

Amy Liu

Aliu66@gatech.edu

CS 3251 Computer Networking I – A

Programming Assignment 1: Basics of Socket Programming

February 8, 2017

smsengineTCP.java

Server application based on TCP. To compile, launch the command line and navigate to the directory that contains smsengineTCP.java and type

```
javac smsengineTCP.java
```

To run the server program, type into the command line

```
java smsengineTCP <port number> <suspicious-words.txt>
```

The text file of suspicious words needs to be in the same directory as smsengineTCP.java. Run the server application before the client application or an error will be thrown

```
Could not get I/O for the connection to <server IP address>
```

The application will check if the user has included the correct number of command line arguments. It will add the suspicious words from the text file provided to a set. The application will then create a socket, accept a connection, and get the input from the client. If the input from the client is empty, over 1000 characters, or contains a non-ascii character, an error message will be sent back to the client.

0 -1 Error

If the input from the client is valid, the application will split the string provided by the client based on spaces and punctuation marks and send a response back to the client.

The application will keep running until the user press Ctrl + C to exit.

smsclientTCP.java

Client application based on TCP. To compile, launch another command line and navigate to the directory that contains smsclientTCP.java and type

```
javac smsclientTCP.java
```

To run the client program, type into the command line

```
java smsclientTCP <server IP address> <port number> <msg.txt>
```

The port number should be the same as the port number used for the TCP server. The text file with the message needs to be in the same directory as smsclientTCP.java.

The application will create a string from the message in the provided text file, create a socket, and send the message to the server.

The application will close once a response has been received from the TCP server.

smseengineUDP.java

Server application based on UDP. To compile, launch the command line and navigate to the directory that contains smseengineUDP.java and type

```
javac smseengineUDP.java
```

To run the server program, type into the command line

```
java smseengineUDP <port number> <suspicious-words.txt>
```

The text file of suspicious words need to be in the same directory as smseengineUDP.java.

The application will check if the user has included the correct number of command line arguments. It will add the suspicious words from the text file provided to a set. The application will then create sockets that are able to send and receive packets of 1024 bytes, receive a request from a client, and get the message from the client. If the input from the client is empty, over 1000 characters, or contains a non-ascii character, an error message will be sent back to the client.

0 -1 Error

If the input from the client if valid, the application will split the string provided by the client based on spaces and punctuation marks and send a response back to the client.

The application will keep running until the user press Ctrl + C to exit.

smsclientUDP.java

Client application based on UDP. To compile, launch another command line and navigate to the directory that contains smsclientUDP.java and type

```
javac smsclientUDP.java
```

To run the client program, type into the command line

```
java smsclientUDP <server IP address> <port number> <msg.txt>
```

The port number should be the same as the port number used for the UDP server. The text file with the message needs to be in the same directory as smsclientUDP.java.

The application will create a string from the message in the provided text file, create a socket, and send the message to the server.

The application close once a response has been received from the UDP server. If the client has not received a request from the server within 2 seconds, it will retry the same query 3 times. After 3 unsuccessful requests, the client will print an error message and exit.