

An Advanced Networking Outreach Activity for Kids

DESIGN DOCUMENT

sddec20-05

Dr. Tom Daniels

Grayson Cox | UI Developer | Agile Project Manager

Austin Dvorak | Network Systems Manager

Ryan Newell | System Admin

Spencer Parry | UI Developer

Ross Thedens | Hardware Systems Manager | Meeting Secretary

Team Email: sddec20-05@iastate.edu

Team Website: <http://sddec20-05.sd.ece.iastate.edu/>

Revised: March 29, 2020 (V2)

Executive Summary

Development Standards & Practices Used

- Standards
 - Quality management (ISO 9001)
 - Developing information for users in an agile environment (ISO/IEC/IEEE 26515)
- Practices
 - Agile development
 - Code reviews
 - Weekly meetings
 - Rigorous testing for security and reliability

Summary of Requirements

- Support mesh and dynamic ad-hoc networking capabilities
- Ability to stream data between nodes
- Easy to use and set up
- Hardware should be mobile and durable
- Can be operated without the need for an Internet connection
- Range between nodes should support a 40 foot radius (when in eye-sight)
- Nodes will consist of a single-board computer, rechargeable battery, and camera/sensors

Applicable Courses from Iowa State University Curriculum

- CprE 489 - Computer Networking and Data Communications
- CprE 430 - Network Protocols and Security
- CprE 537 - Wireless Network Security
- CprE 230 - Cyber Security Fundamentals
- ComS 252 - Linux Operating System Essentials
- SE 319 - Construction of User Interfaces
- SE 329 - Software Project Management
- ComS 309 - Software Development Practices

New Skills/Knowledge acquired that was not taught in courses

- Web Development with Angular/Typescript
- Ad-Hoc/Mesh Networking and Router Configurations
- Raspberry Pi Configuration

Table of Contents

1. Introduction	4
1.1 Acknowledgement	4
1.2 Problem and Project Statement	4
1.3 Operational Environment	4
1.4 Requirements	5
1.5 Intended Users and Uses	5
1.6 Assumptions and Limitations	6
1.7 Expected End Product and Deliverables	6
2. Specifications and Analysis	7
2.1 Proposed Approach	7
2.2 Design Analysis	7
2.3 Development Process	7
2.4 Conceptual Sketch	9
3. Statement of Work	11
3.1 Previous Work And Literature	11
3.2 Technology Considerations	12
3.3 Task Decomposition	12
3.4 Possible Risks And Risk Management	13
3.5 Project Proposed Milestones and Evaluation Criteria	13
3.6 Project Tracking Procedures	14
3.7 Expected Results and Validation	14
4. Project Timeline, Estimated Resources, and Challenges	15
4.1 Project Timeline	15
4.2 Feasibility Assessment	19
4.3 Personnel Effort Requirements	19
4.4 Other Resource Requirements	19
4.5 Financial Requirements	20
5. Testing and Implementation	20

5.1 Interface Specifications	20
5.2 Hardware and software	20
5.3 Functional Testing	20
5.4 Non-Functional Testing	20
5.5 Process	20
5.6 Results	21
6. Closing Material	21
6.1 Conclusion	21
6.2 References	21
6.3 Appendices	21

Figures and Tables

FIGURES

1: Conceptual Sketch	9
2: System Modules	10
3: Gantt Chart CprE 491 (01/19 - 04/04)	15
4: Gantt Chart CprE 491 (04/05 - 05/02)	16
5: Gantt Chart CprE 492 (08/23 - 10/24)	17
6: Gantt Chart CprE 492 (10/25 - 12/12)	18

TABLES

1: Task Estimated Time to Completion Table	19
--	----

1. Introduction

1.1 ACKNOWLEDGEMENT

We would like to acknowledge our client and advisor Dr. Tom Daniels for his guidance and support on this project.

1.2 PROBLEM AND PROJECT STATEMENT

Wireless networking has become a fundamental part of many people's everyday lives. This is especially true for students across grades 4-12, who are beginning to spend hours each day on school-issued internet-connected devices. However, few students understand the technology that underpins these networks. The internet is often portrayed as an amorphous portal granting access to huge amounts of information. In reality, this information is stored on servers around the world and delivered to end users via routing protocols. By creating a teaching tool that demonstrates routing protocols, we can illustrate the challenges involved in creating a reliable wireless network, increase technology literacy, and stimulate students' interest in computer networking careers.

Our approach involves creating a teaching toolkit consisting of a set of portable wireless network nodes, a network monitor/configuration application, and a set of lesson plans. A grade school instructor with no computer networking training will be able to operate it. The nodes will automatically form an ad-hoc network with one another when powered on (and within range of one another). Network data will be provided by video cameras and various sensors on the nodes, such as temperature sensors. The data will travel the network to reach a master node which is directly connected to a network monitor application running on an instructor's laptop. The monitor application will control which sensor/video streams are delivered to the master node and display the data received. The lessons will challenge students to transmit data across a long distance (beyond the range of a single node; from the main office to a classroom, for instance) to the network monitor application. As students add nodes into the network and position them to attain connectivity, they will observe the routing of the data (which nodes are reached) in the monitor application. They will use this information to figure out where data is unable to travel and rearrange the nodes accordingly.

Ultimately, our toolkit will act as a miniature mock-up of the real internet. Students will be able to appreciate how data from a web server hops from node to node over the internet before reaching their devices. Following the activities, the instructor will discuss grade level-appropriate topics related to networking from the lesson plan with the class. For younger students, this may involve the fact that wireless signals have a limited range, while for older students it may include a discussion of the routing decisions made by the ad-hoc protocol. Our project will provide a flexible way to teach computer networking to grade school students.

1.3 OPERATIONAL ENVIRONMENT

A collection of networking nodes will compose the main aspect of the system. It is expected that these nodes will be used indoors, but they should be able to operate outdoors as well. They do not

need to be weatherproof, however, they will need to be durable enough to protect the internal components.

There will be a graphical user interface aspect to the system, which will allow users to view sensor and video data transmitted from the nodes. This can be accessed via any web browser on a separate computer, and no internet connection should be necessary as long as there is a connection between the computer and the base node.

1.4 REQUIREMENTS

Nodes will need to be powered by rechargeable batteries so that they do not require a power cord when in use. They will also connect wirelessly to other nodes. Both of these requirements ensure that nodes are mobile and can be used without the restriction of cords.

Each node should support ad-hoc networking capabilities. Multiple networks should be able to coincide and stay independent of each other. Nodes should use their respective networks and safely handle interference and other network errors. To demonstrate functionality, nodes will send data across the network. Camera feed, sensor data, and control signals will need to be transmitted through nodes to their destination. The graphical user interface will display this data and give a visualization of the network.

The packaging of each node will need to be durable enough to survive drops since they will be moved around quite frequently. Each node will consist of a single-board computer, a wireless network adapter, and a rechargeable battery. Some nodes will contain sensors and cameras to serve as data sources for the network. Components will be bundled together in a package that makes it intuitive to charge and easy to use and set up.

Besides being wireless and easy to place, the software should also be user-friendly and intuitive to use. The instructor may not have much knowledge on computer systems so the monitoring application should be simple enough that most kids and adults can understand the interface. To aid in the ease of set up, nodes will not need to be connected to an existing network. The only network that they will use is the one they create themselves.

1.5 INTENDED USERS AND USES

The project has three main types of users: students, instructors, and system administrators. The goal of the project is to educate students on how mesh networking works, therefore, the students would be the target audience. The students would not interact with the GUI as much as the instructors or teachers would, but the GUI should be user friendly enough for anyone in the three target audience groups to use. The students largely will be moving or placing the nodes in different places, and the instructors will use the GUI to interpret the data to the class.

The instructors would have knowledge on how to use the nodes and the GUI so they can teach their students some of the principle lessons that these devices could teach, such as range and interference. They would know how to position the nodes to display on the GUI how different forms of interference can cause the video to buffer more or less than other forms of interference. The instructor would also use the GUI to change settings on the nodes, like grouping nodes into two separate groups.

The system administrators would have access to the file system and command line of the nodes to manage updates or to diagnose problems with the nodes. The administrators would only have to be involved if there is a problem with one of the nodes, or a mandatory update needs to take place. Students and instructors would not have access to the system administrator settings; however, system administrators would have full system access, as well as access to the student and instructor GUI.

1.6 ASSUMPTIONS AND LIMITATIONS

Assumptions

- The user owns a laptop computer with a web browser
- Person running this activity is able to charge the nodes when batteries run low

Limitations

- The project will be completed by the end of the fall 2020 semester.
- Nodes will need to be mobile
- Internet connection may not be available
- Each node will be powered by a single-board computer
- Storage of nodes is tight
- Battery life should last at least two hour

1.7 EXPECTED END PRODUCT AND DELIVERABLES

The main deliverables will be a set of at least eight wireless, rechargeable nodes. Some of which will be sensor nodes and the rest will be relay nodes whose purpose is only to conduct data from the sensor nodes. Each sensor node will have one sensor for collecting data from an external source and sending it into the network to be viewed by the users on visual display. We expect that at least one of these sensor nodes will have a camera; the selection of other sensors will be made at a later date. We anticipate delivery of functioning relay nodes by October 1 because of the simplicity in their design. As for the sensor nodes, we expect to deliver a functioning set by October 15.

In order for the users to access the data within the network, there will be a base station, which acts as the gateway between the instructor's machine and the node network. The instructor will be able to connect to this base station and access the user interface component. This base station component is to be delivered along with the set of sensor nodes by October 15.

The user interface component will be in the form of a web-based application that can be accessed through any web browser. The UI will offer at least two displays: one large display for visualizing a specific stream of data and one display for controlling the nodes and the large display. The idea is that the instructor can control the system on a small screen and show the large display on a projector or television for the students to see. We anticipate delivery of a functional UI by the end of October.

Lastly, to assist educators in using this technology, we will prepare a set of lesson plans and activities which demonstrate the workings of wireless networking. This deliverable serves as a vehicle for demonstrating our project, so we set the delivery date to be sometime during the last week before our final presentation of the project.

2. Specifications and Analysis

2.1 PROPOSED APPROACH

Recalling the Summary of Requirements, there are some major objectives that this project needs to meet. The most limiting requirement is the budget. Due to wanting to keep each node at \$65, limits the hardware. This means that our approach to what operating system, networking protocols, etc. we use changes due to the limited processing power of the device.

A single board computer was deemed the correct choice due to the low cost ~\$35 per board, but also the higher processing power and IO abilities over embedded boards, along with the high mobility. To increase this mobility, a battery pack would be included to allow the nodes to work without wall power. The next step was deciding how the networking would be handled. Certain distributions of Linux aimed at computer networking come with support for mesh networking protocols, so the decision has to be made about which one to use. These distributions should allow for automatic connecting and disconnecting of links to aid with ease of use. A web server would be set up on one of the links to allow for a cross-platform GUI that would show all of the required information coming from the network, along with easy configuration of the network and nodes.

2.2 DESIGN ANALYSIS

Deciding on the hardware of the project seemed to be relatively easy. Due to the widespread support and price of the board, Raspberry Pi's will be used as the nodes. This also makes figuring out which mesh-networking protocol and linux distribution easy. After research, it was found that IPFire and OpenWRT were the best distributions to use, as there were already examples of these distributions working on the Raspberry Pi. Selecting the ad-hoc networking protocol was the next step.

There are 3 main types of protocols: proactive, reactive, and hybrid. Proactive protocols create a list of connections and their routes at the expense of latency when adding and subtracting nodes, while Reactive floods the network with packets at the expense of latency when sending packets. Hybrid routing creates a table initially and floods only when it cannot find a path to the destination. Due to how much bandwidth flooding a network stream with video packets would require, the decision was made to go with a proactive protocol.

There are 5 main proactive routing protocols: OSLR, Babel, DSDV, DREAM, and BATMAN. BATMAN and OSLR are the two most prominent protocols and are both supported by OpenWRT and IPFIRE. Further testing is required to decide on which distribution and protocol would be best for our project.

For the GUI, It was decided that Angular/Typescript would be used for the UI due to the ease of use and since the members working on the UI have had previous experience in Angular development.

2.3 DEVELOPMENT PROCESS

Because of the lack of certainty in the design of this project, our team requires an iterative development method that allows for design decisions to take place at any point in the project

lifecycle. And given the vagueness of our initial project description, we need an approach that will assist us in reaching our final design starting with a high-level understanding of the project. Therefore, we will conform to an Agile development process with a top-down design approach because it mitigates the risks associated with the inherent variability of this project.

The development lifecycle of our project will consist of a number of sprints, each lasting two weeks. Prior to each sprint, the team will hold a sprint planning meeting in which we choose tasks/features to implement during the coming sprint. Following each sprint, the team will meet for a sprint retrospective meeting, where we will discuss our progress and think critically about our performance.

We will be using the tools provided by GitLab for our artifacts. Individual units of work (i.e., tasks, features, or user stories) will be represented as Issues in GitLab, which can be created by and assigned to any team member. Each Issue is associated with a single Git branch, so Issues are loosely coupled, allowing team members to work independently.

To control the quality of our implementations, all code will go through peer reviews before going into production. These code reviews will be facilitated via merge requests in GitLab. Once a team member finishes work on an Issue, he will open a merge request and assign at least one other team member to review the work.

To track the progress of the team, Issues will appear on sprint boards. We will use different boards to denote the status of every Issue. The board in which an Issue resides will indicate whether it is *Open*, *InProgress*, *InReview*, or *Closed*.

2.4 CONCEPTUAL SKETCH

Our conceptual sketch of the project is shown in Figure 1.

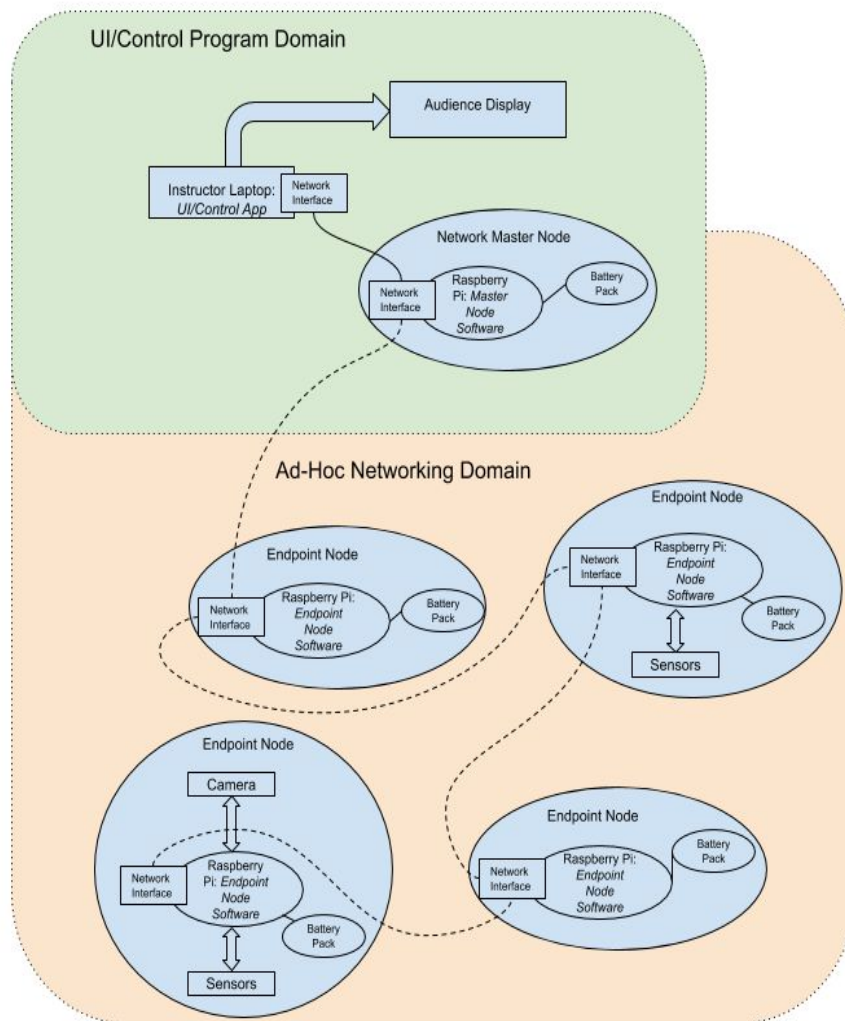


Figure 1: Conceptual Sketch

Our project consists of two major domains: a UI/Control Program domain and an Ad-Hoc Networking domain. In the UI/Control Program domain, the instructor's laptop runs the UI/Control frontend software, which allows users (students and the instructor) to initiate data transfers through the network and observe the routing information (nodes reached, timing information, etc.). The routing information is shown on the Audience Display, which may be the laptop screen or an external display (such as a projector). The instructor's laptop is connected via a dedicated network interface (ethernet or a USB WiFi adapter) to the network interface of the network master node. This allows it access to the ad-hoc network of nodes, which are not internet-connected. This connection will be open for the duration of the system's operation. The master node will have a special software load that allows it to participate in the ad-hoc networking

of the other nodes while providing data/controls to the instructor's laptop via the UI/Control backend.

In the Ad-Hoc Networking domain, each endpoint node will consist of a Raspberry Pi board with a wireless network interface that supports ad-hoc networking. Each board will run a Linux distribution specialized for routing, such as OpenWrt. The connections from node to node will be managed by an ad-hoc networking protocol such as B.A.T.M.A.N or OLSR. On top of the Linux operating system, there will be other drivers and software to capture data from cameras and sensors connected to some of the nodes (via USB). This data will be transmitted back to the master node and to the instructor's laptop at the behest of the UI/Control Program, which sends control signals to the nodes in the network to initiate transfers. All nodes (endpoint and network master) will be powered via rechargeable battery packs and stored in portable, durable, drop-proof casings. Each set of nodes distributed to instructors will be programmed with a unique identifier so that all node sets form separate ad-hoc networks, even when used within range of one another.

Figure 2 (next page) shows the system modules involved in our project for each physical computing device involved. This includes the endpoint nodes, the network master node, and the instructor laptop.

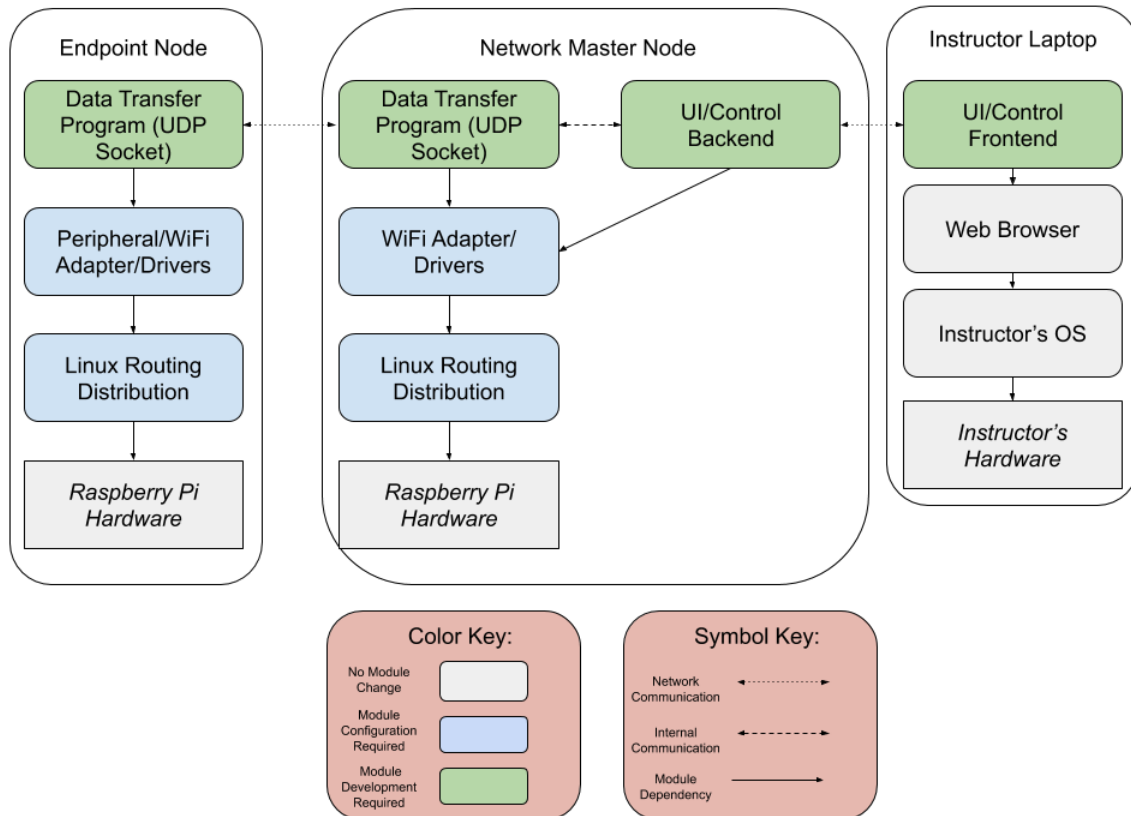


Figure 2: System Modules

The sharp-cornered blocks represent the base hardware layers of each device, while rounded-cornered blocks represent primarily software modules. As shown by the color key, the gray modules already exist and require no special configuration. The blue modules exist but will need to be configured/installed as part of the project. The green modules do not exist and will need to be developed as part of the project.

The arrows indicate the dependencies between layers. For endpoint/network master nodes, the data transfer program will rely on device drivers for the network interfaces and sensor/camera peripherals. The drivers will be installed on top of a Linux routing distribution, which will be installed on the Raspberry Pi hardware. Network communication will be initiated in the data transfer program, although the actual data transfer and routing is handled by the routing distribution and network interfaces. There will be a special connection between the data transfer program and the UI/Control program backend on the network master node; this will allow the UI/Control program to send and receive network data. The data transfer program and UI/Control program backend will run concurrently on the network master node. Since the frontend runs in a browser, no special configuration is required on the instructor's laptop. Once the master node is connected and started, the instructor will simply have to launch a web browser and load the app's webpage.

3. Statement of Work

3.1 PREVIOUS WORK AND LITERATURE

Several alternatives already exist for ad-hoc networking instruction. These primarily consist of computer programs designed to simulate various network types. The ns-3 project [1], supported by the University of Washington NS-3 Consortium, provides a simulated networking environment. Events can be scheduled in ns-3 to cause signals to be transmitted from various nodes at various times. The ns-3 simulator is also capable of interfacing with both virtual and real devices. The mesh package in ns-3 can be used to provide a MAC-layer ad-hoc routing capability for network simulations. Ns-3 is an appropriate tool for graduate and possibly undergraduate study of networking, but it does not effectively meet the needs of a grade-school user base. Ns-3 is primarily configured via the command line, which would be difficult and frustrating for an instructor or grade-school student to use. Students may also have trouble visualizing a network of simulated, virtual nodes; using all physical nodes would provide students with a tangible networking experience that feels more relevant to them.

While our project consists of fully-physical nodes, it shares some points in common with previous academic experiments both physical and virtual. E. Biagioni [2] from the University of Hawai'i at Mānoa created an ad-hoc wireless network using four Raspberry Pi Zero W computers and a laptop. The network was achieved using WiFi and not Bluetooth, similar to our planned approach. We are wary of Bluetooth, as it may be difficult to obtain the bandwidth needed to stream video through the network. The nodes were placed in rooms around a university building, and a utility similar to traceroute was used to test the network strength. However, the goal in this instance was to evaluate a specific ad-hoc networking protocol, AllNet. In order to keep students' attention and provide a

demonstration that feels relevant to students, we need to send live data through the network, such as a video feed.

Another experiment by Sharma and Nekovee [3] is similar to our proposed approach. Unlike our approach, they used virtual nodes. However, their approach does involve sending video over the network. It also centers on the automatic configuration of new nodes as they are added into the network (i.e. assigning IP addresses). This is critical, as we want to avoid having instructors enter arcane commands into a terminal to set up the nodes properly. It also contains a controller with a GUI application, which is very similar to our UI/Control program concept. In completing our project, we hope to combine the aspects of these two prior projects that will benefit students the most and lead to the simplest kit for instructors to set up.

3.2 TECHNOLOGY CONSIDERATIONS

Raspberry Pis are great for this application due to the amount of support they have. If we were to go with a different board we may not be able to find a suitable linux distro that supports mesh networking. They are also cheap enough for our budget and have a wide array of accessories designed specifically for the device. The trade off with this is that because the boards are so cheap, they have low power cpus and low amounts of ram, along with cheaper wifi modules. The range of these modules is not very high and we may have to buy better modules, but, since the boards are so cheap, it will not make the final design infeasible.

For the software, there were a variety of routes we could go with. There was a possibility of using bluetooth mesh networks to design the system, however we found that bluetooth LE might make that infeasible for the timeframe and budget given. Our best option would be to use either BATMAN or OLSR for our mesh networking protocol over some linux distribution, and were able to narrow it down to OpenWRT, due to its inherent Raspberry Pi support, and small size. Both BATMAN and OLSR possess the same performance traits, so it will just be down to picking one and going with it for the rest of the project.

3.3 TASK DECOMPOSITION

- Network Nodes
 - Support Ad-Hoc Networking
 - Configure OpenWrt
 - Configure BATMAN
 - Battery Circuit
 - Rechargeable Battery Installation
 - Battery Level Reporting
 - Endpoint Node
 - Socket Program
 - Case
 - Sensor Nodes
 - Stream Camera Feed
 - Case
- Network Master Node
 - Host UI Component
 - Host Backend Component

- Socket Program
- Connect/Disconnect Nodes
- Case
- User Interface
 - Audience Display
 - Node Selection Page
 - List of Active Nodes in Network
 - Topological Network Graph (Wishlist Item)
 - Node Detail Page
 - Node Information Display
 - Node Information Configuration
 - Data Stream Visualizations
 - Sensory Data (for Sensor Nodes)
 - Network Statistics
 - Battery Level
 - Data Model
 - Nodes
 - Stream
 - Network Layer (i.e., Services for communicating with backend)
 - Graphic Design
 - Artwork
 - Application Icon
 - Endpoint Node Icon
 - Sensor Node Icons
 - Camera Node Icon
 - Icons for any other sensor nodes we make
- Backend
 - Node Microservice
 - Stream Microservice
- User Documentation
 - User Manual
 - Lesson Plans

3.4 POSSIBLE RISKS AND RISK MANAGEMENT

The main risk involved in this project is the college moving to fully online. While our project can be completely done outside of the labs it will be difficult to do group work such as bi-weekly client meetings. We can use Zoom for the meetings, however if someone wants to test some things on the hardware we have to pass the hardware along rather than get to meet in a team. To ensure that everyone is on track with what they need to do, we will have weekly 'touch-base' meetings where we check in each member of the team through Zoom.

Another risk would be that the hardware that we use is incompatible with the software/unable to perform the design requirements. In this case, we have found multiple possible technologies to use and different hardware components that are advertised to work, so we would just try each one until one of them works as needed.

3.5 PROJECT PROPOSED MILESTONES AND EVALUATION CRITERIA

Our first milestone, the completion of one or more endpoint nodes, will be evaluated by our ability to send and receive data in a network of endpoint nodes. The endpoint nodes should also be able to report network statistics. This does not include any other components of the system, and the networking is not required to be ad-hoc.

The second milestone will be marked by the completion of a functional network master node, a functional backend, and a partial implementation of the user interface. With the network master node, we should be able to connect new endpoint nodes to the network. The network master node should detect all the endpoint nodes in the network, and the user interface should be able to query that information and show all active nodes on a display.

Our third milestone will be the completion of a functional camera node and video streaming functionality. We expect to be able to connect a camera node to the network via the network master node and then view live camera feed in the user interface.

Our fourth milestone will include the ability to connect multiple sensor nodes to our network. This will include updates to the user interface allowing the user to select from a list of nodes in the network and view the corresponding sensory data, node information, and reported network statistics in a detail page.

Our fifth milestone will be the completion of the battery circuits. Each network node should be connected to a rechargeable battery and should also be capable of reporting its remaining battery life to the network master node. This change should be accompanied by changes to the user interface which allow the user to see the battery level of each node in the network.

Lastly, the sixth milestone will be the completion of our project, accompanied by a set of lesson plans and a user manual. Additionally, we anticipate the creation of one or more new types of sensor nodes that can be connected to the network.

3.6 PROJECT TRACKING PROCEDURES

To track our progress throughout the lifetime of the project, we will use the project management tools offered in GitLab. We have set up Issue boards to denote certain Issues as Open, Blocked, InProgress, InReview, or Closed. These boards will be used to track the progress of every iteration (represented with a Milestone in GitLab) of the project. For every Milestone, GitLab offers a burndown chart showing the amount of work completed over time. These burndown charts will also show progress for the entirety of the project as we complete our deliverables.

3.7 EXPECTED RESULTS AND VALIDATION

The determination of our desired outcome comes down to these three main goals: functional lesson plans, an easy-to-use user interface, and lack of interference from other networks. The project should include lesson plans that will teach students how mesh networking works, and those should all work seamlessly with the provided equipment. The user interface needs to be designed in a way that an instructor or a student can operate it with minimal knowledge on how it all works. Finally, in order for the first two goals to work properly, the system should not encounter any outside interference from other types of networks.

We have a number of test cases that we are going to use to confirm the results work at our expectations. We intend on testing the system in multiple environments, such as inside one of the university buildings where there would be other networks that could potentially cause interference, or outside where we could test maximum range of the product. We also plan on having a test user from outside of the project test the user interface, making sure that an outside individual can operate the lesson plans, user interface, and system with ease.

4. Project Timeline, Estimated Resources, and Challenges

4.1 PROJECT TIMELINE

Figures 3, 4, 5, and 6 below show the entire Gantt chart for our two-semester senior design project. The figure captions state the time period each image represents. Tasks with subtasks that are not present in a given time period have been collapsed. Figure 3 shows the exploratory steps we are performing prior to the receipt of our official hardware for the project.

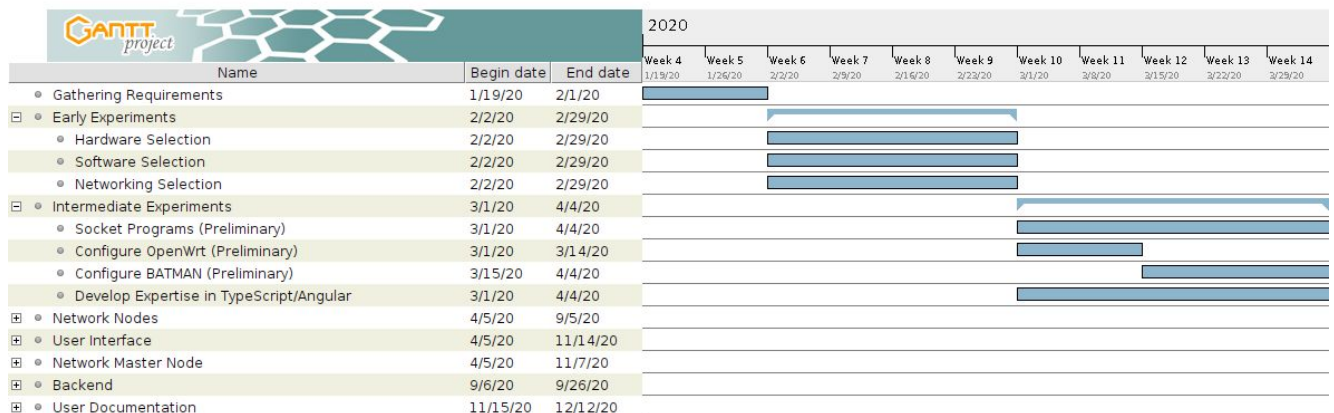


Figure 3: Gantt Chart CprE 491 (01/19 - 04/04)

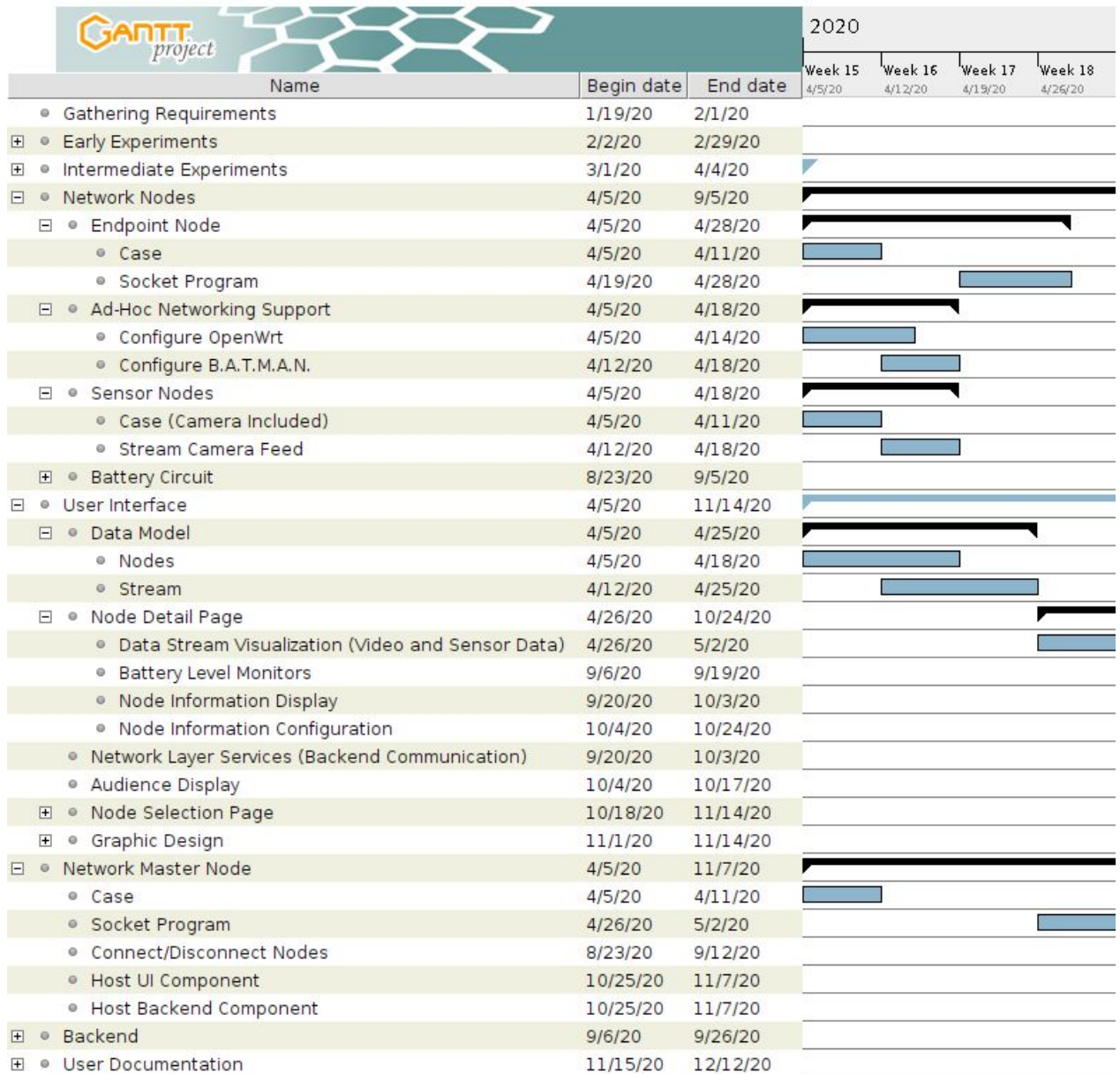


Figure 4: Gantt Chart CprE 491 (04/05 - 05/02)

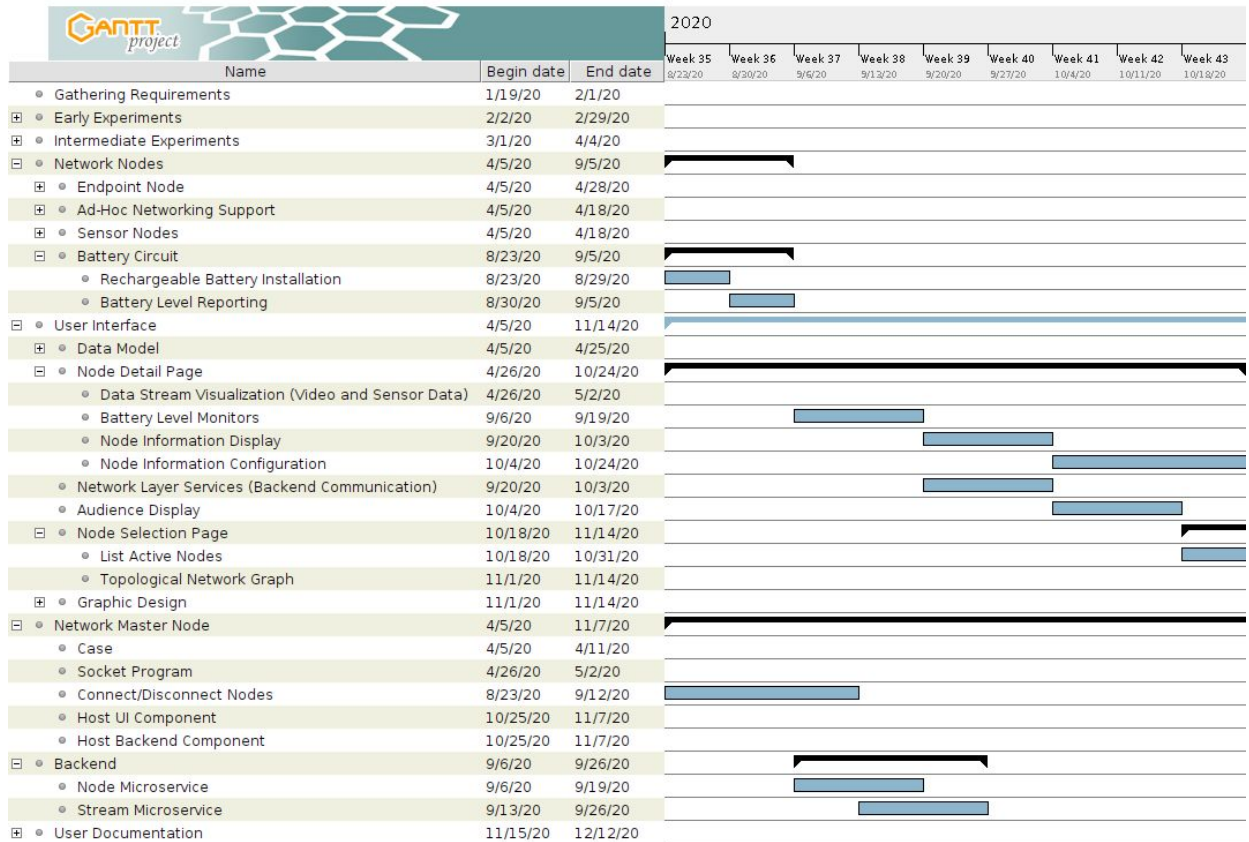


Figure 5: Gantt Chart CprE 492 (08/23 - 10/24)

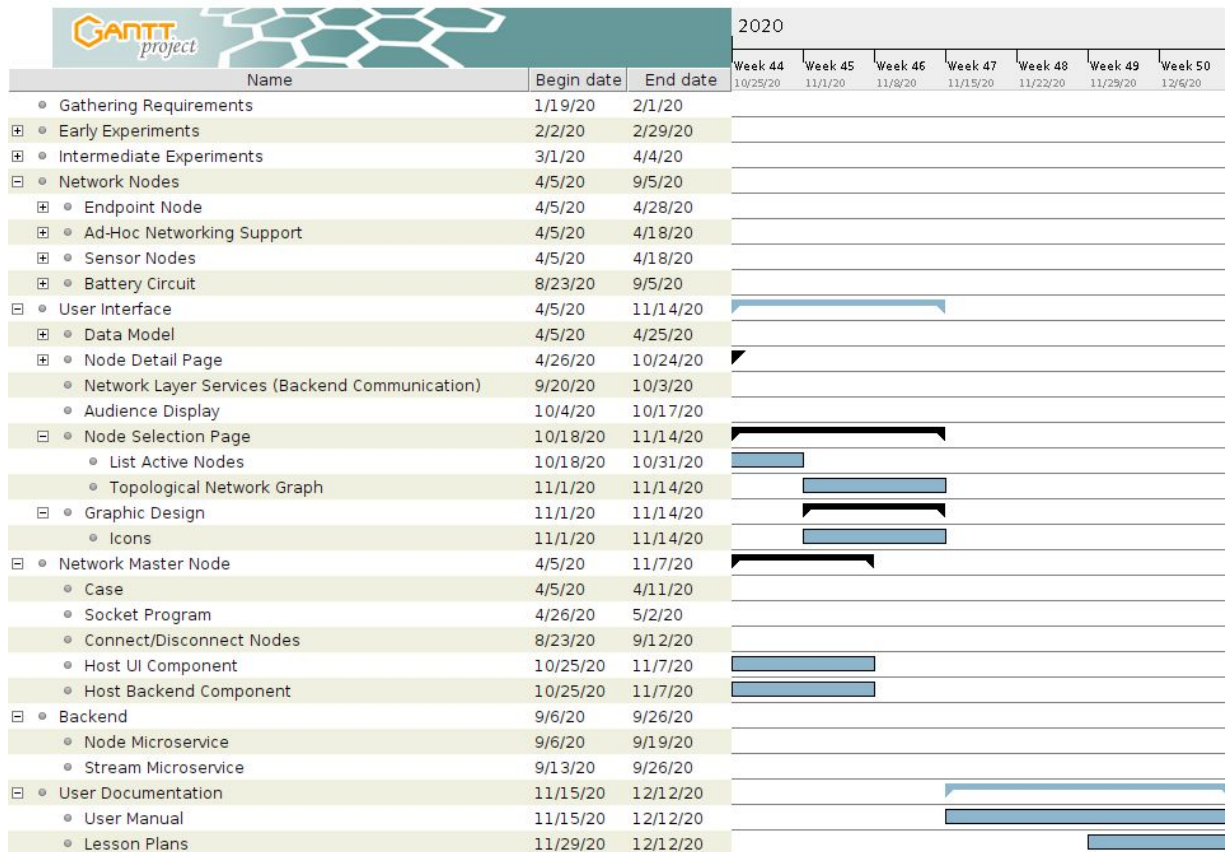


Figure 6: Gantt Chart CprE 492 (10/25 - 12/12)

Our Gantt chart reflects several primary goals. Most importantly, we are pushing to complete the initial ad-hoc networking prototype by the end of CprE 491 (see Figure 4). This requires the proper configuration of OpenWrt and B.A.T.M.A.N., as well as the streaming and forwarding of data in the network. At the same time, we need to develop the model for the node and video stream data within the UI/Control program. Prioritizing this goal will help us take our biggest risks first, when there is less pressure. One of these risks is the streaming from the camera module in OpenWrt. If we hit a roadblock due to an incompatibility, it will be easier to rebuild our schedule before CprE 492 starts. Once we get the network prototype working, we anticipate that later tasks will be more straightforward.

We are also being cautious with the development of our user interface and backend (see Figure 5). These tasks will require a lot of coordination, especially when different team members are working on each part. This will also require some work on the network node software, i.e. to obtain the data requested or set the desired node configuration. We have scheduled most of the backend development before the user interface development to ensure that the interfaces between the UI and network are well defined before UI development begins. We may produce some

documentation internal to our team to capture the details of these interfaces and guide our UI engineers during development.

The end of our timeline (Figure 6) consists of producing two items of documentation: lesson plans and a user manual. Putting these tasks last helps us avoid the possibility of a last-minute technical flaw upending the entire project. These tasks also serve as an assessment of our project goals. By writing our user manual, we will be able to assess whether our project is easy enough for an instructor to use. By writing the lesson plans, we will be able to assess whether our project has real teaching value. It will be helpful to keep these goals in mind throughout CprE 491 and CprE 492.

4.2 FEASIBILITY ASSESSMENT

Since the goal of our project is to showcase mesh networking, it makes sense that the technology already exists. The ability to set up mesh networks on the Raspberry Pi 4 is one of the reasons we chose the Raspberry Pi 4 to base our nodes on. The Raspberry Pi 4 hardware is more than powerful enough for networking and streaming video and will be able to fulfill what is required of it.

The power consumption of the Raspberry Pi 4, while a little higher than older models, is still relatively low. Reaching the minimum of two hour runtime should be no problem with a rechargeable battery pack.

4.3 PERSONNEL EFFORT REQUIREMENTS

Table 1 shows our estimates of hours required to complete the major tasks of the project.

Task	ETC (hrs)	Task	ETC (hrs)	Task	ETC (hrs)
Configure OpenWRT	10	Battery Level Reporting	18	Assemble Battery Pack	2
Configure BATMAN	20	Program to Stream Video	6	Assemble Nodes	2
Build UI for Master Node	40	Build Backend Component for Master Node	8	Build microservice for streaming video	8
Build microservice for mesh	8	Create lesson plans	8	Create user manual	8

Table 1: Task Estimated Time to Completion Table

4.4 OTHER RESOURCE REQUIREMENTS

This project will only need materials and parts to build each node. Each node consists of a Raspberry Pi 4, an SD card, a rechargeable battery pack (including the charger), and a case. In addition to that, the sensor node will also have the Raspberry Pi Camera Module attached. To

accommodate the camera, the sensor node will reside in a case that includes a spot to hold the camera.

4.5 FINANCIAL REQUIREMENTS

The budget for the project has been estimated at around \$500.00. We plan on developing a system of eight nodes, at a price point of \$60.00-\$65.00 per node. This would put the expected cost of the system at \$480.00-\$520.00, meeting our budget. The cost of each node will vary depending on what type of accessories, if any, that they will have. Some may include a camera module or other sensor, which could make the cost of the node higher.

5. Testing and Implementation

Testing is an extremely important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)
2. Define the individual items to be tested
3. Define, design, and develop the actual test cases
4. Determine the anticipated test results for each test case
5. Perform the actual tests
6. Evaluate the actual test results
7. Make the necessary changes to the product being tested
8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you've determined.

5.1 INTERFACE SPECIFICATIONS

- Discuss any hardware/software interfacing that you are working on for testing your project

5.2 HARDWARE AND SOFTWARE

- Indicate any hardware and/or software used in the testing phase
- Provide brief, simple introductions for each to explain the usefulness of each

5.3 FUNCTIONAL TESTING

Examples include unit, integration, system, acceptance testing

5.4 NON-FUNCTIONAL TESTING

Testing for performance, security, usability, compatibility

5.5 PROCESS

- Explain how each method indicated in Section 2 was tested
- Flow diagram of the process if applicable (should be for most projects)

5.6 RESULTS

- List and explain any and all results obtained so far during the testing phase
 - Include failures and successes
 - Explain what you learned and how you are planning to change it as you progress with your project
 - If you are including figures, please include captions and cite it in the text
- This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place
- Modeling and Simulation: This could be logic analyzation, waveform outputs, block testing. 3D model renders, modeling graphs.
- List the implementation Issues and Challenges.

6. Closing Material

6.1 CONCLUSION

Summarize the work you have done so far. Briefly reiterate your goals. Then, reiterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 REFERENCES

- [1] NS-3 | *A Discrete-Event Network Simulator for Internet Systems*, 2020. Accessed on: Mar. 29, 2020. [Online]. Available: <https://www.nsnam.org/>
- [2] E. Biagioni, "A Network Testbed for Ad-Hoc Communications using Raspberry Pi and 802.11," in *Proc. of the 52nd Hawaii International Conference on System Sciences*. Accessed on: Mar. 29, 2020. [Online]. Available: <http://hdl.handle.net/10125/60191>
- [3] S. Sharma and M. Nekovee, "Demo Abstract: A demonstration of automatic configuration of OpenFlow in wireless ad hoc networks," *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. Accessed on: Mar. 29, 2020. [Online]. doi: [10.1109/INFOCOMW.2019.8845307](https://doi.org/10.1109/INFOCOMW.2019.8845307)

6.3 APPENDICES

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.