**Name:** Hung Viet Luu, William Kiplinger

**Link to Project:** <https://github.com/hvluu/CS380/tree/master/Projects/Project3>

**IPv4Client.java**

*import java.io.IOException;*

*import java.net.Socket;*

*import java.net.UnknownHostException;*

*public class IPv4Client*

*{*

*private static Socket socket;*

*public static void main(String[] args)*

*{*

*connect();*

*}*

*/\*\**

*\* Connects the client to the server and*

*\* creates a Listener thread.*

*\*/*

*public static void connect()*

*{*

*String hostName = "18.221.102.182";*

*int portNumber = 38003;*

*try*

*{*

*socket = new Socket(hostName, portNumber);*

*System.out.println("Connected to server.");*

*new Connection(socket).start();*

*}*

*catch (UnknownHostException e) {*

*System.err.println("ERROR: Unknown host " + hostName + ".");*

*} catch (Exception e) {*

*System.err.println("ERROR: Could not connect to " + hostName + ".");*

*}*

*}*

*/\*\**

*\* Disconnects the client from the server.*

*\*/*

*public static void disconnect()*

*{*

*try {*

*socket.close();*

*} catch (IOException e) {*

*System.err.println("ERROR: " + e.getMessage() + ".");*

*}*

*}*

*}*

**Connection.java**

*import java.io.\*;*

*import java.net.Socket;*

*import java.nio.ByteBuffer;*

*public class Connection extends Thread*

*{*

*public volatile static boolean endThread = false;*

*private Socket socket = null;*

*public Connection(Socket socket)*

*{*

*super("Connecting Thread");*

*this.socket = socket;*

*}*

*/\*\**

*\* Override run() function in the Thread class.*

*\* The function handles the communication between the server and the client.*

*\*/*

*public void run()*

*{*

*try*

*{*

*socket.getInputStream();*

*socket.getOutputStream();*

*ipv4();*

*IPv4Client.disconnect();*

*}*

*catch (IOException e) {*

*System.err.println("ERROR: Connection lost.");*

*}*

*}*

*private void ipv4() throws IOException*

*{*

*byte version = 4;*

*byte hLength = 5;*

*byte tos = 0;*

*short length;*

*short ident = 0;*

*short flags = 2;*

*short offset = 0;*

*byte ttl = 50;*

*byte protocol = 6;*

*short checksum = 0;*

*int source = 1414;*

*int destination = socket.getInetAddress().hashCode();*

*byte [] data;*

*short dataLength = 2;*

*int counter = 0;*

*InputStream inputStreams = socket.getInputStream();*

*InputStreamReader inputStreamReader = new InputStreamReader(inputStreams);*

*new BufferedReader(inputStreamReader);*

*OutputStream outputStream = socket.getOutputStream();*

*while(counter < 12)*

*{*

*checksum = 0;*

*data = new byte [dataLength];*

*for (int i = 0; i < dataLength; i++)*

*data[i] = 1;*

*length = (short)(hLength \* 4 + dataLength);*

*byte[] packet = new byte[length];*

*byte[] header = new byte[hLength\*4];*

*ByteBuffer byteBuffer = ByteBuffer.wrap(packet);*

*ByteBuffer bufferChecksum = ByteBuffer.wrap(header);*

*byteBuffer.put((byte)((byte)(version & 0xf) << 4 | (byte)hLength & 0xf));*

*bufferChecksum.put((byte)((byte)(version & 0xf) << 4 | (byte)hLength & 0xf));*

*byteBuffer.put(tos);*

*bufferChecksum.put(tos);*

*byteBuffer.putShort(length);*

*bufferChecksum.putShort(length);*

*byteBuffer.putShort(ident);*

*bufferChecksum.putShort(ident);*

*byteBuffer.putShort((short)((flags & 0x7) << 13 | offset & 0x1fff));*

*bufferChecksum.putShort((short)((flags & 0x7) << 13 | offset & 0x1fff));*

*byteBuffer.put(ttl);*

*bufferChecksum.put(ttl);*

*byteBuffer.put(protocol);*

*bufferChecksum.put(protocol);*

*bufferChecksum.putShort(checksum);*

*bufferChecksum.putInt(source);*

*bufferChecksum.putInt(destination);*

*checksum = checksum(bufferChecksum.array(), bufferChecksum.array().length);*

*byteBuffer.putShort(checksum);*

*byteBuffer.putInt(source);*

*byteBuffer.putInt(destination);*

*byteBuffer.put(data);*

*outputStream.write(byteBuffer.array());*

*System.out.println("data length: " + dataLength);*

*System.out.println("good\n");*

*dataLength = (short)(dataLength\*2);*

*counter++;*

*}*

*}*

*public static short checksum(byte[] array, int length) {*

*long sum = 0;*

*for (int i = 0; length > 0; i++) {*

*sum += (array[i] & 0xff) << 8;*

*length--;*

*if (length == 0)*

*break;*

*sum += (array[i] & 0xff);*

*length--;*

*}*

*sum = (~((sum & 0xFFFF) + (sum >> 16))) & 0xFFFF;*

*short cs = (short) (sum & 0xffff);*

*return cs;*

*}*

*}*