Alexicia Richardson, Thomas Baker, Nirmit Patel

Lab 1

March 1, 2017

Reliable and Mobile Communication Hub for First Responders Using WiFi Network Emulation

* Project Overview and Requirements
  + For this project, we were tasked with developing a system that communication over the network. The system’s job was to send a specific set of data across the network from the first responders to the Fire Chiefs. The objective of this assignment was for students to learn how important it is to have a reliable communication system and severity of them.
* Architecture
  + For the gremlin function, the user is prompted to enter a number from 0 to 100. Then, when program is about to send a packet to the server, it will generate a random number from 1-100. If the generated number is above the user-entered threshold, then the packet will be sent no problem. If, however, the generated number is below the user-entered threshold, then the packet will be sent to an invalid address, simulating a dropped packet.
  + Likewise, to simulate sensors for tank level, location, and heart rate, we use random number generators to generate values for each only a portion of the time.
* Design
  + We decided to complete our code in Java. Java offered the necessary tools that would make the completion of the project easier for us. We implemented a client and server file that were used for the communication between the two. To begin, our client class declares the DatagramSocket to prepare for the connection that will occur. The user is prompted for the IP address of the machine that it plans to communicate with in order to send the proper information to the server. Following that, the user is then prompted to enter a Gremlin Function value, which is the probability that a percentage of data being lost through the communication. The user is also prompted decide which option they want to test. Once the program obtains all the data from the user, it then randomly generates the data that will be sent over the network and packs it together. Once it packs it, the packet is now sent over the network to the server class that was implemented, and now the client begins to listen and waits to receive some data back. The server file immediately opens up the three sensor windows when ran, displaying the Sensor Label and what will be sent to that window. The server is also waiting to receive data during that time. Once the sever receives the data, it opens the packet in the main window and send the information the appropriate windows. Once the server is done with the packet, it sends a packet of data(ACK) back to the client to be read.
* Issues
  + When we first began the project, our idea was to implement the code in C++. As we began to work on it, we found it very easy to accomplish the communication aspect of the project. While that part was not a challenge for us, the challenge came when it was time to display the sensor data in three separate windows. This seemed to be a huge challenge and caused a halt in our progress. There was not a simple way in C++ to open up three new windows from the terminal in order to accurately display the data to the Fire Chief.