CS380 — Project 2

January 18, 2017

Due: Friday, January 27, 2017 (50 points)

Description

In this project, we will be simulating the physical layer of communication. To do this, I will have a server running that will send "signals" to you by sending numbers in the range [0, 255] (i.e., unsigned bytes) and you need to implement the client that properly interprets them.

The numbers will fluctuate but all be either slightly higher or slightly lower than some average. The average value will be randomly chosen each time you connect. This means to send 10101 directly, It might look like this one time: 65, 43, 66, 40, 71 and the next time it might be 181, 167, 188, 160, 179. However, to add another layer of complication I will also be implementing NRZI on top of the actual bits being sent.

When you first connect, the server will send a *preamble* of 64 alternating high and low signals so that you can establish the baseline. After this point, it will send you 32 bytes of randomly generated data encoded using 4B/5B with NRZI. You need to properly decode this data and record the 32 bytes sent to you.

Once you have figured out the 32 bytes I sent, you must reply it back to me directly, don't use any encoding. Just send me the array of 32 bytes. Once I receive the response, I will send a final byte (not encoded) that is either 0 or 1. If it is 0, I didn't receive the correct 32 bytes. If it's 1, then I did and your program worked correctly.

The 4B/5B table is below:

4-Bit Data	5-Bit Code	4-Bit Data	5-Bit Code
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

The server will be running at `codebank.xyz` on port 38002

A few test runs of your client program might look like this:

```
$ java PhysLayerClient
Connected to server.
Baseline established from preamble: 102.84
Received 32 bytes: E695AB1F3D6D4BFB06A06200F378C1EF3772E67DF3EF7FCFD26FFD496881CDC7
Response good.
Disconnected from server.
```

```
$ java PhysLayerClient
Connected to server.
Baseline established from preamble: 216.06
Received 32 bytes: 221BB41C341531437CF4C500A3FC6194435D839D79AB6261629C22EE9E79D3B4
Response good.
Disconnected from server.

$ java PhysLayerClient
Connected to server.
Baseline established from preamble: 199.78
Received 32 bytes: 0704000434B278D75F124DF7CDD2B9931CDA90F2573C3ED0F45D401628DB1849
Response good.
Disconnected from server.
```

Submission

1. On codebank.xyz, create a project as your user named `CS380-P2`. Follow this naming convention precisely including case.
2. Make sure you add the reference to the remote repository in your local project with:

   ```
   $ git remote add origin https://codebank.xyz/username/CS380-P2.git
   ```

3. Your project should have a main class named `PhysLayerClient` in a file named `PhysLayerClient.java`. You can have other files or classes, but it should successfully compile and run by simplying using:

   ```
   $ javac PhysLayerClient.java
   $ java PhysLayerClient
   ```