Carson Wittwer: bbplayer@vt.edu
Jason Nelson: njason18@vt.edu
MH Charles-Etuk: mh6698@vt.edu
ECE 4564, Assignment 3

Objectives

The objective of this assignment was to gain experience in programming with zeroconf, RESTful communications, HTTP authentication, and further expertise with GPIO and mongoDB programming, coupled with network tools such as cURL; as well as their practical use in facilitating the use of data-driven network applications. In this project we designed an advanced bookkeeping system that could still handle basic requests on the part of the user. We used two Raspberry Pi 3+s outfitted with Flask; with two using zeroconf to serve as the Service & Storage, and a third Raspberry Pi 3+ using the GPIO functionality to control LEDs. The Client would request an action, such as selling a book to the Service Pi, who would relay this request to the Storage Pi, who would use the mongoDB framework to carry out said action against the existing database, and then reflect those changes using the LEDs to signify what had just been done. Of note is the fact that the client would require HTTP authentication to carry out these requests

Design

Service:

The Service Pi is not complicated. It uses mongoDB to create a dynamic user database. It is then able to both add users to this database as well as carry out client requests, by way of bash script files which it sends to the Storage Pi, that can either affect the LEDs or the book database itself.

Storage:

The Storage Pi is fairly straightforward, essentially unchanged in core functionality. It uses cURL to execute the commands issued by the client against the mongoDB database, as opposed to RabbitMQ as was the case in the previous project.

It uses Flask to acquire the commands from cURL and creates a web microframework that it can access to carry out its assigned tasks.

LED:

The LED Pi uses Flask to acquire cURL commands from the client and utilizes the GPIO functionality in Raspbian to change the color of connected LEDs depending on what the client requests.

Conclusions

By the end of this assignment we had augmented our understanding of how to create continuous links between our Pis and send information between them. This assignment taught us another way to achieve the same functionality we had reached by the end of the last one, with the added benefit that fewer hurdles in between the Pis amounts to. We learned how to authenticate users using HTTP, how to send bash files from Pi to Pi, and how to use Flask to transmit cURL commands between Pis.

This served as a substantial boost our knowledge of Python overall and worthwhile experience to add to our repertoires.