# Problem Set 2

## CS352 Internet Technology

## Instructions Please read and follow these instructions carefully. 1. You must work on this problem set individually. 2. Please complete your answers to the questions below and upload your responses to Canvas as a single PDF file. 3. Your answers (preferably typed up rather than handwritten) must be clear, legible, and concise. If we cannot understand your answer with reasonable effort, you will get zero credit. 4. If you leave a question out or clearly state "I don't know", you will receive 25% of the points for that question. 5. We care not just about your final answer, but also how you approach the questions. In general, if you show us your reasoning for your answers, we will do our best to provide partial credit even if your final answer is incorrect. However, if you provide no reasoning, and your answer turns out to be incorrect, you will typically receive zero points. So, please explain yourself. 6. You are free to discuss the problem set on Piazza or through other means with your peers and the instructors. You may refer to the course materials, textbook, and resources on the Internet for a deeper understanding of topics. However, you cannot lift solutions from other students or from the web. Do not post these problems to question-answering services like Chegg. All written solutions must be your own. We run sophisticated software to detect plagiarism and carefully monitor student answers. 7. There is a due date/time but no "time limit" on problem sets. That is, you may take as long as you need to work on problem sets, as long as you submit them on time. Further, you may make an unlimited number of submissions on Canvas. 8. As a response to the last question of this problem set, please specify who you collaborated with, and also include all the resources you consulted to answer questions in this problem set, including URLs of pages you visited on the Internet. Also specify which question and aspect you got help with. Please be as thorough and complete as possible here. It is mandatory to answer this question. 9. If you have any questions or clarifications on the problem set, please post them on Piazza or contact the course staff. We are here to help.

# Questions

1. **TCP vs. UDP (2 points)**. When is it beneficial for an application to use TCP as its transport, and when UDP?

```
UDP is suitable for loss-tolerant and speed critical applications, whereas TCP is
best used when data must be transmitted without loss.
```

2. **Computing the checksum (2 + 2 = 4 points)**. In this problem, we will get practice with computing the one's complement of the one's complement sum over a sequence of data values. We will use the exact same algorithm used to compute TCP/UDP checksum, but over a sequence of 4-bit chunks rather than 16-bit chunks.

   a. Consider the following data bits. What is the (4-bit) checksum of this (8-bit) data? (2 points)

   `0101 1100`
   $0101 + 1100 = 10001$
   Extract 1 bit on the left end.
   $0001 + 1 = 0010$
   The complement of $0010$ is $1101$, and therefore the checksum is $1101$.

   b. Consider the following data bits. What is the 4-bit checksum of this (12-bit) data (Note: you may be able to reuse calculations from part (a).) (2 points)
   `0101 1100 1001`

Reuse $0101 + 1100 = 10001$.
$10001 + 1001 = 11010$
Wrap around leftmost bit to right side and add.
$1010 + 1 = 1011$
Find the complement / checksum = $0100$.

3. **Listing and identifying sockets. (5 points)** As we saw in a demo in class, the ss command may be used to display the listening connections and the established connections on a machine. Suppose you see the sample output shown below.
   | Netid | State | Local Addr:Port | Peer Addr:Port | Users |
   | ----- | ------ | ------------ | --------- | -------------------- |
   | udp | UNCONN | 0.0.0.0:8003 | 0.0.0.0: | *(pid=29604,fd=3)* |
   | *tcp* | *LISTEN* | *0.0.0.0:8003* | *0.0.0.0:* | (pid=28832,fd=3) |

| Netid | State | Local Addr:Port | Peer Addr:Port | Users |
|---|---|---|---|---|
| tcp | ESTAB | 127.0.0.1:8003 | 127.0.0.1:47468 | (pid=28832,fd=4) |
| tcp | ESTAB | 127.0.0.1:47468 | 127.0.0.1:8003 | (pid=29910,fd=3) |

The output above lists all the connections on the machine where the ss command is run. Helpfully, the last column of the output, entitled Users, also shows the process ID (pid) of the application process associated with the socket and the number of the file descriptor (fd) associated with the socket

(a) Suppose a TCP packet enters the machine, destined to TCP port 8003 corresponding to an established connection. Can you identify the pid and socket fd, if any, corresponding to the socket where this packet is demuxed to? (1 point)

$(pid = 28832, fd = 4)$

(b) Suppose a TCP packet enters the machine, destined to TCP port 8003, corresponding to a fresh connection just being initiated by a client through a connect() call. Can you identify the pid and socket fd, if any, corresponding to the socket where this packet is demuxed to? (1 point)

$(pid = 28832, fd = 3)$

(c) Suppose a TCP packet enters the machine, destined to TCP port 47468, corresponding to a fresh connection being made by a client through the connect() call. Will this client's connect() succeed? Why or why not? (2 points)

```
This connect() call will fail because the port is already in use by another socket.
```

(d) Suppose a UDP packet enters the machine, destined to UDP port 8003. Can you identify the pid and socket fd, if any, corresponding to the socket where this packet is demuxed to? (1 point)

$(pid = 29604, fd = 3)$

4. **When are retransmission timeouts useful? (0.5 * 4 = 2 points)** Say YES or NO to each part of the question below. Do retransmission timeouts (followed by retransmission) help a sender ensure that a receiver was delivered a piece of data when:

(a) the data was lost? **YES**

(b) the data was corrupted? **YES**

(c) the ACK was lost? **YES**

(d) the ACK was delayed? **NO**

5. Efficiency gains with pipelined reliability protocols (2 points). Why do pipelined reliability protocols provide more throughput than stop-and-wait reliability protocols?

```
Pipelined reliability protocols allow for multiple packets to be in-flight at once,
without needing to wait for an ACK. Stop-and-wait protocols wait for a response
from the receiver before sending the next packet.
```

6. Sender and receiver view in sliding window protocols (8 points). Suppose a sender and receiver agree to use a window size of 4. Further, assume that the set of available sequence numbers goes from 0 to 9, and then rolls back to 0.

(a) What was the latest sequence number for which the sender has received an ACK from the receiver? (1 point)
**6**

(b) What is the latest sequence number that the sender has transmitted? (1 point)
**0**

(c) What was the latest sequence number whose ACK was sent by the receiver? (1 point)
**6**

(d) What is the latest sequence number that will be accepted to be buffered by the receiver? (1 point)
**0**

(e) Is it possible that the following view of the window at the receiver can happen simultaneously with the sender's view shown above? Why or why not? (1 + 1 = 2 points)

```
Yes, it is possible when the sender's window contains [7,8,9,0] and the receiver
has ACKed all, but the ACKs have not yet reached the sender. Therefore, the
receiver has moved the window forward, but the sender still believes the data is in
flight.
```

(f) Is it possible that the following view of the window at the receiver can happen simultaneously with the sender's view shown above? Why or why not? (1 + 1 = 2 points)

```
No, it is not possible. The receiver window can not be behind the sender window
because it only advances upon receiving data from the sender.
```

7. **TCP byte-based sequence numbers (8 points)**. Consider the TCP segment structure shown in slide 15/16 of lecture 13. Suppose the sequence number field on a TCP packet is 46005. The packet carries 102 bytes of (application-layer) data. The TCP header size on this packet is 20 bytes.

(a) What is the sequence number of the last application byte contained in the packet above? (2 points)
**46087**

(b) What is the value of the *acknowledgment number* field on the ACK that the receiver generates for the packet above? (2 points)
**46088**

(c) Do you have sufficient information in this question to determine the value of the *sequence number* field of the ACK packet that the receiver generates for the packet above? If so, what is the sequence number? If not, why not? (2 points).

```
No, because the ACK value is not included in the packet received.
```

(d) If the size of the TCP header were larger (e.g., 24 bytes, due to additional TCP options), would your answers to parts (a)–(c) change? Why or why not? (2 points)

```
Yes. The packet would hold less data.
Therefore, 7a and 7b would change, but 7c would not.
```

8. **TCP retransmission strategies (5 points).**
   (a) What is the advantage of selective repeat over a go-back-N retransmission strategy? What is a disadvantage? (2 points)

```
With selective repeat, only corrupted data will discarded by the receiver, and then
retransmitted by the sender. Any other data is stored in the buffer. The main
advantage is that it reduces of unnecessarily transmitted data. The main
disadvantage is that there is a greater overhead that comes from transmtting the
sequence numbers that were or were not received.
```

(b) Among the selective repeat strategies, what is an advantage of selective acknowledgments over cumulative acknowledgments? What is a disadvantage? (3 points)

```
The advantage of selective ACKs is that they are more accurate. However, it has
more overhead and as a result, uses more memory in the header of the packet.
```

9. **Which data will be retransmitted? (1 + 2.5 + 2.5 = 6 points)**
   (a) Suppose the sender and receiver use a go-back-N retransmission strategy. Which sequence numbers are retransmitted by the sender after the RTO is triggered?
   **503**

(b) Instead, suppose the sender and receiver use a cumulative ACK retransmission strategy. Assume that the sender receives (correspondingly-numbered) ACKs from the receiver for each sequence number successfully delivered. Further, assume that after t = 0, there are no drops in the network. At what time does the sender know that it has successfully recovered from all the losses? Explain your answer.

```
The sender knows that it has recovered from the losses once it receives a non-
duplicate ACK.
```

(c) Instead, suppose the sender and receiver use a selective ACK retransmission strategy. Assume that the sender receives (correspondingly-numbered) ACKs from the receiver for each sequence number successfully delivered. Further, assume that after t = 0, there are no drops in the network. Note that the sender may use the information in all the selective ACKs so far to retransmit more eagerly, after an RTO. At what time does the sender know it has successfully recovered from all the losses? Explain your answer.

```
The sender knows that it has recovered from the losses once the receiver ACKs the
next packet.
```

10. **The impact of packet reordering (2 points)**. Why might heavy packet reordering in the network reduce the throughput of TCP connections?

```
Reordering packets uses the buffer memory, which has a limited capacity. If too
much reordering occurs, the amount of unused buffer memory drops, which lowers the
advertised window, the sender's window, and consequently, the throughput.
```

11. **Stream-oriented transfer (2 points)**. Suppose a TCP sender pushed two pieces of data through a socket send() call. Each piece of data is of size 100 bytes. The data is delivered reliably to the TCP receiver. Now the receiving application performs a recv() system call on its socket, and gets some data in return. What are the possible sizes (in bytes) of this returned data?
    $0 < size < 99$

12. **Flow control and congestion control (4 points).**
    (a) Why is flow control necessary in TCP? (2 points)

```
Flow control helps prevent packet loss by matching the sending rate to the
receiver's read speed.
```

(b) What is the difference between flow control and congestion control? (2 points)

```
Congestion control ensures that no bottleneck link rates are exceeded, while flow
control ensures that packets are ordered and do not exceed the receiver buffer.
```

13. **Collaboration and References (mandatory).** Who did you collaborate with on this problem set? What resources and references did you consult? Please also specify on what questions and aspects of the problem set you got help on. If you did not consult any resources other than the lecture slides and textbook, just say “no collaboration”.

```
I collaborated with my classmate Akash Shah on this problem set. He helped confirm
my understanding of checksum addition.
```