Christina Nguyen: [cpn716@vt.edu](mailto:cpn716@vt.edu)

Nathaniel Hughes: [njh2986@vt.edu](mailto:njh2986@vt.edu)

ECE 4564, Assignment 1

Section 1 – Objectives

Assignment 1 aimed to use Python network programming to have two Raspberry Pi’s communicate with each other through sockets. The first Raspberry Pi (RPi 1) was to act as a client that takes in a user’s Tweets and parses it accordingly. The parsed Tweets, which contained instructions, were processed and sent to RPi 2, the server side, to control an LED. The format of the Tweet is as follows:

*@UserAcc #ECE4564\_IPByte1\_IPByte2\_IPByte3\_IP Byte4\_PortNo\_GPIOinstruction*

The GPIO instruction portion could be one of three things: *LEDON*, *LEDOFF*, and *LEDFLASH*. Turning on and off the LED are straightforward. For flashing, the LED was to remain flashing until the Twitter user says otherwise. Thus, the flashing was done on a separate thread.

Twython, an open-source API, was used to accomplish the task. The client Pi streamed the Tweets in real-time. After having sent out the GPIO instruction to Pi 2, it sent a Tweet to the Network Application course handle (Twitter account) when it received an acknowledgement.

Section 2 – Team member responsibilities

Christina programmed the server side (RPi 2), which took in the GPIO inputs and controlled the LED accordingly. First, the LED was turned on in the Python code to ensure it was hooked up to the RPi 2 correctly. Once the base code was written out, the client Pi was used to send out plain GPIO instructions to the server Pi to test that it properly received the instructions. It also tested whether or not the acknowledgements were being sent back to the client. Originally, for the flash instruction, the LED merely turned on and off once. After initial server side testing, the flashing was done as specified by the assignment—continuously until a different instruction and on a separate thread.

Section 3 – Conclusions