

## CPSC 441: Assignment 2 – Transport Protocols

User Manual of the programs written by Josh Vanderstoop

### Program Features:

client.c - Accepts User commands through the terminal and sends them via TCP connection to the indirection server.

indirection-server.c – Receives commands from the client and acts as a middle-man between the client and micro-servers. The commands and replies are sent between the services through a UDP connection and to clients a using TCP connection. The indirection server does not anticipate any missed responses from the UDP servers, and thus will not time out.

translator-service.c – Sends the indirection server a database of translatable words upon a connection, and then receives an English word form the indirection server and returns the French translation of that word to the indirection server.

converter-service.c – Sends the indirection server a format for proper conversion commands, as well as the available currencies to convert form CAD. The clients command will be processed and the converted value will be calculated, which is then sent back to the indirection server.

voting-service.c – Based on the UTC time, the server will send one of two messages to the indirection server. If the indirection server connects before the arranged time to close the polls, the instructions will prompt for a voting key from the client. This will be encrypted with a key that is specific to the client to ensure privacy. If the polls are closed (ie. The user connected after the voting time limit), the user will be shown the results of the election.

### Usage:

First, move onto any suitable university computer using the commands:

```
ssh your.username@linux.cpsc.ucalgary.ca
```

```
YourPassword1235
```

now, move client.c, indirection-server.c, voting-service.c, converter-service.c, and translator-service.c onto the computer using any method you would like, I use FileZilla. Then compile and run each of the three servers using the usage tips in the initial description found in each .c file on their own window. Now, after reading the terminal of each service, read the beginning of the indirection-server.c file to understand which IP address to comment out of the indirection-server.c file, and follow the instructions. In another window connected to the university servers, compile and run indirection-server.c as per its instructions. Lastly in another window, run and compile client.c using the directions found at the beginning of the file, and be sure to use the proper command line arguments when running the file. From this point, you can follow the prompts of the client window and use the application as directed. When you are finished, type ctrl + c into every terminal to end the programs running on that terminal.