

Introduction to Computer Networks

Lab2 – Socket Programming

111062109 黃庭曜

Contents

§1 Implementation	2
§2 Report	3

§1 Implementation

1. Connect the server and client to the TCP socket and successfully send a message.

Server-side:

```
● crazytingyao@huangtiyaodeAir lab2 % ./server
New connection
```

Client side:

```
● crazytingyao@huangtiyaodeAir lab2 % ./client
Hi, I'm server 111062109
```

2. Create a TCP header (without checksum) using l4info.

Client-side: "test"

```
● crazytingyao@huangtiyaodeAir lab2 % ./client
Hi, I'm server 111062109
server: Enter "test" to start or "quit" to leave!
test
Receive Data from server...
Receive information:

          Layer 3 information:
          Source IP: 53.34.79.160 , Destination Ip: 199.151.130.68
          Protocol: 6 (TCP)

          Layer 4 information:
          Source port: 42961 , Destination port: 35171
          Seq number: 2121043209 , Ack number: 1167626588
          Header length: 5 (bytes) , FLAG: 0x10 (ACK)
          Window size: 16324
```

Client side: "quit"

```
● crazytingyao@huangtiyaodeAir lab2 % ./client
Hi, I'm server 111062109
server: Enter "test" to start or "quit" to leave!
quit
close Socket
```

Client side: {wrong_input}

```
● crazytingyao@huangtiyaodeAir lab2 % ./client
Hi, I'm server 111062109
server: Enter "test" to start or "quit" to leave!
111062109
server: Wrong input! Enter "test" to start or "quit" to leave!
3r23
server: Wrong input! Enter "test" to start or "quit" to leave!
quit
close Socket
```

3. Complete the header (with checksum)

Client side:

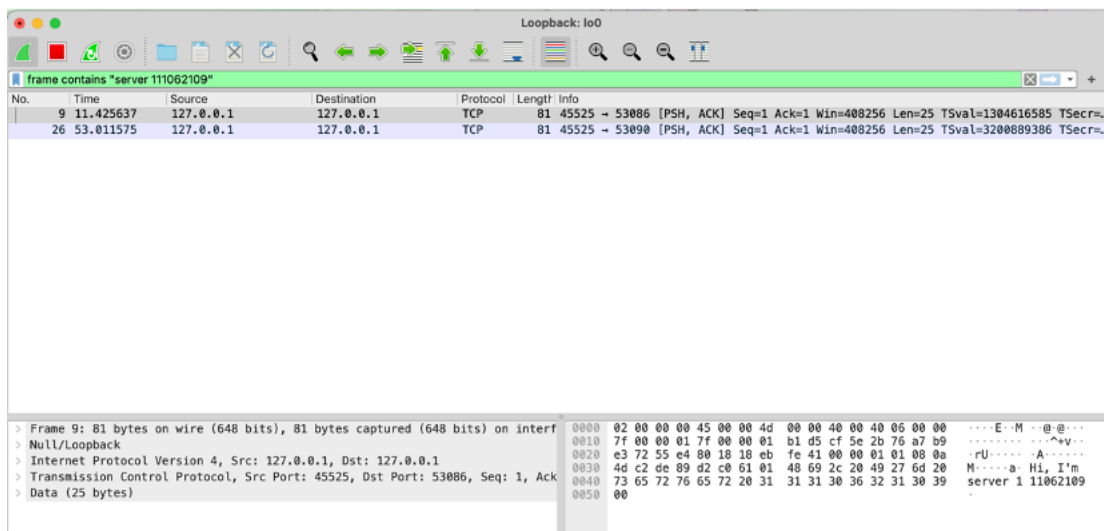
```
● crazytingyao@huangtiyaodeAir lab2 % ./client
Hi, I'm server 111062109
server: Enter "test" to start or "quit" to leave!
test
Receive Data from server...
Receive information:

      Layer 3 information:
      Source IP: 53.34.79.160 , Destination Ip: 199.151.130.68
      Protocol: 6 (TCP)

      Layer 4 information:
      Source port: 42961 , Destination port: 35171
      Seq number: 2121043209 , Ack number: 1167626588
      Header length: 5 (bytes) , FLAG: 0x10 (ACK)
      Window size: 16324

The header is:
A7 D1 89 63 7E 6C 8D 09 45 98 91 5C 50 10 3F C4 8D D2 00 00
```

4. Screenshot the TCP Packet contains “Server {Your_student_ID}”



§2 Report

1. What does INADDR_ANY mean?

A: INADDR_ANY is a constant that represents the IP address 0.0.0.0. This IP address binds a socket to all available network interfaces on a machine.

2. What's the difference between `bind()` and `listen()`?

A: The function `bind()` is used for initializing and setting the socket and for associating a socket with a specific port number. The function `listen()` makes the socket start to wait for the client's request.

3. Usually, we set up the server's port and exclude the client's. Who determines the client's port, and what is the benefit?

A: The client's operating system will determine the client's ports, called dynamic ports.

There are several benefits, such as the clients can connect multiple applications to the server, the wide range of the dynamic ports allows the operating system to distribute the ports, and the chance of being attacked is reduced since the ports are unpredictable.

4. What is Little Endian and Big Endian? Why do most network byte orders use Big Endian?

A: In Little Endian, the most significant bytes are stored in the highest memory address, while in Big Endian, the most significant bytes are stored in the lowest memory address.

An example is illustrated below. The network uses the Big Endian system since the designs in the early network protocols, such as TCP/IP, choose it. When designing the protocols, they think humans are more used to reading data from the highest to the lowest in numbers. The example illustrated below shows that reading data in Big Endian is more intuitive.

Data	0x12345678			
Memory address	Lowset	->		Highest
Little Endian	78	56	34	12
Big Endian	12	34	56	78

5. Why do we need a pseudo-header?

A: A pseudo-header provides extra error-checking. For example, it protects misrouted segments and prevents IP Spoofing. When attackers pretend to be a legitimate source to send data to us, we can check whether the IP address in the pseudo-header is the same as the IP header.

6. For the code below, what's the difference between `client_fd` and `socket_fd`?

```
"client_fd = accept(socket_fd, (struct sockaddr *)&clientAddr,
(socklen_t*)&client_len);"
```

A: The `socket_fd` is the socket that listens to the connection request, while `client_fd` is the socket that communicates between the server and the client.

7. When using the `send()` function and `recv()` function, why don't we need the address?

A: In TCP/IP, we have already set the connection by the functions `connect()` on the client side and `accept()` on the server side. The system will know its destination and source if we use the connected socket to send or receive the data.

8. Write about what you have learned from Lab 2.

A: First, I learned the calculation process for the checksum. When calculating the checksum, the "TCP header length should be calculated in bytes instead of rows (which requires $\times 4$). Next, I've learned that we use Little Endian in computer systems and Big Endian in network systems. When calculating the checksum, I must carefully assign the data to the header fields. Last, I have learned the basics of socket programming. Although socket programming has some tricky parts, I still learned much from it.