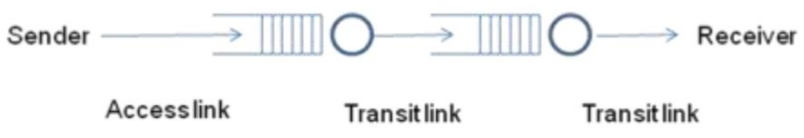
COL 334 - Assignment 3

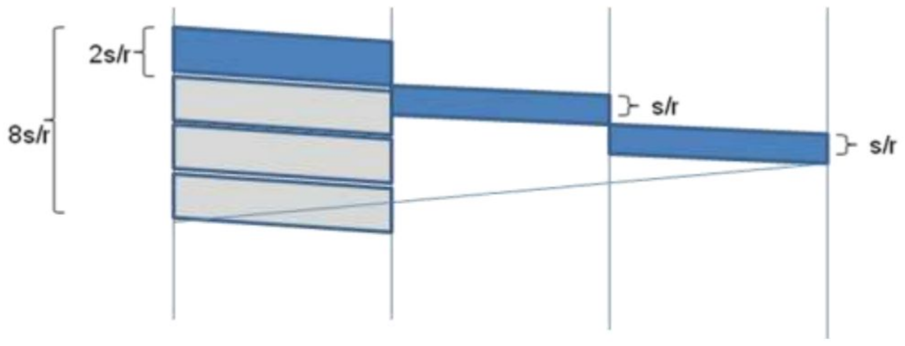
Akshay Kumar Gupta **Barun Patra** **Haroun Habeeb**

2013CS50275 **2013CS10773** **2013CS10225**

QUESTION I

1. We have the following topology: a sender communicating to a receiver via a series of two routers. Packets are of size s, the transit links have a transmission rate of r, while the access link operates at half the rate of the transit links. The round trip propagation delay is 4s/r, hence within an RTT of 8s/r the access link can just about support a window size of 4 packets. Ignore acknowledgement sizes.

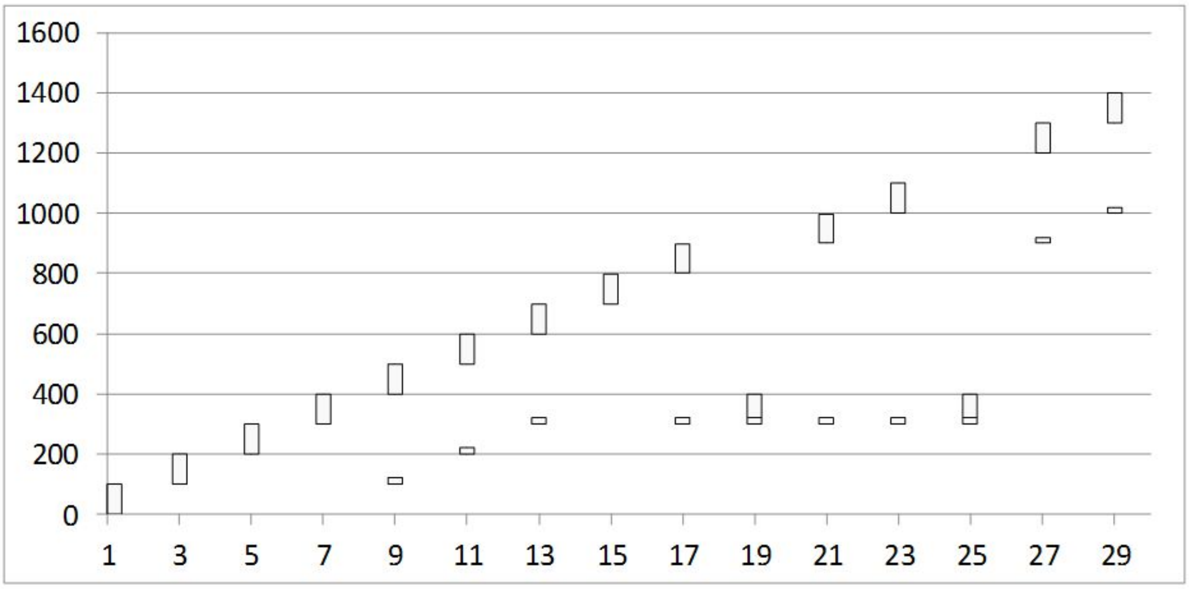




Consider the following packet trace at the sender using a transport protocol similar to TCP. The y-axis indicates sequence numbers in bytes – long vertical rectangles are packets and each packet is 100 bytes long, thus the first packet contains data from sequence number 0 to 99, the second packet from sequence number 100 to 199, etc. The x-axis indicates time in units of s/r. The small stubs are acknowledgement numbers – thus, the stub at time 9 (after one RTT) is the cumulative acknowledgement for the first packet with sequence number 0 to 99, the stub at time unit 11 is the cumulative acknowledgement for the second packet, etc.

Assume the congestion window size to be fixed and greater than 4 packets – this implies that the sender will try to push out a packet every 2 time units, which is the maximum transmission rate its access link allows.

Also assume for simplicity that no acknowledgements are lost and no reordering occurs. Now explain the packet trace below.



i. **Packet 300-399 seems to have been lost. What triggers the retransmission of the lost packet?**

A. Acknowledgements with sequence number as 300 are sent thrice by the receiver. This triggers a fast retransmit of the packet 300-399.

ii. **The acknowledgement received at time 21 was generated at the receipt of which packet at the receiver?**

A. It was generated by packet with sequence number 600-699.

iii. **Why is the acknowledgement at time 21 still referring to the lost packet even though it has been retransmitted?**

A. The receiver has not yet received the packet 300-399 even though it has been retransmitted. Thus the acknowledgement at time 21 still refers to that packet.

iv. **Why is the lost packet again retransmitted at time 25?**

A. After the first retransmits, three more ACKs with sequence number 300 are sent by the receiver. This causes another retransmit of the lost packet.

v. **Why is there a sudden jump in the acknowledgement number at time 27? The acknowledgement was generated at the receipt of which packet?**

A. By time 26, the receiver has received every packet sent by the sender till sequence number 899 except for 300-399. Thus at time 26, the receiver was still sending an ACK with sequence number 300. At time 27, it receives packet 300-399, which causes the receiver to generate a cumulative ACK for packets till 899, so the sequence number it sends back is 900. So there is a jump in acknowledgement number from 300 to 900 at time 27.

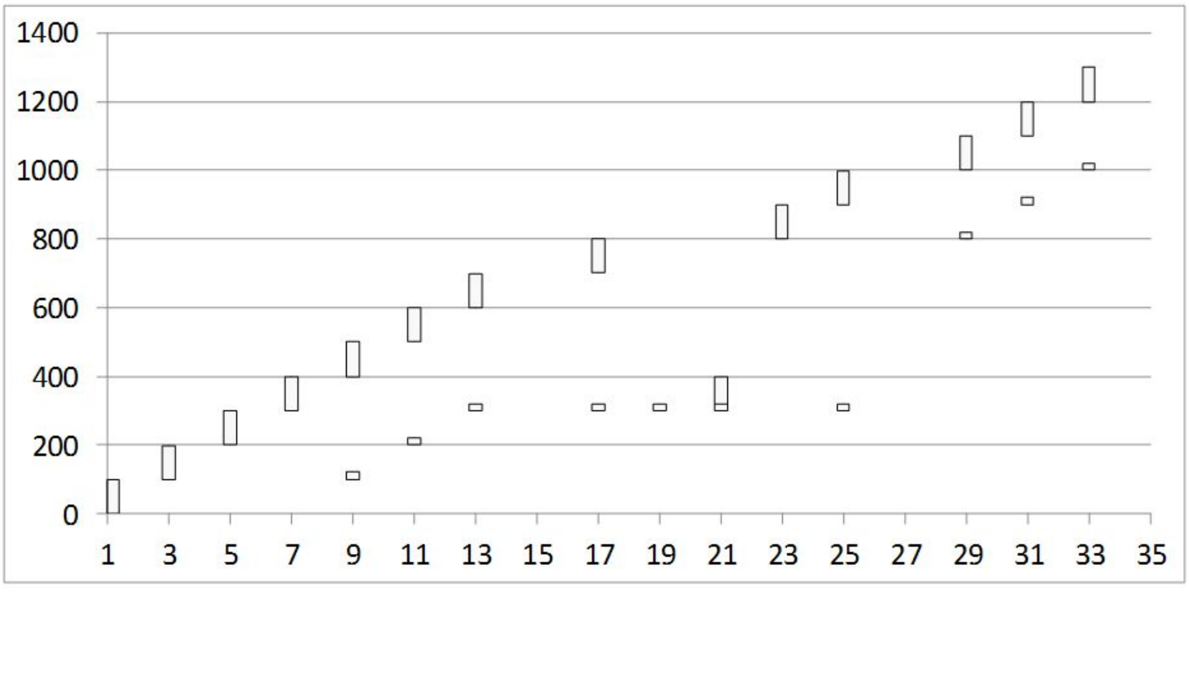
QUESTION II

2. In another variant, the initial congestion window size is started at 4 packets, and the window is incremented fractionally by (1 / int(current window size)) upon receiving an acknowledgement. Thus, a window size of 4 will increment to 5 after having received 4 acknowledgements (4.25 after the first ack, 4.5 after the second ack, 4.75 after the third, and 5 after the fourth ack). Note that packets are dispatched only if they can be accommodated fully within the window, ie. even with a window of 4.75 only four outstanding packets will be allowed.

The window size also reduces to half upon receiving triple duplicate acknowledgements which is seen as an evidence of packet loss. Note that receipt of the third dup ack will not increment the window, ie. if the window is 5 when the third dup ack arrives, it will just be reduced to 2.5, and not add another 1 / int(2.5) increment for this ack as is done for other acks. Also note that the event of a triple dup ack is also interpreted as a loss, and hence the number of outstanding packets will be assumed to be one less than what was it estimated to be earlier. This is almost identical to TCP operations in the congestion avoidance phase.

Answer the following questions.

Hint: Mentally maintain two variables for congestion window and the outstanding data to understand what is happening.



i. **What is the window size at time 17?**

A. Window size at time 17 is 5 because 4 ACKs have been sent by time 17.

ii. **What is the window size at time 19? Why is no packet pushed out at time 19? What is the window size at time 21? What is the outstanding data estimated by the sender at time 21?**

A. Window size at time 19 is 2.5. No packet is pushed out at time 19 because from the sender's point of view there are 2 unacknowledged packets in the stream at time 19 so sending out another packet would exceed the window size of 2.5. The window size at time 21 is 3. The outstanding data estimated by sender at time 21 is 2.

iii. **Why is a packet pushed out at time 23 even though no ack is received at that time?**

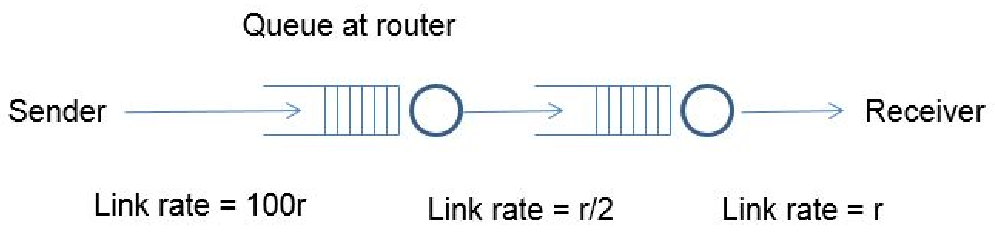
A. The outstanding data estimated by sender before time 23 is 2 while the window size is 3. So the sender pushes out a packet at time 23 because doing so does not exceed the window size.

iv. **What is the window size at time 31?**

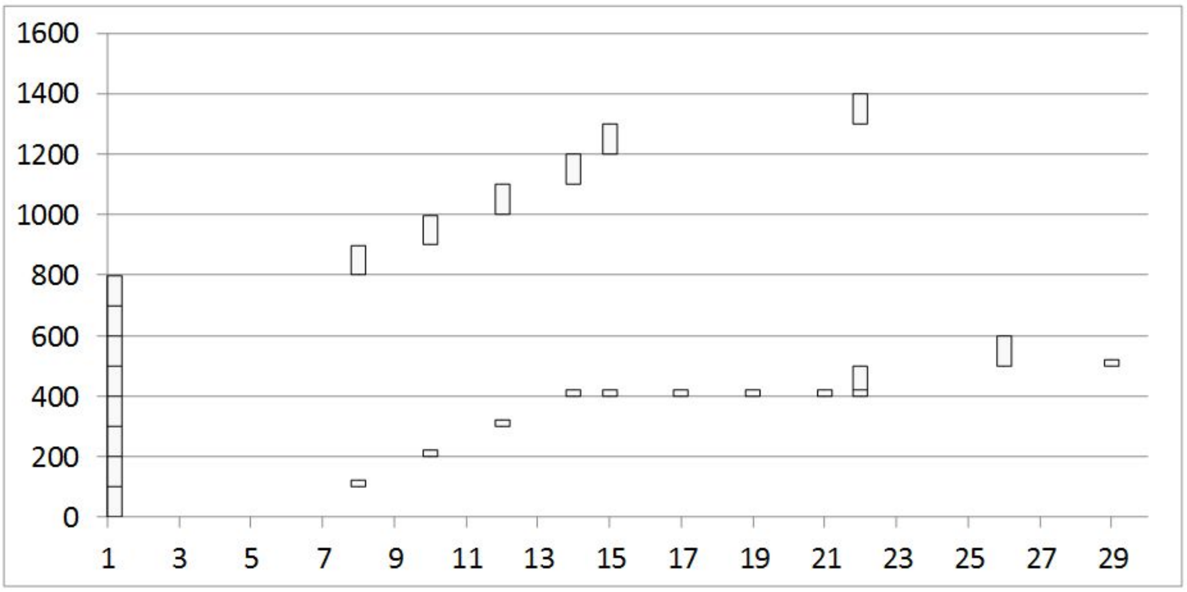
A. Window size at time 31 is 4.

QUESTION III

3. Now consider a different scenario where the access link is very fast but the next link is slower.



Assume an initial window size of 8 this time. As in the previous question, the window size is reduced to half upon receiving triple duplicate acknowledgements, and it is incremented by 1 / int(congestion window) upon receiving an acknowledgment. A timeout occurs if the last unacked packet goes unacknowledged for more than 25 time units. The buffer size at the first router is limited, and it follows a drop-tail policy. Assume that all packet losses happen due to buffer overflow at the first router. Answer the following questions:



i. **What is the RTT in this case?**

A. The link rates effect transmission delay. We assume that the propagation delay is the same as that in Q1, i.e., $4\frac{s}{r}$. The transmission delay in this case would be $\frac{s}{100r} + \frac{s}{0.5r} + \frac{s}{r} = 3.01\frac{s}{r}$. Hence, RTT becomes $7.01\frac{s}{r}$. Note, we assume that transmission delay for ACK is 0 since the packet size of an ACK is negligible. This is a fair assumption since its the same as what happens in Q1.

ii. **Can you infer the buffer size at the first router? How?**

A. Since all drops occur due to buffer overflows at the first router, and the first link rate is far greater than the other link rates, we will look for the first repeated ACK. The first repeated ACK is at time 15, and from it we infer that packet 400-499 was lost. Hence, the 5th packet caused a buffer overflow, which means that the buffer size at the first router is 4.

iii. **Why is there no retransmission at time 17 despite a triple dup ack? Why does this retransmission happen later at time 22? Why do two packets get fired off at time 22?**

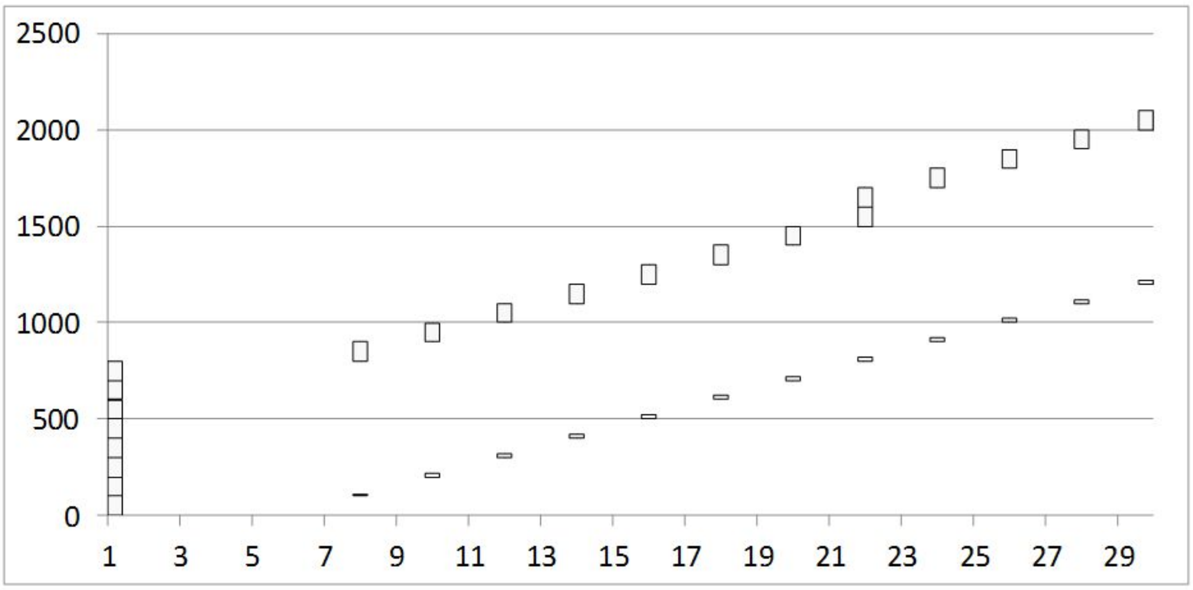
A. This is due to the number of packets in the network being larger than the congestion window. At time 17, the third duplicate ACK is received which causes the window size to be cut in half. Up until time 17, 5 ACKs have been received, hence window size was 8.625 . Upon receiving 3rd duplicate ACK, the window size becomes 8.625/2 = 4.3125. At this point, from the point of view of the sender, there are 6 packets in the network. This is because it has sent out 13 packets, and received 5 ACKs and the third duplicate ACK at time 17. Hence, number of active packets = 13 - 5 -1 - 1 = 6 > 4.3125. Since the number of active packets is greater than the congestion window size, the sender doesn't transmit. At time t=22, the window size equals 5.0625, while the number of outstanding packets equals 3. Thus, the sender can send two packets and not overflow the window size. Hence, it sends packets 400-499 (the last requested packet) and 1300-1399 (the packet following the last sent packet).

iv. **When did a timeout occur? Which packet gets timed out?**

A. A timeout first occurs at time 26. All the packets sent at time 1 get timed out. i.e., packets with sequence numbers 500-599, 600-699, 700-799. However, due to the congestion window size restricting the packets it can transmit, the sender transmits the packet with sequence number 500-599.

QUESTION IV

4. Consider now a scenario where the buffer size at the first router is very large so that no drops occur. Answer the following questions:



i. **What is the round trip time clocked for the first packet? For the fifth packet? For the eighth packet?**

A. RTT for 1st packet - 7s/r, RTT for 5th packet - 15s/r, RTT for 8th packet - 21s/r

ii. **When is the window size increased to 9?**

A. At time = 22. That is when the 8th ACK is received. The consequence is that two packets are transmitted at that point.

iii. **Since the buffers are assumed to be large enough, there will be no drops and the window size will keep increasing each time a complete round of acknowledgments for the current window are received. What is the problem with such a scenario?**

A. The queueing delay at a buffer increases as the number of packets in the buffer increases. If the congestion window size keeps increasing, the sender pumps more packets into the buffer. The buffers queue increases in size and consequently, the RTT increases and hence, the network becomes slower and more congested.

iv. **Suggest a method to trigger a window size reduction in such a scenario, without witnessing any loss events.**

A. If the RTT exceeds a certain threshold(timeout threshold) , then reduce/ keep constant the window size.

QUESTION V

5. Show that in a steady state TCP connection working in the congestion avoidance phase, the throughput $\sim 1.22\frac{MSS}{RTT\sqrt{L}}$, where *RTT* is the round trip time, *MSS* is the maximum segment size, and *L* is the loss rate.

Note that in the congestion avoidance phase, all losses are assumed to be detected through fast retransmits and not timeouts, hence the congestion window rises additively and falls to half its value in a saw-tooth pattern.

Hint: If N packets are sent between two consecutive packet loss events, assume that the events happen due to the loss of only one packet in each event, hence the loss rate can be written as 1/N.

A. Let *W* be the congestion window size when a loss event happens. Then, the window size drops to *W*/2. Window size then increases until *W* and then drops again. As the window size increases linearly from *W*/2 to *W*, the average throughput,

$$S = 0.75W\frac{MSS}{RTT}$$

Now in 1 cycle (from *W*/2 to *W*), one packet loss occurs. The number of packets sent in 1 cycle is

$$\frac{W}{2} + \left(\frac{W}{2}+1\right) + \left(\frac{W}{2}+2\right) + \ldots + W = \frac{3}{8}W^2 + \frac{3}{4}W \sim \frac{3}{8}W^2$$

Hence,

$$L \approx \frac{1}{\frac{3}{8}W^2} \text{ where } L \text{ is the loss rate.}$$

$$\Rightarrow W \approx \sqrt{\frac{8}{3L}}$$

This is true because $L = 1/N$ where *N* is number of packets sent between consecutive loss events.
Substituting W in the original equation, we get,

$$S \approx 1.22\frac{MSS}{RTT\sqrt{L}}$$

QUESTION VI

6. Two TCP flows are running through the same bottleneck router through a shared buffer, and competing with each other. When the buffer gets congested, it drops a packet each from both the flows, and both the flows halve their windows simultaneously. The RTT of the first flow is half the RTT of the second flow. Show that the first flow settles at twice the bandwidth of the second flow at steady state.
Hint: Start with writing recursive equations for the window sizes of each flow in terms of the ith congestion event.
$W^1_{i+1} = (W^1_i + T) / 2$
$W^2_{i+1} = (W^2_i + T/2) / 2$

Where T is the number of transmission rounds between two consecutive loss events in which a window size amount of data is dispatched.

A. We will first derive the window size as a function of the number of loss events. Then, we will represent bandwidth as a function of window size and prove the desired results.

Let the window size right after the ith loss event be $W^1_i and\ W^2_i$. Let T be the number of transmissions by the first flow until the (i+1)th loss event. Since RTT of the first flow is half the RTT of the second, the second flow would've transmit $\frac{T}{2}$ times.
Since every transmit causes the window size to increase by 1, the window sizes just before the (i+1)th loss event are $W^1_i + T\ and\ W^2_i + \frac{T}{2}$. Upon the (i+1)th loss event, the window sizes becomes half. By our definition, the new window sizes are $W^1_{i+1} and\ W^2_{i+1}$.
Also,

$$W^1_{i+1} = 0.5 * (W^1_i + T)\ and\ W^2_{i+1} = 0.5 * (W^2_i + \tfrac{T}{2})$$

We now have the window sizes as a function of the number of loss events. We will now derive the bandwidth as a function of the number of loss events.
Also, let steady state start at n . Hence,

$$\forall i \geq n \Rightarrow W^j_i = W^j_{i+1}$$

For convenience of notation, we will show that the bandwidth of the first flow is double the bandwidth of the second when $i \rightarrow \infty$. This is equivalent to showing the same $\forall i \geq n$ for some n.

Let B^j_i be the bandwidth of j at time i. Since Bandwidth is $MSS * W$ and we can assume MSS to be the same for both flows, we can write $B^j_i = W^j_i$.
At steady state, $W^j_i = W^j_{i+1}$. Substituting these in the recursive formulae for W^j_i, we get,

$$\lim_{i\to\infty} W^1_i = T \text{ and } \lim_{i\to\infty} W^2_i = \tfrac{T}{2}$$

Hence, $\lim_{i\to\infty} W^1_i = \lim_{i\to\infty} 2 * W^2_i \Rightarrow \lim_{i\to\infty} B^1_i = 2 * \lim_{i\to\infty} B^2_i$, i.e, The first flow has a bandwidth that is twice the bandwidth of the second flow at the steady state.