In the name of Allah

بسم الله‌الرحمن الرحیم

OSPF and Socket Programming Laboratory Manual

University of Tehran
دانشگاه تهران

School of Electrical and Computer Engineering
دانشکده مهندسی برق و کامپیوتر

Computer Network Lab
آزمایشگاه شبکه‌های کامپیوتری

Dr. Ahmad Khonsari - احمد خونساری
a_khonsari@ut.ac.ir

Amir Haji Ali Khamseh'i - امیر حاجی علی خمسهء
khamse@ut.ac.ir

Sina Kashi pazha - سینا کاشی پزها
sina_kashipazha@ut.ac.ir

Mohammad Ali Shahsavand - محمد علی شاهسوند
mashahsavand@ut.ac.ir

Amirahmad Khordadi - امیر احمد خردادی
a.a.khordadi@ut.ac.ir

December 10, 2018

۱۹ آذر ۱۳۹۷

Socket programming exercises 1

Examine the UDP socket programs `/home/guest/UDPserver.c` and `/home/guest/UDPclient.c` to learn how to write a UDP socket program. Compile the C programs using `gcc -o UDPserver UDPserver.c -lnsl` and `gcc -o UDPclient UDPclient.c -lnsl`.
Start **wireshark** to capture packets from or to a remote host.
On the remote host, start the UDP server by **UDPserver server_port**. Then, start the UDP client on your host by **UDPclient remote_host server_port a_message**. You may execute the UDP client program on other hosts to connect to the same UDP server. Terminate **wireshark**, examine its output and compare the output with the UDP server and client outputs. Repeat the above experiments, but now use the `TCPserver.c` and `TCPclient.c`.

Socket programming exercises 2

Execute **man setsockopt** to display the various socket options and how to set them. Examine the **netspy** and **netspyd** source code in Appendix C.2 to see how to create a multicast socket and how to set the TTL value for the packets.

Socket programming exercises 3

This is an optional exercise on socket programming. Or, it can be assigned as a take-home project for extra credits. Note that familiarity with C programming is required.

PROBLEM

Examine the message exchanges of FTP. Write a FTP client program which takes a file name as input, and upload the file to a standard FTP server on a remote machine.

HINTS

- First you need to set up the control connection to Port 21 of the remote machine, using a TCP socket.
- When the control connection is established, you need to exchange FTP commands with the remote FTP server, as given in Table 5.1.
- You can first run **telnet remote_host 21**, then type **help** to list all the FTP commands. Also, you can try the commands out in the **telnet** window, e.g. use **USER guest** to send the user ID and **PASS guest1** to send the password to the FTP server. To terminate the **telnet** session, type **QUIT**.
- In your program, these messages should be sent to the FTP server by calling the **send()** function of the local TCP socket.
- Also your program needs to parse the server responses (some examples are given in Table 5.2) to find out the status of the previous FTP command.
- The FTP data connection should be established using the **PORT** command (see Chapter 5).