【2019 Advanced Computer Networks Homework 5】

Rules:

1. Please use C language in this homework and run your program on Ubuntu 18.04.
2. Please provide Makefile to compile your homework; otherwise, you will get ZERO.
3. Do not copy others homework.
4. If you have any question, please send email to net_ta@net.nsysu.edu.tw or come to F5018, but TA does not help to debug.

Upload:

1. Please compress your homework into zip or tar archive.
2. Naming rules: “StudentID_TCPIP_HW5.zip”. For example: D083040001_TCPIP_HW5.zip
3. Upload your homework to NSYSU Cyber University.
4. Deadline: 2019/11/27 23:59; if it is overdue, you will also get ZERO.

Section 61.13.1 noted that an alternative to out-of-band data would be to create two socket connections between the client and server: one for normal data and one for priority data. Write client and server programs that implement this framework. Here are a few require:

Create two hosts using the mininet, one host executes the server program and the other host executes the client program. The server needs some way of knowing which two sockets belong to the same client. One way to do this is to have the client first create a listening socket using an ephemeral port (i.e., binding to port 0). After obtaining the ephemeral port number of its listening socket (using getsockname()), the client connects its“normal” socket to the server’s listening socket and sends a message containing the port number of the client’s listening socket. The client then waits for the server to use the client’s listening socket to make a connection in the opposite direction for the “priority” socket. (The server can obtain the client’s IP address during the accept() of the normal connection.) Implement some type of security mechanism to prevent a rogue process from trying to connect to the client’s listening socket. To do this, the client could send a cookie (i.e., some type of unique message) to the server using the normal socket. The server would then return this cookie via the priority socket so that the client could verify it. In order to experiment with transmitting normal and priority data from the client to the server, you will need to code the server to multiplex the input from the two sockets using select() or poll() (described in Section 63.2).