Senior Design Progress Report

|  |  |  |  |
| --- | --- | --- | --- |
| **Student**: | Brian Dye | **Team**: | 20 ENIGMA |
| **Semester**: | Spring 2022 | **Position**: | Team Leader |
| **Week**: | **3** | **Hours**: | 20-25 |

# Progress Description

Diagram

Description automatically generatedDuring this week, I review our past project on the Spanning Tree Protocol. I came to understand that we will need to make modifications in the algorithms we used in our project due to our objective of creating a wireless land area network. Our past project that utilized the Spanning Tree Protocol operated on the assumption of a **wired ethernet connection** to the ports on the switch. This is illustrated in Figure 1:

Figure 1: Switch Diagram

For each switch in our project, we don’t have data arriving and exiting through multiple different ports. We have 1 point of entry, and 1 point of exit. To overcome this problem, I believe the solution is to abstract “ports” with the MAC addresses of the sender to each switch. This abstraction is illustrated in Figure 2:

Wireless Switch

Figure 2: MAC address substitution for port

Diagram

Description automatically generatedBy knowing which neighboring node a packet was received from, we can turn the abstraction of mapping ports and links to **sender addresses** and **links**.

# Ordering Parts

This week my goal was to order electronic components that we can prototype on the ESP-32 development board and use on our Printed Circuit Board. The list of components is the following:

1. 4 Duck Radio Antenna’s
2. 3-4 RFM-69 Radio’s
3. 4 SMA Connectors
4. 4 GPS SAM-M8Q modules

We will be using antennas/connectors to increase the RFM-69 wireless transmission signal’s range and reliability. The GPS modules will used for a satellite application that demonstrates network status and node locations.

# RFM-69 Radio API

I wanted to make sure that team members have a clear vision of their individual responsibilities. I spoke with Hanyu and Henry about the expectations of interfacing with the RFM-69 radio module. I gave them the following RFM-69 API to implement to facilitate creating a wireless network:

|  |  |
| --- | --- |
| RFM69(addr) | Create an RFM69 object with address addr  Instance Variables:   * SPI object associated with RFM-69 * Node Addr = addr   Configure the RFM-69 with the highest bitrate and transmission power consumption |
| **void** sendTo(self, Addr, DATA) | Send Data to Addr using the RFM-69 |
| **bytearray** recv(self) | Get the data from the receive buffer of RFM-69 |
| **void** writeToReg(self, register, value) | Write value to register in RFM-69 |
| **int** readReg(self, register) | Read value from RFM-69 register |
| **bool** checkRecv(self) | Check if RFM-69 has received data |
| **void** lowerPowerMode(self) | Turn on the low power features of RFM-69 |
| **list** search(self, seconds) | Return the address(es) of other RFM69’s broadcasting using the same network ID |

Example API usage:



I helped Henry and Hanyu research previous implementations for interfacing with the RFM-69 to help them create our own implementation that will align with our objective of creating a wireless land area network. We found several open source RFM-69 modules written in variety of different programming languages.

# Network Applications and User Interface

I spoke with Nathan about the react native application that will implement a couple different sub applications that utilize the mobile ad-hoc network. We discussed the welcome page of the application and agreed on prompting the user to enter an identifier, and then allowing them to select which sub application they would like to use. Nathan and I also spoke about companies that occupy similar markets such as goTenna which has civilian and military applications in mesh networking.