Wireless IntraNetwork

ECE 445 Design Document

Daniel Gardner and Jeeth Suresh
Group 1
TA: James Norton
2/29/16

1 Introduction

1.1 Objective

The internet is such an important part of modern life that the UN has defined it as a basic human right. Unfortunately, 60% of the human race (over four billion people) still lack any kind of connection to this invaluable service. There is a giant gap in internet availability between the developed and developing world today. In regions such as Sub-Saharan Africa, less than 0.5% of people [1] have a fixed internet broadband subscription. This leads to social, educational, and economic isolation. LTE and HSPA coverage, the two leading standards of mobile connectivity, have a combined coverage of just 25% of the region [2]. In comparison, in the US a phone will spend over 85% of its time covered by an LTE signal [3] and 70% of adults have an active broadband subscription [4]. A contributing factor to this massive difference in connectivity may be that sub-Saharan Africa is significantly less urbanized than the US - 64.2% of residents in the United States live in an urban environment versus 10.6% in Zimbabwe and 2.8% in Uganda [5]. Studies show that a 10% increase in internet access in a developing country brings a 1.7% increase in exports and 1.1% increase in imports [6]. Without reliable access to the wealth of educational, medical, and economic resources provided by the internet, WiFi enabled devices donated to villages in the developing world by organizations such as OLPC (One Laptop Per Child) will never achieve their full potential.

Our goal is to bring the internet to the developing world using an entirely new form of infrastructure. Instead of installing miles of fiber optic cables, our 'infrastructure' is almost entirely wireless. We will use a large number of solar-powered nodes to build a mesh network capable of transmitting data throughout a community, creating an intranet. These nodes will use WiFi so any device (whether it's been donated or purchased by a villager) can connect to the intranet and will adapt to the failures of other nodes using a reactive routing protocol.

1.2 Background

Efforts by Facebook, SpaceX, and Google attempt to solve the internet problem through low-flying satellites or drones, which suffer the same throughput bottlenecks as traditional satellite communication [7]. A mesh network that is easily-installed and expanded, while existing independently of expensive global connections, is the solution to connect remote areas of the world. Other attempts have been made to connect the world with remote, portable mesh networking [8]. Unfortunately, these have failed in the long term due to costs of over $200/node.

Our nodes must be as affordable as possible, to ensure that they can reasonably be purchased with the disposable income a rural villager might earn. A subsistence farmer in Nepal, for example, has a disposable income of about $5 per year [9]. We also plan to partner with NGOs to provide a subsidized distribution program so the node is cheaper (or even free) for the customer.

1.3 High-Level Requirements

- Nodes must be able to connect to each other automatically, allowing data stored on any node in the network to be available from any access point.
- Nodes must be able to operate indefinitely on solar power.
- Nodes must be as low-cost as possible, ideally under $20.

2 Design

Nodes require three sections for successful operation: a power supply, a control unit, and a WiFi module as shown in Fig. 1. The power supply ensures that the system can be powered continuously all day and night with the proper 3.3V. The control unit contains up to 16GB of storage for educational materials and other resources, as well as a microcontroller to handle this data. Lastly, a WiFi module connects this control unit to a standard IEEE 802.11b/g/n WiFi network. Each node will need to cover an area approximately 600-800ft in every direction with 4MB/s access. Nodes will also store 16GB of data for access by other users.

A central server is necessary for an optimized network, but its design is outside the scope of this class.

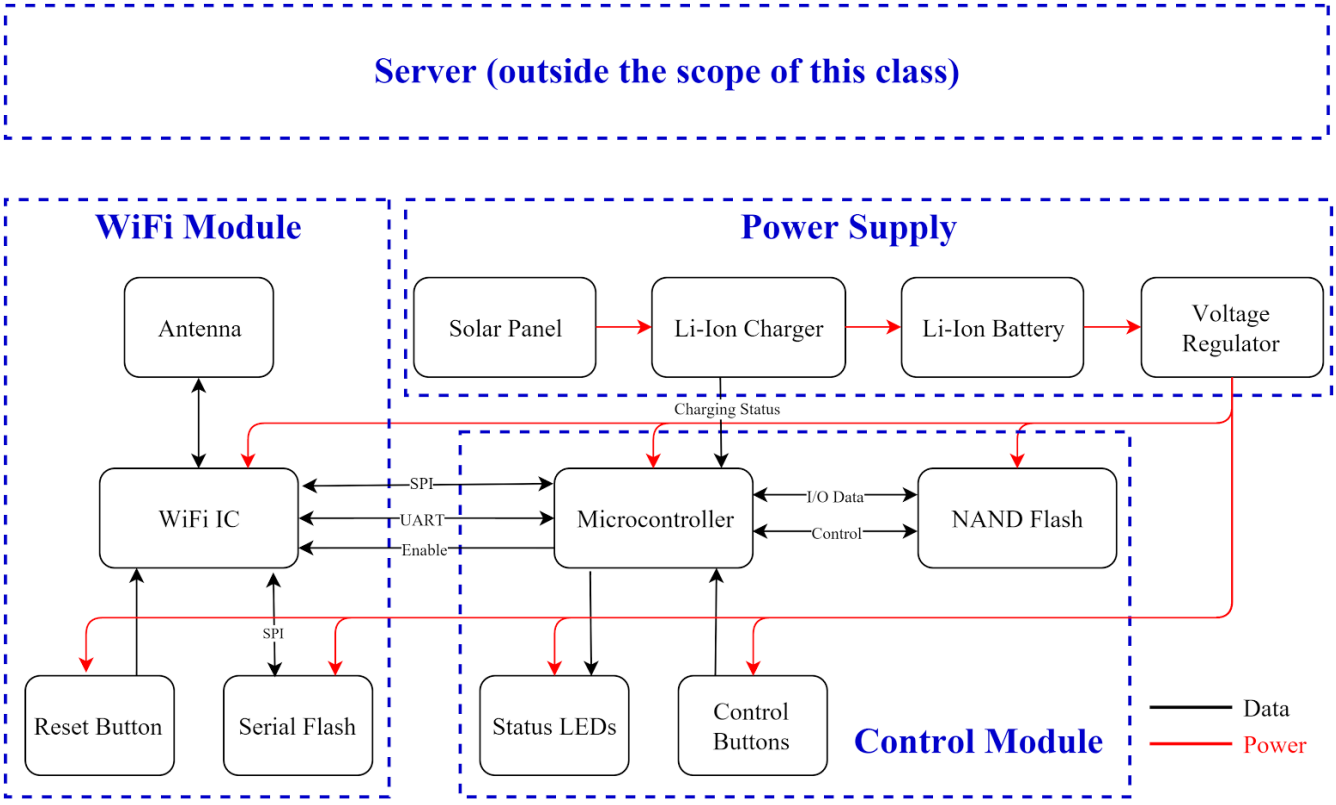

Fig. 1. Block Diagram

The physical design, shown in Fig. 2, will consist of a thin, waterproof case holding the battery pack and the main PCB. A ~15x15cm solar panel (depending on the environment) will sit on top, encased in epoxy. The node will be approx. 6cm thick.

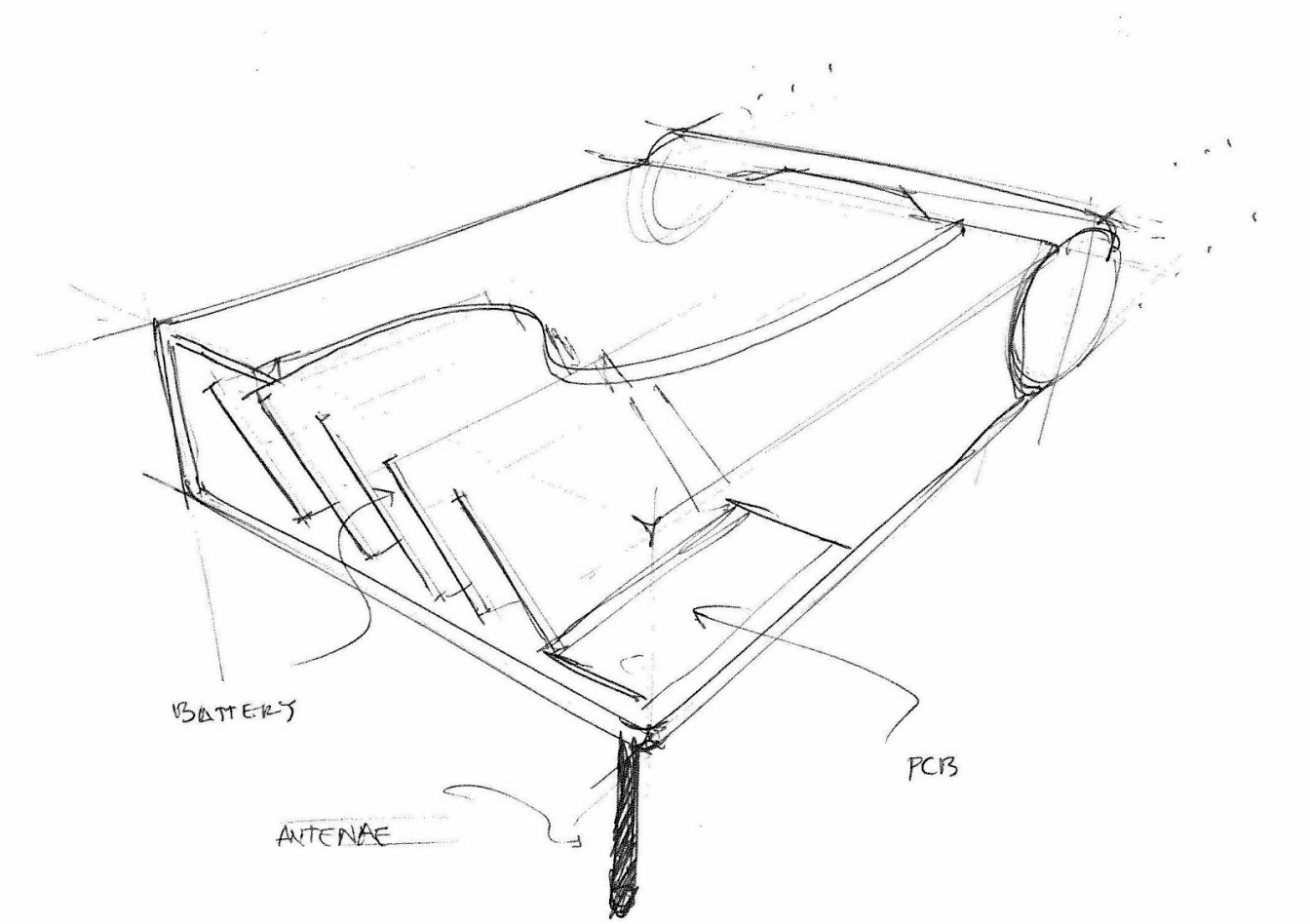

Fig. 2. Physical design sketch

2.1 Power Supply

The power supply, Fig. 5, provides the circuit with 3.3V at all times. It is powered by a solar panel, which stores excess power overnight with a li-ion battery to keep the node continuously powered. Our power budget is 150-200mA (average) at 3.3V, mostly consumed by the microcontroller (~50mA) and WiFi IC (~150mA).

2.1.1 Solar Panel

Nodes will be powered by a solar panel. These will provide the charging IC with enough voltage to charge the battery, and enough power to keep the node continuously powered off of 10-12 hours of sunlight each day- this requires an average output of at least 720mA throughout the day. This, we chose a 2.5W 5V solar cell for its low cost and ability to meet our power requirements.

Requirement	Verification
Outputs 300mA-1A between 4.40V-7.50V in no more than full sunlight (110,000 lux [10])	A. Place a solar panel in 110,000 lux. Confirm that the sunlight has no more than this intensity using a 1% tolerance photoresistor B. Measure the open-circuit voltage with a voltmeter, ensuring that it is below 7.50V C. Terminate the solar panel with a resistive load such that the voltage drop is 4.40V D. Ensure that the current through the load is above 300mA using an ammeter in series

2.1.2 Li-ion charger

The node will charge the li-ion battery through a charging IC, the AAT3693. It is powered by the solar panel. This chip was chosen for its affordability and thermal stability. This chip will charge the battery and be able to (thermally and electrically) charge the battery fully in only 10-12 hours. A li-ion charging cycle consists of constant-current to a specified voltage, followed by constant-voltage "float" charging to 4.2V. This second stage relies on the internal resistance of the battery to regulate current.

Requirements	Verification
1. Li-ion battery charges to 4.16-4.23V when a continuous 4.4-7.0V input voltage is applied 2. Charging at maximum current and voltage can be sustained below 125°C	1. A. Discharge a li-ion battery to 3.7V cell voltage. B. Charge the battery at the output of the AAT3693 from an input of 7V, without limiting current. C. At the termination of the charge cycle, signified when the "charge status" pin of the AAT3693 goes high, we will ensure that the battery is charged between 4.16-4.23V 2. A. Throughout the charging cycle outlined in verification 1.B-C, observe the temperature. Use an IR thermometer to ensure that the IC does not reach temperatures greater than 125°C.

2.1.3 Li-ion battery

The charging IC will power the battery, which will feed into the voltage regulator. The lithium-ion battery must be able to keep the circuit continuously powered, even at night with no power supplied by the solar panel for up to 14 hours. This demands a capacity of approximately 2.1AH. We will plan for a cushion of 2x for cloudy days, and thus require 4.2AH. There will be a temperature sensor (thermistor) physically next to the battery, which will

provide a temperature-correlated analog voltage for the AAT3693 charging IC to manage potential overheating issues. The thermistor, connected between the AAT3693 sense pin and the negative terminal of the battery, changes resistance in response to the temperature so that the cell temperature can be monitored as in the application circuit of the AAT3693 [11].

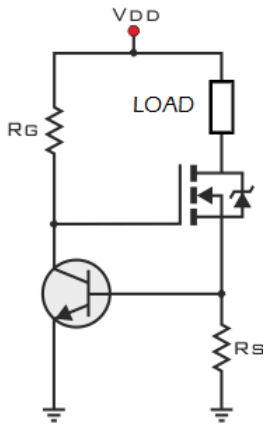

Fig. 3. Constant-current test circuit, $I_L=0.7/R_s$

Requirement	Verification
Stores >4.2AH of charge	A. Connect a fully-charged (4.2V) lithium-ion battery with the positive terminal at VDD and the negative at ground, as defined in figure 3. B. Discharge the battery at 300mA for 14 hours C. Use a voltmeter to ensure that the battery voltage remains above 3.7V

2.1.4 Voltage regulator

This low-dropout regulator supplies the required 3.3V to the node system from the 3.7-4.2V battery. This chip, the classic 1117 regulator, must be able to handle both the peak input from the battery (4.2V) and minimum input from the battery (3.7v) at the peak current draw (300mA).

Requirements	Verification
1. Provides 3.3V +/- 5% from a 3.7-4.2V source 2. Can operate at currents within 0-300mA 3. Maintains thermal stability below 125°C	1,2. A. Use the constant-current circuit in Fig. 3, connecting the output of the voltage regulator to “VDD” in the image, and draw 300mA B. Measure the output voltage using an oscilloscope, ensuring that the output voltage stays within 5% of 3.3V 3. A. During verifications A and B, use an IR thermometer to ensure the IC stays below 125°C

2.2 Control Module

The control module, Fig. 7, handles the node’s cache and hosts applications though SPI and UART communication with the WiFi module. It is powered by the power supply, and will disable the circuit if the battery voltage reaches critical levels. The microcontroller and flash memory together consume approximately 50mA.

2.2.1 Microcontroller

The microcontroller, a Freescale Kinetis KV30P64M100, handles memory allocation for the cache. It communicates with the WiFi chip via UART, and reads the NAND flash cache through SPI. This microcontroller was chosen for its affordability and SPI clock speeds of 12.5MHz, which forms the basis for the WiFi module data transfer so users can access data as quickly as possible. The chip also monitors battery charge status and will disable the WiFi module and itself if the battery voltage is low enough to risk damaging the battery. The microcontroller is programmed through a JTAG interface by a Freescale Freedom development board.

Requirements	Verification
1. Can both receive and transmit over SPI at speeds greater than 4.5Mbps 2. Can both receive and transmit over UART at a speed of at 115.2kbps	1. A. Connect microcontroller to USB SPI bridge, such as FT4222, and to a terminal such as Putty B. (Start timer) send a 0.45Mbit block of random data from the USB bridge into the Kinetis C. Echo data back, this time transmitting over SPI from the Kinetis D. (Stop timer) ensure that data received matches data sent, and that time elapsed does not exceed 100ms 2. A. Connect microcontroller to USB UART bridge, such as FT4222 or FT232, and to a terminal such as Putty B. Set up terminal at 115.2kbaud C. Send and echo back 100 characters D. Ensure that all characters match those sent

2.2.2 Parallel flash

The parallel NAND flash interfaces with the microcontroller to store the node's cache. It is an affordable way to store large amounts of user-accessible data. We will be using a 16Gb IC initially.

Requirements	Verification
Provides read and write speeds above 4.5Mbps	We will write a program to test this with the Kinetis application processor A. Write a 4.5Mbit block of random data to the flash B. (Start timer) Read the same block of data from the flash into the application processor C. Write the same block of data back to flash D. (Stop timer) Ensure that sent and received data match E. Ensure that the time elapsed between the initialization and full received data block is under two seconds.

2.2.3 Status LED

The status LEDs, powered through the microcontroller, will display to the operator whether the node has been able to connect to the network.

Requirement	Verification
Must be visible from 3 meters away with a drive current of 10mA	A. Adjust R_s in the constant current circuit in Fig. 3 to deliver 10mA to the load (supply 5V to VDD) B. Connect the LED in the 'LOAD" position in Fig. 3 C. Measure 3 meters distance from LED circuit D. Ensure that LED is clearly visible when pointed in the viewer's direction

2.2.4 Connect button

The two control buttons connect to the microcontroller to interface with the user to set up a network. They also help us debug while the node is in development.

Requirement	Verification
Must be easily-pressable	Press button and ensure that it can be done without strain

2.3 WiFi Module

The WiFi module, Fig. 6, sends and receives information from the microcontroller, through UART and SPI, to communicate with the wireless network. It contains a secondary microcontroller as well, which allows the node to keep up a mesh network through a dual-processor system, so the main Kinetis microcontroller is not needed for simply forwarding data. The WiFi module consumes approximately 100-150mA, depending on traffic.

A larger separation results in a lower throughput, shown in Fig. 4, which limits usefulness based on range. We will aim for a 10Mbps connection at 800ft.

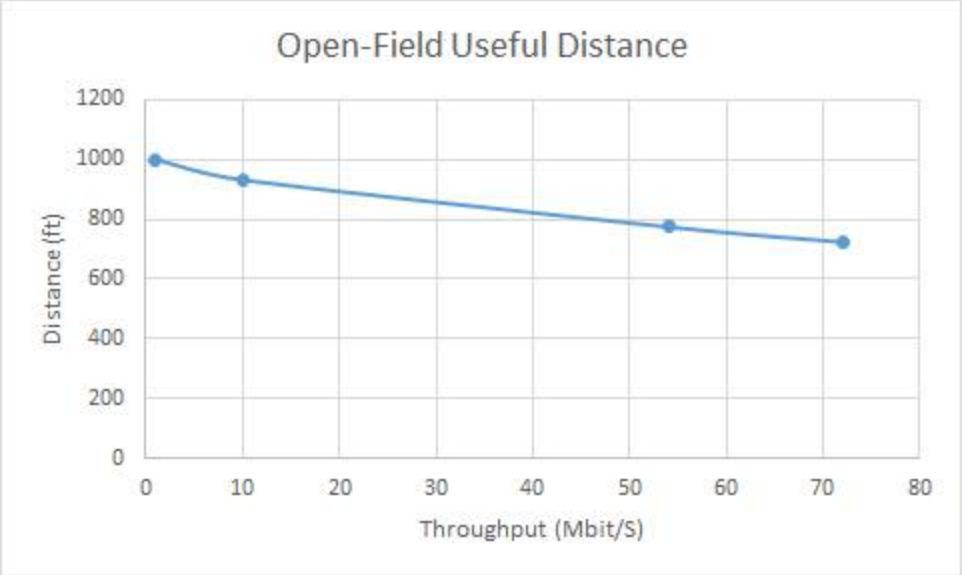

Fig. 4. Experimental Throughput vs. Open-Field Separation [12]

2.3.1 Antenna

The WiFi IC will communicate with the network via a 2.4GHz PCB trace (inverted F-type) antenna. It will be optimized to allow for maximum throughput and a maximum range. We will aim for 10Mbps access at 800ft. This is within the capabilities of the WiFi IC and is dependant on the match and design of the antenna. In the future, we will test the throughput and range of various antennas.

Requirements	Verification
1. Antenna must be matched at 50Ω +/- 5% between 2402-2484MHz 2. Provides >-91dBm received signal strength at 800ft with +19.5dBm input power 3. Antenna must be omnidirectional to within 6dB	1. A. Test the antenna match to the trace antenna with a network analyzer and a coaxial pigtail B. Ensure that the impedance is within required range at IEEE 802.11 frequencies 2. A. Antennas should both be oriented vertically. B. Read the RSSI with an identical WiFi board/antenna at 800ft open-field, given an input of +19.5dBm from the WiFi IC, ensuring that it is above -91dBm. 3. A. At 800ft open-field, rotate antenna through all six 90° orthogonal orientations and ensure that the RSSI does not vary by more than +/-6dB from the measurement in step 3.

2.3.2 WiFi IC

The WiFi IC communicates with the Kinetis microcontroller through UART to establish modes of operation and SPI to transfer data to and from the cache. It runs off program memory stored within serial flash, and is programmed via an ICSP header. It communicates with the network through the antenna. This chip includes a 32-bit microcontroller and WiFi transceiver, and can keep up the network without any correspondence with the main Kinetis microcontroller. We have chosen our WiFi IC, the ESP8266, with cost in mind. This chip includes a 32-bit microcontroller and WiFi transceiver. This was chosen since it costs 5-10 times less than competitors, but it sacrifices some performance and is a bottleneck in our design. It operates at 160MHz (overclock) and has a data input communication with the Kinetis microcontroller via UART and SPI.

Note: The ESP8266, was chosen due to significant cost differences, which allows us to aim for impoverished target markets where any other IC would not. For this reason, the majority of the circuit has been designed around this IC. Cost and the ability to uphold a WiFi network are the primary requirements for this IC. Using a different IC would add ~$10 to our cost (for example, if replaced with a common Broadcom chipset, WM-N-BM-xx), which affects the relative cost in different ways. One potential application is in areas with electricity; without solar panels or a battery, this would triple the approximate $5 main PCB cost. If the device is fully solar-powered, this would add 50% to the $20 target price point. This is unacceptable for any potential small performance upgrades. The ESP8266 offers speeds in excess of 3Mbps, plenty of speed for smooth web browsing.

Requirements	Verification
1. The WiFi IC must be able to communicate over IEEE 802.11b/g/n at 4.5Mbps with a 50Ω nominal RF connection. 2. It must be able to communicate over both SPI and UART.	1. A. Assemble WiFi IC on PCB as specified in the datasheet as the basic application schematic. B. Note default WiFi network on a mobile device 2. A. Connect to the ESP8266's UART port with an FT232 UART bridge, as per the FT232 reference diagram, and a computer. This can be done on a through-hole breadboard with an ESP-01 module B. Program 4.5Mbit HTML page (large photo) to SPI flash (program memory) C. Connect to default network with a mobile device, navigate to webpage from step 4 D. Time loading process, ensure that it is less than one second

2.3.3 Serial flash

The serial flash, connected to the WiFi IC through SPI, holds the program memory for the WiFi IC. It must operate at 80MHz for the WiFi microcontroller to operate at full speed. Although we do not know our current program size for this microcontroller, we will prototype a size of 1MB and downsize (for cost savings) if possible.

Requirements	Verification
1. Operates reliably at 80MHz 2. Flash must have >=1MB of storage	1. A. Operate the flash at 80MHz, and run a known program B. Confirm that the program works as expected 2. A. Fill 1MB of memory with verifiable, documented data B. Read data back and echo into UART terminal provided by an FT232 device, ensure that data matches completely

2.3.4 Reset button

The reset button connects the WiFi IC's reset pin to ground when pressed. It allows the user to reset that particular network connection without losing any applications running on the microcontroller- it only resets the WiFi IC.

Requirement	Verification
Must be easily-pressable	Press button and ensure that it can be done without strain

2.4 Case

The case, shown in Fig. 2, will protect all electronics from moisture, dust, and weather in its outdoor environment. To withstand harsh environments, the case must be able to pass IP66 environmental standards.

Requirements	Verification
Passes IP66 environmental codes	A. Fill inside of case with a few grams of a dyed anhydrous powder such as Jello or Kool-Aide B. Spray the case with water from a 12.5mm nozzle at the case from any direction from 3 meters, for 3 minutes C. Shake case thoroughly D. Ensure that all powder remains white and dry- any powder turning into the intended dye color has been hydrated

2.5 Server

The server design will consist of a Raspberry Pi, a cellular receiver, a WiFi dongle, and an external hard drive. We will not consider it in the scope of this class, though it is integral for our project to connect to the greater internet.

2.6 Schematics

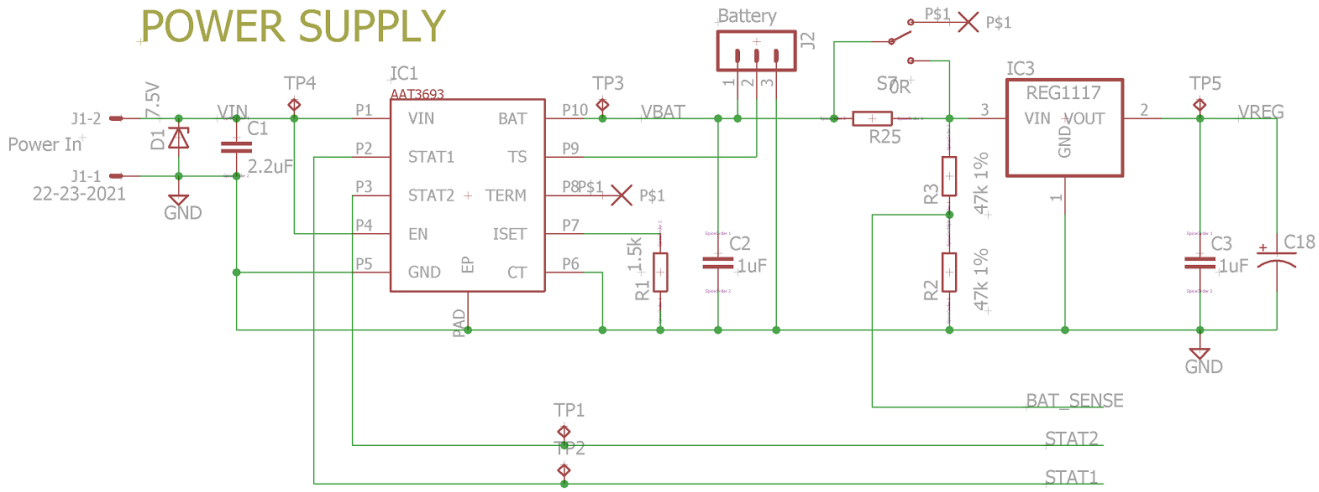

Fig. 5. Power supply Schematic

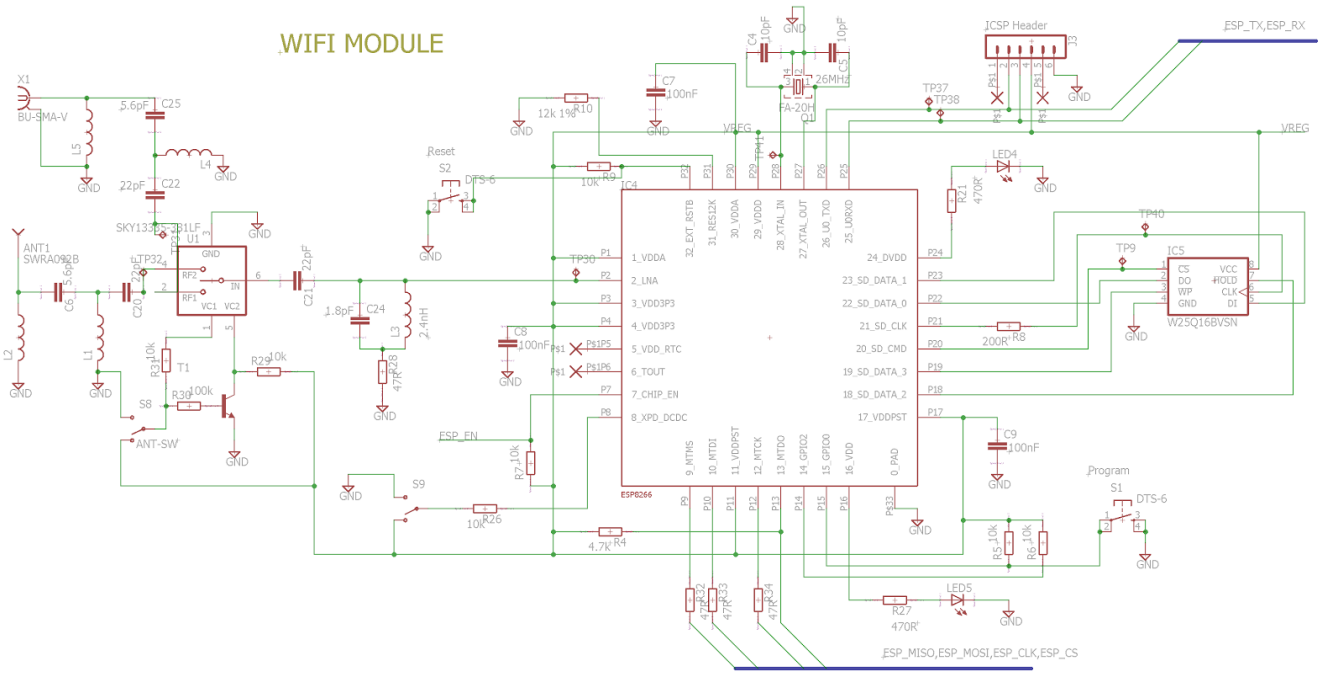

Fig. 6. WiFi module schematic [11]

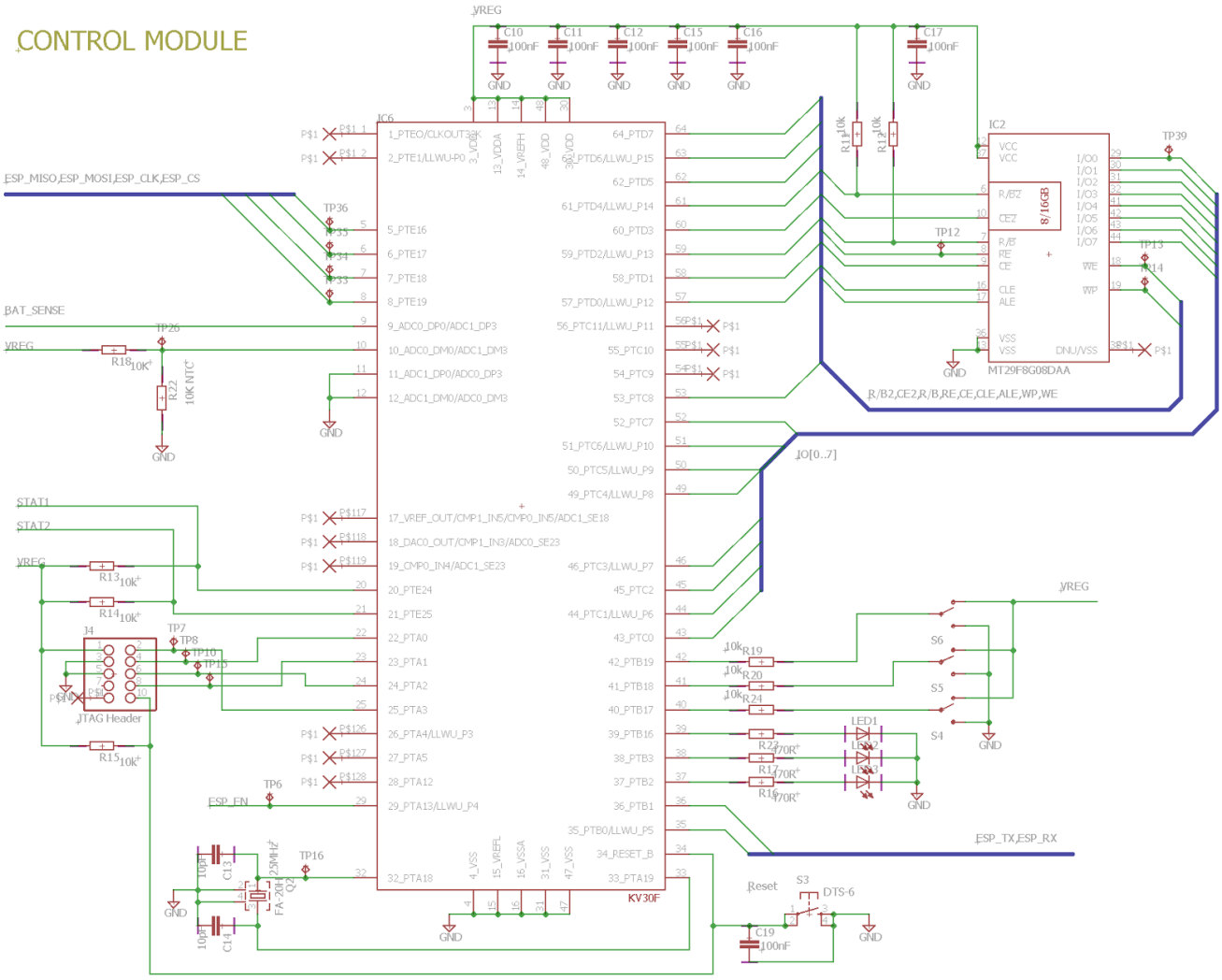

Fig. 7. Control module schematic

2.7 Board Layout

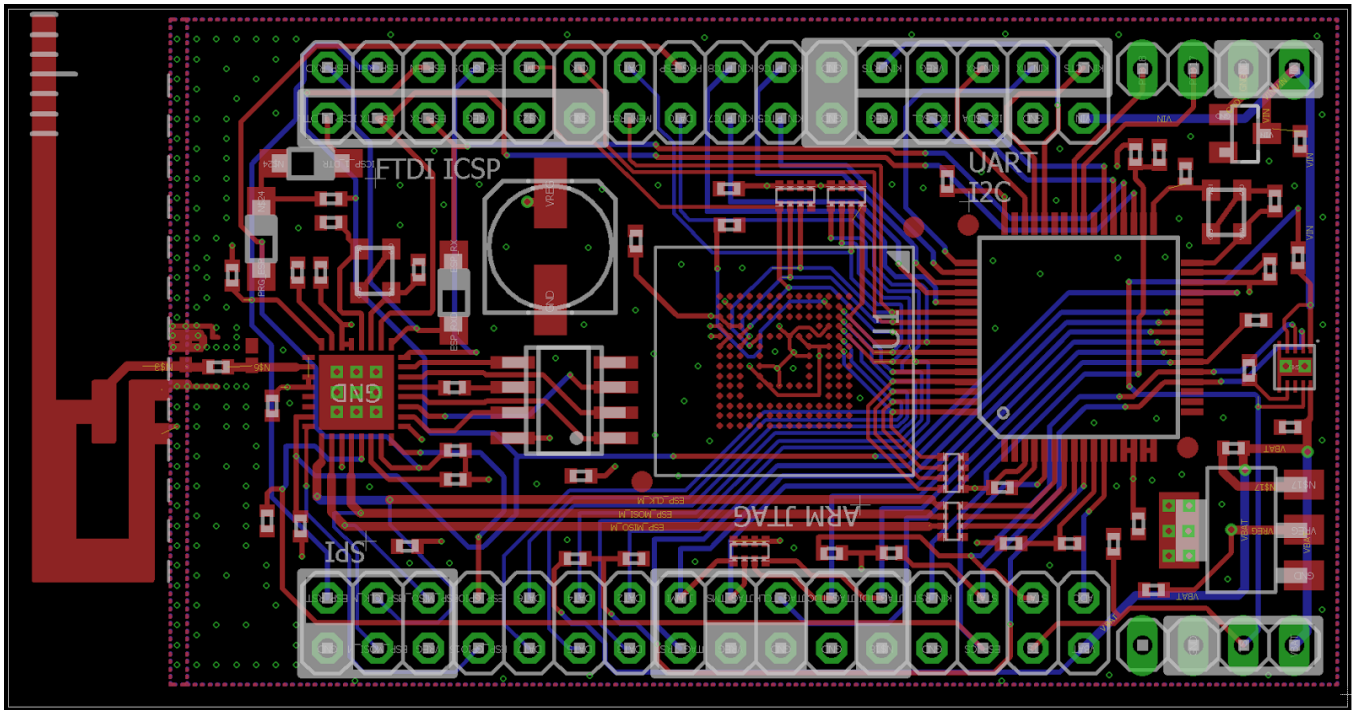

Fig. 8. Eagle PCB Layout

2.8 Software

The software routes information through the mesh network, and can be divided into two parts. The routing algorithm (Fig. 9), which will handle the pathing of packets through nodes, is a variation on the DSR reactive routing algorithm running in promiscuous mode. The data transmission algorithm (Fig. 10) is custom-built and based on UDP, which does not require a consistent connection between a source and a destination.

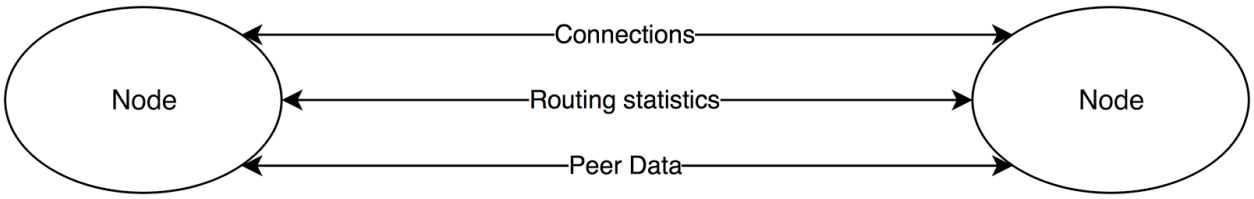

Fig. 9. Promiscuous Mode Diagram

2.8.1 Promiscuous Mode

The nodes will share data with their peers to ensure optimal packet distribution. Each node will share a peer list with all of its peers to reduce the number of situations where packets are flooded through areas of the network that do not contain the destination node. This peer list will contain a small amount of information about each node - whether or not it's hosting a service, the number of users it serves directly, it's battery level and health, and whether or not certain components have failed. This information will allow nodes to build a small outlook on their section of the network, and enable the network as a whole to more easily recover from a node failure by

taking on the paths of the failed node. Routing statistics are also shared between nodes, as this is the only tool the network has of evaluating itself. If a subnet of nodes is passing a lot of traffic back and forth for other nodes, it becomes critical that if one of them fails the others are able to pick up the slack.

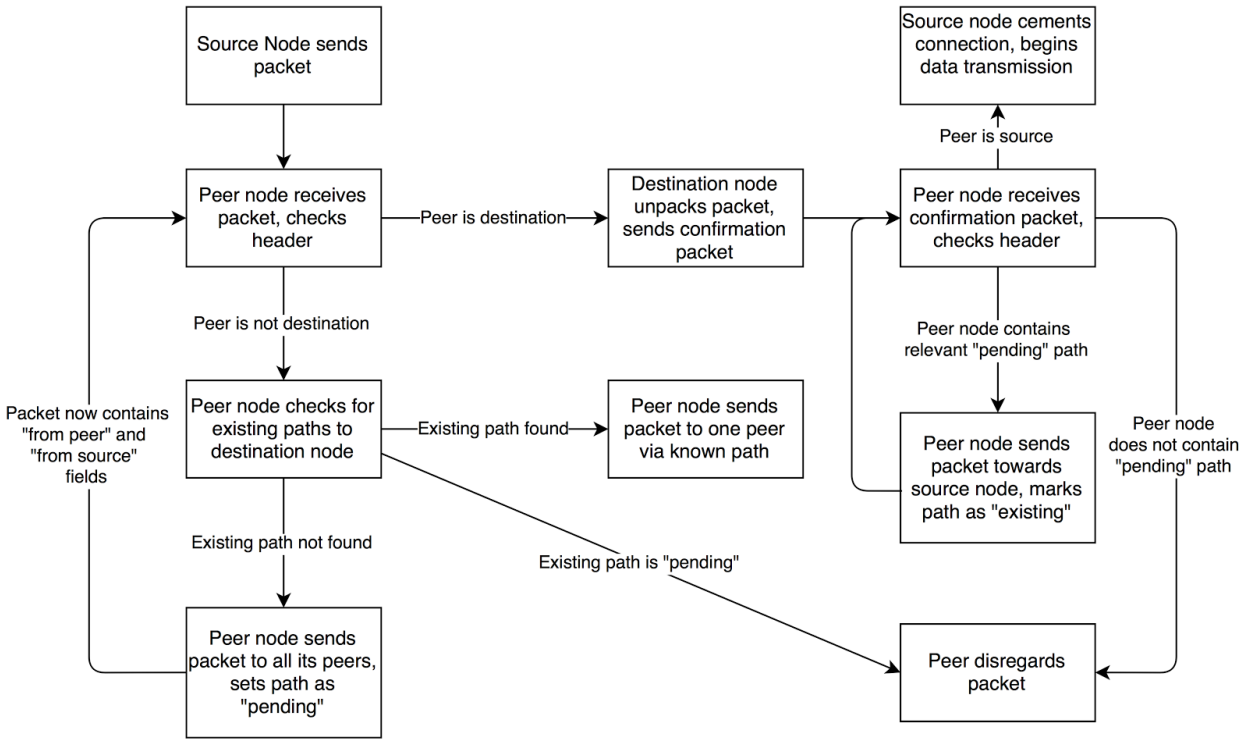

Fig.10. Routing Algorithm

2.8.2 Routing Algorithm

The routing algorithm operates on a reactive routing protocol. This means that instead of containing a complete map of the network to reference when it wants to send a message, the node will instead “flood” the network with so-called handshaking packets in an attempt to find a path to the node it wishes to connect to. This flooding will create transient paths that packets may use to traverse the network. Because these paths are not permanent (and are stored individually on each node), it is difficult for malicious programs to trace the routes used by a packet to traverse the network. The path creation and routing algorithm is outlined in Fig. 10. This process is handled entirely by the WiFi IC, as transmitting information to the microcontroller would take too long for every packet a node receives to be properly processed.

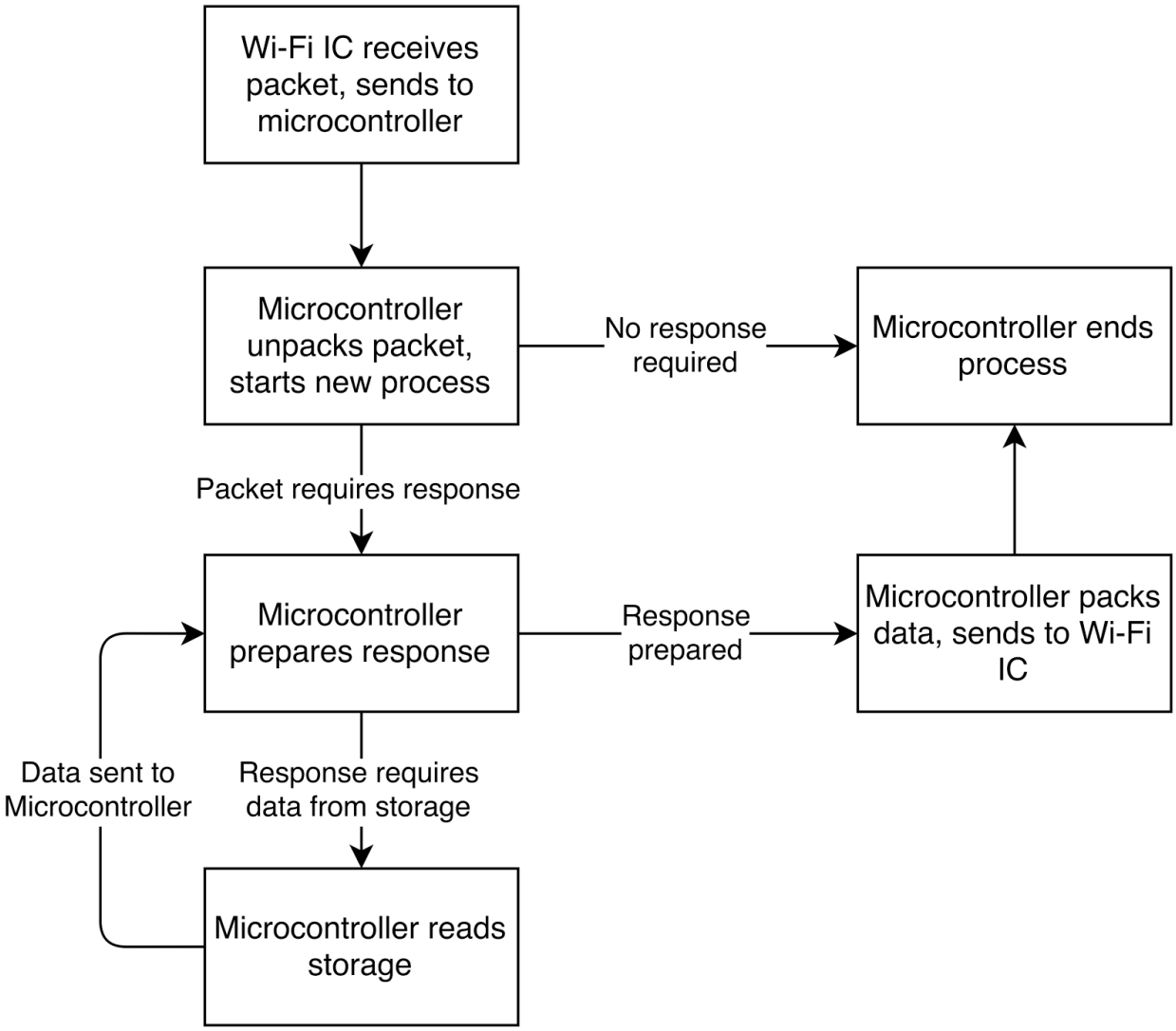

Fig. 11. Data Transmission Algorithm

2.8.3 Data Transmission Protocol

Data transmission is the other major protocol at work in the mesh network. It's built on top of UDP - a protocol built to send large amounts of data over a potentially unstable network. To this end, it leverages the complexity of the routing protocol to keep it's own mechanisms simple. The process (outlined in Fig. 11) begins with the microcontroller receiving a data packet (a process handled by the routing algorithm in Fig. 10). This packet is then processed by the system, and if a response is required, it is prepared. To do this the microcontroller may need to read the cached storage multiple times; because the microcontroller has no part in the handling of pass-through packets (packets that are not destined for the current node) the node can use a relatively slow storage method to store large amounts of data. Pass-through packets are described in Fig. 10 where the peer is not the destination node, and do not enter the decision tree of Fig. 11. This response is then packaged and sent back to the WiFi IC for transmission over the network, and the process is closed.

2.9 Tolerance Analysis

One important tolerance we must maintain is the antenna match to the WiFi IC. The target impedance is the characteristic impedance, 50Ω, within the IEEE 802.11 frequency band, 2402-2484MHz, to optimize the RF output power. The reflection coefficient, Γ, is the fraction of the signal amplitude that is reflected back to the source (in this case, the WiFi IC), and can be calculated as:

$$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0} = \frac{Z_L - 50\Omega}{Z_L + 50\Omega} \quad \text{Eq. 1}$$

Ideally, the reflection coefficient should be minimized with the network impedance approaching 50Ω, meaning that a minimal amount of power is dissipated as noise. Based on the above equation, the reflection coefficient is zero if Z_L=Z_0, which is 50Ω in this case. Using Eq. 1, we can calculate the transmitted power ratio below in terms of the mismatch ratio. The transmission coefficient τ represents the ratio of voltage that is transmitted to the antenna with respect to the applied voltage. The value of tau τ can be solved for through the conservation of power, which is proportional to the square of voltage.

$$\tau^2 = 1 - \Gamma^2 = 1 - \left(\frac{Z_L - 50\Omega}{Z_L + 50\Omega}\right)^2 \quad \text{Eq. 2}$$

$$Power\ loss\ (dB) = 10log_{10}\left(1 - \left(\frac{Z_L - 50\Omega}{Z_L + 50\Omega}\right)^2\right) \quad \text{Eq. 3}$$

As our goal is to keep the transmission through the antenna at a maximum, we hope not to reach this point. We will test the impedance match with a small coaxial cable ("pigtail") soldered to the LNA (RF I/O) ESP8266 trace as close to the chip as possible. This pigtail will be connected to a network analyzer, which will display the distance-corrected complex impedance on a Smith chart. This data will allow us to modify the inductors and capacitors that make up the 2-stage RF match. The match will likely be of a high-pass topology, meaning a capacitor in series and an inductor in parallel with the load. There is a larger selection of capacitors around j10-50Ω, at 2.4GHz (1.3-6.6pF), which are most useful for matching at 50Ω, than inductors around j10-50Ω at 2.4GHz (0.6-3.3nH). This leads us to match with a higher-impedance inductor in parallel and a lower-impedance capacitor in series.

Our impedance requirement to meet power transmission specifications is 50Ω +/-2.5Ω (+/-5%). The worst-case scenario is if the antenna's unmatched impedance approaches zero or infinity, which can be proven by the limits of Fig. 2. At zero (short), the circuit is essentially the capacitor as a load. Thus, the tolerance on the capacitor in this case must be 5%, though this is an unreasonable case as the real impedance can never equal 50Ω with only a capacitor. If the load is open (infinite impedance), the circuit consists of the matching capacitor in series with the matching inductor. Here, the total error must stay below 5%. The error magnitude on either component can add together with respect to the total impedance, and must stay below 2.5Ω total. Therefore, regardless of nominal values, the total error will stay within a 5% tolerance if both components are within +/-1.25Ω impedance of the nominal value. This 1.25Ω tolerance can yield more useful percentages using the below equations for capacitor and inductor impedance once nominal values are found.

$$Z_L = j\omega L \quad \text{Eq. 4a}$$

$$Z_C = \frac{-1}{j\omega C} \quad \text{Eq. 4b}$$

3 Costs

Our fixed development costs are estimated to be $40/hour, 10 hours/week for two people. We consider approximately 60% of our final design in this semester (16 weeks), neglecting the central server, mesh network optimization, and partnerships with NGOs:

$$2 \cdot \frac{\$40}{hr} \cdot \frac{10\ hr}{wk} \cdot \frac{16\ wks}{0.6} \cdot 2.5 = \$53,333 \qquad \text{Eq. 5}$$

Our parts and manufacturing prototype costs are estimated as $86 each:

Part	Cost (prototype)	Cost (bulk)
5V 5W solar panel (Amazon; generic/Talesun)	$13.98	$3.75
2.1AH Li-ion battery * 2 (AA Portable Power; 565068/Panasonic)	$13.98	$3.08
PCBs (PCBWay)	$3.10	$0.10
Microcontroller (Digikey; MKV30F64VLH10)	$3.10	$1.95
Assorted resistors, capacitors, ICs, crystals, sockets (Digikey; est.)	$10.00	$0.40
16Gb NAND Flash (Mouser; TH58NVG4S0FTA20)	$17.18	$4.18
WiFi IC (EspressIF; ESP8266)	$1.75	$1.75
Total	**$56.10**	**$15.21**

Since we will build ten boards, this yields a total development cost of $53,894. Our server parts and manufacturing costs are estimated at $130/node, considering commercial modules. Depending on production size, we may choose to develop a custom PCB. Since each network would have a maximum of one server, we do not expect significant bulk.

4 Schedule

Week	Daniel	Jeeth
2/7/16	Stress-test version 1 nodes, complete power tests	Initiate version 1 node programming, test wireless signal broadcast
2/14/16	Complete ESP secondary SPI speed tests, finalize version 2 node design considerations	Begin version 1 routing protocol design and programming
2/21/16	Finalize version 2 schematic	Continue version 1 programming, research data transmission protocols
2/28/16	Revise node and server schematics for efficiency, robustness and FCC/UL regulations	Continue version 1 programming, consult on potential version 2 PCB
3/6/16	Version 2 PCB design	Bugfix routing protocol for version 1, complete final design for data transmission protocol
3/13/16	Finish and order version 2 PCBs	Finalize routing protocol for version 1 node, begin work on data transmission protocols
3/20/16	Collect detailed power vs. throughput data on version 1 nodes to assist with separation planning	Continue work on data transmission protocol, bugfix routing protocol
3/27/16	Prototype version 2 nodes	Continue work on data transmission protocol, begin tests on version 2 wireless signal broadcast
4/3/16	Complete version 2 antenna match	Continue work on data transmission, begin translating routing protocol to version 2 prototypes
4/10/16	Extensive range vs. throughput experimentation with external antenna designs	Continue work on data transmission, finalize routing protocols for version 2 prototypes
4/17/16	Prototype case	Port data transmission protocols to version 2 prototype
4/24/16	Conduct environmental testing	Bugfix any problems caused in the transition between version 1 and version 2
5/1/16	Prepare final presentation	Begin final report

5 Ethics and Safety

There are several potential safety hazards with our project. Lithium-ion batteries can explode if overcharged or brought to extreme temperatures [13]. A li-ion cell can experience thermal runaway, where a positive feedback loop between cell temperature and discharge rate can lead to battery failure and potential explosion. To close this feedback loop, we will closely monitor cell temperature with a thermistor and isolate the battery from both

the charging circuitry and the node hardware if it reaches a temperature of above 45C or below 0C. Additionally, over-charging the battery can lead to a breakdown of the cathode, a highly exothermic process. Before attaching a battery, we will thoroughly verify our charging circuitry as per section 2.1.2 and the ECE445 battery safety document in [14] to ensure that the battery will not be charged over 4.21v under any circumstance.

As an outdoor electrical device, moisture could cause damage to the nodes leading to short-circuits. The case will need to adhere to strict IP66 guidelines, which keeps the internals dry from water jets in any direction.

Working in in an electronics lab carries its own challenges. We will be assembling the boards with soldering irons and hot air, as well as powering our boards with lab power supplies. We have attended lab safety training to learn how to use this equipment safely to avoid risk of burns, electrical shorts, and electric shock.

We are responsible for the information that is sent through our technology. This spread of valuable knowledge is an implementation of the IEEE Code of Ethics, #5: “To improve the understanding of technology; its appropriate application, and potential consequences” [15]. We hope to bring education and communication to the most remote corners of the world.

Unfortunately, risks surrounding the spread of information include piracy and mental health. Every day, people pirate music, movies, and even books via the conventional internet - and there is no reason to believe that our network will be any different. We are not explicitly giving out the tools to commit piracy or copyright infringement of any kind, but in a decentralized network it is impossible to track with any degree of certainty what information is shared. This would go against #7 and #9 of the IEEE Code of Ethics - the people committing piracy are not properly crediting the work of others, and they could be injuring the copyright holders by sharing content without paying for it [15]. We do not currently have a solution to this - we do not believe it would be the right course of action to limit the utility of our network simply because we anticipate a small subset of our users engaging in piracy.

On the internet, where a certain degree of anonymity is assured, there are fewer barriers to behaviors like cyberbullying. This type of harassment will adversely affect the mental health of those on the receiving end. It is entirely possible that the network will be used to discriminate by race, gender, or sexual orientation, violating #8 of the IEEE Code of Ethics, “to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression” [15]. We plan to introduce the ability for node owners to “ban” certain devices from services hosted on their node. The “banned” user would have no knowledge of this action; their packets would simply not return a response as the node hosting the service would throw them away instead of processing them. We believe this is the best course of action - any harassment can be stopped by an automated system, and the harasser(s) will never know that their messages aren’t being delivered.

Our mitigation techniques align with the IEEE Code of Ethics, #1: “To accept responsibility...” [15]. There are many risks that present themselves as a consequence to access and free communication, but we believe that the advantages of open resources, which include free education and the potential for economic development, far outweigh the potential negative effects.

References

[1] Google Public Data Explorer, 'World Development Indicators', 2014. [Online]. Available: http://www.google.com/publicdata/explore?ds=d5bncppjof8f9_#!ctype=l&strail=false&bcs=d&nselm=h&met_y=it_net_bbnd_p2&scale_y=lin&ind_y=false&rdim=region&idim=region:SSF&ifdim=region&hl=en_US&dl=en_US&ind=false. [Accessed: 28- Feb- 2016].

[2] Ericsson, 'Sub-Saharan Africa - Ericsson Mobility Report Appendix', 2014. [Online]. Available: http://www.ericsson.com/res/docs/2014/emr-june2014-regional-appendices-ssa.pdf. [Accessed: 28-Feb-2016].

[3] OpenSignal, 'The State of LTE February 2014', 2014. [Online]. Available: http://www.ericsson.com/res/docs/2014/emr-june2014-regional-appendices-ssa.pdf. [Accessed: 28-Feb-2016].

[4] Pew Research Center, 'Broadband Technology Fact Sheet', 2014. [Online]. Available: http://www.pewinternet.org/fact-sheets/broadband-technology-fact-sheet/. [Accessed: 28-Feb-2016].

[5] F. Brown, 'Percentage of global population living in cities, by continent', *The Guardian*, 2009. [Online]. Available: http://www.theguardian.com/news/datablog/2009/aug/18/percentage-population-living-cities. [Accessed: 28-Feb-2016].

[6] PewGlobal.org, 'Communications Technology in Emerging and Developing Nations', 2015. [Online]. Available: http://www.pewglobal.org/2015/03/19/1-communications-technology-in-emerging-and-developing-nations/. [Accessed: 02- Oct- 2015].

[7] Internet.org, 'Our Approach', 2016. [Online] Available: https://info.internet.org/en/approach/. [Accessed: 24-Dec-2016]

[8] Renewable Energy World, 'Solar-powered Internet Connectivity in Lascahobas, Haiti', 2012. [Online]. Available: http://www.renewableenergyworld.com/ugc/articles/2012/01/solarpowered-internet-connectivity-in-lascahobas-haiti.html. [Accessed: 24-Dec-2016]

[9] IFAD,'Rural Poverty in Nepal', 2011. [Online]. Available: http://www.ruralpovertyportal.org/country/home/tags/nepal. [Accessed: 28-Feb-2016].

[10] Web.archive.org, "Petzl reference system for lighting performance", 2016. [Online]. Available: http://web.archive.org/web/20080620123040/http://en.petzl.com/petzl/frontoffice/Lampes/static/referentiel/present_referentiel_en.jsp. [Accessed: 29- Feb- 2016].

[11] Skyworks, Inc., "AAT3693: 1.6 A Li-Ion/Polymer Battery Charger," 2016 [Online]. Available: http://www.skyworksinc.com/uploads/documents/AAT3693_201896E.pdf. [Accessed: 9-Feb-2016].

[12] Espressif, “ESP8266 Module Reference Design,” V1.3, 2016 [Online]. Available: http://espressif.com/en/support/download/documents?keys=ESP8266+hardware+resources [Accessed: 24-Jan-2017]

[13] D. Doughty and E. P. Roth, "A General Discussion of Li Ion Battery Safety," The Electrochemical Society Interface, Summer 2012.

[14] “Safe Practice for Lead Acid and Lithium Batteries,” University of Illinois ECE445 Course Staff, 2016. [Online]. Available: https://courses.engr.illinois.edu/ece445/documents/GeneralBatterySafety.pdf. [Accessed: 18-Jan- 2017].

[15] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 29- Feb- 2016].