Mount Royal University
Department of Mathematics and Computing

COMP 3533 Network Infrastructure
Lab 6: Socket Programming
October 25, 2021

Given: October 25, 2021

Due: November 8, 2021, 11:59pm

Exercise Instructions: All lab tasks must be attempted. **The assignment for this lab is given at the end of this document.** Student can work as part of their lab group to complete the assignment. A zip file containing all the source code files must be submitted. The code must follow good formatting and naming practices, and must be well documented. In addition to the Java source files, a statement of contribution document is expected in which the contribution of each member of the group to the solution is stated. Satisfactory completion of both the programming exercise and the statement of contribution are required to get your exercise mark.

The Socket programming API allows application layer programs to communicate with one another using the interface provided by the transport layer. Transport layer communication comes in two flavors: Transport Control Protocol (TCP) and User Datagram Protocol (UDP). TCP and UDP are used for different purposes and both have unique constraints:

- TCP is the more reliable protocol that enables a client to establish a bidirectional connection to a server, then send and receive messages over a byte stream. When using TCP, each entity knows that its messages are received in order, or it gets a connection error.
- UDP is a connectionless protocol and is good for scenarios where you do not necessarily need every packet to be received in order at its destination, or arrive at all.

In this lab, we will study the basic building blocks for setting up and using TCP connections with Java socket programing.

Webcasts for lab 6 are posted on blackboard and they cover socket programming basics, and walks through an example client server program. The slides correspond to section 2.7 of the Chapter 2: Application Layer (slides 104-117) posted under course material.

Lab Tasks:

You must first watch the lab webcasts and study the example client and server code provided. You can then start working on extending the code to implement the assignment given below.

Lab Assignment:

In this lab, you are asked to implement a server code and a client code in Java. The server maintains a codebook in its directory. The codebook.txt file can be downloaded from the blackboard. In this codebook, each line contains a code word and a decoded string separated by a TAB (i.e., /t). While your server is running, you can start a client program from a second machine to communicate with the server. At the prompt, user can type a confusing message like this:

GM, RU COMING B4 NOON? PLZ WB ASAP. TYVM. GTG, TTYL.

Then, the message will be sent over the network to the server. The server will go through the message word by word, and check them with its codebook. Upon a match, the coded word will be replaced by its decoded string in the original message. After all the possible coded words are replaced, the whole decoded message will be sent back to the client, and be displayed on the user's screen like this:

GOOD MORNING, ARE YOU COMING BEFORE NOON? PLEASE WRITE BACK AS SOON AS POSSIBLE. THANK YOU VERY MUCH. GOT TO GO, TALK TO YOU LATER.

User can request the decoding procedure over and over until a special character or phrase such as "exit" to stops the client program, depending on your design (just state it clearly in your documentation). At the mean time, the server will keep running forever.