

EE / CprE / SE 491 – sddec20-05

An Advanced Networking Outreach Activity for Kids

Bi-Weekly Report #5

3/16/2020 – 4/12/2020 (includes Spring Break)

Client & Faculty Advisor: Dr. Tom Daniels

Team Members

Grayson Cox | UI Developer | Agile Project Manager

Austin Dvorak | Network Systems Manager

Ryan Newell | Hardware Systems Admin

Spencer Parry | UI Developer

Ross Thedens | Communication Systems Manager | Meeting Secretary

Reporting Period Summary

This was our first reporting period during the virtual instruction period. Grayson, Austin, Ryan, and Spencer are in Ames, while Ross is about two hours away. We received all of our hardware except for the camera module. We have begun setting up the ad-hoc networking configuration in OpenWrt, but we have run into some roadblocks that may prevent us from using the distribution. Besides this, we have been working out the details of video streaming from the network to the backend/frontend and basic control/data flow between the network master node and the relay nodes. As part of our design document, we established a set of concrete goals for the project and a timeline to accomplish them on. As we proceed towards the end of 491, we will continue working on setting up our ad-hoc network, establishing a communication architecture, and making revisions to the design document until our remaining path towards implementing the project is well-defined.

Reporting Period Accomplishments

- A more concrete design plan for the Network Master Node (All)
 - We've decided there will be three software components on the network master node:
 - Socket Program (Ross & Ryan)
 - Handles direct connections with network nodes.
 - Transmits data to backend/Spring application.
 - Supports connecting and disconnecting nodes in the network.

- This will be done with the hardware alone, independent of the user application.
 - Backend/Spring Application (Grayson)
 - Receives data from socket application.
 - Stores relevant data to be queried by User Application.
 - Examples:
 - Current nodes in network
 - Connection statistics for each node
 - Sensory data streams for each sensor node
 - Provides HTTP API for User Application.
 - Provides means of controlling the network through HTTP.
 - Possibly supports web socket connections with the User Application.
 - User Application (Web App) (Spencer & Grayson)
 - Sends HTTP requests to backend and shows responses in GUI.
 - Provides GUI for users to control the network.
 - Capable of visualizing live data streams.
- Kafka/WebSocket prototyping (Ross)
 - Researched Apache Kafka
 - Kafka can be used to create a publisher-subscriber communication model with ease
 - Topics can be created; producers can post messages to topics; consumers can read messages from specific topics
 - This will be useful for issuing commands from the network master node to the relay nodes
 - It is flexible enough to allow us to send data from the relay nodes to the network master node as well
 - Experimented with Kafka
 - Created a simple application to send data between a Spring Boot application and a Python program using Kafka
 - Spring Boot has built-in Kafka support
 - Python has several Kafka implementations; settled on the open-source kafka-python
 - Prototyped with Kafka
 - Wrote some code to simulate three relay nodes and a network master node
 - Network master requests data from relay nodes using one topic; relay nodes send messages back on a different topic
 - This will be integrated into the socket program that transfers video streams.
 - Experimented with WebSockets
 - Thought this might be a simple (high-level) way to transfer video data from python program to Spring Boot on network master node

Individual Contributions

Team Member	Contribution	Reporting Period Hours	Total Hours
Grayson Cox	Experimented with Spring Boot and Angular to test means of connecting the backend to the user application. Started experimenting with VLC and WebRTC to create a basic video streaming application consisting of a Spring Boot server and Angular client.	20	68
Austin Dvorak	Started working on the mesh network configuration testing.	30	78
Ryan Newell	Picked up hardware from ETG and started experimenting with OpenWRT. Experimented with configuring mesh networks on a Raspberry Pi.	18	66
Spencer Parry	Started experimenting with streaming UDP streams into a VLC plugin on HTML.	22	70
Ross Thedens	Experimented with Kafka to simulate control data flowing between relay nodes and network master node. Experimented with WebSockets for data transfer between python/Spring Boot applications on network master node. Contributed research and initial schedule to design document.	24	72

Plans for Next Period

- Decide on library for showing video streams in Angular (Grayson & Spencer)
 - Find ways to encapsulate the use of the chosen library to support modular design in the project.
 - Find out how the chosen library will affect the design of the backend/Spring application in terms of video format.
 - Make demo applications using the library to learn more about how it works.
 - Maybe include unit tests for the library.

- Continue testing the mesh network to find a feasible solution that can be implemented (Austin)
- Camera Testing (Ryan & Ross)
 - Once camera module is received, we need to begin testing it in OpenWrt, or another distro if we have to switch
 - Figure out if we can install the raspicam package in the distribution we are using, or if there is another way to make this work
- Consider use of status LED for indicating network connections (Ryan & Ross)
 - LED would turn on if connected to a network master node
 - Need to assess how this would fit with the Raspberry Pi case we have currently
- Get input from Dr. Daniels on whether Kafka is a good fit (better than ZeroMQ) (Ross)
 - Dr. Daniels used ZeroMQ successfully on a previous Pi project
 - However, we are planning to use Spring Boot, which has no existing ZeroMQ integration
- Update Design Document (All)
 - Ensure that diagrams, schedule, details are up to date with latest team decisions and planning before submitting V3
 - Write Testing & Implementation section (section 5)