EE / CprE / SE 492 – sddec20-05

An Advanced Networking Outreach Activity for Kids

Bi-Weekly Report #2
08/28/2020 – 09/13/2020
Client & Faculty Advisor: Dr. Tom Daniels

Team Members

Grayson Cox | UI Developer | Agile Project Manager
Austin Dvorak | Network Systems Manager
Malcolm Johnson
Ryan Newell | Hardware Systems Admin
Spencer Parry | UI Developer
Ross Thedens | Communication Systems Manager | Meeting Secretary

Reporting Period Summary

This week we wanted to focus finishing the video streaming feature for both the network side and the user application. However, the process ended up being more difficult than planned, so while Ross focused on it, the rest of us worked on laying the foundation for our entire project. We initialized the backend application with a simple REST controller for fetching node information from the server, the user application with some initial pages for viewing network information, and the network script. We now have the root project structure of our repository built, and we are now focusing on completing a functioning version of our project.

We also got a new team member, Malcolm.

Reporting Period Accomplishments

- Initialize UserApplication (Spencer & Grayson)
  - Create an initial Angular application
    - Create an empty tabbed Angular application
    - Implement sample data to test the overall look of the application
    - Add plugins to capture camera data
  - Upload and merge into Git
    - Confirm we have a working application for our Master branch
    - Create and merge our pull request into Git
- BackendApplication (Grayson)

  - Initialize the SpringBoot project
    - Created an empty SpringBoot application with SpringInitializr.
    - This was the first commit to the repository, so I also added a .gitignore file.
    - Opened a merge request to add in to the repository. The project is in the BackendApplication.
  - Create REST controller for nodes
    - I created a package in the BackendApplication called network_manager containing the classes that make up the node REST API.
    - Right now, this REST API returns static test data so that we can connect the UserApplication to it and see the data in the UI.
- Video Streaming (Ross)
  - Created a Docker container with all the dependencies for streaming H.264 video from a Raspberry Pi camera via WebRTC
    - Automatically loads all dependencies for the software involved
    - Uses Janus WebRTC gateway to send the WebRTC stream over the network
    - Uses FFmpeg to transcode the stream to H.264
    - Docker configured to allow easy access to host network interface and camera
    - Serves Janus demo webpage to test stream display for now
- Network Scripts (Austin)
  - Pushed network configuration script to repo and merged with develop branch
  - Added support for environment variables
- Python REST API for individual nodes (Austin)
  - Created an API to allow for network configuration from any node as well as getting all of the information about specific nodes.

Pending Issues

- Video streaming over mesh network
  - We still have to test the video stream in the real mesh network
  - Based on the performance, we may decide to run Janus on the network master node and receive the streams remotely, or have Janus run on all of the camera nodes, providing a direct connection to the instructor’s station

Individual Contributions

Team Member	Contribution	Reporting Period Hours	Total Hours
Grayson Cox	● Create SpringBoot application for network master node ● Write REST API for fetching nodes in the current network ● Assist in creating the initial user application in Angular ○ Reviewed merge request ● Assist in planning of the repository structure	20	32
Austin Dvorak	● Write python api for network configuration and viewing ● Modified network configuration file to use environment variables	18	30
Malcolm Johnson	● Added to team late in reporting period, nothing to report yet	-	-
Ryan Newell	● Narrowed down battery solutions that are rechargeable and meet other requirements ● Researched methods for gathering other stats (temperature, power usage, etc.)	12	24
Spencer Parry	● Made some minor bug fixes to the UI ● Assist in creating initial Angular application and created merge request for that	12	12
Ross Thedens	● Create and test Docker container for camera nodes to transmit WebRTC streams from Pi Camera	16	28

Plans for Next Period

- Establish remaining video streaming pathway (Ross)
    - Instead of using the demo webpage, figure out how to use the JavaScript API of Janus to manage a WebRTC connection
    - Expose this capability to the Angular webpage to minimize the need for UI developers to learn the WebRTC protocol and API
        - This likely involves creating a library that makes using the Janus API easier to use
- Containerize the code run on the network master node. (Grayson or Spencer)
    - Make a Docker image that holds the UserApplication and BackendApplication.
    - Ideally, this will allow us to test all the hardware.
- Create local configuration website for each node to allow for configuration of network when node is not connected to master node (Austin)
- Finalize battery and temperature sensor research (Ryan)
- Get familiar with project concept and technical work so far, determine role in project (Malcolm)