Total time: 50 minutes

Name: _____

WPI username: _____

CS 3516 – Mid-Term Exam - A Term, 2018

This exam is worth **32 points + 3 bonus points**. Please write concisely, using only the space provided below. Extremely long answers to questions will be interpreted as guessing and will be penalized. Please specify the question/sub-question you are answering.

1. (2 points, 3 minutes) **What is propagation delay and what is transmission delay?**

   Propagation delay is the **amount of time it takes for data to go from one entity to another (1 point)**. Transmission delay on the other hand is **the time it takes for a host to push the data on to the link (1 point)**

2. (2 points, 3 minutes) **What are persistent HTTP and what are non-persistent HTTP connections?**

   Persistent HTTP connection **gets all the objects on a page with a single TCP connection (1 point)**, while non-persistent HTTP connection **creates a new TCP connection for each object on a page (1 point)**.

3. (2 points, 3 minutes) **Give two issues with having sender timeout value too long for RDT3.0 protocol?**

   The issues are, the protocol will be:
   - Slow to react (retransmit) to packet loss (1 point)
   - Reduce overall throughput/utilization of the protocol (1 point)
   - Please see the page 21 in the following slides: http://users.wpi.edu/~yli15/courses/CS3516Fall18A/slides/CS3516-12-rdt1-3.pdf

4. (2 points, 5 minutes) **Using a couple of sentences, discuss why we need both source and destination port numbers in transport layer protocols, e.g., in UDP.**

   The port numbers are needed to allow the destination host to pass the application data to the correct process running on the destination end system to perform the demultiplexing function (1 point). The source port allows the destination to create response messages to the correct port on the source side (1 point).

5. (19 points, 25 minutes) **In modern packet-switched networks, including the Internet, the source host segments long, application-layer messages (for example, an image or a music file) into smaller packets and sends the packets into the network. The receiver then reassembles the packets back into the original message. We refer to this process as *message segmentation*. Figure 1 illustrates the end-to-end transport of a message with and without message segmentation. Consider a message that is $8 \cdot 10^6$ bits long that is to be sent from source to destination in Figure 1. Suppose each link in the figure 1 is 2 Mbps. Ignore propagation, queuing, and processing delays.**

   (a) (4 points) Consider sending the message from source to destination without message segmentation. How long does it take to move the message from the source host to the first packet switch? Keeping in mind that each switch uses store-and-forward packet switching, what is the total time to move the message from source host to destination host?

   (b) (4 points) Now suppose that the message is segmented into 800 packets, with each packet being 10,000 bits long. How long does it take to move the first packet from source host to the first switch? When the first packet is being sent from the first switch to the second switch, the second packet is being sent from the source host to the first switch. At what time will the second packet be fully received at the first switch?

   (c) (5 points) How long does it take to move the file from source host to destination host when message segmentation is used? Compare this result with your answer in part (a) and comment.

Total time: 50 minutes

(d) (2 points) Will the delay calculated in parts (a), (b), and (c) alter if Source was the destination and Destination was the source? Justify your answer.

(e) (2 points) In addition to reducing delay, what are reasons to use message segmentation?

(f) (2 points) Discuss the drawbacks of message segmentation.

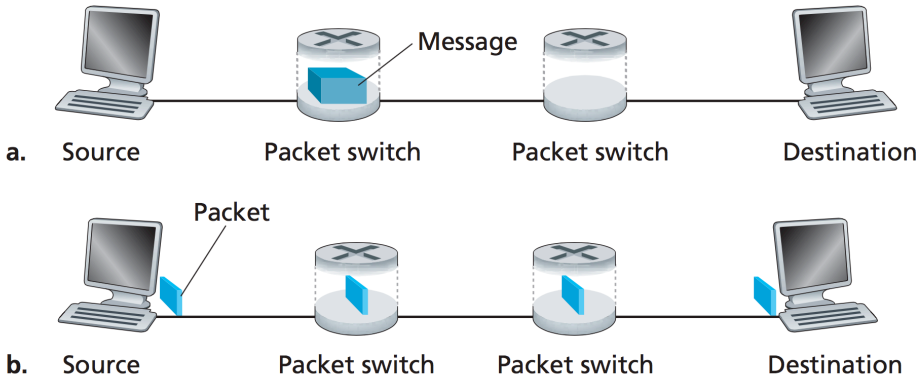

Figure 1: End-to-end message transport: (a) without message segmentation; (b) with message segmentation

(a) Time to send message from source host to first packet switch = 8×10^6 / 2×10^6 sec = 4 sec (2 points). With store-and-forward switching, the total time to move message from source host to destination host = 4 sec × 3 hops = 12 sec. (2 points)

(b) Time to send 1st packet from source host to first packet switch = 1×10^4 / 2×10^6 =5 ms (2 points). Time at which 2nd packet is received at the first switch = time at which 1st packet is received at the second switch = 2 × 5 ms = 10ms. (2 points)

(c) Time at which 1st packet is received at the destination host = 5ms × 3 hops = 15ms (2 points). After this, every 5 ms one packet will be received; thus time at which the last (800th) packet is received = 15 ms + 799 × 5ms = 4.01s (2 points). It can be seen that delay in using message segmentation is significantly less (almost 1/3) (1 point).

(d) Not really, since the transmission delays are the same. (2 points)

(e) i. Without message segmentation, if bit errors are not tolerated, if there is a *single bit error*, the whole message has to be retransmitted (rather than a single packet) (1 point). ii. Without message segmentation, huge packets (containing HD videos, for example) are sent into the network. Routers have to accommodate these huge packets. Smaller packets have to queue behind enormous packets and *suffer unfair delays* (1 point).

(f) i. Packets have to be put in sequence at the destination (1 point); ii. Message segmentation results in many smaller packets. Since header size is usually the same for all packets regardless of their size, with message segmentation the total amount of header bytes is more (1 point).

For (e). and (f), please see the page 8 in the following slides:
http://users.wpi.edu/~yli15/courses/CS3516Fall18A/slides/CS3516-13-midterm-review.pdf

6. DNS and Locality-Aware Load Balancing

(a) (5 points, 5 minutes) **Suppose that a user accesses google.com through your local DNS server, describe in your own words the process of resolving the domain name google.com to IP address of google server.**

Taking Google as an example, once the hostname www.google.com is loaded, the browser first resolves the hostname, by sending the query to the local DNS server (LDNS) (1 point). If LDNS has a cached google

Total time: 50 minutes

server IP, it returns it to the user (1 point). The LDNS will query the root name server, TLD DNS server, and authoritative name server of Google to get the Google server IP address (3 points).

(b) (3 bonus points, 5 minutes) **Web services such as Google have employed DNS to perform locality-aware load balancing. For example, when you send a search query to www.google.com, Google will use the DNS queries to figure out that you likely reside in Worcester, and respond with an IP address of a (front-end) server that is close to you (say, located in Boston metro area) to handle your search query. Suppose that you access google.com through your local DNS server, Google servers can only see the IP of your Local DNS server instead of your machine, thus can only return an IP address of a google web server that is geographically closer to the LDNS is returned to the user.**
(Here you can assume that given an IP address, Google can figure out where the machine with that IP address is approximately located (see e.g., www.maxmind.com).)

Can you come up with a scheme with which Google can utilize the IP address of your machine to determine where you are located, and then direct your query (or other requests) to a server closer to you? Briefly describe how your scheme works. [Hint: HTTP re-direction]

After the DNS resolution, the user obtains the IP address of the server (e.g., a web or video server) (1 point). Then it performs an HTTP requests directly to the server. Via the HTTP request, Google knows the IP address of the user. Then, it can identify the user's location, thus closer servers to the users (1 point). By performing HTTP redirections, the users can be redirected to a closer server (1 point).