

Homework 7 (Transport Layer)

Fill in your answers in the space provided and submit your modified copy to Connex as an attachment.

Concepts

- Port Multiplexing and Demultiplexing
- Sockets and Port numbers
- Checksum
- TCP Sequence and acknowledgement Numbers

Q1

Consider a **TCP** connection between host **A** and host **B**. Suppose that the **TCP** segments traveling from **A** to **B** have source port number **x** and destination port number **y**. What are the source and destination port numbers for the segments traveling from **B** to **A**?

the source port will be: y

the destination port will be: x

Q2

Suppose a process on host **C** has a **UDP** socket with port number **6789**. Suppose both host **A** and host **B** each send a **UDP** segment to host **C** with destination port number **6789**. Will both of these segments

be directed to the same socket at host C? If so, how will the process at host C know that these two segments from two different hosts?

both segments from A and B will be directed to the some socket at C
Both A and B have their own sockect and these individual sockets will have different IP address.

At C the process will know that the segments are from 2 different host because they will have different IP addresses even thought the A and B segments will have the some destination port number.

Q3

UDP and **TCP** use 1's complement for their checksums. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100. What is the 1's complement of the sum of these 3 bytes. (Note: although **UDP** and **TCP** use 16-bit word sum, we are only computing with 8-bit for this problem.) Show all steps.

Sum of the first 2

01010011

(+)01100110

10111001

Result of the sum of the first two and the last byte

10111001

(+) 01110100

$$100101101 + 1 = 00101110$$

1's complement = 11010001

1. Why is it that **UDP** takes the 1s complement of the sum? Why not just use the sum instead?

This is because the 1's complement will be used as the checksum and the checksum will be used to check for errors by the receiving host.

2. With the 1s complement scheme, how does the receiver detect checksum errors?

it adds all the bytes including the checksum and the sum is used to check for errors that is if the sum is all ones then there is no error but if one of the bits is 0 then there will be errors

Q4

Host **A** and **B** are communicating over a **TCP** connection, and host **B** has already received from **A** all bytes up through byte 126 inclusively. Suppose host **A** then sends two segments to host **B** back-to-back. The first and second segments contain 80 and 40 bytes of data respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host **B** sends an acknowledgement whenever it receives a segment from **A**.

1. In the second segment sent from A to B, what are the sequence

number, source port number, and destination port number?

sequence number= $127+80=207$

source port=302

destination port=80

2. If the first segment arrives before the second segment, in the acknowledgement of the first arriving segment, what is the acknowledgement number, the source port number, and the destination port number?

the Ack is for the next expected byte therefore the Ack number will be 207

source port: 80

destination port:302

3. If the second segment arrives before the first segment at **B**, in the acknowledgement of the first arriving segment (i.e., the second segment), what is the acknowledgement number?

the Ack is for the next expected byte therefore the Ack number will be 127

4. Suppose the two segments sent by **A** arrive in order at **B**. The first acknowledgement is lost and second acknowledgement arrives after the first timeout interval at **A**. Draw a timing diagram (Message Sequence Chart), showing these segments and all other segments and acknowledgement sent. For each segment in your figure, provide the sequence number, the

number of bytes of data; for each acknowledgement that you add, provide the acknowledgement number.