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 -- General Software Developer, Data Visualization
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

- (This will probably be filled with every acronym and technical term we use after doing the second iteration)

1 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 a 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.

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.

- Prototype - December 5th, 2018
  - Prototype will demonstrate the use of RQlite 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.
  - A demonstration should be available to show the client the results of this prototype.
- 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.
  - A demonstration of the complete system’s facilities should be available to prove the results to the client.
- 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.

2 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 considerable different form 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.

2.2.2 Standards

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

- Projects Planning
    - Tasks and duties will be distributed and controlled through Trello
    - Communication on the status of activities will also be reported though our team Slack channel
- Project Version Control
    - Git will be used to keep each iteration of the code base
    - Google Drive will be used to keep each iteration of each of our project documents
- Software Testing Suite
    - To be determined once our primary language has been firmly decided

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 RQLite, a system based on SQL that allows copies of the database to be easily distributed across nodes in a network was an immediately attractive option, and one that we decided to use going forward.
- For Linux we plan to use the arm port of (Arch/Alpine TBD) because it is a very stripped down base install. Allowing 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 there are a few options that potentially fulfill the requirements that we need. They are ZigBee, Bluetooth, and Wifi. Currently the project is leaning towards using Wifi as the base for our network as it is already used extensively and the ability to make the hardware act as a router is relatively simplistic. However one possible obstacle is effective range. This is where the other network protocols come into play. Each have their own pros and cons so more research is needed for a confident decision.

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.

Once we begin to implement our own wi-fi based solution, we will begin to look at other mesh networks that utilize wi-fi and attempt to learn from their implementations.

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 rqlite or store data.
- Rqlite Prototype
    - This milestone will be met when we successfully incorporate the rqlite 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 sqlite database.
- Base Prototype
    - Data should be collected from the interface used by the work equipment
    - The collected data should be stored in the rqlite database on the device
    - The network should propagate the information to all other devices connected to to the network
- Minimum Viable Product
    - This milestone will 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.

- Alpha Testing
  - The MVP will be given to the client to use will the actual equipment
  - The client will not any issues and report them back for debugging and patching
- Cloud Storage and Analytics
  - When a Internet connection is available the database should be backup up on a more permanent location
  - The data should be made available for another application to consume and used for Analytics.
- 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 RQLite 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.

<Insert Architecture diagram here>

2.10 Task Approach

The first step to accomplish the task at hand is to decide on a network protocol that will fulfill our requirements the best. This requires some extensive research in the possible technologies that we have chose.
<insert tech stack here>
<insert db scheme here>

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.

3 Estimated Resources and Project Timeline

3.1 Personnel Effort Requirements

Task	Description	Estimated Hours
Setup RQlite	• Install RQlite • Create RQlite database • Create database schema	5 hours
Database storage	• Develop a script to store collected data in the RQlite database	8 hours
Database distribution	• Configure RQlite to store and distribute information to a database on other devices	10 hours
Pis are setup with OS and needed software	• RQlite • 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

*Schedule only accounts for first semester tasks.

*Hours are low due to expected setbacks with technologies not being compatible.

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 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 RQLite will be required on both the desktop application and the nodes themselves.

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.

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
dhcpd (wifi access software)	$0.00
bind (dns software)	$0.00
arch/alpine	$0.00
rqlite	$0.00
Electron (desktop app)	$0.00

3.4 Project Timeline

Deliverable	**Start Date**	**End Date**
Setup Rqlite tables	10/5/2018	10/12/2018
Configure Rqlite to store and distribute information	10/12/2018	10/26/2018
Pis are setup with OS and needed software	10/5/2018	10/12/2018
Pis are able to connect to other nearby devices	10/12/2018	10/19/2018
Pis are able to interpret data from sensors and CAN bus	10/12/2018	11/2/2018
Pis are able to store their own data history in a database	11/2/2018	11/9/2018
Pis are able to send data to and receive data from other devices	11/9/2018	11/16/2018

Pis are able to store data from all devices up to one month	11/16/2018	11/23/2018
Pis broadcast their data to other devices at appropriate intervals	11/23/2018	11/30/2018
Pis route data correctly through the network to the central hub	11/30/2018	12/5/2018
Central hub is able to connect to a device	10/31/2018	11/6/2018
Central hub is able to pull a device's data history from the network	11/6/2018	11/13/2018
Central hub displays all data in a primitive way	11/13/2018	11/20/2018
An intermediary "pylon" is capable of extending the network	11/14/2018	11/28/2018

4 Closure Material

4.1 Conclusion

At this point in time, our team has firmed up our list of deliverables with our client and received preliminary hardware to begin work with. From this stage, we will begin loading up our Raspberry Pi’s with the base software to get started, and begin looking into possible languages to use. By week 10, we hope to move into a phase where we can begin splitting off into hardware and software teams, and move into testing other wireless networks types by Danfoss’ request. Once the network types are finalized and we decide on what to use to build the front end, we will have a lot of work in front of us, but we should have the time to accomplish all of this. At the moment, things look hopeful.

4.2 References

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

4.3 Appendices

<will be done later, libraries for each of the languages / frameworks/ tools will be included here.>