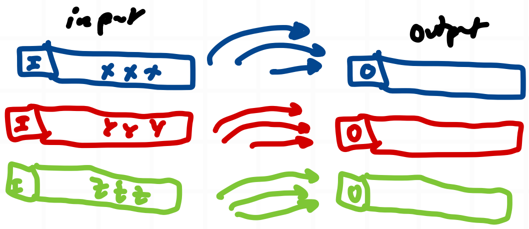
(b) **Assuming that the virtual queuing approach to avoiding head-of-the-line blocking has been applied at each queue** (as discussed in lecture), determine the minimum number of time slots needed to transfer the packets shown from the input ports to the output ports (assuming optimal scheduling) and list, in order, the transfers you used to achieve this minimum.

a)



Moving only the head at a time,
A minimum of $\boxed{5 \text{ timeslots}}$

b) Head of the line blocking



with VC Head of the line blocking
a minimum of $\boxed{3 \text{ timeslots}}$
are needed

Course: Communication Networks

Date: 10/23

Topics Covered: HW #4

- MATTHEW KRUEGER -

4. A TCP payload with 10,000 bytes (9980 bytes of data and 20 bytes of TCP header) is passed to IP for delivery across a network that has an MTU of 4000 bytes. Assuming that each IP datagram will have 20 bytes of header, list the length, offset, and fragmentation flags ([length= , offset= , flag=]) of each fragment delivered by the network layer at the destination host. *(Recall that IPv4 headers allocate only 13 bits to the offset field, so offsets are counted in whole multiples of 8 bytes).*

TCP payload : 9980 data + 20 header = 10000 bytes

Network maximum transmission unit = 4000 bytes

IP header size: 20 bytes per fragment

Fragment offsets units: multiples of 8 per IPv4

maximum data = MTU - IP header = $\left\lfloor \frac{3980 \text{ bytes}}{8} \right\rfloor \cdot 8 = 372 \cdot 8 = \underline{2976}$

num fragments : $\left\lceil \frac{10000}{2976} \right\rceil = \lceil 3.36 \rceil = 4$ fragments

F1 :
- length: 2976+20
- off: 0
- flag: 1

F2:
- length: 2996
- off = $\frac{2976}{8} = 372$
- flag=1

F3:
- length : 2996
- offset : $\frac{2976+2976}{8} = 744$
- flag=1

F4:
- length : 1072 + 20 = 1092
- offset : $\frac{2976+2976+2976}{8} = 1116$
- flag= 0

Fragment 1 :
- length : 2996 bytes
- offset : 0 (first)
- flag : 1 (more frags)

Fragment 2 :
- length = 2996
- offset = 372
- MF = 1

Fragment 3 :
- length = 2996
- offset = 744
- mf = 1

Fragment 4 :
- length = 1092
- offset = 1116
- flag=0

Course: Communication Networks

Date: 10/23

Topics Covered: HW #4

- MATTHEW KRUEGER -

5. You are the manager of a company in need of 1000 new IP addresses. A local IP provider, Goss IP, sells CIDR blocks of address space. To minimize cost, you wish to buy the smallest number of addresses possible.

   (a) If you buy a block of addresses from Goss IP, how many bits are there in the appropriate block's mask (e.g., is it a /8 block, a /22 block, etc.)? How many addresses are not used?

   (b) Goss IP allocates you a block of addresses that begin with address 128.212.80.0. You further subdivide this block into three subnets containing, respectively, 510, 254, and 236 addresses each. For each of these subnets, list the first and last IP addresses assigned (the IP range) and the subnet mask in CIDR notation.

   (c) The output of a "netstat -nr" command on a host in your network is shown below.

Dest/Netmask	Gateway	Flags	Interface
127.0.0.1/255.255.255.255	127.0.0.1	H	lo0
128.96.39.0/255.255.255.128	128.96.39.13		lan2
128.96.39.128/255.255.255.128	128.96.15.23	G	lan1
128.96.40.0/255.255.255.128	128.96.10.30	G	lan0
192.4.153.0/255.255.255.192	128.96.15.7	G	lan1
default/0.0.0.0	128.95.15.1	G	lan1

Indicate to which gateway and interface the router will forward a packet addressed to each of the following destinations: (i) 128.96.40.14, (ii) 128.96.40.184, and (iii) 192.4.153.17.

a) $2^n > 1000 \rightarrow 2^{10} = \boxed{1024}$; $\boxed{\text{10 bits appropriate with 24 unused}}$

mask $32 - 10 = /22$

b) 128.212.80.0

subnet a: 510 → $2^9 = 512$ → subnet mask : $32 - 9 = /23$

subnet b : 254 → $2^8 = 256$ → subnet mask : $32 - 8 = /24$

subnet c: 236 → $2^8 = 256$ → subnet mask: $32 - 8 = /24$

	Addresses needed	Block size	Subnet Mask	First IP	last IP
Subnet a	510	512	/23 ; 255.255.254.0	128.212.80.0	128.212.81.255
subnet b	254	256	/24 ; 255.255.255.0	128.212.82.0	128.212.82.255
subnet c	236	256	/24 ; 255.255.255.0	128.212.83.0	128.212.83.255

c)

H : host

G : use gateway

Destination	Gateway	interface
128.96.40.14	128.96.10.30	lan 0
128.96.40.184	128.95.15.1	lan 1
192.4.153.17	128.96.15.7	lan 1