Name_____ Student ID_____ Department/Year_____

2nd Examination

Introduction to Computer Networks (Online)
Class#: EE 4020, Class-ID: 901E31110
Fall 2022

10:20-12:10 Wednesday
November 17, 2022

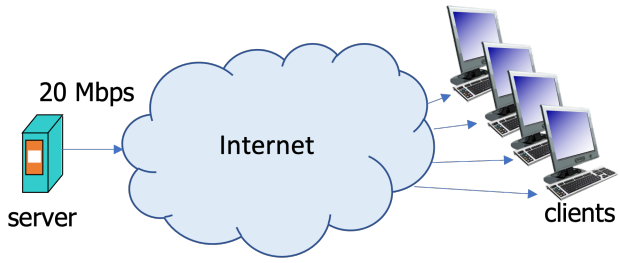
Cautions

1. There are in total 200 points to earn. You have 90 minutes to answer the questions. Skim through all questions and start from the questions you are more confident with.
2. Use only English to answer the questions. Misspelling and grammar errors will be tolerated, but you want to make sure with these errors your answers will still make sense.

1. (ch26, 4pt) Below is the video resolution to bitrate table for H.264. 360p is the minimum resolution possible.

Resolution	Bitrate
2160p (4K)	16 Mbps
1440p (2K)	9.6 Mbps
1080p	5 Mbps
720p	2.5 Mbps
480p	1.2 Mbps
360p	800 Kbps

Consider a network as follows. In that, a server and 4 clients are connected to the Internet via their own access networks. Let's assume the bottleneck is always the access link between the server and the Internet and the uplink bandwidth is 20 Mbps.



The 4 clients begin to request a video one after another. According to DASH, which is a greedy mechanism, the 1st client will get the 2160p video and the available bandwidth will be 4 Mbps. The 2nd client will get 720p video and the remaining bandwidth will be 1.5 Mbps.

(1) What will the 3rd client's video resolution be? (1pt)
(2) How much available bandwidth will remain after the 3rd client's request? (1pt)
(3) What will the 4th client's video resolution be? (1pt)
(4) How much available bandwidth will remain after the 4th client's request? (1pt)

Sample Solution:

(1) 480p
(2) 0.3 Mbps
(3) can't get video (or some resolution lower than 360p)
(4) 0.3 Mbps (not sure)

2. (ch26, 4pt) Continue from Problem Set 1. The resolution-bitrate table and the server-client network remain the same, and the 4 clients begin to request a video one after another. We use the DASH-fair mechanism for a change. It is greedy and fair at the same time. All ongoing videos' resolutions are the same and the total bandwidth consumption

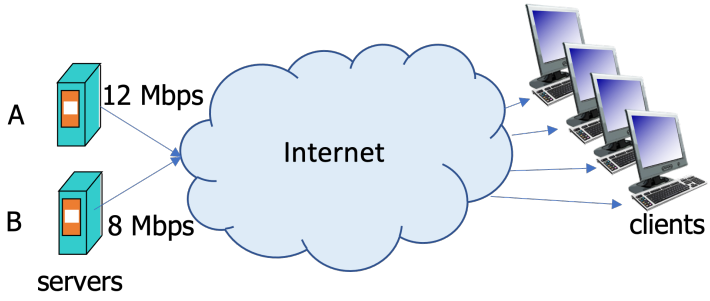
needs to be below 20 Mbps. When the 1st client requests a video, it receives a 2160p video and the remaining available bandwidth is 4 Mbps. Subsequently, after the 2nd client requests a video, both the 1st and 2nd client receive the same 1440p video and the remaining available bandwidth is 0.8 Mbps.

(1) What will the videos' resolution be after the 3rd client requests a video? (1pt)
(2) How much available bandwidth will remain after the 3rd client's request? (1pt)
(3) What will the videos' resolution be after the 4th client requests a video? (1pt)
(4) How much available bandwidth will remain after the 4th client's request? (1pt)

Sample Solution:

(1) 1080p
(2) 5 Mbps
(3) 1080p
(4) 0 Mbps

3. (ch26, 6pt) The resolution-bitrate table remains the same as Problem Set 1. The server-client network is similar except for the server end. There are now 2 servers connected to the Internet and the bottleneck uplink bandwidth is 12 and 8 Mbps respectively.



The 4 clients begin to request a video one after another. We use the DASH-balance mechanism. This mechanism is greedy first choosing the server with the highest available bandwidth and greedy again requesting the best resolution just below the available bandwidth. When the 1st client requests a video, it chooses server A and receives the 1440p video. The remaining available bandwidth is 2.4 Mbps. Subsequently, after the 2nd client requests a video, the 2nd client receive a 1080p video from server B and the remaining available bandwidth is 3 Mbps.

(1) Which server will the 3rd client request the video from? (1pt)
(2) What will the 3rd client's video resolution be? (1pt)
(3) How much bandwidth will remain in server A and B after the 3rd client's request? (1pt)
(4) Which server will the 4th client request the video from? (1pt)
(5) What will the 4th client's video resolution be? (1pt)

(6) How much bandwidth will remain in server A and B after the 4th client's request? (1pt)

Sample Solution:

(1) B
(2) 720p
(3) 2.4 Mbps, 0.5 Mbps
(4) A
(5) 480p
(6) 1.2 Mbps, 0.5 Mbps

4. (ch26, 12pt) Continue from Problem Set 1, 2, and 3 and compare the DASH, DASH-fair, and DASH-balance mechanism. Grading policy: pts will be credited only when Problem Set 1-3 are completed and the justifications make sense.
   (1) Between DASH and DASH-fair which one will allow more video streams and why? (2pt)
   (2) Between DASH and DASH-fair which one will be easier to implement and why? (2pt)
   (3) Between DASH and DASH-balance which one will allow more video streams and why? (2pt)
   (4) Between DASH and DASH-balance which one will be easier to implement and why? (2pt)
   (5) Between DASH-fair and DASH-balance which will allow more video streams and why? (2pt)
   (6) Between DASH-fair and DASH-balance which one will be easier to implement and why? (2pt)

Sample Solution:

Anything that's genuinely from you and makes sense

5. (ch26, 6pt) There are in general two types of CDN.
   (1) Describe the CDN that is the enter deep type. (2pt)
   (2) Describe the CDN that is the bring home type. (2pt)
   (3) Compare and contrast the two types of CDN. (2pt)

Sample Solution:

Anything that's genuinely from you and makes sense

6. (ch26, 6pt) There are in general two ways to redirect a video request to a CDN.

(1) Describe how a video request is redirected via DNS. (2pt)

(2) Describe how a video request is redirected via a cloud service (such as Netflix via Amazon). (2pt)

(3) Compare and contrast the two ways of request redirection. (2pt)

Sample Solution:

Anything that's genuinely from you and makes sense

7. (ch3, 8pt) For each of the functions below, tell whether it is implemented in (a) TCP, (b) UDP, (c) both, or (d) none.

   (1) bit error detection (1pt)

   (2) packet loss detection (1pt)

   (3) bit error recovery (1pt)

   (4) packet loss recovery (1pt)

   (5) multiplexing (1pt)

   (6) demultiplexing (1pt)

   (7) byte counting (1pt)

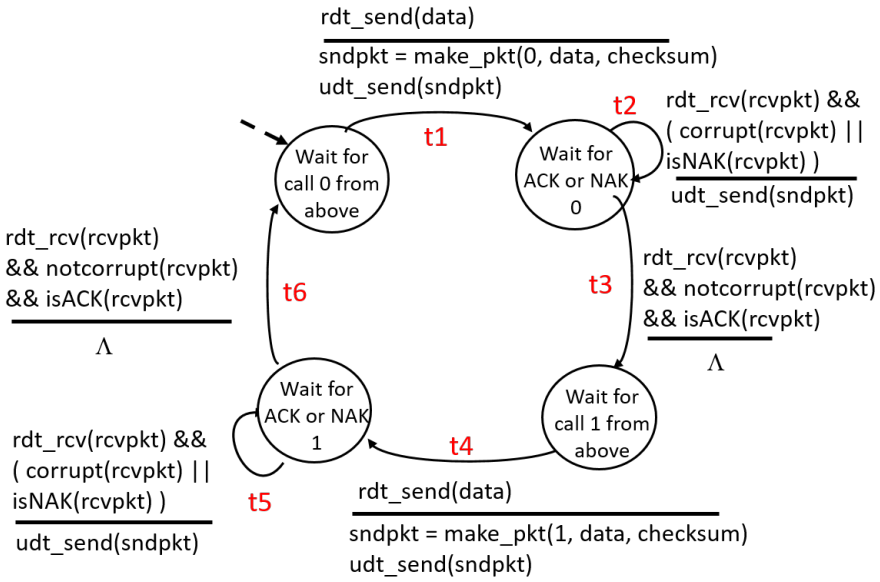
   (8) security (1pt)

Sample Solution:

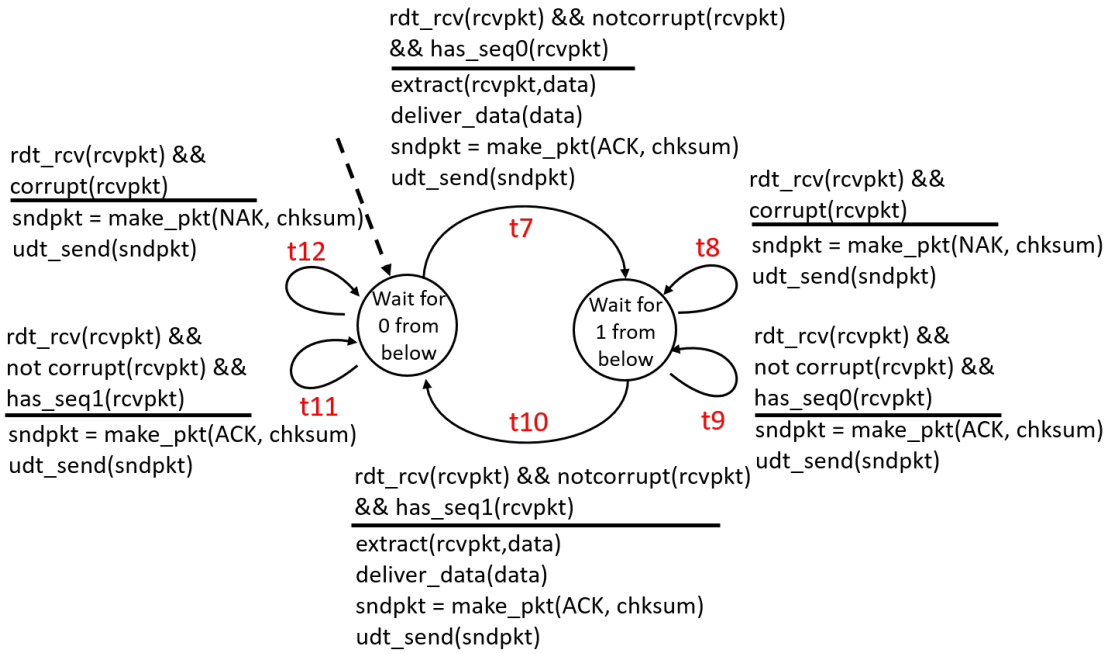
(1) c
(2) a
(3) a
(4) a
(5) c
(6) c
(7) a
(8) d

8. (ch34, 7pt) Below is the FSM of rdt 2.1. Tell the sequence of transitions for the following scenarios.

   <u>rdt 2.1 sender:</u>



rdt 2.1 receiver:



(1) Both the sender and receiver start from the initial state. The sender sends one packet. There are no bit errors at all. (1pt)

(2) Continue from (1). The sender sends another packet. There are no bit errors at all. (1pt)

(3) Continue from (2). The sender sends one more packet. There is a bit error in the packet and no more bit errors afterwards. (1pt)

(4) Continue from (3). The sender sends one more packet. There is a bit error in the NAK packet coming back and no more bit errors afterwards. (1pt)

(5) Continue from (4). The sender sends one more packet. There is a bit error in the ACK packet coming back and no more bit errors afterwards. (1pt)

(6) Why do we need to use sequence number 0 and 1 in the rdt 2.1's finite state machines? (2pt)

Sample Solution:

(1) t1, t7, t3

(2) t4, t10, t6

(3) t1, t12, t2, t7, t3

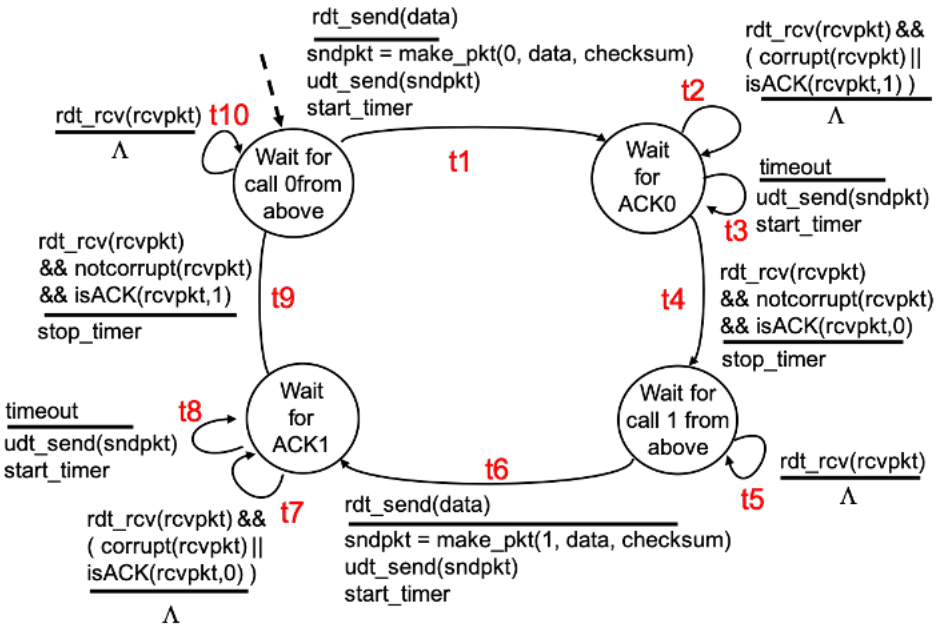
(4) t4, t8, t5, t10, t6

(5) t1, t7, t2, t9, t3

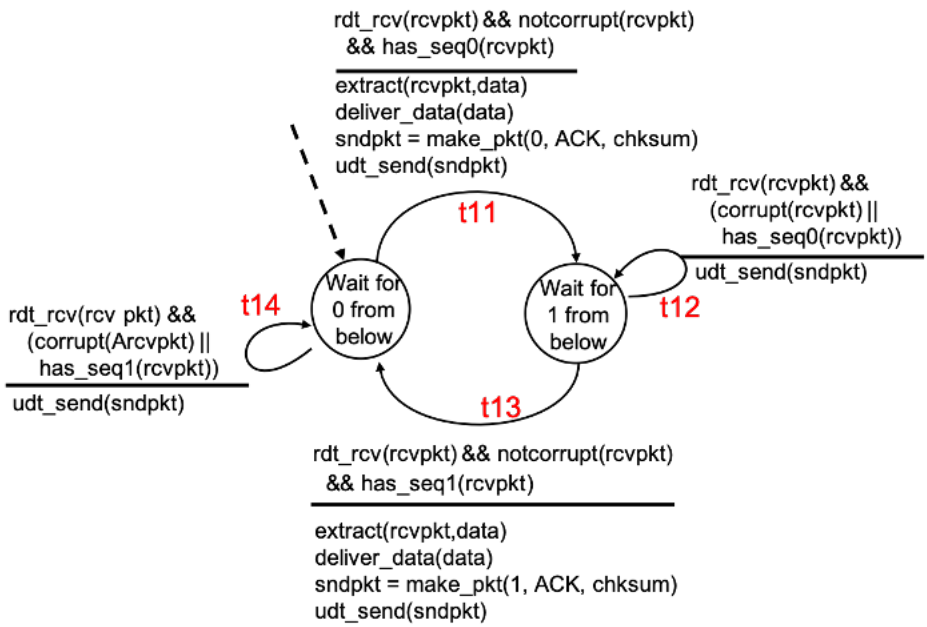
(6) There's a chance a packet is sent twice in rdt 2.1. To tell if a packet is a duplicate, we need a sequence number space that can tell the new and old packet apart. (avoid sending duplicate data to the application layer.)

9. (ch34, 7pt) Below is the FSM of rdt 3.0. Tell the sequence of transitions for the following scenarios.

rdt 3.0 sender



rdt 3.0 receiver



(1) Both the sender and receiver start from the initial state. The sender sends one packet. There are no bit errors or packet losses at all. (1pt)

(2) Continue from (1). The sender sends one more packet. There is a bit error in the packet and no more bit errors or packet losses afterwards. (1pt)

(3) Continue from (2). The sender sends one more packet. The packet is lost but no more bit errors or packet losses afterwards. (1pt)

(4) Continue from (3). The sender sends one more packet. There is a bit error in the ACK packet coming back and no more bit errors or packet losses afterwards. (1pt)

(5) Continue from (4). The sender sends one more packet. The ACK packet coming back is lost and no more bit errors or packet losses afterwards. (1pt)

(6) In rdt 3.0, sender FSM t2 does not do anything when the ACK packet received is corrupted. Do you like this? If yes, why? If not, why not? (2pt)

Sample Solution:

(1) t1, t11, t4

(2) t6, t12, t7, t8, t13, t9

(3) t1, t3, t11, t4

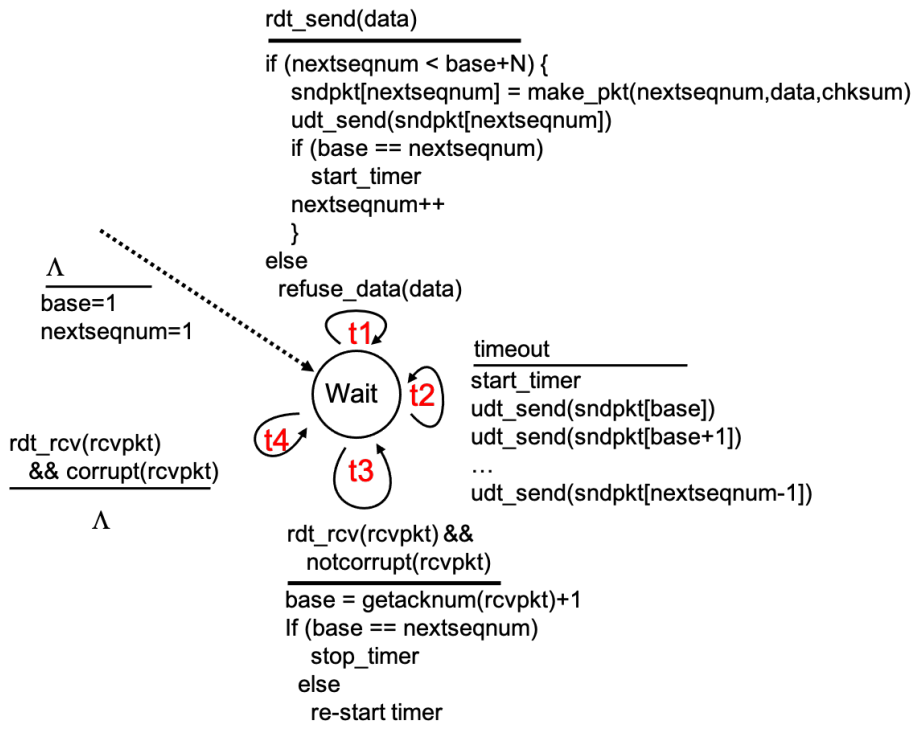
(4) t6, t13, t7, t8, t14, t9

(5) t1, t11, t3, t12, t4

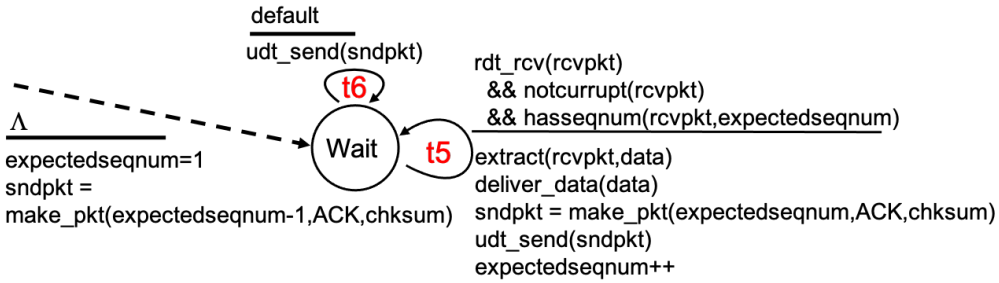
(6) take your pick and justify your own answers

10. (ch34, 11pt) Below is the FSM of GBN. Let N = 10.

GBN sender



GBN receiver



(1) There is this "if (nextseqnum < base+N)" condition in sender FSM's t1. Tell why it is necessary. (2pt)

(2) There is this "if (base==nextseqnum)" condition in sender FSM's t1. Tell why it is necessary. (2pt)

(3) There is this "if (base==nextseqnum)" condition in sender FSM's t3. Tell why it is necessary. (2pt)

(4) Suppose the sender gets the call from above to send 3 packet – pkt1, pkt2 and pkt3. Tell the sequence of transitions triggered sending these 3 packets. (1pt)

(5) Continue from (4). Suppose the 3 packets arrive at the receiver in this order – pkt1, pkt3, pkt2. Tell the sequence of transitions triggered receiving these 3 packets. (1pt)

(6) Continue from (5). Arrival of the pkt1, pkt3, and pk2 will generate ack packets. Tell the sequence numbers of the ack packets generated. (1pt)

(7) Continue from (6). Arrival of the ack packets at the sender will trigger a sequence of transitions. Tell the sequence of transitions. (1pt)

(8) Continue from (7). Tell the value of the send_base (base in the FSM). (1pt)

Sample Solution:

(1) to check if the sending window is used up

(2) to check if the sender needs to start a timer

(3) to check if the sender needs to stop or restart a timer

(4) t1 t1 t1

(5) t5 t6 t5

(6) ack1, ack1, ack2

(7) t3 t3 t3

(8) 3

11. (ch34, 7pt) Consider the FSM of SR. Let N = 10.

SR sender

t1: data from above:
- if next available seq # in window, send pkt

t2: timeout(n):
- resend pkt n, restart timer

t3: ACK(n) in [sendbase,sendbase+N-1]:
- mark pkt n as received
- if n is the smallest unACKed pkt, advance window base to next unACKed seq #

SR receiver

t4: pkt n in [rcvbase, rcvbase+N-1]
- send ACK(n)
- out-of-order: buffer
- in-order: deliver (also deliver buffered, in-order pkts), advance window to next not-yet-received pkt

t5: pkt n in [rcvbase-N,rcvbase-1]
- ACK(n)

t6: otherwise:
- ignore

(1) Suppose the sender gets the call from above to send 3 packet – pkt1, pkt2 and pkt3. Tell the sequence of transitions triggered sending these 3 packets. (1pt)

(2) Continue from (1). Suppose the 3 packets arrive at the receiver in this order – pkt1, pkt3, pkt2. Tell the sequence of transitions triggered receiving these 3 packets. (1pt)

(3) Continue from (2). Arrival of the pkt1, pkt3, and pkt2 will generate ack packets. Tell the sequence numbers of these ack packets. (1pt)

(4) Continue from (3). Arrival of the ack packets at the sender will trigger a sequence of transitions. Tell the sequence of transitions. (1pt)

(5) Continue from (4). Tell the value of send_base (sendbase in the FSM). (1pt)

(6) Consider how GBN and SR address the out-of-order delivery scenario in Problem Set 10 and 11. SR works better handling out of order deliveries. Why? (2pt)

Sample Solution:

(1) t1 t1 t1

(2) t4 t4 t4

(3) ack1, ack3, ack2

(4) t3 t3 t3

(5) 4

(6) SR buffers out of order packets at the receiver. When a gap is filled by pkt2, the whole train of packets can be acknowledged. On the other hand, GBN drops the out of order packets, the GBN sender therefore needs to retransmit out of order packets such as pkt3.

12. (ch34, 5pt) Compare and contrast the 3 pipelined rdt protocols, GBN, SR and TCP (i.e., TCP's rdt with fast retransmission and delayed ack).
    (1) Which method(s) use more network bandwidth to retransmit for losses and why? (1pt)
    (2) Which method(s) require more memory space at the receiver end and why? (1pt)
    (3) Which method(s) require more memory space at the sender end and why? (1pt)
    (4) Which method(s) require more timers at the sender end and why? (1pt)
    (5) Which method(s) require more timers at the receiver end and why? (1pt)

Sample Solution:

(1) GBN the most and TCP the 2nd.
GBN retransmits only a flight of packets.
TCP retransmits only 1 packet but can generate quite a bit of duplicate ACKs.

(2) TCP and SR.
Both will buffer out-of-order packets. TCP needs timer for delayed ack. The memory consumption at the receiver side might be slightly more than SR.

(3) GBN, SR, or TCP all about O(cwnd)

(4) SR. one timer per packet

(5) TCP. Need one timer to ensure sending of a delayed ack

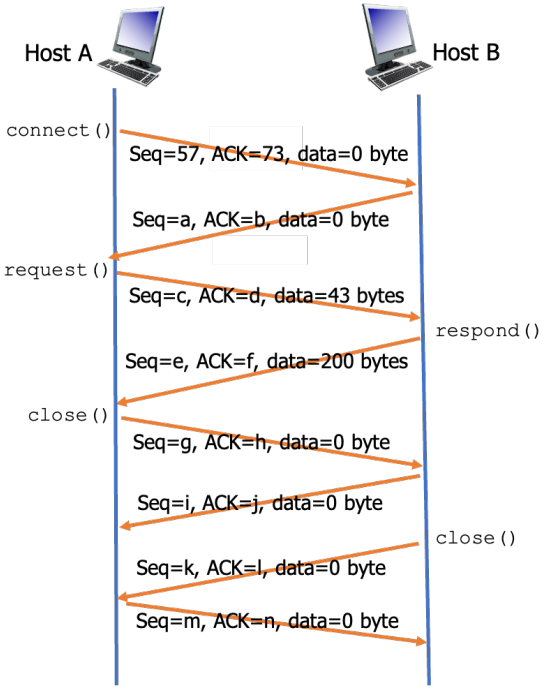
13. (ch35, 4pt) Packet loss can be detected in two ways – 3 duplicate acks and timeout.
    (1) Which of them is more serious and why? (2pt)
    (2) Which of them takes longer to detect and why? (2pt)

Sample Solution:

(1) Timeout. 3 dup acks means there are several packets arriving at the receiver triggering at least 3 dup acks. Although these packets are out of order, they pass the network and reach the receiver. Timeout means very few packets (in order or out of order) get through the network.

(2) Timeout. Timeout duration is set to estimatedRTT+ 4 devRTT, while dup acks come back typically in 1 RTT. The former is substantially longer.

14. (ch35, 14pt) Depicted below is a sequence of packet exchange in a TCP connection between Host A and B. Host A sends the SYN packet to establish connections and then generate request messages. Later, A sends a FIN packet to initiate closing of the connection. Host B sends a FIN packet as well. Tell the value of a, b, c, d, e. f, g, h, i, j, k, l, m, n.



Sample Solution:

(a, b, c, d, e, f, g, h, i, j, k, l, m, n) =

(73, 58, 58, 74, 74, 101, 101, 274, 274, 102, 274, 102, 102, 275)

15. (ch36, 7pt) There are two general approaches to congestion control – end to end and network assisted.

(1) Describe how the end-to-end approach works. (2pt)

(2) Describe how the network-assisted approach works. (2pt)

(3) Which way is taken by TCP? (1pt)
(4) Which way would you prefer and why? (2pt)

Sample Solution:

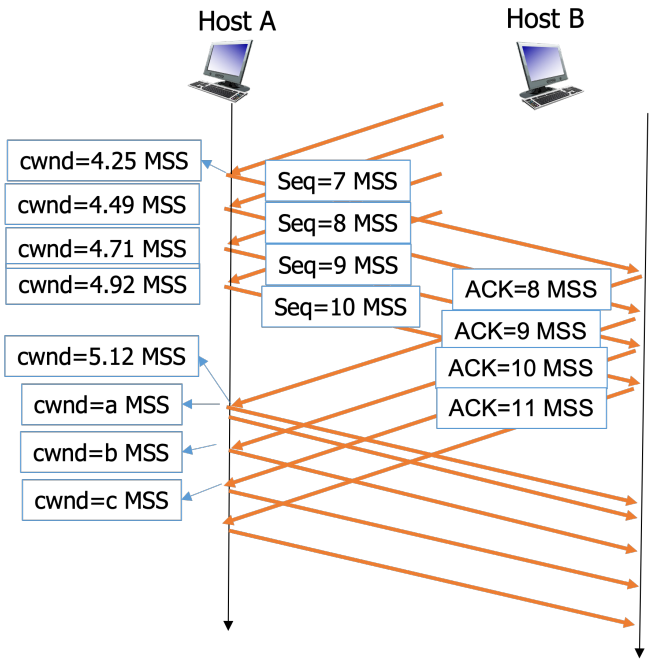
(1) Having the end hosts to detect packet loss or delay and inform the traffic source to set the sending rate accordingly
(2) Having routers to track the free buffer space and inform the traffic source to set the sending rate accordingly
(3) end-to-end approach
(4) take your pick and justify

16. (ch36, 4pt) TCP's congestion control mechanism defines not only how cwnd changes, but also how ssthresh changes.
    (1) Describe how ssthresh is changed when 3 duplicate acks are detected. (1pt)
    (2) Describe how cwnd is changed when 3 duplicate acks are detected. (1pt)
    (3) Describe how ssthresh is changed when a timeout is detected. (1pt)
    (4) Describe how cwnd is changed when a timeout is detected. (1pt)

Sample Solution:

(5) ssthresh = cwnd/2
(6) cwnd = cwnd/2
(7) ssthresh = cwnd/2
(8) cwnd = 1

17. (ch37, 5pt) A TCP connection starts with cwnd = 0 MSS and must transmit data packets at a constant size - 1 MSS. The figure below illustrates the data and ACK packet exchange in a straightforward scenario - no packet loss. Note that the exchanges in the earlier and later rounds are omitted for brevity. Address the questions below assuming the delayed ack mechanism is disabled. Grading policy: show derivation of cwnd to earn points.

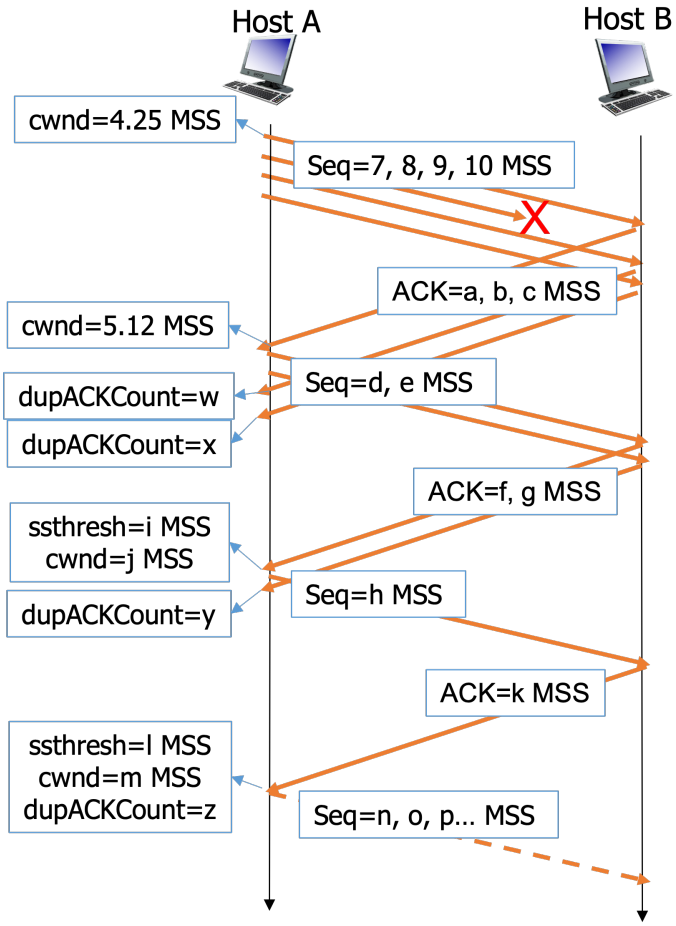


(1) Give the value of a (1pt)
(2) Give the value of b (1pt)
(3) Give the value of c (1pt)
(4) 5 packets are sent in the round starting with cwnd=5.12 MSS. Estimate the number of packets to send in the next round and tell how you come to the estimate. (2pt)

Sample Solution:

(1) 5.12+1/5.12 = 5.32
(2) 5.32+1/5.32 = 5.51
(3) 5.51+1/5.51 = 5.69
(4) 6. This is because after receiving these 5 packets, the cwnd will definitely grow higher than 6, allowing 6 packets to go the next round.

18. (ch37, 19pt) Similar to Problem Set 17, the scenario below is different in that there is a packet drop – the packet with sequence number 8 MSS. The figure below illustrates the data and ACK packet exchange in such a packet drop scenario. Note that the exchange in the earlier rounds is omitted for brevity. Address the questions below assuming the fast recovery mechanism is disabled. Grading policy: show derivation of cwnd to earn points.



(1) Packet 7, 9, 10 MSS should trigger 3 ACK packets to return. Tell the ACKs' sequence numbers – a, b, and c. (3pt)

(2) The 1st ACK returning to the sender trigger more data packets. Tell these data packets' sequence numbers – d, e. (2pt)

(3) The 2nd and 3rd ACK returning to the sender should update the dupACKCount variable. Tell the value of dupACKCount – w, x. (2pt)

(4) The data packets with sequence number d and e will trigger ACK packets. Tell these ACK packets' sequence numbers – f, g. (2pt)

(5) The f, g ACK packets will change dupACKCount, ssthresh, and cwnd of the sender. Tell the value of i, j, and y. (3pt)

(6) The f ACK packet in particular will generate a retransmission of the lost packet. Tell

the sequence number of the retransmission. I.e., the value of h. (1pt)

(7) The retransmission will generate an ACK back to the sender. Tell the sequence number of the ACK packet. I.e., the value of k. (1pt)

(8) The k ACK packet will change dupACKCount, ssthresh, and cwnd of the sender. Tell the value of l, m, and z. (3pt)

(9) Continue from (8). When the state of the sender changes, there might be cwnd space to send more data packets. If so, tell the sequence number of the data packets generated, i.e., the value of n, o, p, … (2pt)

Sample Solution:

(1) 8, 8, 8

(2) 11, 12

(3) 1, 2

(4) 8, 8

(5) 2.56, 2.56, 1

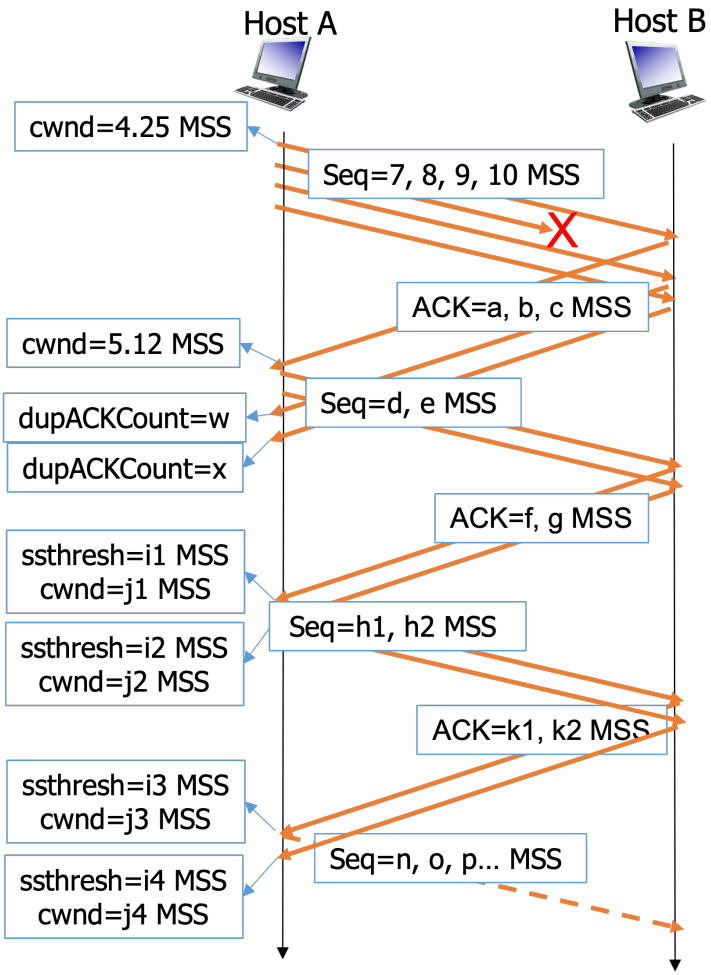
(6) 8

(7) 13

(8) 2.56, 2.56+1/2.56=2.95, 0

(9) 13, 14

19. (ch37, 14pt) The scenario is the same as Problem Set 18. Address the questions below assuming the fast recovery mechanism is enabled. TCP with fast discovery behaves differently when 3 duplicate ACKs are detected. That is approximately after ACK f or g MSS arrives at the sender. Grading policy: show derivation of cwnd to earn point.



(1) Tell the value of i1 and j1 after receiving ACK packet f MSS. (2pt)

(2) Tell the value of i2 and j2 after receiving ACK packet g MSS. (2pt)

(3) Tell the sequence number h1 and h2 for the two data packets triggered by ACK f and g MSS. (2pt)

(4) Tell the sequence number k1 and k2 for the two ACK packets triggered by packet h1 and h2 MSS. (2pt)

(5) Tell the value of i3 and j3 after receiving ACK packet k1 MSS. (2pt)

(6) Tell the value of i4 and j4 after receiving ACK packet k2 MSS. (2pt)

(7) After i3, j3, i4, j4 are changed, more data packets might be generated. Tell the sequence number of the data packets generated, i.e., the value of n, o, p, … (2pt)

Sample Solution:

(1) 5.12/2=2.56, 5.12/2+3=5.56.
f is the 3rd duplicate ACKs. The sender enters the fast recovery state, where ssthresh=cwnd/2 and cwnd=ssthresh+3 MSS.

(2) No change (2.56), 5.56+1=6.56.
The sender in the fast recovery state continues to increase the cwnd by 1 MSS when a duplicate ACK is received.

(3) 8 MSS – 3 duplicate ACKs trigger retransmission of the lost packet
13 MSS – with cwnd at 6.56, the sender can send one new packet, which carries a sequence number of 13 MSS

(4) 13 MSS – when the 8 MSS retransmission is successful, the receiver sends an ACK with the next expected sequence number, which is 13 MSS.
14 MSS – 13 MSS data packet is received. The ACK number is therefore 14 MSS.

(5) No change on ssthresh (2.56), cwnd set back to ssthresh =2.56.
13 MSS is a new ACK. The sender therefore exits the fast recovery state and, in the meantime, reset the cwnd to ssthresh.

(6) No change on ssthresh (2.56), cwnd=2.56+1/2.56=2.95
In the congestion avoidance state, cwnd is incremented by MSS/cwnd per ACK packet received.

(7) 14 MSS, 15 MSS
When 13 MSS ACK is received, the cwnd is 2.56. Having 13 MSS pkt not yet acked, the sender can only send one more packet. Therefore, 14 MSS.
When 14 MSS ACK is received, the cwnd is 2.95. Having 14 MSS pkt out and not yet acked, the sender sends one packet. Therefore, 15 MSS.

20. (PA, 25pt) Please go on the PA workstation and work under the exam2 directory for this problem set. Grading policy: pts for later problems will be given only when the formers are completed.

    (1) Develop exam2-p20-1.go such that it connects to the server running on port 11991 and then close the connection. (3pt)
    (2) Develop exam2-p20-2.go such that it connects to the server running on port 11992, sends “play\n” and then closes the connection. (3pt)
    (3) Extend exam2-p20-2.go to exam2-p20-3.go such that it connects to the server running on port 11993, sends “play\n”, prints the response from the server, and then closes the connection. (4pt)

    You should see the response from the server on port 11993. It says it is a Bulls and Cows game engine. In the game, the player guesses a 4-digit number. Each digit is a number in 0-9 and there’s no repeating digits. The game engine responses with #B#C. #B tells the # of digits being correct and at the right position. #C tells the # of digits being correct but at the wrong position. The game ends when 4B0C response is accomplished.

    (4) Extend the code to exam2-p20-4.go such that it connects to the server running on port 11993, sends “play\n”, prints a line of response from the server. And then, prompts the user for a 4-digit number, sends the number in one line (+’\n’ at the end of the message), receives a line of message from the server, prints the line of message on screen, and then closes the connection. (6pt)
    (5) To complete the game client, extend the code to exam2-p20-5.go such that it repeatedly prompts the user for another 4-digit number, sends the number in one line (+’\n’ at the end of the message), receives a line of message from the server until the server response is 4B0C. (6pt)
    (6) Run exam2-p20-5.go and play the game until 4B0C. Tell the final 4-digit number that gives 4B0C. (3pt)

Sample Solution:

Whatever that works.

21. (PA, 25pt) Please go on the PA workstation and work under the exam2 directory for this problem set. Grading policy: pts for later problems will be given only when the formers are completed.

    (1) Develop exam2-p21-1.go running on your exam port # such that it allows multiple clients such as your exam2-p20-1.go to access concurrently. (3pt)
    (2) Develop exam2-p21-2.go such that it allows multiple clients such as your exam2-p20-2.go to connect, send “play\n”, and close concurrently. (3pt)
    (3) Develop exam2-p21-3.go such that it allows multiple clients such as your exam2-p20-3.go to connect, send “play\n”, receive a short response from your exam2-p21-3.go saying “alright\n” and close concurrently. (4pt)
    (4) Develop exam2-p21-4.go such that it allows multiple clients such as your exam2-p20-4.go to connect concurrently, send “play\n”, receive a short response from your exam2-p21-4.go saying “alright\n”. Afterwards, allow the client to send a 4-digit number followed by “\n”, receive a short response from your exam2-p21-4.go saying the result of Bulls and Cows (assuming 6130 as the answer) and in this form “#B#C\n”. Then, close the connection. (9pt)
    (5) Develop exam2-p21-5.go such that it allows multiple clients such as your exam2-p20-5.go to connect concurrently, send “play\n”, receive a short response from your exam2-p21-5.go saying “alright\n”. Afterwards, allow repetition of – the client to send a 4-digit number followed by “\n”, receive a short response from your exam2-p21-5.go saying the result of Bulls and Cows (assuming 6130 as the answer) and in this form “#B#C\n”. The repetition ends till the short response is “4B0C\n”. That is also when you close the connection to the client. (6pt)

Sample Solution:

Whatever that works.