Project Mesh Network

A local information monitoring service

Project Plan

Team Number: SDMAY19-21
Client: Danfoss/Radek Kornicki
Advisor: Craig Rupp

Team:
Collin Vincent -- Dev Ops, Network Engineer
Cody Lakin -- Software Developer, Data Acquisition
Gage Tenold -- Engagement Lead, Test Engineer
Colton Smith -- Project Manager, Hardware Integration
Will Paul -- System Architect, Hardware Integration
Ryker Tharp -- Documentation Review, Backend Developer

Team Email: sdmay19-21@iastate.edu
Team Website: http://sdmay19-21.sd.ece.iastate.edu
Revised: September 28th 2018 / Version 1.0

Tables of Contents

List of Figures and Tables

List of Symbols and Definitions

- Raspberry Pi (Pis): A small fully functional computer.
- Mesh Network: A local network of connected nodes that connect dynamically to one another and transfer data back and forth.
- Arch: A specific linux distribution that has a good ARM port for the raspberry Pi
- Distro: A set of user space packages that are included with the linux kernel in the base installation.
- Electron: A software framework for creating desktop GUI applications. Applications that are built using this framework include: Slack, Discord, and WhatsApp.
- Wifi: wireless radio wave base networking that is implemented based on IEEE standards
- CAN bus: CAN bus, or Controller Area Network is a widely implemented bus standard used to allow microcontrollers and devices to communicate with each other in applications without a host computer.
- Danfoss: A company based in Nordburg that specializes in creating products and services for a plethora of electronics and machinery related goods.
- SQLite: A sql database that is intended to be used in applications and has no server client model.
- Pylon: A piece of hardware that serves as a moveable access point that can extend the mesh network.
- Central Hub: The heart of the Mesh Network, the access point for reading the collected data and where all data is ultimately directed towards.
- Node: A point in the network that is taking in data and sending it further down the line to get to the Central Hub.

I. Introduction

1.1 Acknowledgement

The Mesh Network team would like to show appreciation to Danfoss for providing us with the expertise and hardware necessary for working on this project. We would also like to thank Craig Rupp for providing additional insight as our advisor and proposing a variety of useful technologies. As students we want to thank all involved for this opportunity to put our skills to use in a practical application, and to experience project development and management in a professional capacity.

1.2 Project Statement

Using a mesh network in a jobsite involving heavy equipment and/or a variety of vehicles is already a common practice. Collecting information and distributing it over a network allows managers to be better informed of the condition of their work which allows them to make better decisions regarding various operations. Information such as vehicle fuel level, location, etc. can be monitored to optimize the workflow of the job. These networks are only applicable in areas that support internet connections. This projects goal is to give job sites that lack this internet connection and provide them with the same information collection so they can improve those locations as well.

With internet each vehicle can connect directly to a service to transmit its collected data to a point where it can be processed and redistributed to those users that can utilize that information. Our mesh network solution is going to have those vehicles connect to each other rather than a global connection, and form a chain of communication connecting back to the users that require that information. In this way, work areas that don't have the conditions necessary for a data network can achieve the same benefits as those that do.

There is a market for companies that go out to under-developed countries or regions that don't have easily accessible internet/cell service. These companies are the target users of this project with our intention being to help them improve their operations through data collection and predictive services being performed locally. Users of this system will be able to install devices on their equipment that collect data and collate it into a hub where they can view that data in a meaningful format. Information such as fuel level for each vehicle can allow those users to optimize the paths that fuel delivery vehicles take which will decrease the slack time of each vehicle, decreasing the total time taken in a project. This is one of many aspects our users can take advantage of with this system.

1.3 Operating Environment

The operating environment will be the construction worksites that have limited access to network connections including cellular. Raspberry Pis will be used to facilitate the data collection and communication. These worksites can vary dramatically in size in terms of vehicles in use which has important design implications. Vehicles in use by the environment are varied in terms of operation data and importance. Work conditions within each environment can be quite different from other environments, resulting in different complications for each location.

1.4 Intended Users

This system is designed to provide a method for collecting and using information in an area with only the use of local networking. Therefore the intended users for this system are primarily companies/operations that are having to work in an environment where there is little to no internet or cellular service. An example of this would be in an underdeveloped country where a company may be creating roads to connect places of interest. In this scenario there wouldn't be reliable connection to the internet without the use of a satellite phone. To that end the network will connect the vehicles being operated by the company in this area through wifi and share the information they are collecting with each other and take this information back to a hub. There the manager(s) can utilize this data to track resource expenditure, equipment condition, and work being performed.

1.5 Assumptions and Limitations

Assumptions:

- After the members of this team have graduated, the project will be handed to Danfoss to be maintained and improved.
- End users will be able to understand the information collected and its implications.
- End users will have a device that will have software installed on it to operate as the central hub.
- The system will be modifiable to fit a wide array of applications/work.
- Each device installed on a vehicle will attempt to collect data 24/7.
- Connecting to any device should grant most if not all the information contained in the network.

Limitations:

- The project will use a Raspberry Pi device.
- Since collecting actual vehicle data is impractical, this data will be simulated.
- The system must function on a variety of devices.
- Only one month of data per vehicle will be stored on each device.
- For our purposes, a range of at most 100 meters will be tested

1.6 Expected End Product and Deliverables

There are three major deliverables to this project. At the end of the first semester a functional prototype that demonstrates the technologies we have chosen work together to achieve the goals of this project will be delivered. Towards the middle of the second semester a Proven Concept will be delivered, which should have all major components of the system functioning and working in tandem. Our final product delivered at the end of the second semester before we graduate will be a Hand-Off version of the system, so that Danfoss can take the project and modify/improve it for use in the field, and develop a marketable system for the project.

- Prototype - December 5th, 2018
    - Prototype will demonstrate the use of a distributed database to store and distribute the information.
    - Each Raspberry Pi will be able to connect to each other and transmit data.
    - A central hub will be able to connect to any Raspberry Pi and collect the data.
    - The central hub will have a preliminary means to display the data to an end user.
    - To demonstrate the success of this stage, the prototype will be shown in a small environment, connecting each of the six Pis contained in the system. Each Pi will collect and share information between themselves, which we will demonstrate through the use of the initial front-end application. Our client should be able to see that the information collected by each Pi persists across the network, and is visible and readable to the end user.
- Proven Concept - April 1st, 2019
    - Each Raspberry Pi will simulate field data and connect to each other device when necessary.
    - The central hub will have an complete front end application that displays the data in readable and meaningful manner.
    - Predictive analysis is being used to take the data collected and further assist end users in utilizing the information to improve their operations.
    - For the proven concept the Pis will be distributed around a geographic area, establishing the required 100m distance for the node-to-node connections. We will establish the ability of the system to ignore faults, having the devices turn off and on to simulate disconnects. The system at this point should be able to route around the disconnects, or alert the end user that a node is out of range.
- Hand-Off Version - May 1st, 2019
    - After all required aspects of the project have been completed, the focus will be to completely document the system and make it easily understood and configurable by the teams at Danfoss. If possible, devices should be interchangeable and the network should be flexible in terms of size, data, and device type.
    - All documentation will be prepared for transfer.
    - All code will be finalized and documented well.
    - Modification and manuals for both software and hardware information will be prepared and available.

- For the testing of this phase, our client will have to be involved in a review of our documentation, and they should be able to operate and work on the system without our assistance. The system should be packageable, that is to say we should be able to hand our client a complete and working system for them to examine and improve upon.

II. Proposed Approach and Statement of Work

2.1 Functional Requirements

The Mesh Network's functional requirements are abstract since the environments in which it will be operating are considerably different from each other. The main expected use case is for users to view the information collected by the network. To accomplish this use case, the functional requirements of the system are:

- Set up the local network on a user's device.
- Add/Remove devices from the network.
- Delete a devices information from the network storage.
- View the information of all the vehicles.
- View the information of a specific vehicle in detail.
- View predictions made by the system for each vehicle.
- Upload the information collected to a server.

2.2 Constraints and Considerations

2.2.1 Non-Functional Requirements

Scalability - Our network needs to allow for easy addition for up to one hundred nodes to be connected through it.

Availability - The data collected through our machines should be collected every few seconds, and the portal through which the information can be viewed for all machines should have no more than a 5 minute latency.

Reliability / Recoverability - Data being collected through our hardware should be stored on every adjacent node, and should contain a log of the given vehicle reading for the last 30 days.

Maintainability - Once the final solution is in place, replacement for the network adapters should be as simple as copying over our program onto the hardware, and setting the new Pi up to the machines CAN bus link.

Security - For this project, at the moment there are no real security plans set in stone. The telemetry data is rather harmless and has no real malicious use available. A base encryption for the data will need to be considered for the future if any government or other types of sensitive work is undertaken.

Data Integrity - The data will be reliably recorded every second, and will only be transmitted through reliable vehicles connected to the mesh network.

Usability- Once the solution is in place, using the system should be as simple as driving a piece of machinery with its hardware in place onto a pre approved worksite, and opening up the application to view the related data.

Localization- The solution will function without any connection to external networks, except the potential upload of data from the central hub to the cloud.

2.2.2 Standards

The following standards will be met during the course of this project-

- IEEE Wifi standard 802.11b
    - We are using this standard for our ad hoc wifi network
    - The iwconfig tool for configuring the network uses the 802.11 standard
- Http/2 RFC 7540
    - Node js follows the http/2 standard and we are going to use node for our http applications so we will be following the http/2 standard
- J1939 SAE
    - This is a standard developed by SAE to transmit important diagnostic heavy machinery data over CAN bus.
- CAN
    - CAN is a vehicle bus standard designed to allow microcontrollers and other devices to transmit data with or without the use of a host computer.

2.3 Technology Considerations

To best implement this project we had several aspects of the project that had multiple potential solutions. We took some time to deliberate and evaluate each of these potential solutions and decided on which ones would be best able to meet the requirements of the project. There are five sections that required research: database type and distribution, linux distribution, hardware device, frontend software language, and communication method.

- For database type and distribution, we first considered just using local SQL servers to accomplish our data storage needs. However, the alternative of SQLite, or a similar distributive database, a system based on SQL that is lightweight and allows the database on one device to be easily shared across the network, was an attractive option. This will be installed on each of the nodes and on the central hub in the network and will help up propagate the data collected throughout the system.
- For Linux we plan to use the arm port of Arch because it is a lightweight base install. This allows us to keep more resources available for our actual application. Members of our team also have extensive prior experience with network configurations on this specific distro so it should reduce time spent on that portion of the project. On top of that the documentation for the distro is very extensive and the community is particularly active so outside help should be easy to get if something should go wrong.

- For hardware, we were offered the options of Raspberry Pis vs a device that Danfoss makes in house. After talking with our point-of-contact at Danfoss we decided that working with the Raspberry Pis was the optimal solution due to having experience with the devices and having a large amount of public documentation available. Danfoss will have the option of converting the network to their devices after hand off.
- For frontend software language we are currently deciding between electron and java. Electron has the benefit of being easily transported from device to device while with java we have more experience and it is a more documented and well-known option.
- For communication method We are going to use the built in wifi cards in ad-hoc mode to allow the Pi's to communicate.

Table 1

OS / Application / Component	
Node Operating System	Arch Linux
Front End Application	Electron / HTML / JavaScript
Database	SQLite
Version Control	GitLab / GitKraken
Runtime Environment	Python / NodeJS
Package Manager	PIP / NPM

2.4 Safety Considerations

Our solution will consist of Raspberry Pis connected to CAN bus ports on various machinery, that will communicate and share data with a remote desktop application. Thanks to the nature of these deliverables, no harm should come to any personnel or users associated with our solution.

Having a local area network one possible safety concern in unauthorized devices accessing the network. To prevent this from occurring we are going to have a specific set of devices that are allowed to access the network. The access may be granted by MAC address or some other unique identifier that the devices have.

2.5 Previous Work and Literature

Danfoss currently has a solution that acquiesces work sites with cellular connections. This solution functions very well in more developed areas, but serious issues arise in areas where cellular connections become more spotty. When in a position with an unreliable connection to

the mesh network, valuable machinery can be put at risk with the current network type and this is unacceptable. Our project will not be a continuation of Danfoss's current solution, but it will serve the same purpose.
Ad Hoc networks can solve this issue by allowing direct communication between devices that are near each other. There is a IEEE paper, *A review of current routing protocols for ad hoc mobile wireless networks*, that discusses how ad hoc networks work that will be the base of our implementation.

Example 1
Example 2
Example 3
Example 4
Example 5

2.6 Possible Risks and Risk Management

Since our team is entirely comprised of software engineering students issues that pertain to hardware will be particularly concerning. Collin, Colton, and Will each have a decent of understanding of the hardware we are working with which puts a heavier burden on them in regards to hardware problems.

We each have a decent understanding of front end languages and GUIs but depending on the language we end up choosing having to create a useful and professional visualization of the data will require us to undergo a learning period. That learning period could potentially upset the flow of our schedule, so considerations will have to be made in the planning.

Depending on the type of connections we create between the nodes/hub in our system the network protocols being used could cause problems transferring data. To this end we are going to keep the other network protocols open for review and plan to test them out if given enough time. In addition the types of networking we have discovered that would be feasible all suffer from the limitation of range. This is planned to be mitigated by getting higher power hardware after handing the project off to Danfoss.

2.7 Project Proposed Milestones and Evaluation Criteria

- Networking Prototype
    - This milestone will be met when we successfully get the 6 Raspberry Pi devices to successfully connect to each other when in range, and pass simple messages through the connection.

    - The devices will need to keep track of what other devices are connected to the mesh network and make sure that they are connecting to the optimal device to allow maximum inclusivity to the network.
    - This stage will not use a database or store data.
    - Testing for this phase will consist of manual tests that showcase the individual devices ability to connect to the others on-the-fly, by turning devices off and on in random sequences where we will examine and record the number of nodes still connected. We will also test our ability to introduce nodes into the network, changing configuration information on the Pis and then re-integrating them into the network.
- Distributed Database Prototype
    - This milestone will be met when we successfully incorporate the database connections on our mesh network from the previous prototype.
    - At this step we will need to ensure all devices in the network reach consensus on the current state of the database.
    - To test this phase, the Networking prototype must be operational. Testing of this phase will consist of recording data on each of the devices then connecting the nodes to see if they will receive the new information. The time it takes for devices to share this information will be recorded as well as the accuracy of shared information. Additionally, nodes will be introduced to the network and the result should be that those nodes are “caught-up” on the information of the other devices, and that other devices begin to collect the new node’s information. This information will be discussed with our client and when satisfactory numbers are achieved this phase will be complete.
- Base Prototype
    - Data should be collected from the interface used by the work equipment.
    - The collected data should be stored in the database on the device.
    - The network should propagate the information to all other devices connected to the network.
    - The testing of this phase will consist of collecting real/simulated data for each node, and then modifying the network’s size and distance apart. We will be examining how the network handles various issues, specifically networks of small to medium size, 0-100m distances, and nodes being disabled. These issues should not be able to impact the network’s performance which will be shown by the information for each device being able to reach the central hub, where verification against an expected database will be performed.
- Minimum Viable Product
    - This milestone will be met once we get the base desktop application to display the basic data.
    - This will be the first implementation of the network on a device other than the Raspberry Pi.

    - Once the application is able to collect the data off a Pi in the network and display the data in the database in an easy to use application we will consider the Minimum requirements for the project met.
    - To test this phase our frontend application must be easily readable by the end-users and the client, and each of the functional requirements must be met and demonstrable.
- Alpha Testing
    - The minimum viable product will be given to the client to use with the actual equipment.
    - The client will report any issues for debugging and patching.
- Cloud Storage and Analytics
    - When a Internet connection is available the database should be backed up on a more permanent location.
    - The data should be made available for another application to consume and be used for Analytics.
    - Testing of this phase will consist of periodically connecting a live network to the internet to see if the database will be uploaded without fault to a remote location. In addition rudimentary analytics should be performed and displayed in the front-end application. Such analytics should include the rate of fuel consumption, setting flags within the system when oil temp, pressure, or level approach unsafe levels.
- Beta Testing
    - Issues identified in alpha testing should be resolved.
    - Solution will be returned to the client to further test.
    - Additional findings will be documented and reported back by client.
- Project Completion
    - Issues found during testing will be resolved.
    - Documentation on the state of the product will be produced and turned over to the client.
    - All code and resources will be thoroughly documented and turned over to the client .

2.8 Project Tracking Procedures

Milestones/Deliverables:

- Prototype: Have connections between devices and information being stored and passed between devices.
- Proven Concept: Have the practical application/functional requirements of the device accomplished.

- Hand Off Version: Have all documentation completed and reviewed, have the code refined and componentized to make the Danfoss integration as smooth as possible.

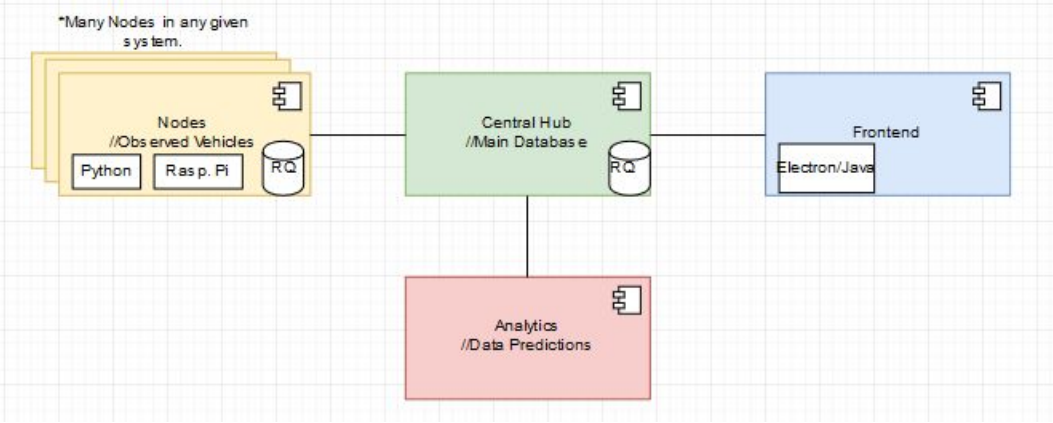
Project Tracking Methods:

- Trello
    - Trello will allow us to track the progress of each individual within the team helping to decrease slack time from dependencies among tasks, and prevent repeated work from occurring since each member will be able to see the todo list of each other member.
- Shared Google Drive Repository
    - Any updates to our documentation will be readily available and modifiable by each member of the team.
    - The documents included in here include the project plan, weekly meetings, weekly progress reports, etc.
- Weekly meetings
    - With our client and advisor we will meet each week to discuss the progress made, modifications to the schedule, and clarifications as the project progresses.
    - With each other, we will meet weekly to discuss progress on our individual tasks and blockers that have arisen. We will check the performance of each individual and provide help to improve the other's performance week-to-week.

2.9 Objective of the Task

Create a system that can connect at least 6 devices to each other or the central hub and collect data from those devices. That information will be collated by the database into a front end application that will allow the end user to modify the contents of the network, view the information from each device in the network, and view predictive information about the devices in the network. The devices will consist of Raspberry Pis and some set of sensors, while the central hub will be a personal device running the hub software from one of the team members.

Figure I: System Architecture Diagram



2.10 Task Approach

Figure II: Design Process Diagram

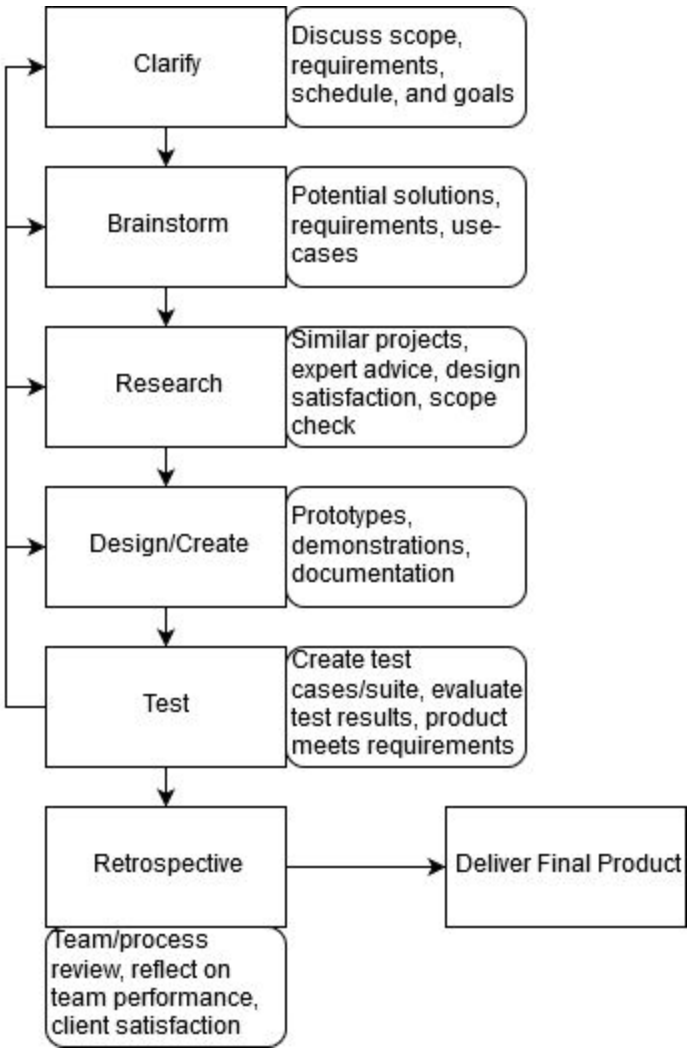
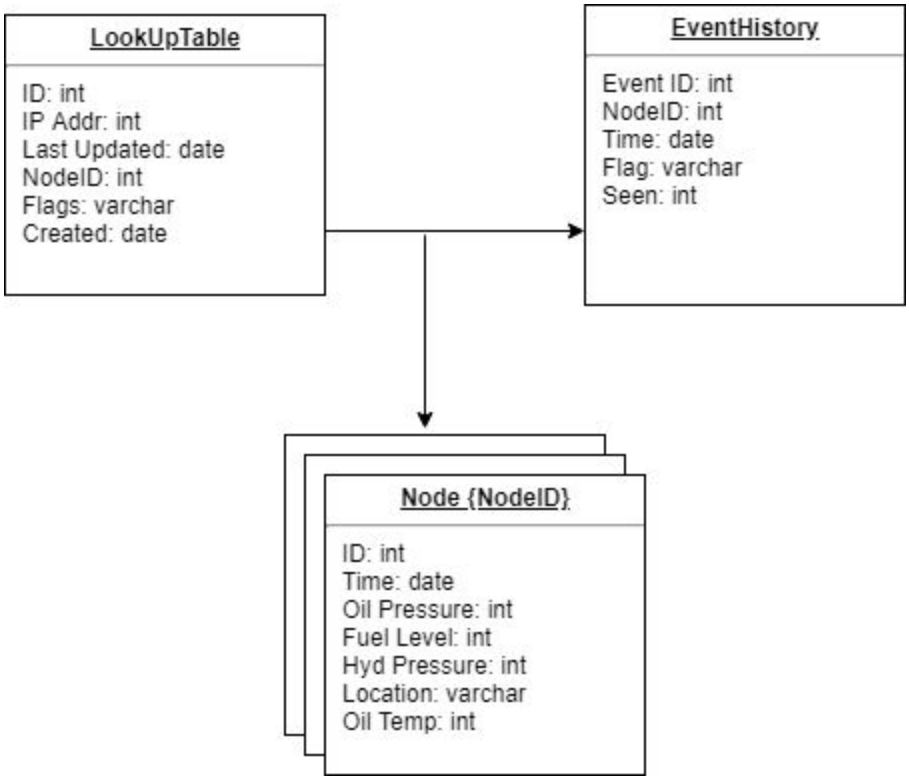


Figure III: Database Schema Diagram



The database schema is broken into 3 main sections. The first table, the LookUpTable (LUT) is where each vehicle's configuration information is kept, such as its IP address, when each vehicle was last updated on this specific node/vehicle. The flags field for the LUT will indicate any specific concerns or information we need to know about a specific vehicle. NodeID will link the tables, connecting to a dynamic table scheme, N{NodeID}. This set of tables will contain the recorded data for each vehicle, and time-stamp it. The timestamps of LastUpdated and the timestamps on N{NodeID} will be used to coordinate the updates between each node. EventHistory is the record of any specific events or flags that are raised for each vehicle. This table will be used during the analytics portion of the project. This is an early sketch of the database schema, and additional fields and tables are expected at this point.

2.11 Expected Results and Validation

The end goal of the project is to create a local network that will collect information on various vehicles operating within the system. A user's interaction with this system should be to connect devices initially within the system and then to view the data collected from each device. This information will be viewable as an overview of all machines or as an individual view per machine.

To test the network there will be unit testing and benchmark tests to test the reliability, speed, and accuracy of the network. These will be performed in demonstrations of the system that will also showcase how the front end application responds in real time to changes with the system.
Functional Tests:

- CAN bus data collection will be tested using SocketCAN's vcan module and/or PiCAN 2 boards paired with data emulators. Additionally, example output is being provided by both Craig and Radek, which will also be used for testing purposes. Collection scripts will be run and the acquired data will be compared to the inserted values.
- To test the nodes collection ability, we are going to have them simultaneously take in five streams of data, simulating oil pressure, oil temperature, fuel level, hydraulic pressure, gps location, and possibly other sensor data on boards provided by Radek. All data received from a CAN bus or sensors will be formatted and stored. This will serve as a proof-of-concept for a variety of data that the product could be used for.
- To test the nodes ability to connect to each other and the hub, we plan to have at minimum three devices running, and turn their connections on and off to test the system's ability to manage the connections between each device.
- To test the central hub's ability to collate the information we will have at minimum three devices connected in sequence collecting and transferring data which we will make tests to verify that data is the same throughout each device.
- To test the system's ability to dynamically add nodes we will have a single device connected to the central hub and then start activating and deactivating nodes in sequences that will be specified at a later time.

Non-Functional Tests:

- Performance will be stressed here. We will have to test the system latency to make sure the system responds to vehicle changes and updates the hub in a reasonable amount of time. Radek (our client) did not give us a specific time requirement, but he did say it should be, more or less, "Live".
- It is required that the solution works locally. Devices will not be configured to access networks other than that which the devices compose (no external networks). Successful operation of the solution will prove that it works locally.
- The frontend will have to be extremely usable, because it will be monitored and operated by a construction site foreman. We can test this by letting friends/family use the software and give feedback. In addition to an easy to use frontend, we also need to make sure the hardware is simplistic and usable as well. Having an employee go out to connect our Pi to a port in a piece of machinery should be about as complex as it will get.
- Security is going to be another important area to test. We need to ensure that the only devices that are authorized are attached to the network and all others are barred.

III. Estimated Resources and Project Timeline

3.1 Personnel Effort Requirements

Table II: Task Breakdown Table

Task	Description	Estimated Hours
Setup Database	• Install database software • Create database • Create database schema	5 hours
Database storage	• Develop a script to store collected data in the database	8 hours
Database distribution	• Configure to store and distribute information to a database on other devices	10 hours
Pis are setup with OS and needed software	• Database software • Libraries for working with sensors and CAN bus • Networking Protocols	10 hours
Pi-to-Pi connection	• Pis are able to connect to other nearby devices	10 hours
Data Collection	• Pis are able to interpret data from sensors that are provided by Danfoss • Pis are able to interpret data from CAN bus connections • All collected data is organized into a standard format and made ready to store or export	35 hours
Store Self-history	• Pis are able to store their own data history in a database • Pis store data as it is collected from sensors and CAN bus connections	10 hours
Pi-to-Pi Data Transmission	• Pis are able to send data to and receive data from other devices	80 hours
Store Complete History	• Pis are able to stored collected data from all devices, creating a complete history of the network on each device • Pis determine which data should be cleared after a certain amount of time, perhaps 30 days.	50 hours

Broadcasting	• Pis broadcast their data to other devices at appropriate intervals	20 hours
Hub-to-Pi Connection	• Central hub is able to connect to a device	30 hours
Hub Pulls History	• Central hub is able to pull a device's data history from the network	20 hours
Pis Route Data to Hub	• Pis route data correctly through the network to the central hub	30 hours
Hub Displays Raw Data	• Central hub collects all known and reachable data • Central hub displays all data in a primitive way	40 hours
Pylons	• An intermediary "pylon" is capable of extending the network	30 hours
Documentation	• All Raspberry Pi scripts are documented • All desktop app code is documented	10 hours

Table II: Task Breakdown Table (continued)

*Schedule only accounts for first semester tasks.

3.2 Other Resource Requirements

Hardware: To create the deliverables for this project Raspberry Pis, sd cards, and a variety of sensors will be required. The specifics of the hardware is being left to Danfoss since the project will eventually be placed into their hands exclusively.

Software: IDEs will be left up the individuals of the team so that their progress will be assisted by their understanding of their personnel software choices. Additionally, installations of software like SQLite will be required on both the desktop application and the nodes themselves. We are also going to be using some free software for this project including avahi, nodejs, iwconfig, systemd, and electron.

Other: Network protocols and specifications will have to be chosen that do not impact the environments around them.

3.3 Financial Requirements

Our project will use only free software. We will use open-source libraries and technologies. Danfoss will be covering our hardware needs. This will include the devices that are to make up the network, sensors and CAN connectors for the devices, and wifi modules for each device. At this time, we are in possession of Raspberry Pi devices and enclosures for

them. This table will also be update in the event that the scope of our project is adjusted and we require additional hardware or upgraded components.

Table III: Cost Table

Item	Cost
Raspberry Pi 3 Model B x 6	~$210.00
32GB Micro sd card's x 6	~$60.00
Can Bus kit x 6	~$300.00

Table III: Cost Table (continued)

3.4 Project Timeline

Table IV: Fall Project/Deliverables Timeline

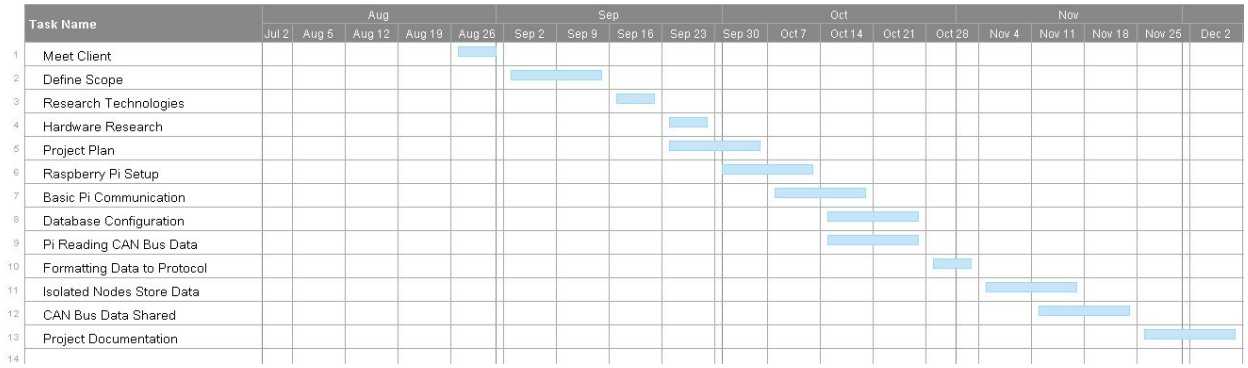
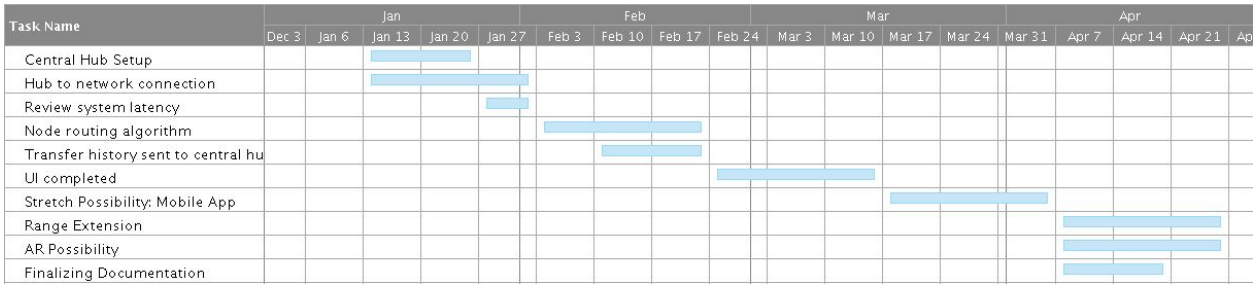


Table V: Spring Project/Deliverables Timeline



4 Closure Material

4.1 Conclusion

At this stage of the project, we've moved past the conceptual stage and have started working on our prototype deliverables. We've got the go ahead from the client to work on specific tasks and have now broken the team down into a Database Team, a Hardware Team, and a person working with Networking and our UI. The Database team is working with distributed databases to come up with a solution to keeping a synched record of the whole network on each node. The Hardware Team is working to learn to read input from CAN bus and how to decode these messages to forward to our databases. On the networking side, we've begun setting up and Ad Hoc network to sync each of the Raspberry Pi's wirelessly. Lastly we're also starting on the UI while waiting for more hardware, and have made a base UI and are trying to re implement it with Electron. For now, we have our goals set and are ready to continue working towards them.

4.2 References

Toh, Chai-Keong, and E. M. Royer. "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks." A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks - IEEE Journals & Magazine, IEEE, Apr. 1999, ieeexplore.ieee.org/abstract/document/760423.

<will be gathered later, will include the website of the technologies used, pricing sources, previous projects examined, etc.>

4.3 Appendices

<will be compiled later, libraries for each of the languages / frameworks/ tools will be included here once we get further into the project.>
Electron, SQLite/alternative, Arch, Raspberry Pi, communication protocols, SocketCAN, pandas, Python