Chap 3 Homework 1

Zilong Wang 18281218 CS1804 wangzilong@bjtu.edu.cn

R1. Consider a TCP connection between Host A and Host B. Suppose that the TCP segments traveling from Host A to Host B have source port number x and destination port number y. What are the source and destination port numbers for the segments traveling from Host B to Host A?

Solu: From y to x.

R2. Suppose the network layer provides the following service. The network layer in the source host accepts a segment of maximum size 1,200 bytes and a destination host address from the transport layer at the destination guarantees to deliver the segment to the transport layer at the destination host. Suppose many network application processes can be running at the destination host.

a. Design the simplest possible transport-layer protocol that will get application data to the desired process at the destination host. Assume the operating system in the destination host has assigned a 4-byte port number to each running application process.

b. Modify this protocol so that it provides a "return address" to the destination process.

c. In your protocols, does the transport layer "have to do anything" in the core of the computer network?

Solu: a. The Simple Transport Protocol takes data not exceeding 1196 bytes at the sender side .It accepts four byte of destination port number and host address. The Simple Transport Protocol gives the destination host address and the resulting segment to the network layer. The network layer sends the segment to Simple Transport Protocol at the destination host.

The Simple Transport Protocol observes the port number. Abstracts the data from the segment in the Simple Transport Protocol. Finally, send the data to the process recognized by the port number.

b. Consider the two header fields in the segment:

Source port field

Destination port field

The Simple Transport Protocol creates application data, source and destination port numbers in the segment. It sends the destination host address to the network layer. Then, The Simple Transport Protocol is receiving host address and provides the process the source port number and the application data.

c. No, the transport layer does not have to do anything in the core. The reason is that, the transport layer "lives" in the end systems.

R3. Consider a planet where everyone belongs to a family of six, every family lives in its own house, each house has a unique address, and each person in a given house have a unique name. Suppose this planet has a mail service that delivers letters from source house to destination house. The mail service requires that (i) the letter be in an envelope and that (ii) the address of the destination house (and nothing more) be clearly written

on the envelope. Suppose each family has a delegate family member who collects and distributes letters for the other family members. The letters do not necessarily provide any indication of the recipients of the letters.

a. **Using the solution to Problem R1 above as inspiration, describe a protocol that the delegates can use to deliver letters from a sending family member to a receiving family member.**
b. **In your protocol, does the mail service ever have to open the envelope and examine the letter in order to provide its service?**

Solu: a. Sender has to provide the address of the destination name. It is written by the delegate to the planet's mail service..
After receive the destination address, the envelop the written on the top details.

b. *No. The mail service ever have not to open the envelope and examine the letter in order to provide its service.*

R4. Describe why an application developer might choose to run an application over UDP rather than TCP.

Solu: When they need to ensure speed rather than accuracy. Some errors have no effect on the application.

R5. Suppose that a Web server runs in Host C on port 80. Suppose this Web server uses persistent connections, and is currently receiving requests from two different Hosts, A and B. Are all of the requests being sent through the same socket at Host C? If they are being passed through different sockets, do both of the sockets have port 80? Discuss and explain.

Solu: For each persistent connection, the Web server creates a separate "connection socket". Each connection socket is identified with a four-tuple: (source IP address, source port number, destination IP address, destination port number). When host C receives and IP datagram, it examines these four fields in the datagram/segment to determine to which socket it should pass the payload of the TCP segment. Thus, the requests from A and B pass through different sockets. The identifier for both of these sockets has 80 for the destination port; however, the identifiers for these sockets have different values for source IP addresses. Unlike UDP, when the transport layer passes a TCP segment's payload to the application process, it does not specify the source IP address, as this is implicitly specified by the socket identifier

R6. Why is it that voice and video traffic is often sent over TCP rather than UDP in today's Internet.

Solu: Most firewalls are configured to block UDP traffic, using TCP for video and voice traffic lets the traffic though the firewalls. So, that voice and video traffic is often sent over TCP rather than UDP in today's Internet.

P1. Suppose Client A initiates a Telnet session with Server S. At about the same time, Client B also initiates a Telnet session with Server S. Provide possible source and destination port numbers for

a. **The segments sent from A to S.**
b. **The segments sent from B to S.**
c. **The segments sent from S to A.**
d. **The segments sent from S to B.**
e. **If A and B are different hosts, is it possible that the source port number in the segments from A to S is the same as that from B to S?**
f. **How about if they are the same host?**

Solu: Serial port number Destination port number

a. A->S	*467*	*23*
b. B->S	*513*	*23*
c. S->A	*23*	*467*
d. s->B	*23*	*513*

e. Yes
f. No

P2. UDP and TCP use 1s complement (反码) for their checksums. Suppose you have the following three 8-bit bytes: 01010101, 01110000, 01001100. That is the 1s complement of the sum of these 8-bit bytes? (Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums.) Show all work. Why is it that UDP takes the 1s complement of the sum; that is, why not just use the sum? With the 1s complement scheme, how does that receiver detect errors? Is it possible that a 1-bit error will go undetected? How about a 2-bit error?

Solu:

```
  01010101
+ 01110000
  11000101
+ 01001100
  00010001
```

To detect errors, the receiver adds the four words (the three original words and the checksum). If the sum contains a zero, the receiver knows there has been an error.

P3. Consider Figure 3.5. What are the source and destination port values in the segments flowing from the server back to the clients' processes? What are the IP addresses in the network-layer datagrams carrying the transport-layer segments?

Solu: Assume the IP addresses of the hosts A, B, and C are a, b, c, respectively. (Note that a, b, c are distinct.)

- *To host A: Source port =80, source IP address = b, destination port = 26145, destination IP address = a*
- *To host C, left process: Source port =80, source IP address = b, dest port = 7532, destination IP address = c*
- *To host C, right process: Source port =80, source IP address = b, dest port = 26145, destination IP address = c*