Computer Networks - Midterm

November 12, 2021

Duration: 2h15m

- This is an open-notes exam.
- It consists of 3 problems. The total number of points is 50.
- Write your answers clearly, in English or in French, using extra sheets if necessary. When you are done, scan or take a picture of your entire exam and upload it to Moodle.
- This document contains 19 pages.
- Please work on your own. Collaboration is not allowed during the exam.

Good luck!

Last Name:
First Name:

SCIPER No:

(answers to the questions are shown in italic and blue) *(grades in red)*

1 Short questions (5 points)

For each question, circle a single best answer.

1. If you connect to the Internet through a 1 Gbps (downstream) cable Internet connection, this means that:
   (a) You will be able to download data from the Internet at a maximum rate (throughput) of 1 Gbps.
   (b) Your download rate may vary over time.
   (c) You will share physical links with other households in your neighbourhood.
   (d) All of the above. *(Correct)*

2. The customers of Internet service provider (ISP) A and ISP B should be able to communicate with each other (exchange traffic) in the following scenario:
   (a) Always. *(Correct)*
   (b) Only if A and B are peering with each other.
   (c) Only if A and B are connected through an Internet exchange point (IXP).
   (d) Only if A and B have a customer/provider relationship.

3. When the network layer of a device (end-system or packet switch) wants to send a packet, it passes the packet to:
   (a) The link layer. *(Correct)*
   (b) The transport layer.
   (c) The physical layer.
   (d) All the layers.

4. If you change the length of a network link, you affect its:
   (a) Propagation delay. *(Correct)*
   (b) Transmission delay.
   (c) All the delay components.
   (d) None of the delay components.

5. Consider a queue that feeds a network link. If you change the transmission rate of the link, you may affect:
   (a) The queuing delay of the packets in the queue.
   (b) The transmission delay of the packets in the queue.
   (c) Both of the above. *(Correct)*
   (d) None of the above.

6. Suppose a DNS server uses UDP for all its communications (even with other DNS servers). What is the minimum number of sockets that this DNS server must keep?

   (a) Four: one to communicate with DNS clients, one to communicate with root DNS servers, one to communicate with top-level-domain (TLD) servers, and one to communicate with authoritative servers.

   (b) Zero.

   (c) Two: one for sending, one for receiving.

   (d) One. *(Correct)*

7. Suppose the network layer of the Internet guaranteed that packets were never corrupted or lost, and that packets were always delivered within reasonable time and in order. How could TCP still be useful?

   (a) It would reduce processing delays.

   (b) It would enable receivers to acknowledge to senders that they received their packets.

   (c) It would provide congestion control.

   (d) It would provide flow control. *(Correct)*

8. What does it mean that TCP "provides the illusion of reliability on top of an unreliable network layer"?

   (a) Applications do not need to worry about recovering from network-layer packet corruption and packet loss (TCP does it for them). *(Correct)*

   (b) TCP guarantees to the applications that it can always deliver all their packets on time.

   (c) TCP can only work on top of a reliable network layer.

   (d) All of the above.

9. What does it mean that TCP does "self-clocking"?

   (a) It determines whether to increase or decrease the congestion window based on the received ACKs. *(Correct)*

   (b) It determines whether to increase or decrease the receiver window based on the received ACKs.

   (c) It performs clock synchronization.

   (d) It keeps its own clock.

10. What happens to a TCP sender's window when there is zero network congestion (no packet corruption or loss)?

    (a) It becomes 0.

    (b) It becomes infinite.

    (c) It is always equal to the receiver's window (the TCP-header field). *(Correct)*

    (d) Who cares? The sender's window is not useful/relevant in this scenario.

2 Web browsing + delays (30 points)

Consider the network in figure 1.

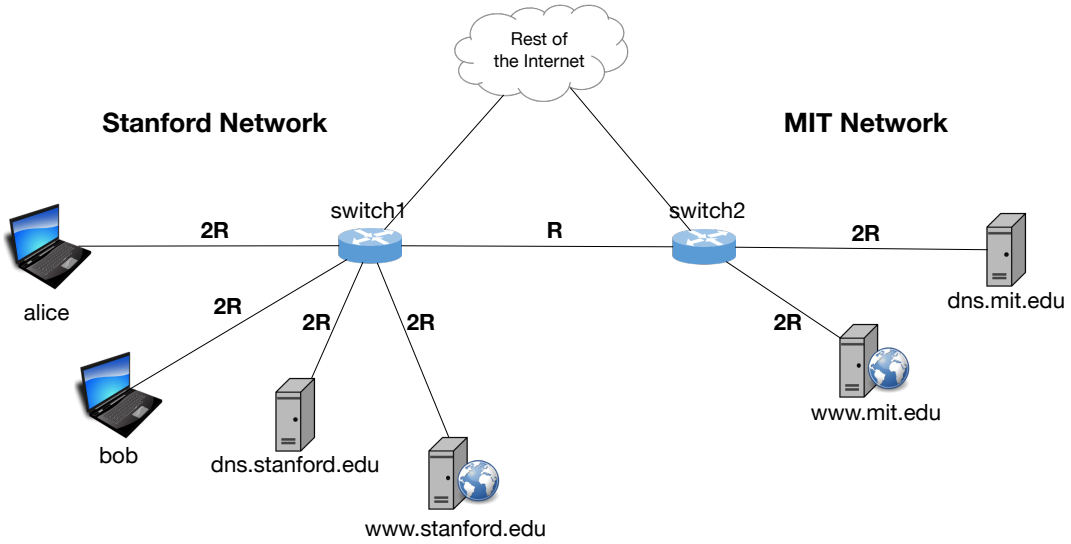


Figure 1: Network Topology for Problem 2

- Each link has propagation delay Dp.
- For each link, the transmission rate has the value shown in the figure in each direction.
- The maximum segment size (MSS) is 1000 bytes (for both the Stanford and MIT networks).
- **switch1** and **switch2** are store-and-forward packet switches with 0 processing delay.
- All Stanford computers use **dns.stanford.edu** as their local DNS server. **dns.stanford.edu** also acts as an authoritative DNS server for **stanford.edu**, and also as a top-level-domain (TLD) DNS server for **.edu**. It performs recursive DNS requests.
- All MIT computers use **dns.mit.edu** as their local DNS server. **dns.mit.edu** also acts as an authoritative DNS server for **mit.edu**. It performs recursive DNS requests.
- Web browsers and web servers communicate through persistent TCP connections (i.e., they reuse each established TCP connection to exchange as much traffic as possible).
- At the beginning of this problem, all caches (of all kinds) are empty, and there are no established TCP connections.
- In all questions except for Question 7, DNS records are cached for a day.
- There is no other traffic on the Internet other than the traffic caused by Alice's and Bob's actions.

Header and object sizes:

- Transport-layer, network-layer and link-layer headers are insignificant (assume 0 bytes).
- Assume that packets that do not carry any application-layer data experience 0 transmission delay. E.g., TCP SYN, TCP SYN ACK, and packets that carry only TCP ACKs experience 0 transmission delay.
- Each DNS request is 50 bytes.
- Each DNS response is 100 bytes.
- Each HTTP GET request is 200 bytes.
- Each HTTP GET response is 200 bytes + the size of the requested object.
- www.stanford.edu/index.html: 800 bytes
- www.mit.edu/tech-news.mp4: 2100 bytes

Question 1 (8 **points):**

Alice types in her web browser http://www.stanford.edu/index.html and presses enter. The requested web page references video www.mit.edu/tech-news.mp4 and no other object.

List all the packets that are exchanged between a Stanford computer and an MIT computer as a result of Alice's actions. (A "computer" may be a personal computer, e.g., Alice's computer, or a server.) Include any TCP connection-setup packets, but ignore any packets that carry only TCP ACKs. For each packet, list which computer sends it and which computer receives it, the transport-layer protocol, the source and destination port number, and briefly its purpose (e.g., it carries a request for...).

If a port number could be anything, use a letter, e.g., "x" or "y," but be consistent, i.e., use the same letter to refer to the same port number.

Packet #	Source	Destination	Source port	Dest. port	Transp. prot.	Purpose
1	dns.stanford.edu	dns.mit.edu	53	53	UDP	DNS request for www.mit.edu's IP address
2	dns.mit.edu	dns.stanford.edu	53	53	UDP	DNS response
3	alice	www.mit.edu	x	80	TCP	connection-setup request
4	www.mit.edu	alice	80	x	TCP	connection-setup response
5	alice	www.mit.edu	x	80	TCP	HTTP GET request for video file
6	www.mit.edu	alice	80	x	TCP	HTTP GET response, part 1
7	www.mit.edu	alice	80	x	TCP	HTTP GET response, part 2
8	www.mit.edu	alice	80	x	TCP	HTTP GET response, part 3

Question 2 (8 **points):**

After Alice receives the last bit of the last packet, Bob types in his web browser http://www.mit.edu/tech-news.mp4 and presses enter.

List all the packets generated by any computer as a result of Bob's actions. (A "computer" may be a personal computer, e.g., Bob's computer, or a server.) Include any TCP connection-setup packets, but ignore any packets that carry only TCP ACKs. For each packet, state which computer sends it and which computer receives it, the transport-layer protocol, the source and destination port number, and (briefly) its purpose.

If a port number could be anything, use a letter, e.g., "x" or "y," but be consistent, i.e., use the same letter to refer to the same port number.

Packet #	Source	Destination	Source port	Dest. port	Transp. prot.	Purpose
1	bob	dns.stanford.edu	y	53	UDP	DNS request for www.mit.edu's IP address
2	dns.stanford.edu	bob	53	y	UDP	DNS response
3	bob	www.mit.edu	z	80	TCP	connection-setup request
4	www.mit.edu	bob	80	z	TCP	connection-setup response
5	bob	www.mit.edu	z	80	TCP	HTTP GET request for video file
6	www.mit.edu	bob	80	z	TCP	HTTP GET response, part 1
7	www.mit.edu	bob	80	z	TCP	HTTP GET response, part 2
8	www.mit.edu	bob	80	z	TCP	HTTP GET response, part 3

Question 3 (3 **points):**

How long does it take from the moment Bob's computer sends the first bit of the first packet until the TCP 3-way handshake between Bob's computer and **www.mit.edu** is completed? There is no packet loss/corruption.

Justify your answer.

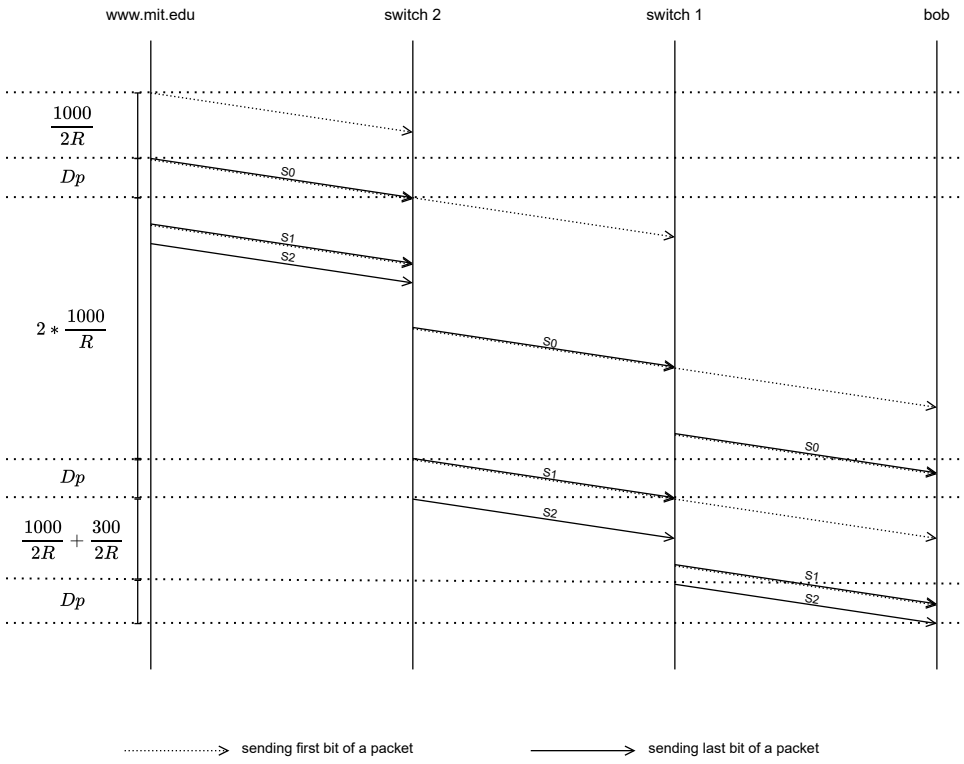
- DNS request $2D_p + 2\frac{50}{2R}$
- DNS response $2D_p + 2\frac{100}{2R}$
- TCP SYN from Bob to www.mit.edu: $3D_p$.
- TCP SYN ACK from www.mit.edu to Bob: $3D_p$.
- HTTP GET request from Bob to www.mit.edu: $3D_p + \frac{200}{2R} + \frac{200}{R} + \frac{200}{2R} = 3D_p + \frac{400}{R}$.
- Total: $13D_p + \frac{550}{R}$.

Question 4 (4 **points):**

Suppose **www.mit.edu** uses a TCP variant where the sender's window is always set to 10xMSS. How long does it take from the moment the TCP 3-way handshake between Bob's computer and **www.mit.edu** is completed until Bob's computer receives the last bit of the last packet? There is no packet loss/corruption.

Justify your answer.

- Propagation delay: $3D_p$.
- First packet of HTTP GET response over first link: $\frac{1000}{2R}$.
- First and Second packets over the bottleneck: $\frac{2000}{R}$.
- Second and Last packet over last link: $\frac{1300}{2R}$.
- Total: $3D_p + \frac{3150}{R}$.

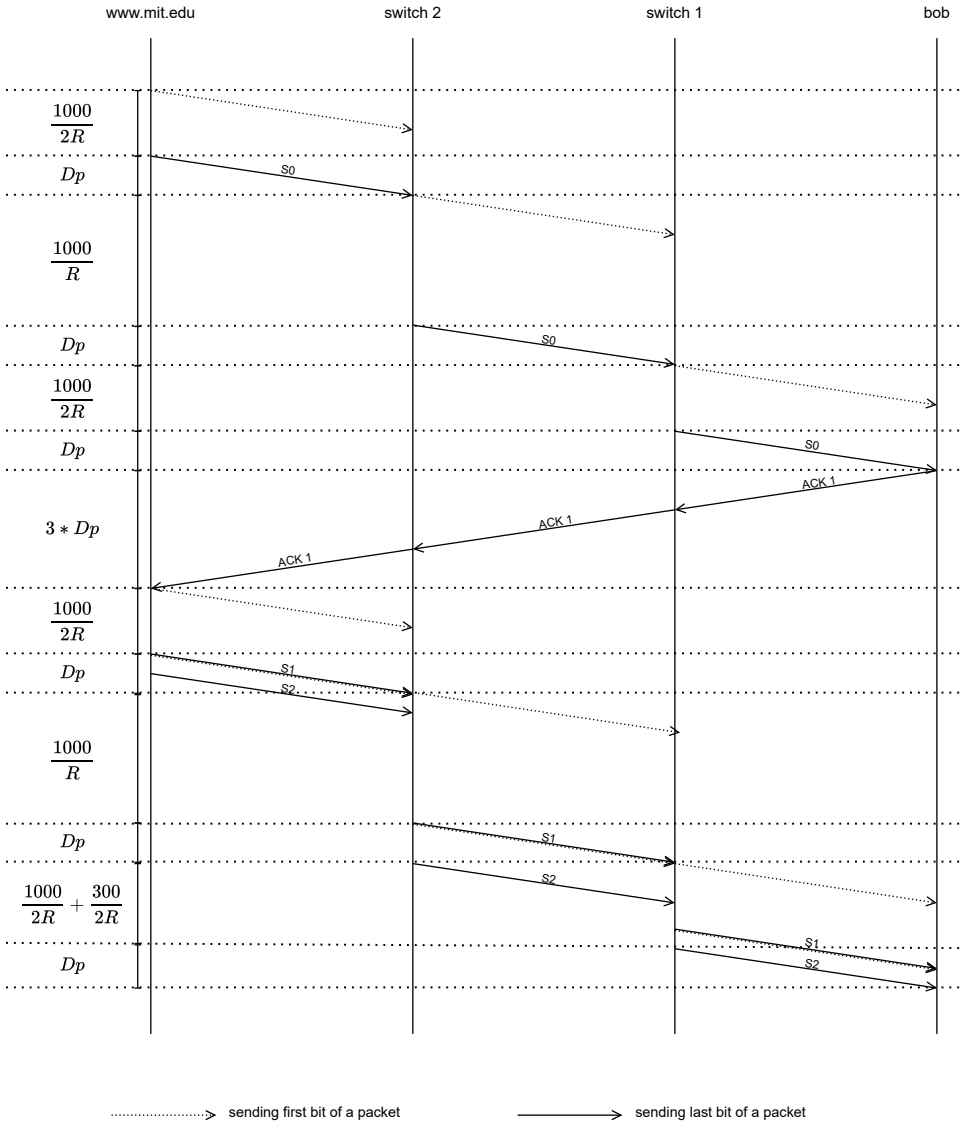


Question 5 (4 **points):**

Now suppose that **www.mit.edu** uses one of the TCP variants that we saw in class (Tahoe or Reno). How does your answer to Question 4 change? There is no packet loss/corruption.

Justify your answer.

- First packet of HTTP GET response: $3D_p + \frac{1000}{2R} + \frac{1000}{R} + \frac{1000}{2R}$.
- ACK: $3D_p$.
- Second and third packets of HTTP GET response together: $3D_p + \frac{1000}{2R} + \frac{1000}{R} + \frac{1300}{2R}$.
- Total: $9D_p + \frac{4150}{R}$.



Question 6 (1 **points):**

a. When there is no packet loss/corruption, which of the two TCP variants (the one from Question 4 or the one from Question 5) results in higher throughput from **www.mit.edu** to Bob's computer? (You don't have to give the actual throughput values, just say which one is higher.) Justify your answer.

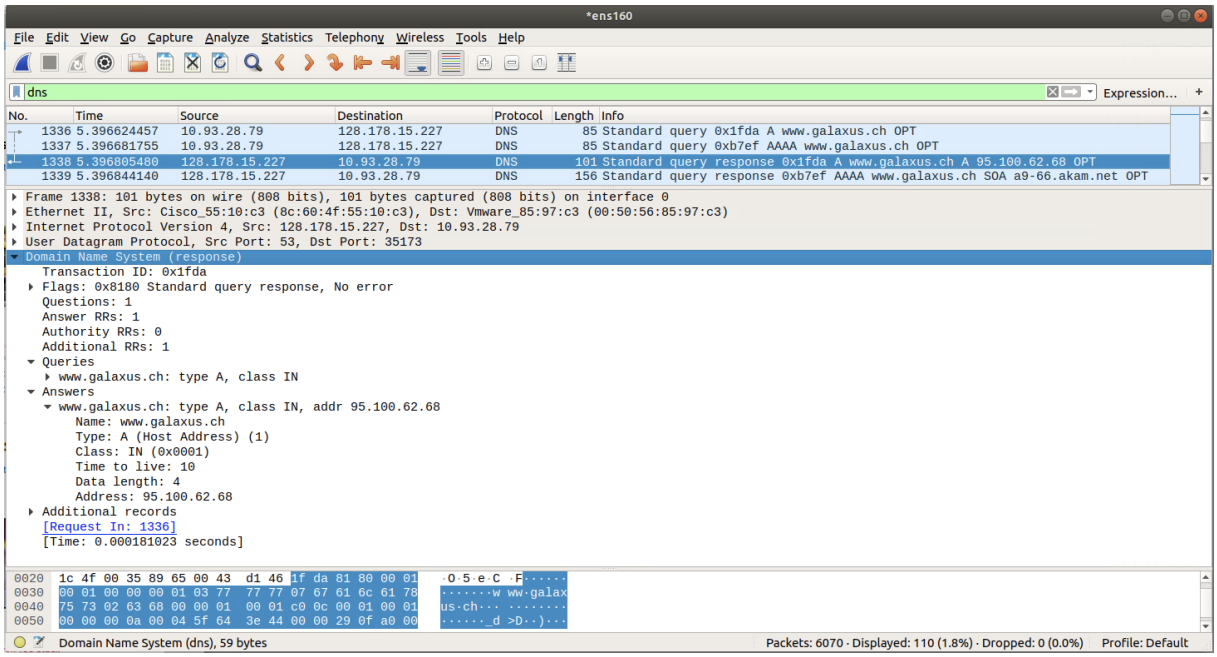
b. Now consider a more realistic scenario where there *is* packet loss/corruption. What is the disadvantage of the first TCP variant (the one from Question 4)? What could go wrong if all computers on the Internet used that variant?

a. The TCP variant with 100xMSS achieves higher throughput because it has a lower transfer time.

b. The disadvantage is that we have neither congestion control nor flow control. So if all computers on the Internet used this variant, we will have more congestion.

(Lab related) Question 7 (2 **points):**

Some time after the events of the previous questions, Alice visits **www.galaxus.ch**. At the same time, she runs Wireshark on the network interface of her computer and gets the following output:



a. What is the IP address of **dns.stanford.edu**?
128.178.15.227

b. What is the IP address of **www.galaxus.ch**?
95.100.62.68

c. Is **dns.stanford.edu** an authoritative DNS server for **galaxus.ch**? Justify your answer.
No, "Authority RRs" is 0.

d. Suppose Alice visits **www.galaxus.ch** again, 30sec later. Will her DNS client make a new DNS request for **www.galaxus.ch**'s IP address or not? Justify your answer.
Yes, "Time to live" is 10 (seconds) and thus the cached record will be expired when Alice revisits Galaxus after 30sec.

3 Transport layer (15 points)

Consider two end-systems A and B. A process running on A has an infinite amount of data to send to a process running on B. The process running on B does not have any data to send to the process running on A.

The two processes are communicating over TCP.

The network never reorders packets.

Figure 2 shows at which points A transmits a segment to B:

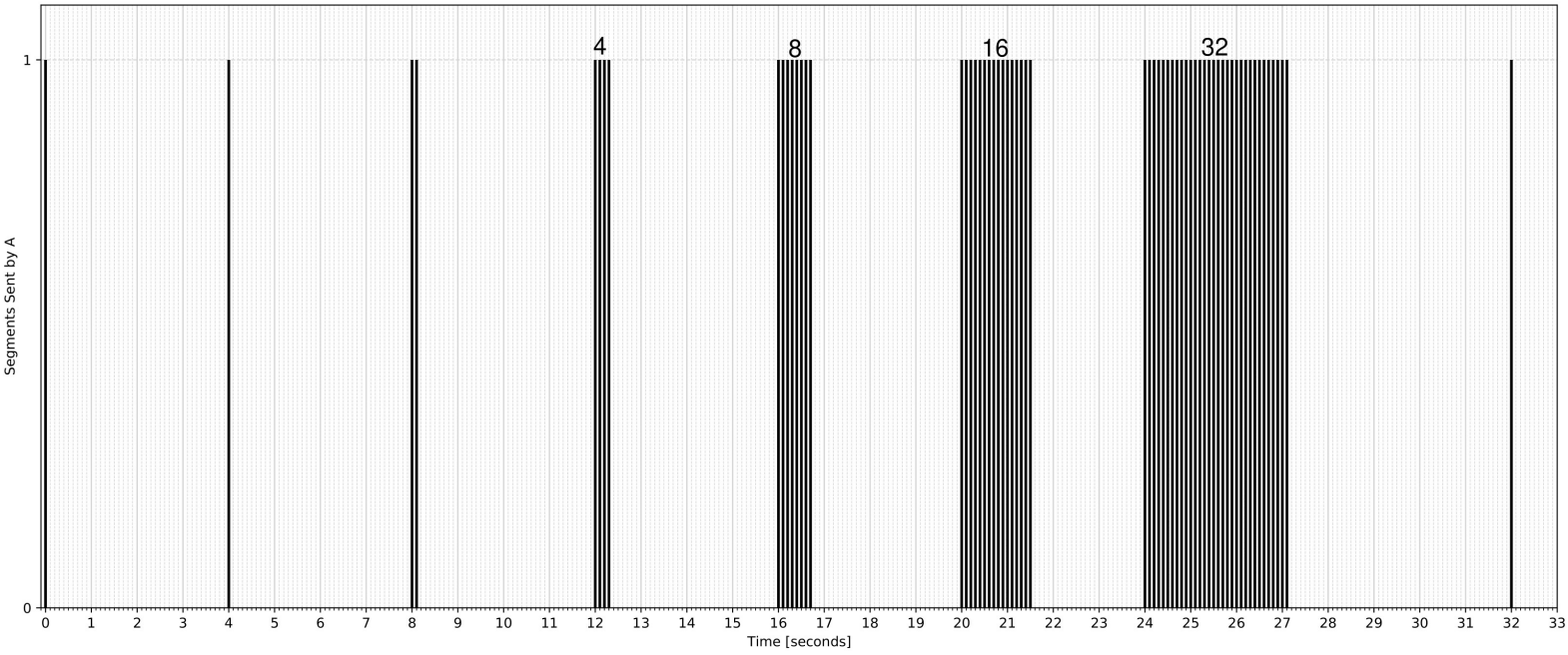


Figure 2: Transmissions of A over time.

For example, at time t=0sec, A's transport layer transmits 1 segment, at time t=4sec 1 segment, at time t=8sec 2 segments back-to-back, and so on. Each segment shown in the figure may be a new segment or a retransmission (you will need to guess).

You may assume that the maximum segment size (MSS) is 1 byte, if that helps you read/interpret the figure.

No TCP connection has been established between A and B before time t=0sec.

Question 1 (0.25 **point):**
What is the role of the segment transmitted at time t=0sec?

It is a TCP SYN segment. A sends it to establish a TCP connection with B.

Question 2 (0.25 **point):**
Which is the first segment that carries data?

The segment transmitted at t=4s.

Question 3 (4 **points):**

a. What is the value of the sender window of A at time t=24sec?

b. Is it possible that any segments have been lost up to this point?

Justify your answers.

a) It is clear that A is in *exponential increase* state and that it doubles its congestion window every RTT (approx. 4 seconds). From t=20s until t=21.5s, A transmits a burst of 16 packets and therefore must have a window size of 16 at t=21.5s. At t=24s, A transmits a new segment and must therefore have received at least one new ACK. This makes A's window size at least 17 at t=24s. It could be up to 32 if for some reason the ACKs from the previous burst arrived faster.

b) Up to t=24s, no data segments can have been lost because the window has not decreased at any point. Instead, it grows exponentially as expected. Similarly, no ACKs can have been lost as the loss of one would make the window grow slower (A increases its window by 1 for each new ACK) or would have resulted in a timeout.

Question 4 (1.5 **points):**

a. What is the value of the sender window of A at time t=32sec?

b. What must have happened between time t=24sec and t=32sec?

Justify your answers.

A's window size is 1. At t=32+s, A transmits a single segment. If A was still in *exponential increase* a new ACK would cause it to transmit 2 segments (one because the window slided and one because it grew). Therefore A is no longer in *exponential increase*. The simplest explanation is that at t=32s a timeout occured, A set its window to 1MSS, and retransmitted the first unaknowledged segment. The timeout can have been caused by packet loss or delay.

Now consider two other end-systems A' and B'. A process running on A' has an infinite amount of data to send to a process running on B'. The process running on B' does not have any data to send to the process running on A'.

The two processes are communicating over TCP.

The network never reorders packets.

Figure 3 shows at which points A' transmits a segment to B':

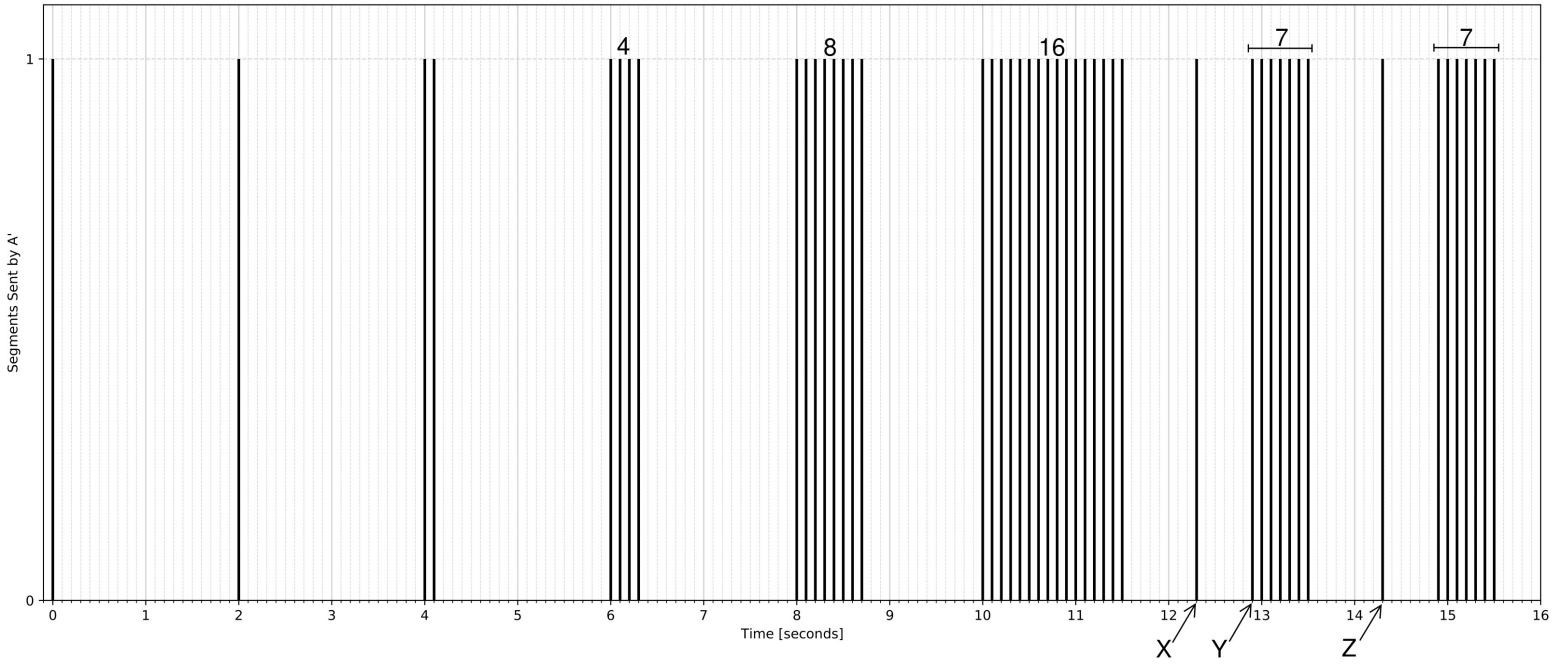


Figure 3: Transmissions of A' over time.

You may assume that MSS=1 byte, if that helps you read/interpret the figure.

No TCP connection has been established between A' and B' before time t=0sec.

Question 5 (4 **points):**

a. Does the TCP variant running on A' use fast-retransmit/fast-recovery (i.e., is it TCP Reno) or not (i.e., it is TCP Tahoe)?

b. What must have happened between time t=10sec and t=12sec?

Justify your answers.

A' uses TCP Reno and between t=10s and t=12s a data packet was lost (the segment which was sent at t=10s). We deduce this because after A' sends the 16-segment batch, the window appears to stop growing exponentially. If A' was running Tahoe, it would stay in slow start mode and set its window to 1. We would then expect A' to transmit one segment, then 2, 4, 8... (exponential increase)
This is not the case and A's behavior can be explained by assuming it uses Reno (see next question). From t=10.1s to t=10.3s, A' receives 3 duplicate acks. A' enters fast recovery and retransmits the lost segment.

Question 6 (4 **points):** What is the value of the sender window of A' at time:

a. t=12.3sec (when A' sends segment X)

b. t=12.9sec (when A' sends segment Y)

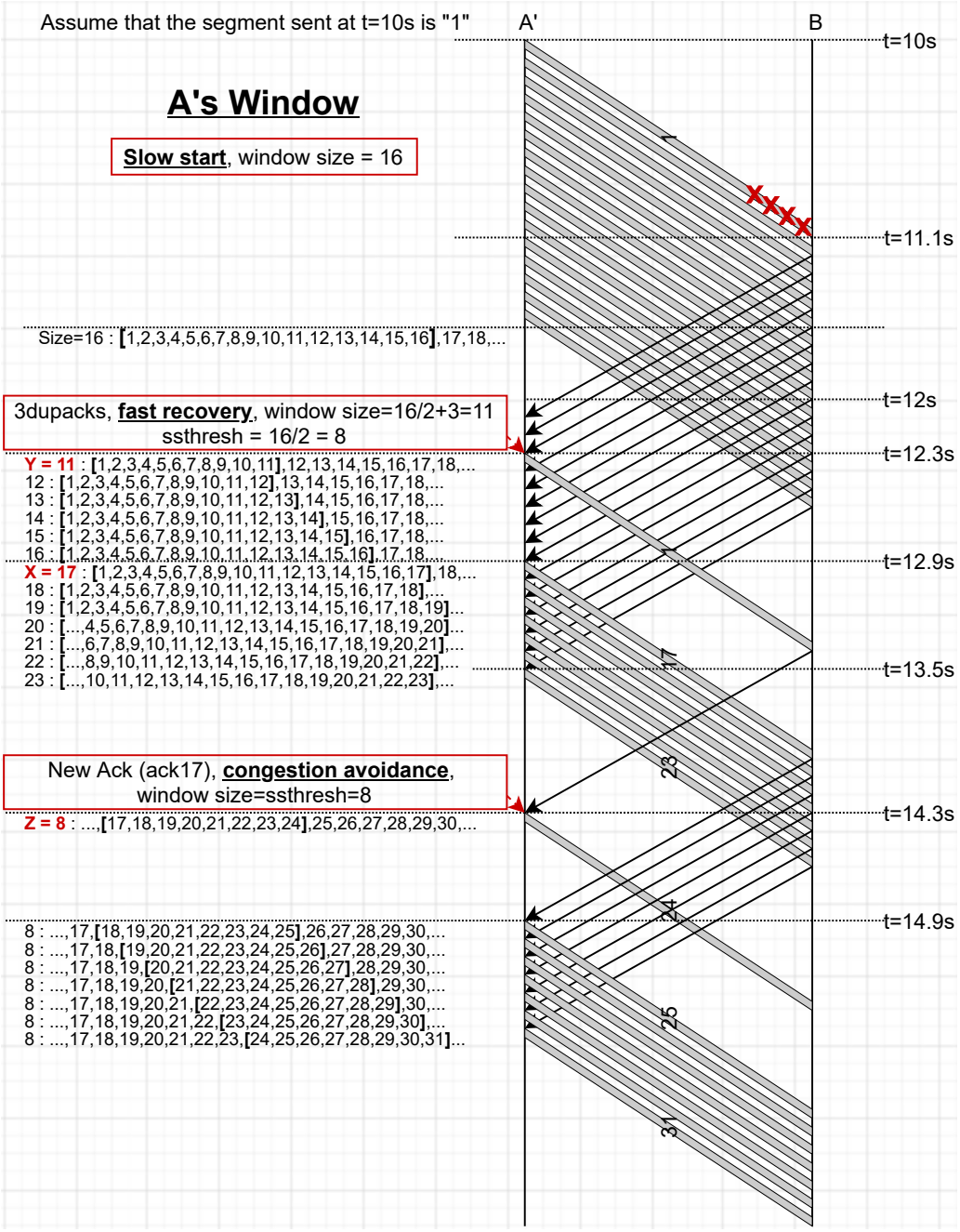
c. t=14.3sec (when A' sends segment Z)

Justify your answers.

See the figure below for an in depth explanation. Assume that segment "1" is the segment transmitted at t=10s.

- X: wnd = 16/2 + 3 = 11. A' just received 3 duplicate acks. It sets ssthresh to half the previous window (16/2 = 8) and sets its window size to ssthresh + inflation = 8 + 3 = 11 MSS.

- Y: wnd = 11 + 6 = 17. A' is in fast recovery. Therefore, for each duplicate ack it receives, it increases its window by 1 MSS. To send the next segment (17), A' needs a window of size 17 (which starts from 1 (as segment 1 is still unaknowledged) and ends at 17). Therefore, at t=Y, since A' sends a segment, it must have received 6 more duplicate acks which makes its window size 17.
- Z: wnd = 8. A' received a new ack for the segment retransmitted at X (segment 1). Therefore, A' exits from fast recovery, enters congestion avoidance, and sets its window to ssthresh = 8.



Question 7 (1 **point):**

At what state is the TCP congestion-control algorithm running on A' at the end of the figure? Justify your answer.

As discussed above, because at t=Z A' receives a new ack, it enters congestion avoidance.