**University of Toronto Mississauga**
Final Test, CSC358H5 Winter 2024
Section: LEC0102 and LEC0103, Erfan Meskar
**Aids: No aid-sheet is permitted. No electronic or mechanical computing devices are permitted.**
**April 25, Time: 9am, Duration: 2h**

---

DO NOT TURN THIS PAGE UNTIL YOU ARE TOLD TO DO SO

---

- The University of Toronto Mississauga and you, as a student, share a commitment to academic integrity. You are reminded that you may be charged with an academic offence for possessing any unauthorized aids during the writing of an exam. Clear, sealable, plastic bags have been provided for all electronic devices with storage, including but not limited to: cell phones, smart watches, SMART devices, tablets, laptops, and calculators. Please turn off all devices, seal them in the bag provided, and place the bag under your desk for the duration of the examination. You will not be able to touch the bag or its contents until the exam is over. If, during an exam, any of these items are found on your person or in the area of your desk other than in the clear, sealable, plastic bag, you may be charged with an academic offence. A typical penalty for an academic offence may cause you to fail the course. Please note, once this exam has begun, you CANNOT re-write it.
- There are 6 questions and 17 pages in this exam, including this one.

| | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Total |
|---|---|---|---|---|---|---|---|---|
| Max | 9 | 27 | 10 | 13 | 10 | 10 | 11 | 90 |
| Score | | | | | | | | |

1. **(9 pts) Multiple Choice**

   **(a) (2 pts)** For each of the following addressing schemes, specify if it is flat or hierarchical?

   | | | | |
   |---|---|---|---|
   | (i) | Media Access Control (MAC) address: | ■ flat | ☐ hierarchical |
   | (ii) | IP address: | ☐ flat | ■ hierarchical |
   | (iii) | Autonomous System (AS) address: | ■ flat | ☐ hierarchical |
   | (iv) | Port number: | ■ flat | ☐ hierarchical |

   **(b) (2 pts)** Which one of the following statements is true about TCP Fast Retransmission? (Select all that applies)

   (i) Fast retransmission is not effective when we have a large congestion window.

   $\boxed{\text{(ii)}}$ Fast retransmission is not effective when packet drops are bursty.

   (ii) Fast retransmission is the exponential growth of congestion window when a new TCP connection is established.

   (iii) Fast retransmission starts after a timeout occurs.

   **(c) (2 pts)** Which of the following is true about layers in the network? (Select all that applies)

   (i) BGP is a link layer protocol and defines how autonomous systems forward traffic to each other.

   (ii) UDP is a transport layer protocol that adds reliability to IP protocol.

   (iii) Ethernet is a link layer protocol and creates a reliable connection for delivery of packets from one host to another.

   $\boxed{\text{(iv)}}$ IP is a network layer protocol, which provides an unreliable, best-effort service.

   **(d) (3 pts)** Consider the situation where a sender A and a receiver B communicate with each other using stop-and-wait ARQ. Suppose that A uses as the initial sequence number (ISN) 200, i.e. we have that ISN=200. Furthermore suppose that the first packet that A sends to B consists of 100 bytes.

   (i) What is the sequence number that A puts into the first data carrying packet? (Choose one)

   ☐ 200
   ■ 201
   ☐ 300
   ☐ 301

   (ii) Assume that B receives the packet in part (i) without any error. What is the ACK number that B uses in the packet that is sends to A in response to the packet it received? (Choose one)

   ☐ 200
   ☐ 201
   ☐ 300
   ■ 301

   (iii) Now, assume that B detects an error in the packet in part (i). What is the ACK number that B uses in the packet that is sends to A in response to the packet it received? (Choose one)

   ☐ 200
   ■ 201
   ☐ 300
   ☐ 301

2. **(27 pts) Short answer**

(a) **(4 pts)** Consider the *Time to Live (TTL)*, *Protocol*, *Type-of-Service*, and *Checksum* field in the IPv4 header. For each of these fields, name the task that it is needed for.
Choose from the following list of tasks: Read packet correctly; Get the packet to the destination and get responses to the packet back to source; Tell host what to do with the packet once arrived; Specify any special network handling of the packet; Deal with potential problems that arise along the path (*e.g.*, Header corruption, Loop, and Packet too large).

| Field | Task |
|---|---|
| TTL | Deal with loop |
| Protocol | Tell host what to do with the packet once arrived |
| Checksum | Deal with header corruption |
| Type-of-Service | Specify any special network handling of the packet |

(b) **(4 pts)** Describe the mapping performed in each of the following tables. Your answer must be in the form of "maps A to B."

| |
|---|
| Router's routing table:<br><br>Maps IP prefix or IP range to outgoing interface OR outgoing port OR IP address |
| Switch's forwarding table:<br><br>Maps MAC address to outgoing port OR outgoing interface |
| Address Resolution Protocol (ARP) cache table:<br><br>Maps IP address to MAC address |
| Network Address Translator (NAT) table:<br><br>Maps pair of IP address and port number to a pair of IP address and port number |

(c) **(3 pts)** In Wireshark Labs, we have seen traces with DHCP, HTTP, and DNS as their application-layer protocols. What is the underlying transport protocol of each of them. Choose between Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

| Dynamic Host Configuration Protocol (DHCP): | Hypertext Transfer Protocol (HTTP): | Domain Name System (DNS): |
|---|---|---|
| UDP | TCP | UDP |

(d) **(2 pts)** Name two programs/command-line utility that are implemented using Internet Control Message Protocol (ICMP).

| |
|---|
| Ping and Traceroute |

**(e)** **(2 pts)** How many unique IP addresses are there in the subnet 83.170.207.160/27, including the broadcast IP address and network IP address? What is the broadcast IP address of this subnet?

| Number of unique IP addresses in this subnet | The broadcast address of this subnet |
|---|---|
| 32 | 83.170.207.191 |

**(f)** **(2 pts)** Explain how Maximum Transmission Unit (MTU), Maximum Segment Size (MSS), and Maximum Frame Size are related to each other.

MTU = Maximum Frame Size - Link layer header

MSS = MTU - IP header - TCP header

**(g)** **(2 pts)** What are the advantage and disadvantage of HTTP being stateless?

**Advantage:**

Improves scalability on the server-side, Failure handling is easier, Can handle higher rate of requests, Order of requests won't matter

**Disadvantage:**

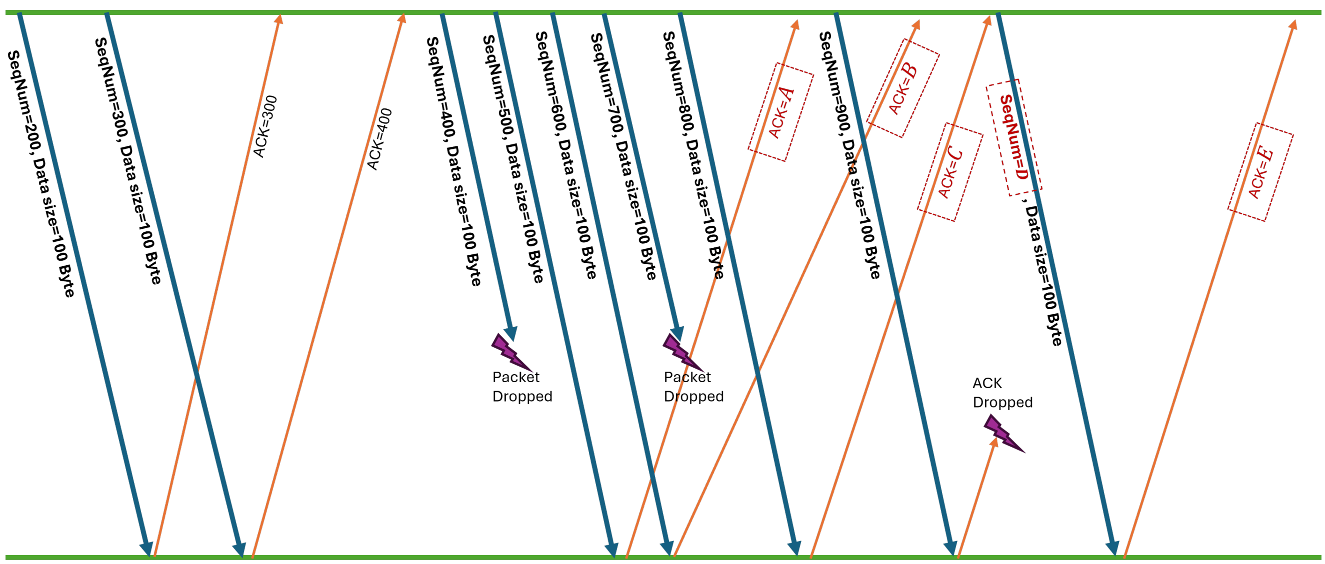
Some applications need persistent state to uniquely identify user or store temporary info *e.g.*, Shopping cart, user profiles, usage tracking

**(h)** **(3 pts)** We discussed in class that at some layers, the addresses need to be translated from one format to another. For these address translations, answer the following questions. Name the layers where addresses are translated from one format to another and name the protocol/mechanism that is used for the translation.

DNS: Translates application layer address to network later address (*i.e.*, domain name to IP address).

ARP: Translates network layer address to link later address (*i.e.*, IP address to MAC address).

**(i) (5 pts)** In figure below, TCP Reno transmits packets with the given sequence numbers and acknowledgment numbers. Assume that the receiving buffer's capacity is infinity and the receiver keeps the out-of-order segments in its buffer. Find the acknowledgment numbers denoted by $A$, $B$, $C$, and $E$, and the sequence number denoted by $D$.




| | | | | |
|---|---|---|---|---|
| $A$ = 400 | $B$ = 400 | $C$ = 400 | $D$ = 400 | $E$ = 700 |

**3.** **(10 pts)** Describe the following pair(s) of terms/concepts clearly and concisely (in at most 4-6 sentences). Explain the key differences – the context they are defined at, protocol(s)/mechanism(s) they are related to, etc.

**(I)** **(5 pts)** Flow Control vs. Congestion Control:

The context they are defined at and mechanisms/protocols they are related to:

- Both are defined at and related to TCP (we also accept transport layer and reliable transmission)
- Flow control is related to the sliding window mechanism.

Key differences:

- Flow control: Prevent sender from overwhelming/overflowing the receiver.
- Congestion control: Prevent sender from overwhelming/congesting the network.

**(II)** **(5 pts)** Data Plane vs. Control Plane of routers:

The context they are defined at and mechanisms/protocols they are related to:

- Both are defined at and related to IP (we also accept network layer, routing, and routers)
- Data Plane is related to longest prefic match mechanism.
- Control Plane is related to Distance Vector (DV) routing protocols (we also accept RIP) and Link State (LS) routing protocols (we also accept OSPF or IS-IS) and BGP

Key differences:

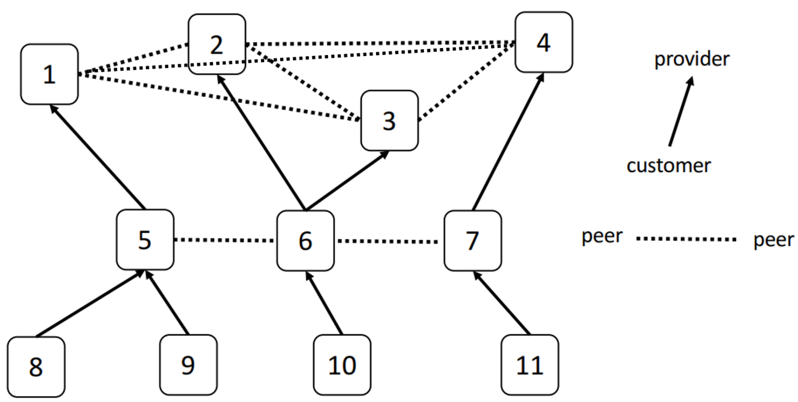
- Data Plane: How to use forwarding/routing Tables
- Control Plane: How to populate forwarding/routing tables (create routes)

**4. (13 pts)** The following parts of this question are independent.

**(a) (3 pts)** Using the local preference to enforce valley-free paths, please fill in whether a route imported from a neighbor of a given type should be sent to another neighbor of a given type or not. Answer by Yes or No.

| Route received from | Route sent to | | |
|---|---|---|---|
| | Customer | Provider | Peer |
| Peer | Yes | No | No |

**(b) (4 pts)** Consider the diagram of Autonomous Systems (ASes) shown here. Arrows point from customer up towards a provider, dashed lines connect peers. What possible valley-free paths are there from AS10 to AS9? Which path will be used for sending traffic?




$10 \rightarrow 6 \rightarrow 5 \rightarrow 9$

$10 \rightarrow 6 \rightarrow 3 \rightarrow 1 \rightarrow 5 \rightarrow 9$

$10 \rightarrow 6 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9$

The path that will be used is: $10 \rightarrow 6 \rightarrow 5 \rightarrow 9$

**(c) (4 pts)** Which of the following statement(s) are true? Circle the true one(s).

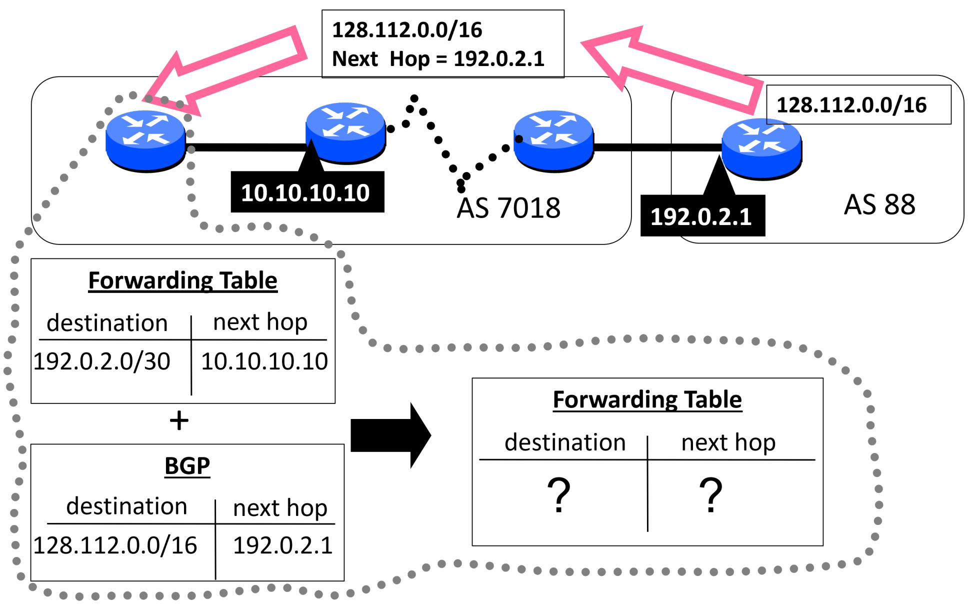
(i) iBGP is used for intradomain routing

$\boxed{\text{(ii)}}$ Avoiding loops is one reason why BGP uses path vector

(iii) BGP always advertises a shortest path

$\boxed{\text{(iv)}}$ BGP route advertisements use classless addressing

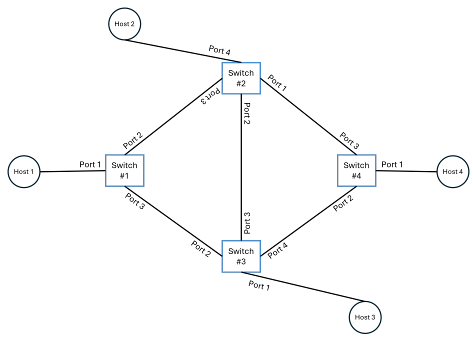
**(d) (2 pts)** In class, we studied how information from the IGP and BGP routing protocols are combined to populate the forwarding table. In the figure below, fill out the forwarding table based on the information obtained from the BGP and IGP protocol.



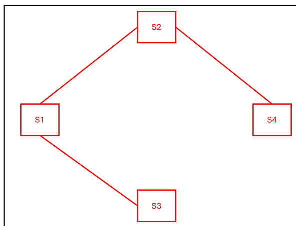

**Forwarding Table**

| destination | next hop |
|---|---|
| 192.0.2.0/30 | 10.10.10.10 |
| 128.112.0.0/16 | 10.10.10.10 |

5. **(10 pts)** In class, we discussed an algorithm to create a spanning tree for a network of switches. Consider the network topology given in figure below, where each square node represents a switch, each circle node represents a host, and cost of each edge is 1 unit. Assume that each switch's ID is equal to its number (*i.e.*, switch #$k$'s ID is set to $k$).




(a) **(4 pts)** Assume the Spanning Tree Protocol (STP) has converged. In the space below, illustrate the resulting spanning tree topology.




(b) **(6 pts)** Once again, assume the Spanning Tree Protocol (STP) has converged. After that, the following events occur sequentially. Assume that forwarding tables of the switches are empty before *Event 1*. What will be the forwarding table for Switch #3, after *Event 3*.

**Event 1:** Host 1 sends a packet to Host 4

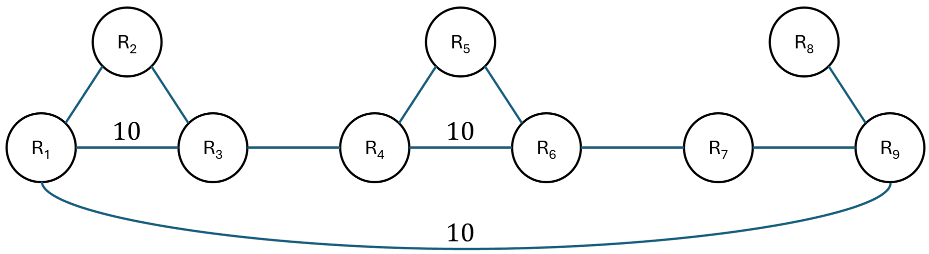
**Event 2:** Host 2 sends a packet to Host 3

**Event 3:** Host 4 sends a packet to Host 1

**S3's forwarding table**

| destination | next hop |
|---|---|
| H1 | port 2 |
| H2 | port 2 |

**6. (10 pts)** Consider the network topology shown below. The topology consists of multiple routers interconnected by full-duplex links. Each link has a static cost of 1 unit associated with it, except for the link between $R_1$ and $R_3$, the link between $R_4$ and $R_6$, and the link between $R_1$ and $R_9$ which have cost 10 unit associated with them.




Assume that routers use Bellman-Ford's distributed algorithm with poisoned reversed. Assume that all routers' routing tables were stable (*i.e.* Bellman-Ford's distributed algorithm had converged) and the link between $R_6$ and $R_7$ fails. Assume that each node broadcasts its distance vector (with possible poisoned entries) every $t$ seconds, starting from the link failure incident. Calculate how long it takes for the routing tables on $R_6$ to become stable again. Write your answer as a multiple of $t$ (*e.g.*, $123 \times t$). Justify your answer.

How long:

Explain:

The update originated from $R_6$ should reach $R_1$ and bounce back to $R_6$ for it to converge.

At time $= 0$, $R_6$'s cost to $R_7$ would be 13 and the update reaches $R_5$.

At time $= t$, the update reaches $R_4$.

...

At time $= 4t$, the update reaches $R_1$. $R_1$'s distance to $R_7, R_8$, and $R_9$ will be updated to $11, 10$, and 11, respectively.

At time $= 5t$, the update from $R_1$ reaches $R_2$ and would result in $R_2$ converging. It also reaches $R_3$, but $R_3$ need one more step to converge.

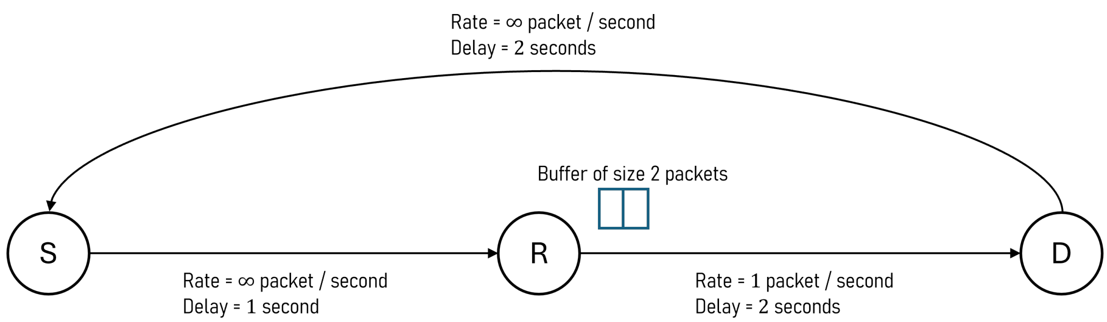
At time $= 6t$, the update from $R_2$ reaches $R_3$ and makes it converge.

...

At time $= 9t$, the update from $R_5$ reaches $R_6$ and makes it converge.

**Depending on how we count the iterations and whether we consider the first update from $R_6$ to be at time = 0 or time = $t$, the answer above can become $10t$ or $11t$.**

**7. (11 pts)** Consider two hosts S and D connected through a router R as depicted in the figure below. The capacity (*i.e.*, transmission rate) and delay of the links (*i.e.*, propagation delay) connecting S to R, R to D, and D back to S are shown in the figure. The source node S starts a TCP connection with destination D. We make the following assumptions:




- The initial congestion window (cwnd) size is 1 MSS.
- We ignore the 3-way handshake.
- The link connecting the router R to D has a buffer of size 2 packets.
- The Retransmission Timeout (RTO) is static and set to 4RTT.
- This TCP is initially in slow-start phase. Furthermore, this TCP only implements slow-start and there is no additive-increase multiplicative-decrease (AIMD) phase.
- Let the current cwnd be $k$ MSS. Upon occurrence of *Timeout* or *Triple Duplicate ACKs*, cwnd will be reduced to 1 MSS and slow-start phase restarts. Note that there is no AIMD phase, so you do not need to worry about slow-start threshold (ssthresh).
- The TCP connection from S to D is used to deliver a very large number of packets (*i.e.* host S's sending buffer is filled with infinite bytes of data).

**Note 1:** TCP does not increase the cwnd upon receiving the $1_{\text{st}}$, $2_{\text{nd}}$, and $3_{\text{rd}}$ duplicate ACKs.
**Note 2:** TCP does not take any action upon receiving the $4_{\text{th}}$, $5_{\text{th}}$, ... duplicate ACKs (*i.e.*, it does not increase or decrease the cwnd or retransmit any packet).
**Note 3:** No further assumptions are required to answer the following questions. In any case, if you feel the need to make additional assumptions, clearly specify them. Based on validity of your assumptions, you may receive partial credit. It is for you to decide how valid your assumptions are.

**(a) (5 pts)** Let $\tau_i$ denote the end-to-end delay of the $i_{\text{th}}$ packet. The average *packet end-to-end delay* of the first seven packet is defined as $\frac{\sum_{i=1}^{7} \tau_i}{7}$.

First, write your definition of end-to-end delay of *a packet* to receive a partial credit.

Then, find the average *packet end-to-end delay* of the first seven packets in this TCP flow and provide your reasoning. You do not need to simplify your final answer. To receive the full credit, it suffices to specify the values for $\tau_1, \ldots, \tau_7$ and provide your reasoning.

| | |
|---|---|
| At $t = 0$ | S receives: NA<br>S sends: $p_1$. [Will be received by R at $t = 1$]<br>R sends: NA, R's Buffer: (_, _).<br>R receives: NA, R's Buffer: (_, _).<br>D receives: NA.<br>D sends: NA. |
| At $t = 1$ | S receives: NA<br>S sends: NA<br>R receives $p_1$.<br>R sends: $p_1$ [Will be received by D at $t = 4$], R's Buffer: (_, _).<br>D receives: NA.<br>D sends: NA. |
| At $t = 4$ | S receives: NA<br>S sends: NA<br>R receives: NA.<br>R sends: NA, R's Buffer: (_, _).<br>D receives: $p_1$.<br>D sends: $a_1$. [Will be received by S at $t = 6$]. |
| At $t = 6$ | S receives: $a_1$. Updates CWND $\leftarrow 2$. Slide window to right by one.<br>S sends: $p_2$ and $p_3$. [Will be received by R at $t = 7$]<br>R receives: NA.<br>R sends: NA, R's Buffer: (_, _).<br>D receives: NA.<br>D sends: NA. |
| At $t = 7$ | S receives: NA.<br>S sends: NA.<br>R receives: $p_2$ and $p_3$.<br>R sends: $p_2$ [Will be received by D at $t = 10$], R's Buffer: (_, $p_3$). [R will send $p_3$ at t=8]<br>D receives: NA.<br>D sends: NA. |
| At $t = 8$ | S receives: NA.<br>S sends: NA.<br>R receives: NA.<br>R sends: $p_3$ [Will be received by D at $t = 11$], R's Buffer: (_, _).<br>D receives: NA.<br>D sends: NA. |
| At $t = 10$ | S receives: NA.<br>S sends: NA.<br>R receives: NA.<br>R sends: NA, R's Buffer: (_, _).<br>D receives: $p_2$.<br>D sends: $a_2$. [Will be received by S at $t = 12$]. |

| | |
|---|---|
| At $t = 11$ | S receives: NA.<br>S sends: NA.<br>R receives: NA.<br>R sends: NA, R's Buffer: $(\_, \_)$.<br>D receives: $p_3$.<br>D sends: $a_3$. [Will be received by S at $t = 13$]. |
| At $t = 12$ | S receives: $a_2$. Updates CWND $\leftarrow 3$. Slide window to right by one.<br>S sends: $p_4$ and $p_5$. [Will be received by R at $t = 13$].<br>R receives: NA.<br>R sends: NA, R's Buffer: $(\_, \_)$.<br>D receives: NA.<br>D sends: NA. |
| At $t = 13$ | S receives: $a_3$. Updates CWND $\leftarrow 4$. Slide window to right by one.<br>S sends: $p_6$ and $p_7$. [Will be received by R at $t = 14$].<br>R receives: $p_4$ and $p_5$.<br>R sends: $p_4$. [Will be received by D at $t = 16$], R's Buffer: $(\_, p_5)$. [R send will $p_5$ send at $t = 14$]<br>D receives: NA.<br>D sends: NA. |
| At $t = 14$ | S receives: NA.<br>S sends: NA.<br>R receives: $p_6$ and $p_7$.<br>R sends: $p_5$. [Will be received by D at $t = 17$], R's Buffer: $(p_7, p_6)$. [R send will $p_6$ send at $t = 15$]<br>D receives: NA.<br>D sends: NA. |
| At $t = 15$ | S receives: NA.<br>S sends: NA.<br>R receives: NA.<br>R sends: $p_6$. [Will be received by D at $t = 18$], R's Buffer: $(\_, p_7)$. [R send will $p_7$ send at $t = 16$]<br>D receives: NA.<br>D sends: NA. |
| At $t = 16$ | S receives: NA.<br>S sends: NA.<br>R receives: $p_6$ and $p_7$.<br>R sends: $p_7$. [Will be received by D at $t = 19$], R's Buffer: $(\_, \_)$.<br>D receives: $p_4$.<br>D sends: $a_4$ [Will be received by S at $t = 18$]. |
| At $t = 17$ | S receives: NA.<br>S sends: NA.<br>R receives: NA.<br>R sends: NA.<br>D receives: $p_5$.<br>D sends: $a_5$ [Will be received by S at $t = 19$]. |
| At $t = 18$ | S receives: $a_4$ Updates CWND $\leftarrow 5$. Slide window to right by one.<br>S sends: $p_8$ and $p_9$. [Will be received by R at $t = 20$].<br>R receives: NA.<br>R sends: NA, R's Buffer: $(\_, \_)$.<br>D receives: $p_6$.<br>D sends: $a_6$ [Will be received by S at $t = 20$]. |

| | |
|---|---|
| At $t = 19$ | S receives: $a_5$ Updates CWND $\leftarrow 6$. Slide window to right by one.<br>S sends: $p_{10}$ and $p_{11}$. [Will be received by R at $t = 21$].<br>R receives: NA.<br>R sends: NA, R's Buffer: $(\_, \_)$.<br>D receives: $p_7$.<br>D sends: $a_7$ [Will be received by S at $t = 21$]. |
| At $t = 20$ | S receives: $a_6$ Updates CWND $\leftarrow 7$. Slide window to right by one.<br>S sends: $p_{12}$ and $p_{13}$. [Will be received by R at $t = 22$].<br>R receives: $p_8$ and $p_9$.<br>R sends: $p_8$ [Will be received by D at $t = 23$], R's Buffer: $(\_, p_9)$ [R will send $p_9$ at $t = 21$].<br>D receives: NA.<br>D sends: NA. |
| At $t = 21$ | S receives: $a_7$ Updates CWND $\leftarrow 8$. Slide window to right by one.<br>S sends: $p_{14}$ and $p_{15}$. [Will be received by R at $t = 23$].<br>R receives: $p_{10}$ and $p_{11}$.<br>R sends: $p_9$ [Will be received by D at $t = 24$], R's Buffer: $(p_{11}, p_{10})$ [R will send $p_{10}$ at $t = 22$].<br>D receives: NA.<br>D sends: NA. |
| At $t = 22$ | S receives: NA.<br>S sends: NA.<br>R receives: $p_{12}$ and $p_{13}$.<br>R sends: $p_{10}$ [Will be received by D at $t = 25$], R's Buffer: $(p_{12}, p_{11})$ [R will send $p_{11}$ at $t = 23$]. $p_{13}$ **is dropped**.<br>D receives: NA.<br>D sends: NA. |
| At $t = 23$ | S receives: NA.<br>S sends: NA.<br>R receives: $p_{14}$ and $p_{15}$.<br>R sends: $p_{11}$ [Will be received by D at $t = 26$], R's Buffer: $(p_{14}, p_{12})$ [R will send $p_{12}$ at $t = 24$]. $p_{15}$ **is dropped**.<br>D receives: $p_8$.<br>D sends: $a_8$ [Will be received by S at $t = 25$]. |

Write your answer to part **(a)** on this page. Define the end-to-end delay of a packet to receive a partial credit. Then, write your final answer and provide your reasoning. You do not need to simplify your final answer. To receive the full credit, it suffices to specify the values for $\tau_1, \ldots, \tau_7$ and provide your reasoning.

End-to-end packet delay: the time between sender sending the packet and the receiver completely receiving that packet.

(0.5 pts) $\tau_1 = 4$.

(0.25 pts) $\tau_2 = 4$.

(0.75 pts) $\tau_3 = 5$.

(0.25 pts) $\tau_4 = 4$.

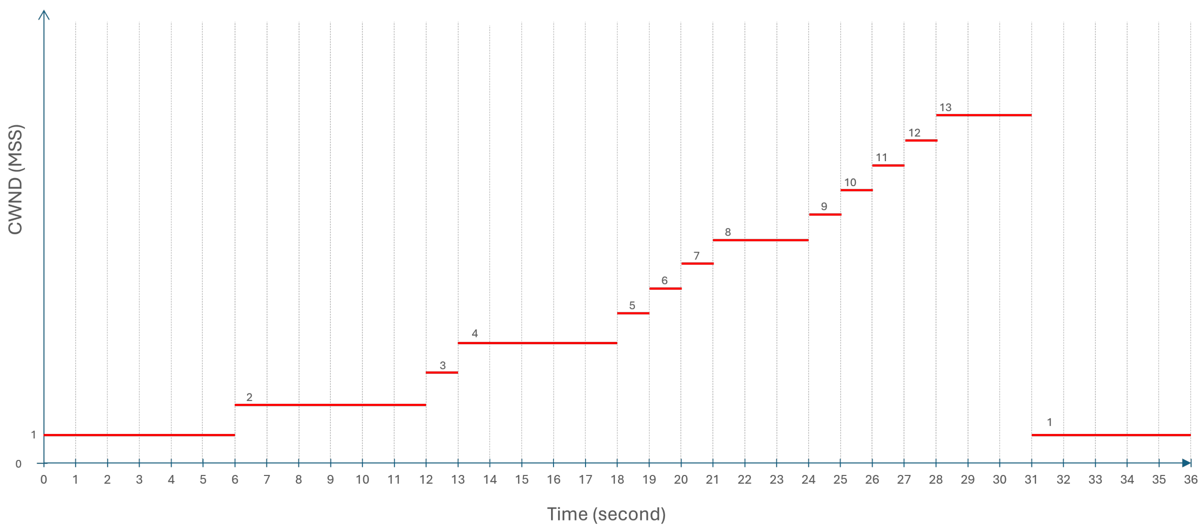
(0.75 pts) $\tau_5 = 5$.

(1 pts) $\tau_6 = 6$.

(1 pts) $\tau_7 = 7$.

Average end-to-end packet delay $= \frac{35}{7}$.

**(b) (4 pts)** Show the evolution of congestion window size (in MSS) as a function of time (in seconds) in the graph below during time $t_0 = 0$ to $t_1 = 23$ seconds. Clearly mark the Y-axis to show the size of congestion window size in packets.



**(c) (2 pts)** What is the average throughput of host S during the time $t_0 = 0$ to $t_1 = 23$ seconds (the duration of this interval is 23 seconds). First, write your definition of average throughput to receive a partial credit. Then, write your final answer as a multiple of MSS (*e.g.*, $\frac{123}{456}$ MSS) and provide your reasoning.

$\frac{15}{23}$ MSS per second.