UNIVERSITY OF TORONTO

Faculty of Arts and Science

DECEMBER 2016 EXAMINATIONS

CSC458H1F – Computer Networks

Duration - 3 hours

Examination Aids:

Non-Programmable Calculators, 1 Double-Sided Page of Notes

Name: **Student ID:**

(i) This exam is closed book and closed notes. However, you may refer to a sheet of 8.5"x11" paper (double-sided) of your own design. You can also use a non-programmable calculator.

(ii) Write your answers in the space provided on this paper. If you need extra space, you can use the reverse side of each page.

(iii) Make sure to write your name and student ID clearly on every page.

(iv) Show your reasoning clearly. If your reasoning is correct, but your final answer is wrong, you will receive most of the credit. If you just show the answer without reasoning, and your answer is wrong, you may receive no points at all.

Question	
Multiple Choice 1-6 (1 point each total 6 points)	
Definitions 7-10 (1 point each total 4 points)	
Longer Questions	
11 (4 points)	
12 (3 points)	
13 (4 points)	
14 (4 points)	
15 (3 points)	
Total 28 points	

Name: **Student ID:**

Part I - Multiple Choice Questions [6 points]

Instructions: In the following questions, check the assertion that appears to be correct. There is exactly one correct assertion per question. Checking the correct assertion will earn you one point. If you check an incorrect assertion, or if you check more than one assertion per question, you will not earn any points for that question.

1. Software-Defined Networking (SDN). Which of the following statements is true about software-defined networking?

(a) In a software-defined network, switches are responsible for choosing the shortest path to reach each destination.
(b) Horizontal open interfaces are one of the main reasons change and innovation is easier in SDN.
(c) In a software-defined network, switches are replaced by a centralized network operating system also called the network controller.
(d) In a software-defined network, a socket is the API between the controller (network OS) and the data plane (switches).

2. TCP. Which one of the following statements is true about TCP?

(a) TCP relies on packet drops to identify receiver window size.
(b) In TCP, the two end-hosts authenticate each other during the three-way handshake.
(c) TCP creates a full duplex connection between the two end-hosts.
(d) When one of the two hosts is finished transmitting data, it sends an RST message to the other side to close the connection.

3. Early Congestion Notification (ECN). Which of the following is true about ECN?

(a) ECN allows the router to notify the source about congestion without dropping the packet.
(b) ECN allows routers to notify the source about congestion by increasing the drop rate.
(c) ECN drops packets with probability 1 when the router's average queue length is greater than the maximum threshold value.
(d) ECN can eliminate congestion in a network.

4. Middleboxes. Which of the following statements is true?

(a) In addition to filtering, some firewalls can also be used for traffic shaping.
(b) An application gateway can be used to support IPv6 in a network that only understands IPv4.
(c) A firewall acts like a server and a client at the same time.
(d) A software-defined network controller is a type of middlebox.

5. Error Correction/Detection. Consider a set of messages which have hamming distance of 3.

(a) We can correct any 2 bit errors.
(b) We can correct any 3 bit errors.
(c) We can detect any 2 bit errors.
(d) We can detect any 3 bit errors.

Name: **Student ID:**

6. Routing. Which of the following is true about routing in the Internet?

(a) Distance vector algorithms find routes in a centralized manner, whereas link state routing algorithms find routes in a distributed manner.
(b) BGP is an example of link state routing protocols.
(c) Distance vector routing relies on the exchange of information between neighboring routers to calculate the routes.
(d) For link state routing algorithms, we do not need to define initial link costs (weights) since the algorithm can compute the weights.

Part II – Definitions [4 points]

Describe each of the following terms/concepts clearly and concisely (in at most 3-4 sentences). For each of these terms, explain the context they are defined at – which protocol(s) they are related to, when/where they are used, etc. – and give examples where possible.

7. Random Early Detection (RED) [1 point]

8. OpenFlow [1 point]

9. Slow Start [1 point]

10. Syn Flooding [1 point]

Part III - Longer Questions [16 points]

11. **TCP (4 points).** Suppose your boss calls you up while visiting a client and says he left a 400KB file on a server at the office, and he needs you to send it to him. The connection speed between the office and his current site is 2Mbps.
    (a) Assuming that all bandwidth is consumed by your data (e.g., there aren't any packet

headers, there's no other traffic on the link, etc.) and that you can immediately start sending data at the maximum rate, how long will it take to transmit the file?

(b) Now assume that you are sending the file via TCP, with a maximum segment size of 1000 bytes. You must follow TCP's transport algorithm, so you start transmitting the file at a low rate. How many network round trip times (RTTs) will it take to transmit the entire file? Again, assume that all bandwidth is consumed by your data.

(c) Let's now consider how TCP reacts when it encounters packet loss. Your boss now wants you to transmit a graphic file for his presentation. The file size is 1GB. When TCP reaches a transmission rate of 32,000 bytes per second, you notice that a single packet is lost. Assuming that you don't currently have any other packets in transmission, what is the instantaneous rate TCP will send at after noticing that loss? How long (in terms of RTTs) will it take to reach 32,000 bytes per second again?

(d) In the previous question c, would anything be different if all packets were lost after you reached 32,000 bytes per second, instead of just a single packet? If so, what is the next transmission rate TCP will send at in this example, and how long (in terms of RTTs) would it now take to reach 32,000 bytes per second again?

Name: **Student ID:**

12. Longest Prefix Matching (3 points). Consider the following routing table, and let us assume the router uses longest prefix matching to forward packets.

IP Address	Netmask	Next Hop	Interface
0.0.0.0	0.0.0.0	5.10.1.1	eth0
12.1.0.0	255.255.0.0	12.1.0.1	eth1
12.1.1.0	255.255.255.0	12.1.1.1	eth2
12.1.1.200	255.255.255.255	12.1.1.200	eth3

Which interface would the router use to forward packets destined to each of the following IP addresses:

(a) 12.1.1.200

(b) 12.1.1.201

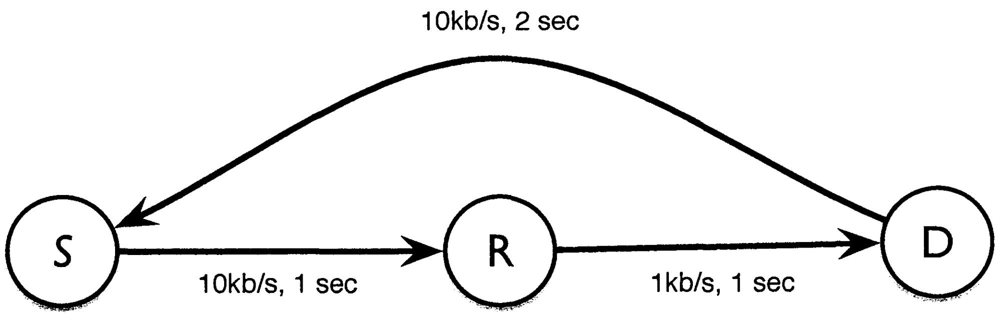
(c) 12.1.0.20

(d) 12.1.1.20

(e) 128.12.92.53

(f) 192.168.0.1

13. Buffer Sizing (4 points). Consider two end-hosts ***S*** and ***D*** connected via router ***R*** as in the following figure. The link connecting ***S*** to ***R*** has a delay of 1 second and a transmission rate of 10kb/s. The link connecting ***R*** to ***D*** also has a delay of 1 second, but the transmission rate is 1kb/s. The link connecting node ***D*** to node ***S*** has a delay of 2 seconds and operates at 10kb/s.



Consider a single TCP flow starting at node ***S***, and sending packets over router ***R*** to destination node

D. The acknowledgement packets come back to ***S*** over the link directly connecting node ***D*** to node ***S***. We assume that each TCP packet (including all the headers) is 1kb in size, and router ***R*** has a buffer that can hold four TCP packets. Also, we ignore any processing delays at any of the nodes in this system. To simplify the analysis, we also assume that the TCP flow is always in congestion avoidance, i.e., it never enters slow-start.

(a) What is the maximum value for congestion window size (in packets) that would be achieved by this TCP flow?

(b) Plot the variations in congestion window size as a function of time. Start at CWND = 0.

(c) From you graph in Part (b) what is the average congestion window size over time?

(d) What is the average throughput (bits per second) of this flow between ***S*** and ***D***?

Name: **Student ID:**

14. Big MAC (4 points). This question explores whether the Internet could be designed using only names (like www.computernetworkingrocks.com) and MAC addresses, without a need for IP addresses. Suppose each network adapter, for any link technology, has a unique MAC addresses from a single address space (such as 48-bit MAC addresses used for Ethernet devices), and these addresses were used Internet-wide for communication between end hosts. Suppose that the Domain Name System (DNS) is changed to return MAC addresses rather than IP addresses, in response to queries.

(a) In today's Internet, why is it difficult to support continuous communication with a host while it moves? Why would an Internet based on MAC addresses have the potential to make mobile hosts easier to support?

(b) Today's local area networks have several "boot-strapping" techniques, such as DHCP, ARP, and MAC learning. In a world without IP addresses, which of these techniques would still be necessary? Explain your answer.

(c) How would moving to an Internet based on MAC addresses affect the size of the forwarding tables in the network nodes (i.e., routers or switches)?

(d) How would the new design, based on MAC address, affect users' privacy?

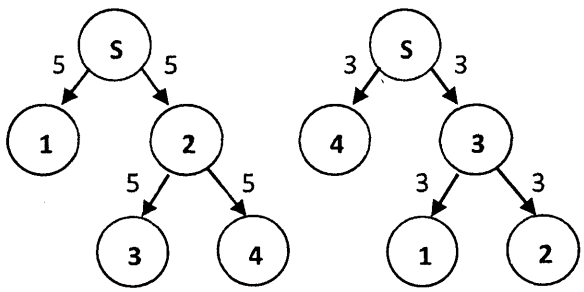
15. Peer-to-Peer Systems (3 points). This question explores the scalability of peer-to-peer systems, compared to client-server systems, for live streaming applications, such as Internet television. The

Name: Student ID:

server wants to deliver a video stream to six (6) receivers, where each receiver receives the stream live at the same streaming rate. A higher streaming rate corresponds to higher video quality, so the system should be designed to stream the video at the highest possible rate. The server has a link connected to the Internet with **30 Kb/second** uploading bandwidth. All receivers have bidirectional links to the Internet–with **50 Kb/second** uploading bandwidth and **50 Kb/second** downloading bandwidth. Assume the core of the Internet has ample bandwidth.

(a) What is the maximum streaming rate that the system can support in a client-server configuration (i.e., where the receivers do *not* upload to each other)? That is, what is the rate of the stream received by each of the six receivers?

(b) Now, suppose the receivers can upload to each other. A receiver can upload a stream of rate ***r*** only if he receives the same stream of rate ***r*** from another receiver or the server. For example, streaming trees shown as follows (***S*** is the source and the other nodes numbered from 1 to 4 represent the receivers):



These trees combined achieve a total streaming rate of 8 Kb/second to four receivers (with two trees of streaming rates 5 and 3 respectively). The server transmits two copies of the stream at rate 5 and two at rate 3, consuming a total of 16 Kb/second of upload bandwidth. In our problem, with the upload and download capacities listed at the beginning of the question, what is the maximum streaming rate that can be achieved? Draw a set of trees (one or more trees) that achieve your streaming rate. You need not worry about minimizing the depths of the trees.

(c) Now, suppose the receivers have a smaller uploading capacity of just **10** Kb/second. What is the maximum streaming rate that can be achieved in this case? Draw one or multiple streaming trees that support your streaming rate.

Name: **Student ID:**

Total Marks = 28 points