Take-Home Quiz I

Due Friday Oct 20 11:59pm

Csci4211: Introduction to Computer Networks
Fall 2017
Prof. Zhi-Li Zhang

Last Name: **First Name:**

Student Id.

Instructions:

1. This is a **open-book** and **open-note** quiz.
2. There are **four** questions in total, each of which has several sub-questions, with a total of 100 points. You have roughly **four days** to answer the questions.
3. Please make sure to write down your name and student id. on your answer sheets.
4. Partial credit is possible for an answer; please include intermediate steps as appropriate. Please try to be as concise and make your exam as neat as possible. We *must* be able to read your handwriting in order to be able to grade your exam.
5. **Please work on the quiz *individually, by yourself only!* No discussion among the students in the class, or with others, is allowed. If you find your answers from the Internet, please cite your sources. Any violation of the University's *Student Conduct Code* will be reported to both the department and the University, and you may be suspended or expelled! Please note that if you let another student to copy your answers, you are also in violation of the University's *Student Conduct Code.***
6. Good luck. Enjoy!

---

1. General True/False Questions: (15 points total. approx. 5 minutes)

Please record your answer to the following **true or false** *questions.*

Note: in answering the "True or False" questions below, explanations are not needed. We provide them here to help you understand why you should choose 'False" to all five questions.

a. (3 points) **True or False:** Circuit switched networks do not suffer from propagation delay.
Answer: False! Propagation is the time it takes for light or electrons to travel across a wire, independent of whether packet switching or circuit switching is used.

b. (3 points) **True or False:** Packet switching is used in computer networks because it is fast.
Answer: False! Packet switching is designed for "bursty" data applications to take advantage of statistical multiplexing gains. It is not necessarily faster than circuit switching.

c. (3 points) **True or False:** That HTTP/1.1 by default uses a *persistent* TCP connection means that HTTP/1.1 is a *stateful* application-layer protocol, i.e., the HTTP/1.1 client and server will maintain some states to remember the previous HTTP requests/replies.
Answer: False. HTTP/1.1 is stateless, as it does not maintain state across multiple HTTP/1.1 requests from the same client.

d (3 points) **True or False:** SMTP is an example of "request-n-reply" application layer protocol.
Answer: False. SMTP uses "session"-based, with explicitly session set-up (HELLO) and termination (QUIT) commands.

e. (3 points) **True or False:** TCP packets carry the transport layer port numbers so that the end hosts can recover lost packets using the port numbers.
Answer: False

2. DNS and UDP vs. TCP (28 points total. Approx. 20 minutes)

a. (3 points) Provide one main reason why the root domain name servers use the *iterative* DNS query mechanisms instead of the *recursive* DNS query mechanism.

Answer:
To reduce the load on the root domain name servers and to increase the scalability of the DNS infrastructure

b. (8 points) When your laptop sends a DNS query to the local DNS server, it uses the *recursive* DNS query. Provide two main advantages of using the recursive DNS query here. What could be a possible disadvantage?

Answer:
Please note that "it" refers to the local DNS server. In this case, *Two Advantages: a) This allows the local DNS reply to cache DNS replies so that it can response to the requests to the cached DNS names faster; b) Caching at the local DNS server also reduces the overall load of the DNS infrastructure*

Disadvantage: the local DNS server may become the potential performance bottleneck.

If you considered "it" to refer to your laptop (or rather, the name resolver daemon on your laptop, and if your answers include the following, you won't be penalized! *Two Advantages: a) This allows the name resolver to cache DNS replies so that it can response to the requests to the cached DNS names faster; b) The name resolver can handle packet losses and reliable data transfer over UDP for applications, relieving them of the complexity of managing reliable data transfer.*

Disadvantage: the name resolver may become the potential performance bottleneck.

c. (4 points) A local DNS server often sends out multiple DNS queries on behalf of many local hosts. When its receives a reply from an authoritative DNS server, how does it figure out which query this reply corresponds to?

Answer:
A 16-bit transaction identifier is included in each query and copied to the response so that a local DNS server can match answers to the corresponding query

d. (5 points) Why does DNS use UDP instead of TCP? What happens if the DNS query sent from your laptop to the local DNS server is lost?

Answer:
For fast look up: UDP is much faster. TCP is slow as it requires 3-way handshake The DNS

client in my laptop will repeat the query, if no response arrives within a short time

e. (8 points) Suppose that instead of UDP, TCP is used to deliver DNS queries and replies between your laptop and the local DNS server. Assume that the propagation delay from your laptop to your local DNS server (and vice versa) is 5 ms. Further assume that the transmission or other delays are negligible. What is the *minimal* time it takes for your laptop to receive a DNS reply for a DNS query it has just issued to the local DNS server? Briefly justify your answer.

Answer:
Assumption: if the DNS reply has been cached by the local DNS server, then the minimal time it takes to receive the DNS reply is the RTT between your laptop and the local DNS server

Two options:

- *DNS Case-1: 3-way handshaking + DNS reply (see Fig. 1):* $3 \times 5 + 5 = 20ms$
- *DNS Case-2: 3-way handshaking + DNS request + DNS reply (see Fig. 2):* $3 \times 5 + 5 + 5 = 25ms$

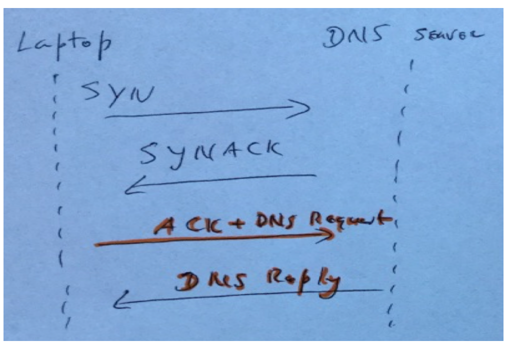

Figure 1: DNS case 1

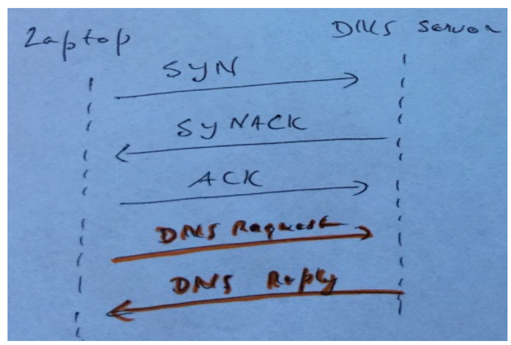

Figure 2: DNS case 2

3. TCP Connection Management (25 points total. Approx. 20 minutes)

The following figure shows the control messages sent among the client and the server under normal operations using the *three-way handshake* protocol. (Note: in `SYNACK(y,x) and ACK(x,y)`, the first number is the sequence number of the message, the second number is the acknowledgment number, i.e., the sequence number of the message being acknowledged.)

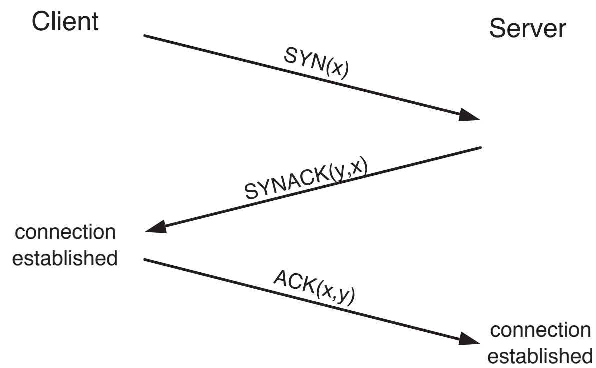

Figure 3: TCP 3-way handshake.

a. (6 points) Consider the following scenario (see Figure 4) where the `ACK(x,y)` message sent by the client is lost during the transmission. What will happen at either the client side or the server side?

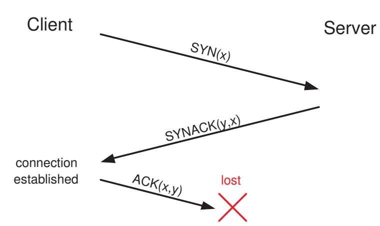

Figure 4: Figure for Question 3a.

Answer:
Two options:

- *Server will timeout for not receiving the ACK. Thus, it resends SYNACK(y,x); Client: receives the retransmitted SYNACK from the server and it retransmits the ACK(x,y)*
- *When the client sends the first data packet the server receives the acknowledgment for that SYNACK packet.*

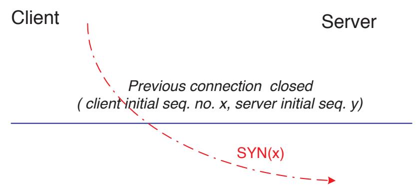

Figure 5: Figure for Question 3b.

b. (7 points) Suppose that by now the connection in **a.** (where the client used the initial sequence no. `x`, and the server used the initial sequence no. `y`) has been closed. An old, duplicate message `SYN(x)` now pops up at the **server** side (see Figure 5). First, can this scenario happen at all? Second, in response to this `SYN(x)` message, what will the *server* do? Briefly explain your answers to both questions.

Answer:
Yes, e.g: a duplicated SYN(x) packet from the previous connection, which experienced some delay due to network congestion. Then, it was received by the server after the previous connection is closed. Server: sends a SYNACK(y', x). The server has no way to distinguish between a SYN packet from a new connection and a duplicated SYN packet from an old connection.

c. (6 points) Suppose that after what happened in **3.b.**, another old, duplicate message `ACK(x,y)` from the previous connection pops up at the **server** side (see Figure 6). In response to this `ACK(x,y)` message, what will the server do? Briefly explain your answer.

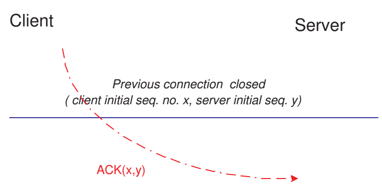

Figure 6: Figure for Question 3c.

Answer:
Server will ignore it. As from the seq. number "y", the server can tell that it's for an old connection.

d. (6 points) Briefly explain why TCP employs the *timed wait* mechanism when tearing down a (duplex) TCP connection. [See, e.g., Slides 27-28 of the lecture notes "Transport Layer Part 1", where the client side sends the first `FIN` message to indicate to the server that it has no more data to transmit; and when it receives the `FIN` message from the server, the client must enter into the `TIME_WAIT` state before fully closing the duplex TCP connection.]

Answer:
The `TIME_WAIT` *is used to address the two-army problem: the client waits for enough time to make sure that the server has received the acknowledgement of its terminating connection request (FIN), in case the ACK from the client is lost, the server resends the FIN, and the client will be able to resend the ACK again.*

4. Reliable Data Transfer: Stop-n-Wait, Selective Repeat and Protocol Efficiency, (32 points total; Approx. 30 minutes)

a. (4 points) In the **Stop-&-Wait** protocol – i.e., the `rdt3.0` protocol discussed in the textbook – (also known as the *alternating-bit* protocol), why is it necessary to introduce sequence numbers?

Answer:
To differentiate between retransmission and new packets.

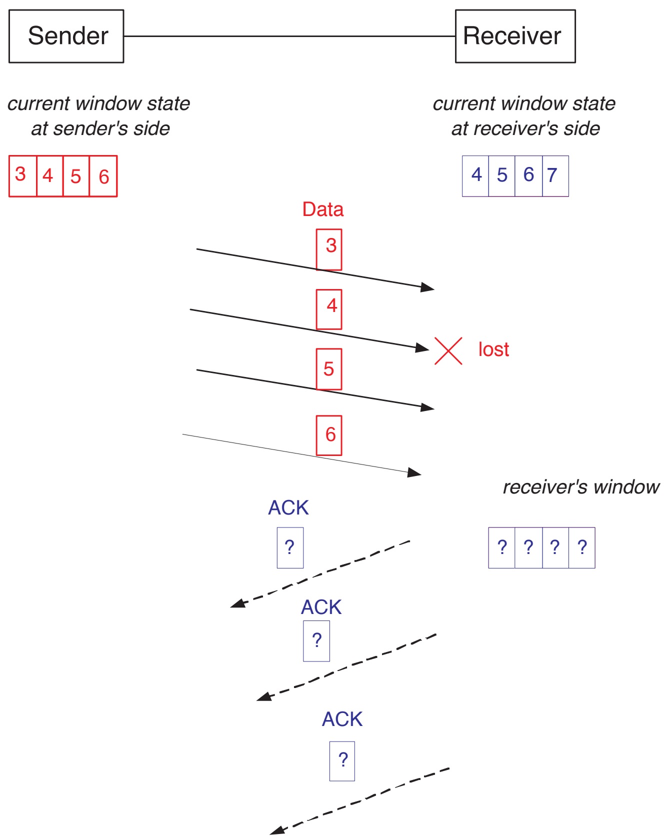

Figure 7: Figure for Questions **4.b** and **4.c.**

b. (12 points) Consider the scenario in Figure 7 where the sender and the receiver are directly connected by a duplex physical link, and they employ the *Selective Repeat* protocol for reliable data transfer across the link. Assume that the sequence numbers are represented with 3 bits, and a window size of $W = 4$ is used in the Selective Repeat protocol. Figure 7 gives a snapshot of the data transfer, where the sender's and receiver's windows at the moment are shown. The sender sends the 4 packets as allowed by the sender's window. The receiver receives three of them, with packet no. 4 lost during the transmission. Describe the action at the receiver. In particular, specify what the receiver will do with packet no. 3, what acknowledgments will be sent by the receiver and what the receiver's window looks like afterward.

Answer:
Packet 3 is discarded (duplicated packets); Receiver sends ACK for packets 3, 5 and 6

4|5|6|7*: receiver window (receiver buffers packets 5 and 6)*

c. (6 points) What will the sender's window look like after it has received the three acknowledgments as indicated in Figure 7? How many *new* packets can it transmit afterward?

Answer:
4|5|6|7*: sender window (ACK received for packets 5 and 6); sender can transmit only 1 new packet*

d. (10 points) Consider the scenario in Problems **4.b** and **4.c** , where the sender and the receiver are *directly connected* by a duplex physical link, and they employ the Selective Repeat protocol for reliable data transfer across the link, with a window size $W = 4$. Suppose that the data rate of the link is 10 Gbps, and the propagation delay is 36 microseconds (μs). The data packets transmitted across the link is of fixed size: 10 KB. [For simplicity of calculation, in the following, please assume that 1 Gbps = 10^9 bits per second, and 1 KB= 10^3 bytes. Also note that 1 byte (B) = 8 bits, and 1 microsecond (μs) = 10^{-6} second. You can assume that the time to transmit the acknowledgment packets are negligible, namely, 0.] Please answer the following questions. i) What is the maximum utilization of the link? ii) If we want to fully utilize the link as much as possible (i.e., with close to 100% utilization), what should the window size be? And iii) what is the minimum number of bits we need to represent the sequence number?

Answer:
i) $w = 4; R = 10Gps = 10^{10} bits/sec; RTT : 2 \times 36 \times 10^{-6} = 72 \times 10^{-6}; L = 10KB = 80,000 bits; L/R = 8 \times 10^{-6}$

$$U = \frac{4 \times L/R}{RTT + L/R} = \frac{4 \times 8}{72 + 8} = 0.4$$

ii) $1.0 = (w \times 8)/(72 + 8) \Rightarrow w = 10$

iii) *5 bits - the minimum number of bits to represent the sequence number space*