

**CS2105 Computer Networking I**  
**Laboratory Experiment 2**

**Socket Programming I**

Objectives:

- (1) Understand socket programming with TCP and UDP through an example client/server application in Java.
- (2) Develop TCP client/server programs and UDP client/server programs for a simple application.

This laboratory experiment is based on **Section 2.7 – Socket Programming with TCP** and **Section 2.8 – Socket Programming with UDP** of the textbook (p. 146 – 164).

**(Total 100 marks)**

**Part A: Demonstrate socket programming for TCP**

**(10 marks)**

This part provides two sample programs, i.e., the client/server program pair, for a TCP implementation of the application that is described in Section 2.7. Do the following:

- (1) Run *sudo lab2* on PC1–“BenchNo” to retrieve the sample client program called **TCPClient.java**.
- (2) Run *sudo lab2* on Router–“BenchNo” to retrieve the sample server program called **TCPServer.java**.
- (3) Study and edit the two sample programs.
- (4) Start up the Ethereal software on PC1–“BenchNo” (shell command: *sudo ethereal*).
- (5) Run the server **TCPServer.java** on Router–“BenchNo” and the client **TCPClient.java** on PC1–“BenchNo”.
- (6) Stop the Ethereal after both the sample programs terminate.
- (7) Debug the programs if there are problems, and run the programs again to obtain results as given in Section 2.7.

**Part B: Develop a simple application over TCP****(50 marks)**

The main socket-related activity of the client and server for the application using connection-oriented transport services is illustrated in Figure 2.28 of the textbook (p. 149 – 150). Now you are going to revise the two sample programs to do the following:

- (8) The server generates two random numbers  $X$  and  $Y$ , and sends the line “ $X + Y = ?$ ” through its connection socket to the client.  
(Reference: */usr/j2sdk1.4.2\_06/api/docs/index.html*)
- (9) The client reads the line from its socket and prints the line on its standard output.
- (10) The client reads a line (your answer to “ $X + Y = ?$ ”) from its standard input and sends the line through its socket to the server.
- (11) The server reads the line from its connection socket, checks the received line against the correct answer, and sends the verdict (right or wrong) through its connection socket to the client.
- (12) After revising the two programs, repeat steps (4) to (6) to demonstrate your results.

**Part C: Redevelop the simple application over UDP****(40 marks)**

From Section 2.8, you have known that the code for UDP is different from the TCP code in many important ways for the same application. Now you will redevelop the same application of Part B, but this time over UDP. Do the following:

- (13) Study the server program **UDPServer.java** and the client program **UDPClient.java** (Section 2.8). Figure out how to change a TCP program to its corresponding UDP program.
- (14) Modify and change your programs developed in Part B to allow both the client and server to communicate over a connectionless (UDP) transport service.
- (15) Repeat steps (4) to (6) for your revised UDP programs.
- (16) Demonstrate the results for steps (8) to (11).

– END –