

**AAST      College of Computing and Information Technology**

***Network Protocols (CS422)***

***Spring 2014***

***Homework 5      (Total of 30 pts)***

***Due on May 25<sup>th</sup>, 2014***

***No late submissions will be allowed***

**Q1. (5 pts)** Write C source code to verify the IP address and ephemeral port assigned to a client UDP socket? Hint: Print this information after calling *socket*, *sendto*, and *connect* in the *connect* variant. What do you conclude regarding the assignment of IP/port to a UDP socket?

**Q2. (5 pts)** Write C source code for a UDP client. This client will send datagrams to 2 different UDP echo servers in order, i.e., send the first datagram to the first server, then wait to receive it back, send the second datagram to the second server, then wait to receive it back. The first and second datagrams are user-provided input. Upon receiving the datagram from the server, the client displays the datagram's data along with the source of the datagram (IP and port). *Hint*: Do you need 1 or 2 sockets in the client?

**Q3. (20 pts)** Write C source code for the following client/server specifications.

*Client*

The client creates 1 TCP socket and 1 UDP socket. It uses its TCP socket to connect to a TCP server. The TCP server will send to the client its own UDP server port (the client only knows the TCP server port and not the UDP server port). The client uses its UDP socket to send 5 user-provided messages to the server. For each message, the client waits for it to be echoed back from the server before sending the next one. The client verifies that it is receiving the echo from the proper server (communication endpoint), otherwise display an error message and exit the program. Upon successful reception from the server, display the datagram's data along with source IP and port. Finally, the client will perform the active close.

*Usage*: client <TCP Server IP> <TCP Server Port>

*Server*

The server creates 1 TCP server socket and 1 UDP server socket. The port for the TCP server socket will be passed as command-line argument, whereas the port for the UDP server socket will be any free available port. When a TCP client connects to its TCP server socket, it accepts the connection and sends back to the client the port for its UDP server socket (dynamically allocated). On the UDP server socket, it receives 5 messages from the client and echoes them back. The server should print suitable diagnostic messages regarding TCP client connection, and the datagrams received from UDP client sockets.

*Usage:* server <TCP server port>

Example for Server Output

Connection from TCP client 128.82.10.15:48967 accepted

“xyz” received from 128.82.10.15:45987

....

For the client and server, your code should provide suitable error checking and diagnostic messages.