Mini Project (due Nov. 2, 2022)

NOTE:

This is team based assignment, and each team has 1-4 people.

Please put your answer in word and name the word file as MP_ID1_ID2_ID3.pdf if your team has three people.

EX: MP_1234567_9876543_3456789.pdf if your team has 3 people.

Please put your implementations, and the pdf file into a folder named as MP_ID1_ID2_ID3 before zip it as MP_ID1_ID2_ID3.zip for submission.

a. Each team can just submit one version (one zip file) to Moodle
b. If you refer to any resources for completing this assignment, please add the reference at the end of your assignment.
c. Late submission policy: No late submission allowed.

The goal of this mini project is to use socket programming for investigating and evaluating the network performance in the presence of delay and loss as some IoT applications have very strict delay and loss requirement. Moreover, this project is designed to brainstorm your ideas improving the network performance.

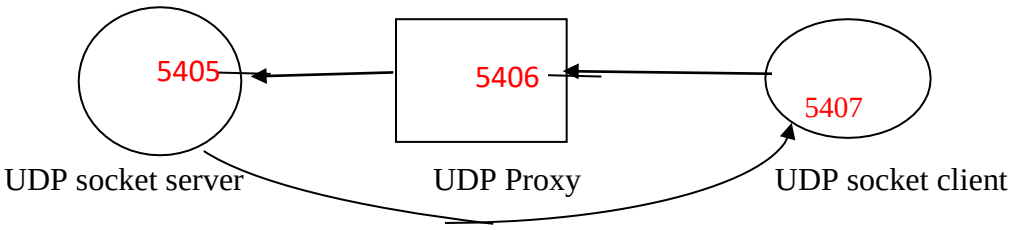

Figure 1. Connection setup.

Figure 1 shows the connection setup among the UDP socket server, UDP proxy and the UDP proxy client by using the port 5405-5407.

The UDP socket client transmits the UDP packet to the UDP socket server whose port listening to the socket client is 5405. Between the UDP socket client and the UDP socket server, the UDP proxy takes care of the packet forwarding. Hence, the UDP socket client first sends the UDP packet to the UDP proxy whose port is 5406, and then the UDP proxy forwards the packet to the UDP socket server whose port is 5405. On the other hand, the UDP socket server can directly send packets to the UDP client whose port is 5407.

Please implement the connect setup shown in Figure 1 and answer the following questions accordingly.

1. Write a client, UDP proxy and server by UDP socket. The server listens to the packets, and the socket client send "Hello" to the server. When the socket server receive a "Hello", it returns a "World" back. During the transmission, UDP proxy directly forwards packets from the client to the server, after receiving the packets.

   Please add indicator "i" along with Hello to show the i-th transmission, and i is from 1 to 10000.

   For example:

   The client sends Hello 1 to the server.

   The server sends World 1 to the client after it receives Hello 1.

   The client sends Hello 2 to the server after it receives World 1.

   … This process repeats till the server sends World 10000 to the client.

   Show your implementation results to prove the implementation is correct.

2. Following 1, we further implement loss and delay functions in the UDP proxy. Instead of directly forwarding packets, the UDP proxy behaves as the following:

   When the UDP proxy receives the "Hello i" message,

   -- it drops each received packet with 10% probability if i is even number.

   -- it delays 100 ms the received packet with 5% probability before forwarding to the server if i is odd number.

   Show your implementation results to prove the implementation is correct.

3. Following 2, you observe transmission loss and delay which are undesirable for IoT applications, such as remote surgery, with hard delay or loss constraints. Think and design any methods which can be implemented in the server or client side to improve the problems following:

   a. Describe your methods

   b. Implement your methods into your socket implementations in 2.

   c. Discuss what happens for the transmission by applying your designed methods.

   NOTE: Please don't change proxy's loss and delay behavior !

4. Share what you have learned/discovered/explored through the current discussions in the lecture and mini project.