12-2013

A Solar-Powered and Multi-Tiered Mesh Node for a Portable In Situ Emergency Response System

Adam Matthews
University of Arkansas, Fayetteville

A Solar-Powered and Multi-Tiered Mesh Node for a
Portable In Situ Emergency Response System

A Solar-Powered and Multi-Tiered Mesh Node for a Portable In Situ Emergency Response System

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Computer Engineering

by

Adam Matthews
University of Arkansas
Bachelor of Science in Computer Engineering, 2003
University of Arkansas
Master of Science in Computer Engineering, 2005

December 2013
University of Arkansas

This dissertation is approved for recommendation by the Graduate Council

Dr. James P. Parkerson Dissertation Director	Dr. Nilanjan Banerjee Committee Member/Co-Director
Dr. Christophe Bobda Committee Member	Dr. Randy Brown Committee Member

ABSTRACT

The aftermath of a natural disaster is typically characterized by lack of a reliable medium for dissemination of information to survivors. Current state-of-the-art emergency response systems rely on satellite radio-enabled devices, but survivors, unlike first responders, do not have access to such devices. To mitigate this problem, we present PERPETUU a solar-powered portable GIS microserver. The microserver node can be deployed in a disaster scene, and can serve maps to survivors viewable on browsers of off-the-shelf mobile systems. A key innovation in the design of the PERPETUU node is a multi-tiered hardware architecture-the system combines a low-power micro-controller, a medium-power Wifi module, and a high-power micro-processor to provide large spectrum of power states. In addition to being able to detect survivors using a low-power Wi-Fi sensing technique, the tiered design leverages hardware-assisted energy measurements, a wakeup controller to balance energy harvested from solar panels with energy consumed by the system, and a future-energy prediction algorithm, in order to provide natural disaster survivors with up-to-date emergency relief information. We evaluate PERPETUU using measurements from our prototype and trace-based simulations, and show that it can function near perpetually while serving GIS maps and other information to a large number of survivors.

ACKNOWLEDGEMENTS

Thanks to all the collaborators on this project in the Mobile Sensor Systems Laboratory. Thanks to William Johnston for his work on designing the GIS map stack, and to Stan Bobovych for his programming expertise in helping to design the storage cache necessary to virtualize the map stack on the microcontroller. Special thanks to my co-advisor Dr. Nilanjan Banerjee for his guidance during this project, and especially to my advisor Dr. James P. Parkerson for his support and guidance throughout the years. It would have been impossible to complete the project without their help.

Also, thank you to my family and friends who have supported me throughout the years. I would not be here without them.

TABLE OF CONTENTS

I. INTRODUCTION

With natural disasters and other calamities on the rise in recent years, there is a growing need for a reliable post-disaster communication infrastructure to provide survivors with emergency relief information. Disasters such as earthquakes, tornados, hurricanes, snow storms, and floods claim the lives of thousands of people (and rising) in the United States every year [11]. The majority of these casualties reportedly occur after the disaster has taken place [61], many of which could be avoided with an appropriate emergency response system in place to provide survivors with pertinent information such as what has happened, when they can expect relief, and how they can escape safely if possible. For example, victims in Mississippi and Louisiana were left stranded without power for months after hurricane Katrina [22], as were those in the wake of Hurricane Sandy [23] and the Tokyo Earthquake [21]. Had there been a post-disaster communication infrastructure in place at the time, each individual survivor would have had the opportunity to find out what to do and where to go for the best chance of escaping the carnage.

Our current overall communication system for emergency relief is lacking in preparedness for disasters in which survivors are left without connectivity to a power grid. Many of our current efforts for providing post-disaster information rely on media such as DSL, cable, mobile phone systems, and other Internet services, which become unavailable during power outages. As an alternative, many emergency relief personnel utilize satellite communications which do remain available during a power outage [6], although the majority of people do not have access to satellite radios. Furthermore, satellite radios cannot provide survivors with critical up-to-date information specific to their current location during a disaster, such as directions to a safe zone. Since neither satellite radios nor emergency sirens can provide such personalized information, communities may desire a more effective and comprehensive post-disaster system for disseminating relief information.

Objective

The objective of this research project has been to design and construct a solar-powered mesh that can be used in the aftermath of a natural disaster to supply survivors via Wi-Fi interface with relief information such as maps, locations and routes to safe zones, and any other pertinent information which could assist survivors in finding safety. In order to successfully remedy the existing problems with current emergency information services, several criteria must met for our design:

1) Our system must allow for *near* perpetual life of each mesh node by efficiently utilizing the solar energy available, by preventing the scenarios of running out of energy and wasting energy, and by prolonging the life of the batteries necessary to store the harvested energy.
2) The mesh should allow for self-stabilization in the event of node failures, which are likely to occur outdoors in the aftermath of a disaster. The nodes must redistribute the data residing at the failed node, requiring a sufficient level of redundancy accomplished primarily by creating overlapping coverage areas in the node placement scheme.
3) The system should be able to serve a maximum number of survivors by minimizing the power consumption. To accomplish this, the system must switch on the high-power Wi-Fi radio only when necessary, yet remain available by detecting when survivors are in the area *without* switching on the high-power server.
4) The system must be compatible with off-the-shelf mobile devices such as smartphones, tablets, and laptops, in order to provide mesh-client communication. The mesh node

must host a lightweight webserver, and through the use of an 802.11b/g radio, allow users to connect and transmit/receive data using the web browser and built-in Wi-Fi on their smart device.

Design Challenges

The design and implementation of a practical system for post-disaster information dissemination presents several major challenges. Since such a system must be available in the absence of power grid access, it must be made self-sustainable by using an alternative power source such as solar energy. Since solar energy harvesting is known to be unpredictable and difficult to model [27], [28], [30], a significant challenge is to balance system functionality and availability with an unpredictable energy supply, especially for a system which has power-hungry resources such as *PERPETUU*.

A second major challenge lies in dealing with the unpredictable and perhaps changing environment of the disaster area. As a consequence, both transient and permanent node failures may occur, requiring information to be rerouted around the failed node and also requiring the retrieval of information that resided on the failed node. Furthermore, changing environmental conditions at one location should be distributed throughout the mesh in a timely fashion in order to be up-to-date and thus helpful to survivors; and without internet connectivity, an alternative communication method must be utilized.

Finally, during an emergency, it is critical that pertinent information be accessible to everyone in the general public with common hand-held mobile devices. In order to provide accessibility to all survivors, the system should be compatible with virtually all "smart" mobile devices in today's market such as phones, tablets, and laptops, by allowing for Wi-Fi communication. Wi-Fi, which is based on the IEEE 802.11 standard, has no inherent mechanism for lowering power consumption [70], which presents a problem with providing constant accessibility to users.

As a solution to this problem and in light of its many challenges, we present *PERPETUU* – a solar-powered emergency mesh that is energy-efficient, self-sustainable, and available to survivors even when the power grid is nonoperational or unreachable. Each node in the mesh can provide users with pertinent information relative to their current location, including alerts, information-rich maps, and routes to safe areas in the region. Each node acts as a wireless access point for Wi-Fi users and is accessible to anyone with an off-the-shelf wireless device. When new information is uploaded to one mesh node it is automatically distributed to all other nodes in the mesh, hence allowing a large number of survivors over an expansive area to have access to critical updates during an emergency.

Motivation for a Multi-Tiered Architecture

The tasks of distributing GIS information among the nodes and to mobile clients require high-bandwidth radios which consume a large amount of energy. Since scavenged solar energy is sparse, a careful plan must be developed that conserves energy by only turning on the high-power radios when absolutely necessary. This need for ultra-low-power consumption by the system is in direct conflict with the need to maximize availability to the users by keeping the system *always on*. While optimization techniques exist for reducing power consumption such as CPU Dynamic Voltage and Frequency Scaling (DVFS) [62] and Power Saving Modes (PSM) [3], these techniques cannot eliminate the irreducible power draw of the system sourced by the CPU clock, RAM, and onboard oscillators. We resolve the conflict between providing high-responsiveness and ultra-low-power by presenting a multi-tiered platform – having a low-power radio with a low-power microprocessor to provide high-responsiveness, and a scheme in which the high-power radio with a high-power microprocessor is turned on *only* when necessary. Here *necessary* means turning on the high-power tier when it fits into the plan to maximize the overall number of clients served. A diagram of the multi-tiered architecture can be seen in Figure 1 in Chapter III.

The alternative to a multi-tiered approach for the GIS micro-server must result in one of two possible negative outcomes: either 1) the high-power radio always remains on, draining (and wasting) precious energy or 2) the high-power radio remains off most of the time and turns on intermittently. The result of number 2 would be frequent unnecessary wakeups as well as often not waking up when clients are present. To alleviate this problem, our multi-tiered system senses the presence of Wi-Fi clients using a low-power radio, and subsequently decides whether or not to wake up the high-power server based on the sensing of a client and the amount of energy available to the system.

II. BACKGROUND

In this chapter we describe some of the background work from which our research emerged. We describe previous and current methodologies for designing low-power photo-voltaic systems, future energy prediction, Wi-Fi sensing, and the infrastructure for contemporary emergency alert systems.

Low-Power Photo-Voltaic Systems

Several factors have led to research in finding new methods for improving energy efficiency in wireless LANs and meshes. First, available energy is limited by the size of costly solar panels and batteries, while the prevalence of solar-powered wireless systems has grown in recent years due to the spread of smart mobile devices. Secondly, there is no inherent mechanism in the 802.11 standard which allows for any power saving method at an access point [70]. This shortcoming of IEEE 802.11 has been a hindrance to the development of practical power saving methods for WLAN infrastructure [69], [70].

Energy-aware resource management has been studied extensively in wireless networks in several different contexts. Node placement and routing algorithms for wireless mesh networks, for instance, have been popular areas of study for some time [66], [68]. Methods for adaptively distributing traffic flows using energy-aware routing schemes, in an attempt to improve performance over shortest path routing, have been proposed in [71], [72]. To accommodate for peak loading requirements in solar-powered nodes, authors have proposed methods for overlapping the coverage area of APs to provide load balancing in [69], [73]. There are many aspects such as these to consider in developing a *comprehensive* plan for power management in a wireless mesh network. For example, in [74], the authors present a method for provisioning resources for a node which considers positional solar insolation awareness, i.e. it assesses and then considers in its energy provisioning the relative position of the node in terms of its relationship to the sun.

Energy Prediction

Harvested-energy management systems must essentially be able to adapt to uncertainty in future energy availability by predicting expected incoming energy. Energy prediction is not a simple task however, as it depends upon varying, and to a large degree unpredictable, conditions such as the efficiency of the PV cells, the non-ideal behavior of the energy harvesting circuit and battery system, and varying irradiance [13], [40]. In the case of solar energy, the 24-hour quasi-repeating cycle is exploited as the foundation for prediction, by considering the current and recent trends in solar availability.

The success of an energy prediction method depends on the observed accuracy weighed against the computational overhead of the algorithm. While fuzzy systems, such as in [28], are a popular area of research and provide relatively accurate results, they require very high computational overhead, making them unsuitable for the *PERPETUU* node and most WSN applications. Simple prediction techniques, such as in [2], [29], and [40], make use of parameters such as number of energy samples taken per day, length of prediction horizon of samples, and window sizes of samples used – all of which determine both the accuracy and cost, in terms of computational overhead and memory, of each technique. These simpler prediction algorithms, which are better suited for a system such as *PERPETUU*, exhibit a tradeoff between the size of the time window used (or frequency of prediction iterations which determines power consumed by the process) and the accuracy of the results.

Wi-Fi Sensing

Wi-Fi sensing usually refers to detecting Wi-Fi APs or hotspots, while for our application we must detect a Wi-Fi *client* whom is searching for an AP. There are many Wi-Fi packet sniffer applications – some hardware and some software. Most of the dedicated hardware tools are found in expensive, bulky, and/or high-power lab equipment. There are several Linux tools that could assist in recognizing Wi-Fi clients, such as WiSPY, WifiScanner, and Kismet. However, these tools require Linux to be up and running, and as later described, the *PERPETUU* node cannot boot the Linux machine on the Gumstix™ until after it has detected a Wi-Fi client – a *catch- 22* for our system. The Wi-Fi client-sensing problem is essentially the crux of our overall problem which is to detect the presence of a *disassociated* Wi-Fi client without switching on our high-power Wi-Fi AP.

Emergency Alert Systems

As mentioned previously, most current emergency information systems are inadequate for providing relief during power outages, which occur often during disasters. Many local areas, such as our city of Fayetteville, allow residents to register for text message or email alerts, but this relies on the power grid and internet connectivity. Fayetteville also has a siren that is supposed to sound during an emergency, but this provides no information except "something is wrong".

There are many web-based services which provide users with alerts via their smart mobile devices. For example, the company Ushahidi [84] makes an app called CrowdMap™ which allows users to download crowdsourced maps from all over the world to get real-time information about their current position. They also market an accompanying "backup generator" to the internet called BRCK™, which basically a portable Wi-Fi hotspot powered by rechargeable batteries, but it relies on mobile phone networks. There are many smartphone apps, such as Disaster Alert™, Earthquake! ™, and StormTracker™, which allow users to get up-to-the-minute details on ongoing natural disasters throughout the world. AT&T and other wireless carriers now provide many of their customers with Wireless Emergency Alerts (WEAs – emergency alerts, AMBER alerts, and presidential alerts) *automatically* [85], pursuant to the Warning Alert and Response Network Act (WARN Act) of 2006, which states cell phone companies have the right to transmit WEAs to their customers phones as a free service [86].

Serval Mesh

> "*Communications should not just be for the geographically, financially otherwise fortunate — for it is the unfortunate who need it most*." [75]

This quote is the first thing that appears on the website for the Serval Project – a non-profit project headed by researchers in Australia who are attempting to create a "disaster proof" mobile network [76]. Funded by a variety of charitable organizations, the Serval Mesh is an Android™ smartphone app which allows Android phones to send and receive data to one another using built-in Wi-Fi without depending on cellular networks, cell phone towers, or the internet. In essence, Android phones running the Serval Mesh app form their own mesh. The way it works is calls and texts are broadcast out to all smartphones within range having the app, and like a stone makes ripples through a pond, once the message is received at its final destination, that device sends back a signal through the "pond" which cancels copies of the message that still may be traversing through the "pond" [79].

Currently, there are barriers holding back the Serval Project from reaching its full potential for which the researchers are striving. First, the range for Wi-Fi in smartphones is only about 170 meters outdoors. To improve upon this, developers are working on a *Mesh Extender* to significantly extend this range [80]. The Serval Mesh Extender consists of a Wi-Fi interface capable of running AP and ad hoc mode simultaneously and an Arduino™ micro-controller to interface with optional additional radios. Secondly, Serval Mesh is being developed for Android devices, but is currently unavailable for other platforms such as Apple. The iPhone™ presents many problems such as providing inadequate support for native code compilation using GCC and restricting iOS app development terms [81]. Another hindrance for the proliferation of the Serval Mesh is sure to be lack of support from the telecommunications companies, since users will be able to communicate with one another without having to go through an established mobile network [78]. The Serval Project team remains confident that these are only temporary roadblocks and at least the Serval concept will eventually become a ubiquitous part of our mobile technologies [79], [81].

The implications of Serval Mesh could extend quite far, not only improving communications for people in disaster areas, but also for people in war zones, and developing and third world countries [76]. By allowing mobile communication even when there is access to neither a power grid nor outside cellular networks, it can provide relief to survivors of a disaster and also improve the quality of life for those in areas without telecommunication service. In war zones, someone could take a picture or video at a protest or uprising and have it dispersed immediately throughout the mesh. This way, even if the person's phone is seized the footage has already been dispersed, making the Serval Mesh a kind of deterrent for social injustice and many forms of violence and abuse.

Research Contributions

Although the trend for emergency information disbursement is moving toward integration with smart mobile technology, for the foreseeable future there will still be usefulness in having stationary access points to assist in disaster relief, even if every phone were equipped with Serval Mesh or a similar mechanism. Although survivors may be able to communicate with each other, they will still benefit from local GIS information which cannot be provided by a mobile mesh without internet access. A stationary mesh node may also offload some of the computing and routing workload, thereby reducing the energy burden of mobile handhelds. Also, much of our research may be applicable to the mobile devices running Serval Mesh, such as making them solar-powered and therefore adaptive to a fluctuating energy source. Our research contributions as a result of the design, implementation, and evaluation of *PERPETUU*, which should be viable to future developers for years to come, are described below.

(1) Hardware architecture for a solar panel-driven Wi-Fi AP: We present a multi-tiered, multi-radio hardware architecture which provides high availability and computational capabilities while minimizing overall power consumption. We accomplish this at the circuit level by using efficient power profiling and power gating techniques, and at the system level by implementing Wi-Fi sensing using a low-power radio in order to notify the system of the presence of a Wi-Fi client. This way, the high-power Wi-Fi server (high-power tier) remains off until both a client in the vicinity is detected *AND* there is sufficient power available to serve the client. At this time, we have found no other system that uses this technique to conserve energy.

(2) Virtualized software architecture: We present software techniques in which the micro-controller virtualizes system resources, and by using intelligent decision-making processes for allocating time and energy to resources, seeks to minimize the use of the high-power radio tier. These software techniques include a novel radio wakeup controller algorithm, a distinctly new future-energy prediction algorithm, a low-power method for detecting Wi-Fi clients, and an algorithm for allocating time and energy resources to the up-and-running high-power Gumstix module.

(3) Prototype implementation: We have implemented a fully functional *PERPETUU* node, and have evaluated the node through both trace-based simulations and actual work loads. The custom power supply board integrates a low-power PIC™ micro-controller, solar energy harvester and battery charger, supporting circuitry and interfacing with the three radios supporting the multi-tiered paradigm. The prototype of *PERPETUU* is unique in its ability to conserve power while providing *nearly* always-on availability, by use of its customized hardware and software modules; and it is ready to be deployed and tested in a real outdoor environment.

(4) Energy prediction algorithm: The energy prediction algorithm used by *PERPETUU* is based on the *Weather-Conditioned Moving Average* (WCMA) algorithm proposed in [40], which is in turn based on the common *Exponentially Weighted Moving-Average* (EWMA) algorithm. Our proposed prediction algorithm adds a new component to the algorithm in [40], and through our simulation results while using real data over the period of one year, as described in Chapter IV, we show an improvement in future energy prediction accuracy.

III. HARDWARE ARCHITECTURE

The hardware architecture for *PERPETUU* is built around five basic hardware pieces needed to achieve the design goals for the project: 1) a multi-tiered, multi-radio platform allowing for a wide range of power states, 2) a PIC micro-controller and its peripheral devices and supporting hardware for operation control, 3) a means of energy harvesting and energy storage, 4) power gating techniques for minimizing power consumption, and 5) the ability to make fine-grained current and voltage measurements.

Multi-Tiered Radio Platform

The hardware architecture is centered, around the multi-tiered platform described which consists of two subsystems: 1) subsystem-1: a low-power PIC micro-controller with two low-power RF transceivers, and 2) subsystem-2: a high-power microprocessor with a high-bandwidth radio on a Gumstix COM module which is hosting the GIS map stack. The low-power subsystem-1 provides high responsiveness being almost always on (often in power save mode), while the high-power subsystem-2 is powered down. The low-power radio (WizFi220) is used to detect the presence of Wi-Fi activity, indicating there may be a user trying to connect to the mesh node. If a potential user is detected, the low-power micro-controller on tier-1 must make a decision as whether or not to wake up tier-2 for the purpose of transmitting data over Wi-Fi. This decision is based on the estimated available energy present, along with the estimated energy expected in the near-future. The other low-power radio on tier-1 is an XBee™ RF module which is used for intercommunication between the nodes.

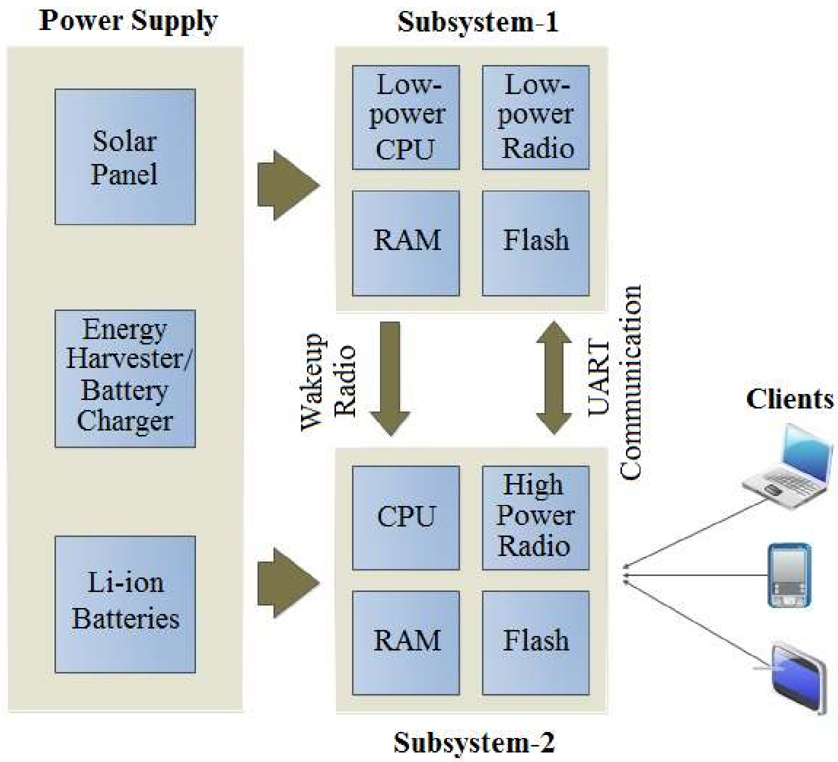


Figure 1: Multi-tiered hardware architecture for a *PERPETUU* node

Micro-Controller and Power Rails

As an intermediary between the power supply and the system load, there are two buck/boost DC/DC power converters – one proving a 3.3V power rail used by the Gumstix COM and low-power Wi-Fi radio (used for Wi-Fi sensing), and one providing a 3.3V power rail used by the rest of the system. The power converter for the Gumstix is controlled by the PIC micro-controller. This converter can be enabled and disabled by the PIC, as can the two analog switches which in turn can force the Gumstix and low-power radio on and off independently.

Solar Energy Harvesting and Storage

When there is ample solar energy available, a battery charger IC provides a constant float voltage for charging the batteries, while allowing the solar panel to also power the load of the system concurrently. It also provides the protection circuitry needed for charging Li-ion batteries, and allows the batteries to power the system load when there is insufficient power from the solar panel. The harvester allows the solar panel to power the system directly, thereby reducing wasted energy from the solar panel. This occurs when either the batteries are nearly full, or when there is energy available from the solar panel, but it is not enough to charge the batteries efficiently. The harvester has a programmable minimum threshold voltage for the solar panel, whereby the harvester is automatically disabled when the solar panel voltage falls below the threshold. The ideal threshold depends on the size of the solar panel and the size of the batteries. For the *PERPETUU* node we use 12V as the minimum threshold voltage for the solar panel. When the solar panel voltage falls below 12V the source of power for the system is a combination of the batteries and solar panel, except when the solar panel voltage is zero, in which case the system is powered by the batteries alone.

Fine-Grained Power Measurements and Power Gating

Our system is capable of making fine-grained power measurements using the onboard ADC module of the PIC micro-controller, as well as power gating for several components including four current-sense amplifiers, temperature sensor, Gumstix module, and low-power WizFi™ module. The supply board can collect measurements for the power harvested from the solar panel, the residual battery charge, the power draw for the PIC, and the power draw for the Gumstix. We use low-power current accumulators to measure the power consumed by different subsystems. The combination of power gating and fine-grain power measurements comprise a power controller capable of balancing energy harvested with energy expended by the system.

IV. SOFTWARE ARCHITECTURE

The design principles for the software architecture of *PERPETUU* are two-fold: 1) Virtualize the high-power subsystem residing on the Gumstix at the low-power subsystem existing between the PIC and WizFi radio, to achieve *nearly* always-on operation at a minimal energy cost. 2) Utilize high-energy demanding resources intelligently by balancing solar energy harvested with the allotted energy for each resource. *PERPETUU* succeeds in meeting the software design goals by using the following essential software modules, an illustrated in Figure 2.

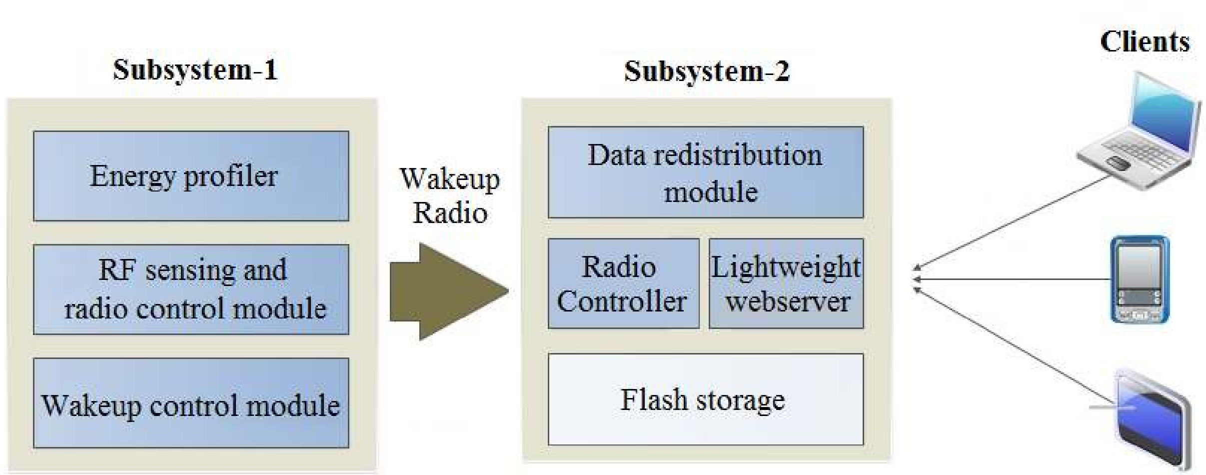
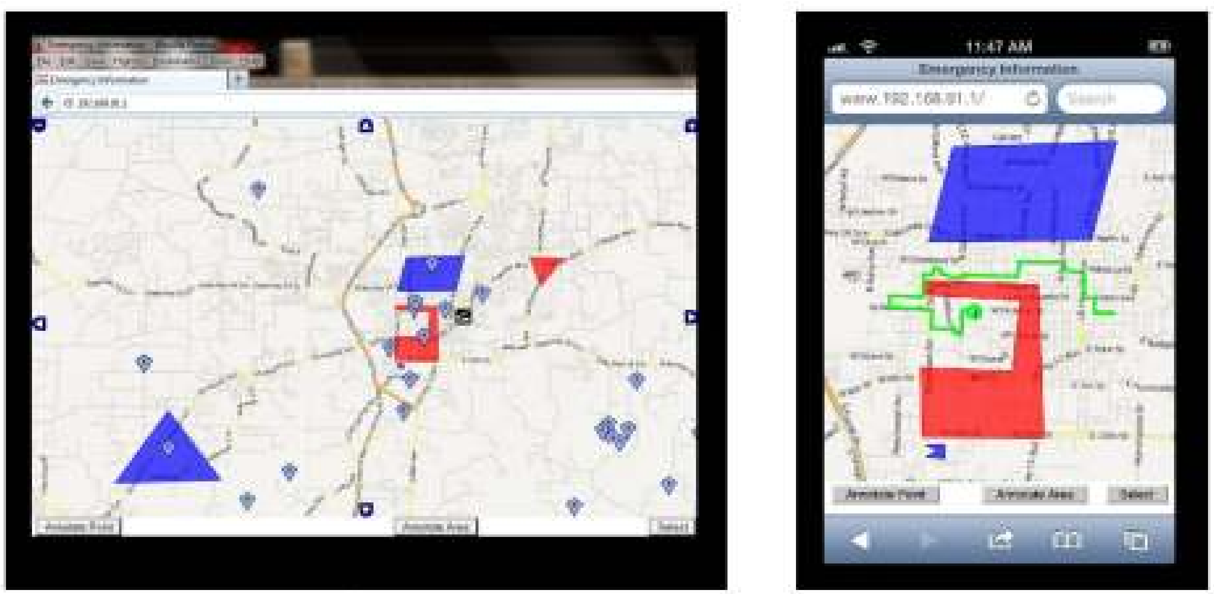


Figure 2: Basic software architecture for *PERPETUU* node

Portable Map Stack

The Gumstix module hosts a portable custom designed map stack module, while the PIC micro-controller runs a virtualized version of the map stack. Map layers are retrieved from a residing postgresql database by the map stack, which also runs mapserver, an apache webserver, and custom software modules that adaptively send different map layers to Wi-Fi clients. The map stack is a key component to making the *PERPETUU* system low-power, adaptive, and accessible to users. Figure 3 shows the viewable map using off-the-shelf wireless devices. The map stack residing on the Gumstix was designed and implemented by Bill Johnston (University of Arkansas).



(a) (b)

Figure 3: User interface for the viewable maps downloaded onto (a) a Dell laptop (b) an iPhone. Printed with permission from developer Bill Johnston (Appendix C).

There are three key components to the map stack making it indispensable to the *PERPETUU* system: 1) Multi-resolution map dissemination: the map stack can disseminate different layers adaptively to users including a *base* layer, a layer showing locations of dangerous and safe areas, a layer showing routes to safe areas, and a *crowd* layer which is annotated by users. The mapserver can be programmed to adaptively choose which map layers are to be disseminated to a user, a decision made primarily based on available energy to the system. For instance, if the Gumstix Wi-Fi server is up and running and a *PERPETUU* node is low on reserve energy, it can transmit only the base layer to a user and opt out of sending the map routing layer, since the base layer takes about two seconds to send compared to the 13 seconds it takes to transmit the route layer. 2) Efficient route generation: the map stack can generate routes to areas of safety for users, considering the users current position, using HTML5 on smartphone browsers, and location of the nearest safe zone. Safe zones and hazardous zones can be annotated by other users, and these zones are used to assign weights to local streets in these areas – giving high weights to roads intersecting with dangerous areas. The algorithm uses mapnik [37] and the A* search algorithm to determine the shortest and safest route to a safe area, which can be displayed on the map on the user's screen. 3) Crowd layers: the third innovative component of the map stack is the ability for users to annotate points and areas using labels and colors to indicate safe and unsafe areas. In an emergency scenario, it is a critical aspect of the system for the users to be able to crowdsource information, since there is no backhaul connection to the internet and no other source from which to get the information. The crowdsource layer is stored on the Gumstix and transmitted to neighboring nodes via the XBee radio which is part of the low-power tier of the *PERPETUU* node.

Storage Cache

The Gumstix database is virtualized on the low-power tier subsystem using an EEPROM peripheral to the micro-controller. This virtualization allows dynamic layers such as crowd layers to be transmitted to neighboring mesh nodes without waking up the high-power tier. The GIS data and crowd layers are sent from the Gumstix to the PIC and stored on the EEPROM. We implement wear leveling in the EEPROM by using a log-structured system, in order to extend the life of the EEPROM. The EEPROM is divided into two halves, while one half is being used to store new data the second half is being garbage-collected. The EEPROM maintains a table of contents which contains a status byte indicating if the data has been sent out to neighboring nodes through the XBee radio and indicating if the data has been synced with its own Gumstix database. New files get an entry in the table of contents and are appended to the log – no useful files ever need be overwritten. The storage cache residing on the EEPROM, as well as the communication between the Gumstix and PIC, were designed and implemented by Stan Bobovych (University of Maryland, Baltimore County).

Wi-Fi Sensing

The *PERPETUU* node utilizes a relatively low-power Wi-Fi module (WizFi220) to sense the presence of a Wi-Fi client. This low-power Wi-Fi module is used to sense whether or not a user is present without waking up the Gumstix. The low-power module sends out the same ESSID as the Gumstix Wi-Fi. A Wi-Fi user can connect to the low-power module, which indicates to the PIC a user is present. If a user is detected, the PIC uses the wakeup controller algorithm to determine whether or not the Gumstix should be woken up to serve map data. This is a critical component to the system – providing the *nearly* always-on availability criteria while consuming only a fraction of the power of the Gumstix. While the WizFi220 has a boot time of only 20ms, it takes about ten seconds, from beginning to end, to boot, broadcast its ESSID, allow for a client to connect to its Wi-Fi server, and notify the PIC that a client has been detected.

Energy Profiler

In assessing the amount of energy available to a *PERPETUU* node, the energy profiler module examines three possible sources of energy: 1) the present energy input from the solar panel, 2) the energy residing on the Li-ion batteries, and 3) the predicted future input from the solar panel. The present input power from the solar panel is measured with a current-sense amplifier and the voltage is measured with the ADC on the PIC, while the predicted future energy from the solar panel is described in the following section entitled "Future energy prediction".

The residual energy on the Li-ion batteries, or state of charge (SOC), is at times estimated by measuring the open-circuit voltage (OCV) across the batteries, and at times measured using a standard discrete-time flow model that describes the input and output energy of the system. The OCV method (described below) is used when the Gumstix is powered off and the batteries are relatively stable. The energy flow model is used when the Gumstix is powered on, as it is not possible to determine the SOC of Li-ion batteries using the OCV method when the voltage is not stable [10]. With the energy flow model [67], [68], and [72], $E_{source}(k)$ is defined as the total energy sourced by the solar panel over the time increment $[(k - 1)\Delta, k\Delta]$. Modeling of photo-voltaic systems is usually done in discrete time increments, often using hourly increments represented by Δ. $B(k)$ is defined as the residual battery energy stored at time $k\Delta$, $\mathbf{B}_{outage}$ is the maximum allowed discharge depth of the battery, and $\mathbf{B}_{max}$ is the maximum battery capacity. If we let $L(k)$ represent the load energy demand over the time duration $[(k-1)\Delta, k\Delta]$, then we can assume the following:

$$B(k) = \min\{\max[B(k-1) + E_{source}(k) - L(k), \mathbf{B}_{outage}], \mathbf{B}_{max}\} \qquad (1)$$

Here, the residual energy on a battery at time k is equal to the energy at time $k-1$ plus the energy supplied by the solar panel minus the energy consumed by the load during that time period. Recall, $B(k)$ is the residual battery energy and $\mathbf{B}_{outage}$ is the maximum allowed discharge depth. When $B(k) < \mathbf{B}_{outage}$, the node must disconnect its load until sufficient energy is once again available. This method is incorporated into the Gumstix wakeup control module, as described in Chapter IV, in order to ensure the batteries do not fall below their minimum threshold.

Since accuracy of the SOC estimation is improved the longer the battery has been at rest, this OCV measurement is the first thing the PIC does when it wakes from sleep. Once the OCV has been determined, a calculation is made based on the graph shown in Figure 4 which shows the conventional relationship between OCV and SOC. While determining the SOC for any type of battery is a notoriously difficult task [3], [10], [42], and there are other more involved and expensive methods for making more accurate estimations, the method shown here is sufficient for the *PERPETUU* node because high accuracy is not required. Since it is our goal to avoid empty and full batteries, we leave a 10-20% cushion on both sides of this spectrum. If the battery enters a state within 8-9% of being empty or full, the life of the battery will not be noticeably affected and therefore the SOC measurement need only be an estimate.

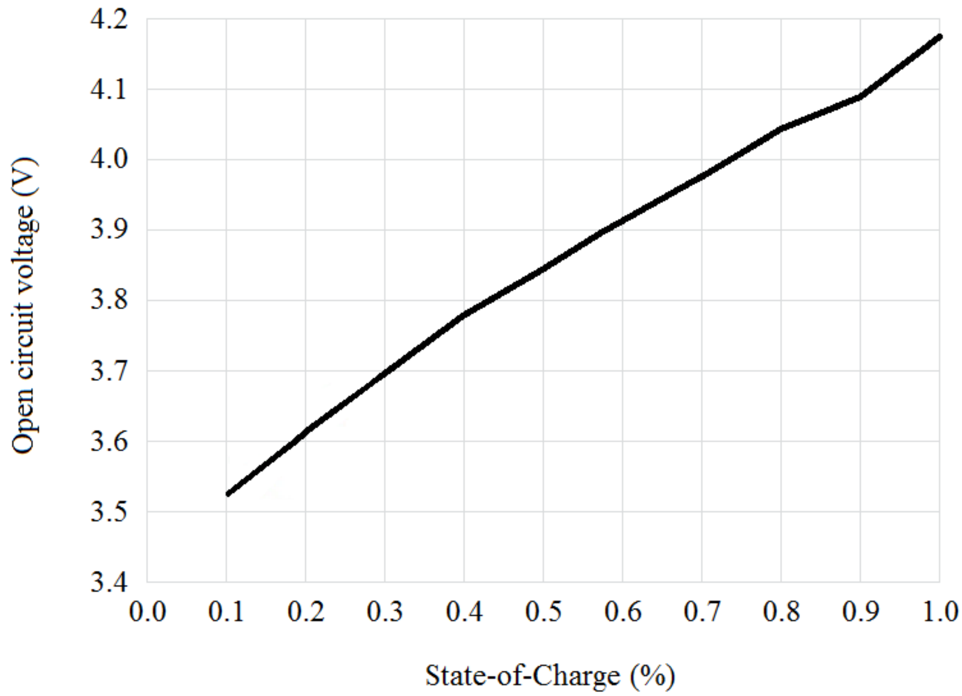


Figure 4: Conventional relationship between open-circuit voltage (OCV) and state-of-charge (SOC) for Li-ion batteries. This data was taken from [10], [35].

Future Energy Prediction

Our energy prediction algorithm is based on the algorithm proposed in [40] called *Weather-Conditioned Moving Average* (WCMA), which is in turn based on the common *Exponentially Weighted Moving-Average* (EWMA) estimation method. In EWMA, the day is divided into slots, and the estimated value for each slot i is stored in a vector $X(i)$. The value of each slot is updated using the following equation:

$$X(i) = \alpha \cdot X(i - 1) + (1 - \alpha) \cdot x(i) \qquad (2)$$

Here, $x(i)$ denotes the value of the real energy observed at the end of slot i, $X(i)$ represents the average harvested energy of past days in the same slot, and α is a weighting factor. WCMA improves upon EWMA by taking into account the solar conditions at a specific time of day and the relative weather conditions for the current day.

WCMA uses a matrix E of size D x N, storing N energy values for D past days, where $E(i, j)$ is the energy stored in the matrix for the j^{th} sample on the i^{th} day, and the predicted next E (energy) is computed according to the following equation:

$$E(d, n + 1) = \alpha \cdot E(d, n) + GAP_k \cdot (1 - \alpha) \cdot M_D(d, n + 1) \qquad (3)$$

$M_D(d, n + 1)$ is the mean energy of D past days at $n + 1$ sample of the day:

$$M_D(d, n) = \frac{\sum_{i=d-1}^{d-D} E(i,n)}{D} \qquad (4)$$

The factor GAP_k measures present-day solar conditions relative to previous days. To compute GAP_k, we first define a vector $V = [v_1, v_2, \ldots, v_K]$ with K elements. Each v is computed using the equation:

$$V_k = \frac{E(d,n-K+k-1)}{MD(d,n-K+k-1)} \qquad (6)$$

V is simply the current energy divided by the average energy for that time slot. So, values of v greater than one represent sunny days, and values less than one represent cloudy days. In order to give more weight to recent v values, a vector P is used, where $P = [p_1,p_2, \ldots, p_K]$ and $p_k = k/K$. Then, the weighting factor GAP_K is computed, where $V{\cdot}P$ is the vector dot product of V and P:

$$GAP_k = \frac{V \cdot P}{\sum P} \qquad (7)$$

WCMA requires calibration and optimization of the parameters D, N, α, and K, and can be accomplished with an error function defined by:

$$Err = \frac{1}{N} \sum_{i=1}^{N} abs(1 - \frac{Ereal}{Epred}) \qquad (8)$$

In addition to minimizing the error, the tradeoff between accuracy and duty cycle must also be considered. That is, as the size of the time slots decrease, the accuracy increases but the power required to wake the PIC and run the program increases. For our prediction algorithm, we make an improvement on the WCMA algorithm by including a second *GAP* factor (GAP_2) in our equation. The WCMA algorithm considers the previous energy value ($E(d, n)$) to be on par with the current energy level, by multiplying $E(d, n)$ by the weighting factor α. In reality, the previous energy value on average will be smaller when the sun is rising and larger when the sun is setting. To account for this deviation, we first calculate $GAP_2 = M_D(d, n+1) - M_D(d, n)$; this is equal to the average energy difference between the current and previous time slots. Our new equation for predicting energy for the next time slot becomes:

$$E(d, n+1) = \alpha \cdot (GAP_2 + E(d, n)) + GAP_k \cdot (1 - \alpha) \cdot M_D(d, n+1) \qquad (9)$$

Figure 5 shows the average error of the WCMA algorithm compared to the average error of our improved algorithm over the complete range of values for *alpha*. As shown, the smallest error for the WCMA algorithm (at alpha=0.1) is 25.0%, while the smallest error for our improved algorithm (at alpha=0.4) is 24.5%. This reveals an increase in accuracy of about 2% over WCMA. Although 2% is a seemingly small improvement, two things should be noted: 1) it is difficult to substantially improve upon a decent weather prediction algorithm because the weather is simply unpredictable, although having said that, the algorithm could possibly be improved upon further, and 2) if the small improvement can increase the number of clients served, even by one, during a disaster, then the modification is justified since the additional computational overhead is insignificant.

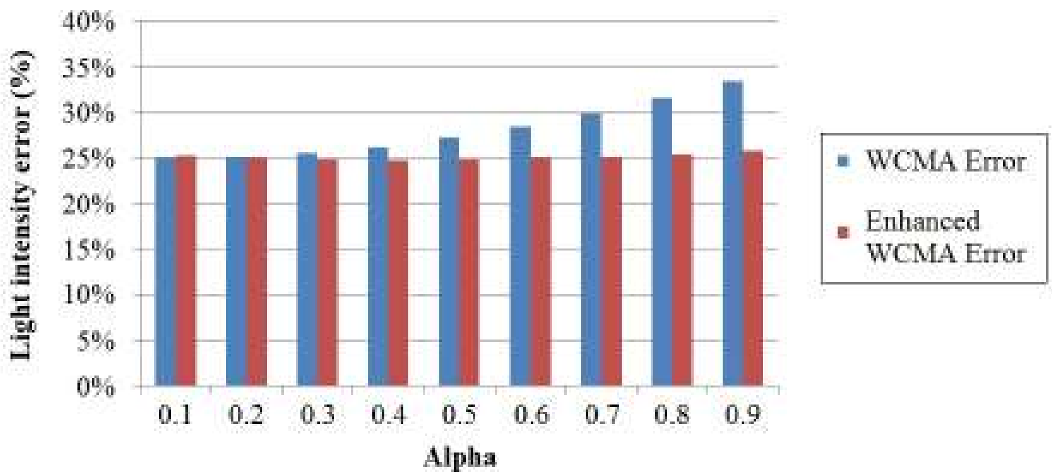


Figure 5: Average energy prediction error for the WCMA algorithm as compared to our improvement on the WCMA algorithm

Wakeup Controller

The wakeup controller plays a vital role in maximizing the number of users served during an emergency scenario, as will be shown in Chapter VI. It uses the predicted energy, hardware measurements on residual battery SOC, and energy consumption of different components of the system to calculate whether the low power Wi-Fi module and the Gumstix should be woken up to serve maps. It also determines the period of time that the Gumstix should remain on to serve users (T_{Wait}).

The goal of the wakeup controller is two-fold: to avoid an empty battery and a full battery, while serving the maximal number of survivors. An empty battery leads to a non-functional node, a full battery leads to wasted harvested energy, and both shorten the lifespan of the battery. A battery is considered empty in a *PERPETUU* node when the residual battery SOC is less than 20% and full when the SOC is at 90%, since full charges and deep charges of the battery reduce battery lifetime. The 20% and 90% thresholds are not chosen arbitrarily, and they also are not chosen as part of a hard and fast rule. The closer a battery gets to the extreme states of full and empty, the more the battery is put under strain [10]. Since our methods for measuring SOC are only estimates, as the characteristics of Li-ion batteries vary as some of their environmental elements vary such as temperature, we allow for a cushion to account for our errors in measurement. The logic behind allowing the battery to reach 20% of its minimum and to reach only 10% of its maximum is three-fold: 1) we have an interest in maintaining as much charge in the battery as possible. When the battery is ninety percent full it is under little strain, even considering the errors in measurement. 2) When the battery is discharging it becomes increasingly difficult to estimate the SOC as it approaches empty [10]. 3) The battery charger/harvester circuit has built-in overcharge protection – as the battery approaches a full state, the charger provides a trickle charge slowing down the charging process automatically. There is no way for the charger to protect against over discharge. The system relies on the energy profiler to cutoff the discharge. Since the energy profiler is much less accurate than the dedicated hardware of the harvester/charger, and if the battery gets too close to an empty state it could be destroyed permanently, we allow for a larger cushion when discharging. When designing the system, it must be considered that only about 70% of the battery's capacity is going to be used normally, so the size of the battery should therefore be chosen accordingly.

The wakeup controller maintains an extensive profile of energy consumed by different components of the system (as can be seen in Table 1 in Chapter VI): energy consumed by the Gumstix node and map dissemination module (P_g) at different resolutions, energy consumed by the Gumstix boot-up process (P_b), energy consumed by the Wi-Fi sensing module at the low-power subsystem ($P_{Wi\text{-}Fi}$), and the idle power consumption of the power supply board and the PIC (P_i). The algorithm uses the energy harvested in the next hour (P_h) and first determines whether the energy harvested is sufficient to wake the Wi-Fi sensing module and the Gumstix. Note that detecting a client using the sensing module is useless if the system lacks sufficient energy to wake the Gumstix to serve maps, so if there is not enough energy to power the Gumstix then the Wi-Fi sensing module remains off. If the energy harvested alone is sufficient, the system wakes up the Wi-Fi sensing module to look for users, since our harvester can power the system directly from the solar panels bypassing the batteries, and it would never make sense to deny a present client in order to charge the battery to serve a future client. If the available energy being harvested is insufficient, the wakeup controller determines if the residual charge on the battery and the energy harvested together is sufficient to wake up the Wi-Fi sensing module and the Gumstix. Consequently, if the Wi-Fi sensing module senses a Wi-Fi user nearby, it must first reaffirm that there is still sufficient energy to power the Gumstix, and if there is it wakes up the Gumstix to serve maps. The next decision made by the wakeup controller is to determine the amount of time that the Gumstix module should remain on. The resolution of the map served as well as the number of users served is a function of this wait time (T_{Wait}).

The *PERPETUU* node uses a constant pre-determined maximum wait time ($T_{WaitMax}$) to determine the actual wait time (T_{Wait}), which is always a fraction of $T_{WaitMax}$. The T_{Wait} fraction (f) is computed using the equation:

$$f = \left(\frac{Qb}{Qmax} + \frac{Pin}{Pmax} + \frac{Ph}{Pmax} \right) / 3 \qquad (10)$$

where Q_b/Q_{max} is the present battery charge divided by the full battery capacity (i.e. SOC), P_{in}/P_{max} is the present energy being harvested from the solar panel divided by the maximum energy that can be harvested from the solar panel, and P_h/P_{max} is the predicted energy harvested in the next hour divided by the maximum energy that can be harvested from the solar panel.

The calculation for the T_{Wait} fraction (*f*) is simpler during nighttime, since there is no energy from the solar panel to consider but only from the battery. At nighttime, the T_{Wait} fraction becomes:

$$f = \frac{Qb}{Qmax} \qquad (11)$$

The fraction *f* captures the present energy state of the system, in terms of how much energy is available for use by the Gumstix. The intuition behind using this fraction is the following: the time the system should keep the Gumstix on to wait for incoming clients should be proportional to the sum of the system energy reserve. We know when the system has *near* maximum available energy, $f = 1$ and therefore $T_{wait} = T_{WaitMax}$, and this is where our reasoning begins. All energy sources are considered in the calculation of *f*. As the available energy from any of the sources decreases, *f* should logically decrease as well. This way, the amount of time the Gumstix waits for additional clients remains proportional to the current energy available to the Gumstix. The only question is: how long should $T_{WaitMax}$ be? The answer will always determine the value of *f* and thereby the value of T_{wait}. The value of $T_{WaitMax}$ is a critical parameter of our overall system. It has a great effect on the total amount of energy drawn by the system as well as on the number of clients served. Since it affects system performance to such a great extent, we performed experiments varying the value of $T_{WaitMax}$ in order to find an optimal solution. We use different values for $T_{WaitMax}$ during the day and during the night. As it turns out, a longer $T_{WaitMax}$ at nighttime produces better results. A more detailed effect T_{Wait} has on system performance is laid out in Chapter VI.

V. IMPLEMENTATION

The *PERPETUU* node is implemented on a custom designed power supply PCB which houses the PIC micro-controller, energy harvester/battery charger, light sensor, and interfaces to the three radios. The system can be powered using a solar panel, Li-ion batteries, an external power supply, or any combination of those. Some key features of the board include: 1) hardware implemented temperature-controlled maximum power point tracking, 2) the ability to power the system directly from the solar panel when the battery is near full, 3) the ability to make fine-grained power measurements, 4) power gating certain subsections of the board to reduce power leakage, 5) the ability to survive a wide range of temperatures, 6) a weather-resistant utility box to protect the susceptible components from outdoor conditions.

Hardware Components – Description

Figure 6 shows the prominent components which comprise the *PERPETUU* node. A detailed description of each component is given below. Other system components not listed include an EEPROM module, an RS232 serial module, current-sense amplifiers, voltage dividers, and many passive devices.

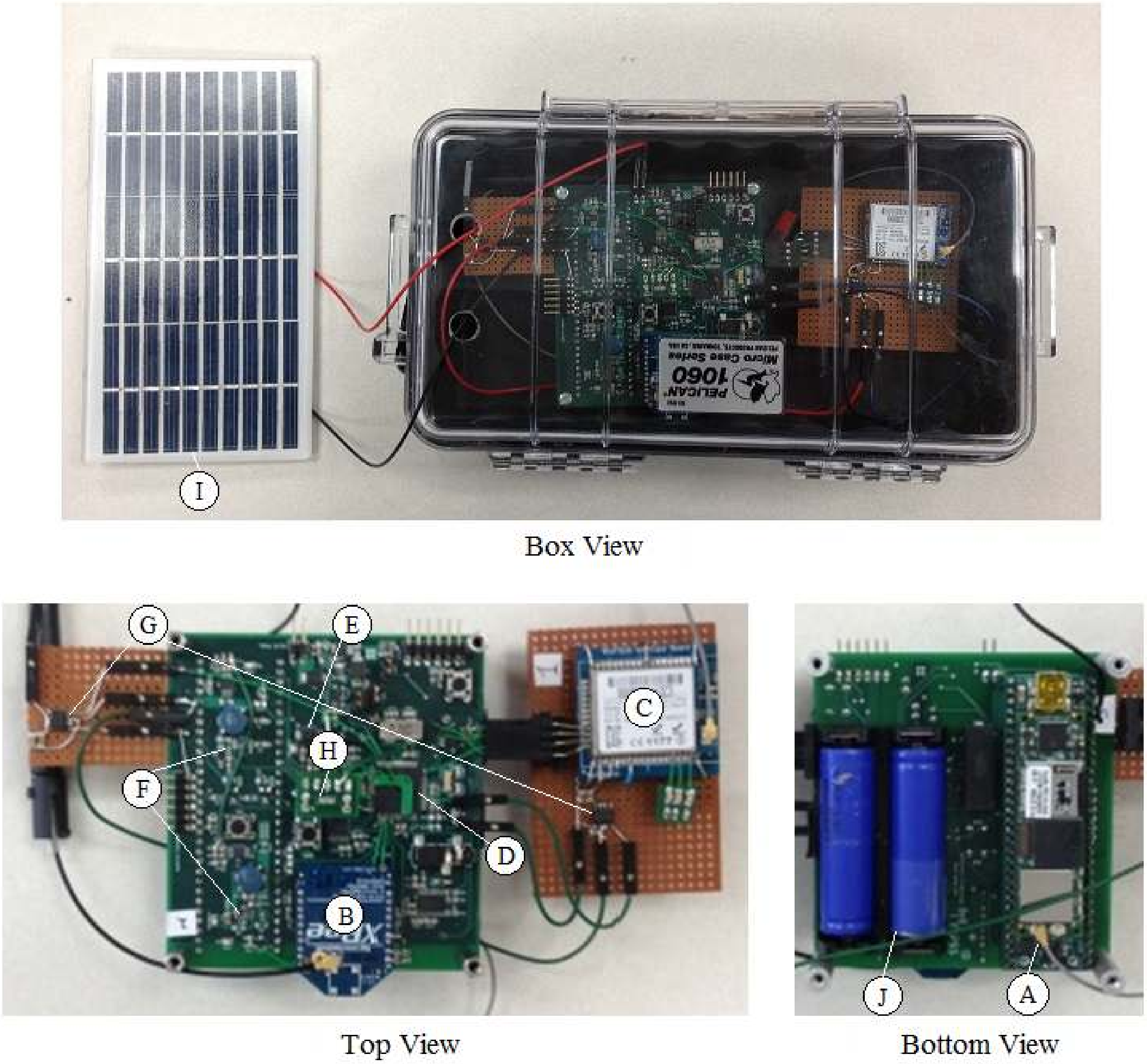


Figure 6: *Photo taken by: Adam Matthews, 8/26/2013, Mobile and Sensor Systems Lab at the University of Arkansas. PERPETUU* node prototype. Top and bottom views are shown, along with the node inside the weather-resistant utility box. The main components are labeled A-J:

A: Gumstix module	*F: DC/DC power converters*
B: XBee Module	*G: Analog switches*
C: WizFi Module	*H: Light sensor*
D: PIC micro-controller	*I: Solar panel*
E: Harvester/Batter charger	*J: Li-ion batteries*

A) Gumstix module: The Gumstix Overo Air computer-on-module is equipped with a 600MHz ARM Cortex-A8 OMAP3503 processor, and acts as our Wi-Fi server via the 802.11b/g protocol.

It communicates with the microprocessor over a UART interface. Having a micro-SD card which hosts the Linux kernel, the Gumstix module stores and serves all GIS data in a multi-resolution scheme representative of the surrounding terrain.

B) XBee module: For intercommunication between the mesh nodes, we use the XBee PRO DigiMesh 2.4 RF module. Programmable through an AT command mode interface, the XBee allows for synchronous sleep cycles among the mesh nodes and for automatic disbursement of changes in any map layer on any node.

C) WizFi module: For our system we use the WizFi210 low-power Wi-Fi module to detect Wi-Fi clients. The low power consumption of the WizFi210 allows the *PERPETUU* node to maximize its availability to clients, while the high-power Gumstix map server wakes up only after a client is detected. Programmable through an AT command mode interface, it provides a standard 802.11b/g access point and has a boot time 20msec.

D) PIC micro-controller: The PIC18F87K22 is a high-performance and ultra-low-power MCU. It hosts the energy profiler, the Gumstix wakeup control module, communicates with all other devices on the board through its MSSP and ESUART modules, and takes all ADC measurements. This micro-controller has several low-power features and can wake from sleep mode by means of a timer using its real-time clock and calendar or externally from its peripheral devices.

E) Energy harvester/battery charger: The mesh board uses the LT3652, a step-down regulating battery charger, as a solar energy harvester. It charges the Li-ion batteries which are used to power the system when there is not enough sunlight available to do so. It has a programmable output voltage, a programmable current limitation, trickle charge, low-current standby mode, and an efficiency of typically around 85% (although with maximum power point tracking the efficiency is around 98%). It operates over 5-32V input range, making it accommodating for many different sizes of solar panels.

F) Power converters: The mesh board uses two high-efficiency DC/DC step-down regulators with fixed 3.3V outputs: the LTC3631 and the LT3971. The mesh board has a separate regulator for each of the two hardware tiers, in order to optimize the efficiency of the regulators which provide for two distinct ranges of power draw. The LTC3631 powers the entire low-tier hardware circuitry, including the micro-controller and the XBee module. This regulator was chosen to provide maximum efficiency for the power demands of our low-tier subsystem. It has a maximum load current of 100mA and an efficiency around 85% for our application. The LT3971 powers the Gumstix module and WizFi module. This regulator can have a maximum load current up 1.2A with an efficiency between 80-85% for our application. Both regulators have a wide input range, suitable for the wide voltage swings incurred by the solar panels and batteries.

G) Analog switches: There are two TPS2033 high-side MOSFET power distribution switches used in our system. These switches allow the Gumstix and WizFi modules to be powered from the same regulator, allowing one to be powered on while the other is powered off, as they are controlled by the PIC.

H) Light sensor: We use the BH1710 light sensor; a 16-bit digital ambient light sensor with an I^2C bus interface. The light sensor is used to assess the potential energy available from the solar panel. As a fast, low power, high precision sensor, it responds approximately to the same spectrum as the human eye and has three modes of resolution.

I) *Solar panel:* For many of our experiments and all of our simulations, we use a SolMaxx 15V/240mA OEM solar panel. Containing 30 crystalline solar cells, it is 210mm x 184mm in size. Some I-V traces showing profile information for this solar panel is shown in Figure 7. The overlapping of traces and their inconsistent shapes show the effects temperature has on the solar panel. Chosen because it is moderately sized and relatively inexpensive, this solar panel is capable of outputting about 3.84W under normal sunny conditions.

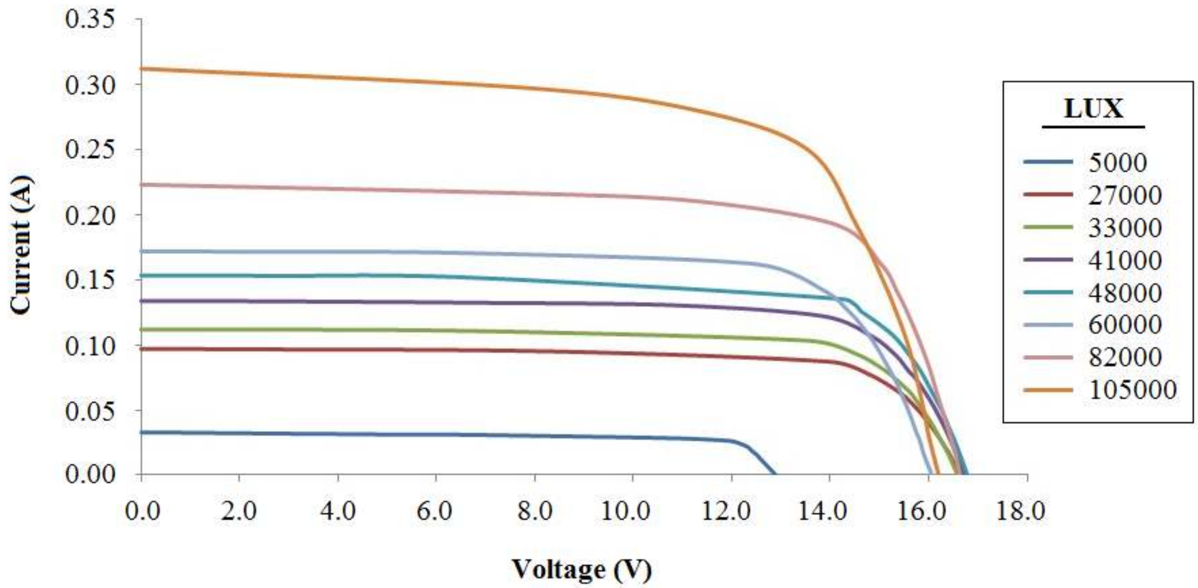


Figure 7: I-V curves for SolMaxx 240mA/15V OEM solar panel

J) Li-ion batteries: Our system uses two AA-sized Li-ion batteries. A lithium-based battery chemistry is preferred for our system, for reasons described later in this chapter. For our experiments, we have used batteries with capacities of 750mAh, 1200mAh, and 2000mAh. The capacity of the batteries provides a tradeoff similar to that of the solar panel, and it should logically be sized to store the harvested energy from the given solar panel.

Power Supply PCB Description

The power supply board is a custom designed 4-layer PCB, manufactured and assembled by Advanced Circuits (4pcb.com). It is 3.68in wide x 3.15in long. In addition to the onboard components already described, the board is equipped with a serial console interface for debugging and a serial in-circuit debugging interface for programming and debugging the PIC micro-controller. The board also has headers fit for each of the three radios, providing power supply and serial interface to the PIC for each radio. By using power gating and other techniques heretofore described, the supply board is a versatile and efficient vessel for accomplishing our design goals. The complete circuit schematic and design layout for the *PERPETUU* node are shown in Appendices A and B.

We call our custom designed PCB the *supply board* because it provides power to every system component and allows the source of power to be automatically selected and seamlessly routed to the power converters which power every system device. This is accomplished by the means of the solar energy harvester/battery charger device (LT3652), which also provides the temperature-controlled maximum power point tracking capability given its supporting circuitry. When there is ample energy from the solar panel, the LT3652 provides a constant float voltage and programmable current with which to power the batteries, and if there is any excess power from the solar panel it is automatically used to power the system. Along with the use of efficient power converters and the power gating technique, it is a primary component in making the *PERPETUU* circuit board energy efficient and reliable.

Solar Panel and Battery Selection

Selecting the appropriate solar panel and battery(s) for a wireless mesh node such as *PERPETUU* can be a non-trivial task, considering all the different options available. The goal is to select a solar panel/battery combination that: 1) satisfies the energy needs of the system given the expected traffic flow, and 2) can survive the unpredictable elements of nature.

The first step is to determine the battery *chemistry* to be used in the system. A lithium-based battery chemistry is the only logical choice for the *PERPETUU* node for a variety of reasons [9], [10]:

- Li-based batteries have a high energy density, making them well-suited for small portable systems.
- Li-based batteries exhibit lower self-discharge or *leakage* as compared with Ni-based batteries.
- Li-based batteries typically allow for 500-1000 charge cycles, while lead acid and Ni-based batteries typically only provide about 200-500 charge cycles.
- While lead acid and Ni-based batteries require deep charge/discharge cycles, Li-based batteries have a prolonged life when smaller, more frequent cycles are used. The *PERPETUU* node frequently switches between charging and discharging states.
- The state-of-charge on Li-based batteries can be estimated simply by measuring the open-circuit voltage across the battery when it is at rest. No other battery chemistry has a simple method such as this for measuring SOC.

Li-ion was chosen as opposed to Li-polymer because Li-ion batteries have a protective encasing which makes them much more durable and safe than Li-polymer batteries. There are distinctions between the different chemistries of non-polymer Li-based batteries, such as Li-ion, Li-Cobalt, Li-Manganese, etc. While some Li-based chemistries, such as Li-Manganese, allow for very high load currents, which is not needed for *PERPETUU*, Li-ion is the most cost-effective choice and totally sufficient for our system.

To select an appropriately sized solar panel for a system such as the *PERPETUU* node, we typically would first determine the expected energy supply needs of the system and select a solar panel capable of harvesting that much solar energy. The major problem with this first step is we do not know a priori how much traffic to expect at any one node during an emergency. So essentially, the first step becomes selecting the size of the solar panel based on the portability requirements and monetary budget of each node, so long as the panel can deliver enough power to serve a *sufficient* number of clients. Our medium-sized solar panel and batteries are capable of serving 1000 or more clients on typical days.

As mentioned, our solar panel is a 15V/240mA crystalline solar panel capable of producing up to about 3.84W of DC power. Based on the real solar energy data from [44], we can calculate the average daily total power we can expect from our solar panel – 20.48Whr per day. Figure 8 shows the average expected power *hourly* from the solar panel, where 20.48Whr is the approximate area under the curve. This is the average amount of energy we can expect to be available to our system daily.

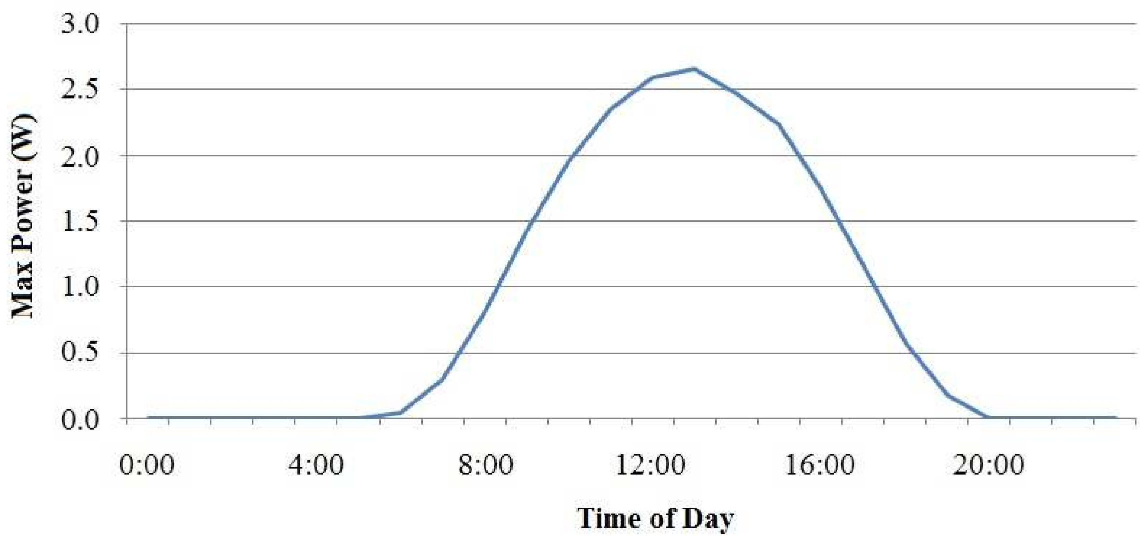


Figure 8: Average hourly DC power delivered by the solar panel chosen for our system. This is based on the actual solar radiation records from the NSRDB [44] and the measured power profile of the given solar panel.

Note this is the power incoming directly from the solar panel, and due to inefficiencies of the circuits and components on the board (primarily the DC-DC power converters and harvesting circuitry), some of this energy will be lost and unavailable for use by system components such as radios. This is however the total power budget of the system (on average). The panel must be used during the day to power the system and charge the battery to a sufficient state such that the battery can power the system throughout the night. Figure 9 shows a visual of the on-average power state progression, for the solar panel and battery(s), over a period of two days.

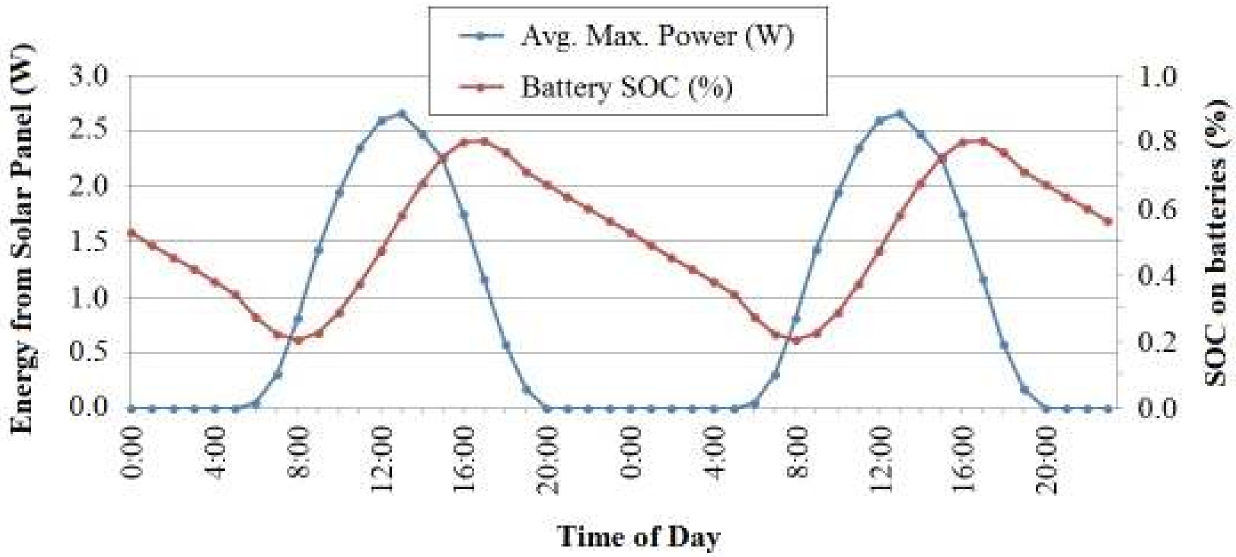


Figure 9: Average maximum power available from the solar panel and the corresponding state-of-charge (SOC) on the batteries over a two-day period.

Figure 9 shows how the residual charge on the batteries fluctuates daily, by charging during the day and discharging by powering the system during the night, and thus illustrating how the system will function near perpetually. Note the SOC *swing* on the battery is from 20% to 90%, as previously described. To determine the size of the batteries needed to power the system throughout the night, we first take the total amount of energy from the solar panel (20.84Whr) and distribute it evenly across the 24-hour period: 20.84/24 ≈ 0.8533Whr/hr. Since there is approximately ten hours each day during which the system depends entirely on energy stored in the batteries, the average amount of energy required to be supplied by the batteries each night is: (0.8533Whr/hr) · (10hr) ≈ 8.533Whr.

After we determine the amount of energy required to be sourced by the batteries (8.533Whr), we must next determine the necessary *capacity* of the batteries. The capacity of the batteries must be substantially higher than 8.533Whr, since we only utilize 70% of the capacity (i.e. the battery starts at 90% and discharges down to 20%). Therefore, the capacity of the batteries must be: 8.533Whr/0.7 = 12.19Whr. We use a two-battery system, so the capacity of each battery must be: 12.19Whr/2 = 6.095Whr. Since battery capacity is usually described in amp-hours (Ahr), we convert the capacity from Whr to Ah. Since the nominal voltage for a Li-ion battery is 3.7V, we calculate the capacity based on $P=IV$: 6.095Whr/3.7V ≈ 1.647Ahr. This is the ideal battery size for the *PERPETUU* node while using the given solar panel. However, since batteries on the market are typically available in 0.5Ahr increments, either 1.5Ahr or 2.0Ahr batteries must be chosen. To err on the side of *more than sufficient* as opposed to *not enough*, we have chosen 2.0Ahr as the ideal Li-ion battery for our system.

VI. EVALUATION

While the foremost goal of *PERPETUU* is to reliably serve the maximum number of incoming Wi-Fi clients during an emergency, it is also desirable to conserve energy as much as possible in order to serve users later and perhaps to prolong the life of each node. We examine a set of micro-benchmarks on energy consumption and latency, as well as present some key attributes of system performance resulting from our evaluation. At the onset of the evaluation, we were focused on answering three key questions: 1) Can a *PERPETUU* node carry out its necessary tasks over a long period of time using energy harvested from a medium-sized solar panel? 2) How many clients can a node serve based on the amount of energy available? 3) To what extent does the *wait time* and *energy prediction* improve *PERPETUU*'s performance? These and other questions will be answered in this chapter.

Experimental Methodology

Evaluating a solar-powered system such as *PERPETUU* involves deploying the mesh outdoors for a long period of time, and testing the system using real workloads. Unfortunately, it is not practical to emulate an actual emergency scenario given the large amount of traffic which would incur. Also, the length of time required to make multiple runs in order to evaluate the system makes actual deployment impractical for evaluation purposes. Therefore, we evaluate *PERPETUU* using traced-based simulations. Written in C++, the simulation emulates all system components in terms of timing and power consumption, under conditions which may presumably occur during an emergency situation.

The available solar energy input for the simulation is real data which comes from a record in the National Solar Radiation Data Base (NSRDB) in which solar energy measurements were taken in southwest Fayetteville, AR [44]. Each simulation runs over the course of a year (2009), taking the available solar energy data from the NSRDB. We are typically interested in the average performance of the mesh node for each day over the entire year, unless we are looking specifically at seasonal changes.

It is assumed, shortly after a disaster occurs, there will be a spike in the number of clients attempting to log on to the mesh. Since a disaster could occur at any time during a 24-hour period, and the number of clients served can change depending on when the peak of client arrival rate occurs. For any given day, we run the simulation six times a day – varying the peak arrival times for each of the six iterations. Since a disaster could occur at any time of day, we are not solely interested in cases where the system would perform best – during midday. We observe the average of these six iterations of the simulation. The six possible peak arrival times are distributed evenly across the day, as shown in Figure 10.

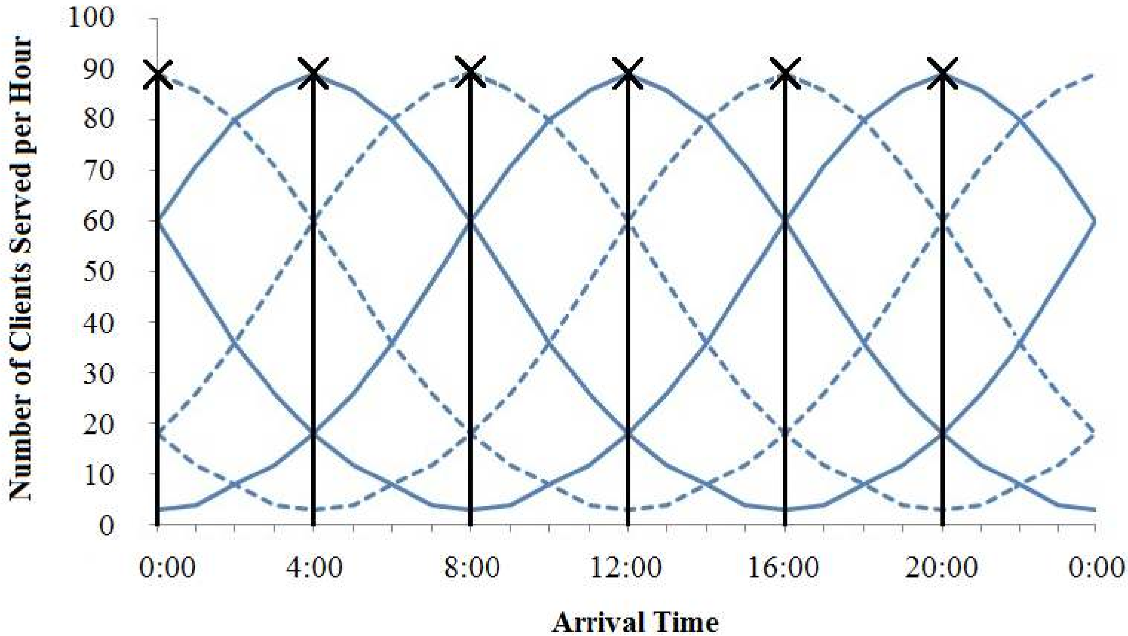


Figure 10: Six different curves representing Wi-Fi client arrival rates. The area under each curve shown is approximately equal to 1000 clients arriving per day. Each curve represents a normal distribution of clients, scaled up from a normal curve having a mean=0, sum=1, and standard deviation=4.5.

We assume a Poisson distribution of client arrivals which has been shown to be an accurate model for arrival processes. Therefore, each of the six client arrival curves in Figure 10 follows an exponential distribution, and as mentioned an average of the six trial runs is taken to evaluate the performance during each day. This simulation of six runs over each day is in turn ran over the course of an entire year (2009), using the available solar energy data from the NSRDB. The advantages of using our trace-based simulation approach has the advantage of enabling multiple runs of controlled experimentation, as well helping us to evaluate the system over a long period of time using realistic data.

Micro-Benchmarks

Here, we present micro-benchmarks for latency and energy consumption for different subsystems of the *PERPETUU* node, over a range of possible power states of the node. Table I shows the average power consumption of different components of *PERPETUU* and Figure 11 illustrates a single run of the system, from the node waking up to the Gumstix transmitting map data, as different subsystems are switched on.

Subsystem	Power (W)
PIC (sleep)	0.005
PIC (active)	0.014
PIC (taking measurements)	0.020
XBee RX/idle	0.241
XBee TX	1.0
Wi-Fi (low power) TX	0.486
Gumstix (bootup)	1.270
Gumstix + Wi-Fi (idle)	1.560
Gumstix + Wi-Fi (transmit)	1.700

Table 1: Power consumption of different *PERPETUU* subsystems.

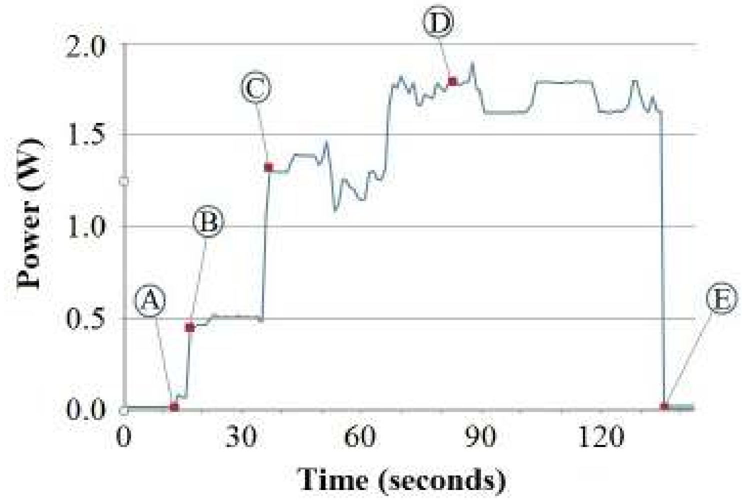


Figure 11: Power trace for a run of the *PERPETUU* node. A denotes the time when the PIC is on, at time B the low power Wi-Fi module is switched on, at time C the Gumstix is switched on, at time D the Gumstix has finished booting, and at time E the Gumstix has served all maps and is switched off

There are several conclusions we can draw from these figures. First, the system provides a power range that spans four orders of magnitude and spans two orders of magnitude in computational power. This illustrates how a multi-tiered approach can provide high computational capability when required while allowing for *nearly* always-on operation at low-power. Second, the results show two bottlenecks in the system. The boot time of the Gumstix module is high, close to 50 seconds and the time to configure the low-power WizFi module is ten seconds. Hence, the cost of booting up these two devices is reduced by serving as many users as possible when the Gumstix is on, showing the need for the adaptive wait time (T_{Wait}) discussed in Chapter IV. Third, the XBee radio used to transfer data across nodes in the mesh is high and consumes close to 1W. Hence, caching data at the micro-controller and transmitting data only when there is a sufficiently large reserve of energy is warranted.

System Performance

Next we perform a set of experiments to determine whether the *PERPETUU* node can serve a large percentage of survivors while maintaining near-perpetual operation. Recall that for all trace-based simulations, unless otherwise noted, we use real solar luminance traces from the NSRDB and a synthetic trace modeled by a Poisson arrival of clients. Figure 12 shows the battery charge of a *PERPETUU* node when the system runs for a period of five days and the rate of client arrival is 1440 per day. As expected, the amount of energy harvested peaks around noon and drops to zero in the evening. From this graph it is clear that the battery charge in our system stays between 20% and 90%, hence, the system avoids a full battery and an empty battery. The result also shows that the system can adapt to energy harvested—it accumulates energy when there is an abundance of energy available from the solar panel, and uses the accumulated energy to serve clients during the night time.

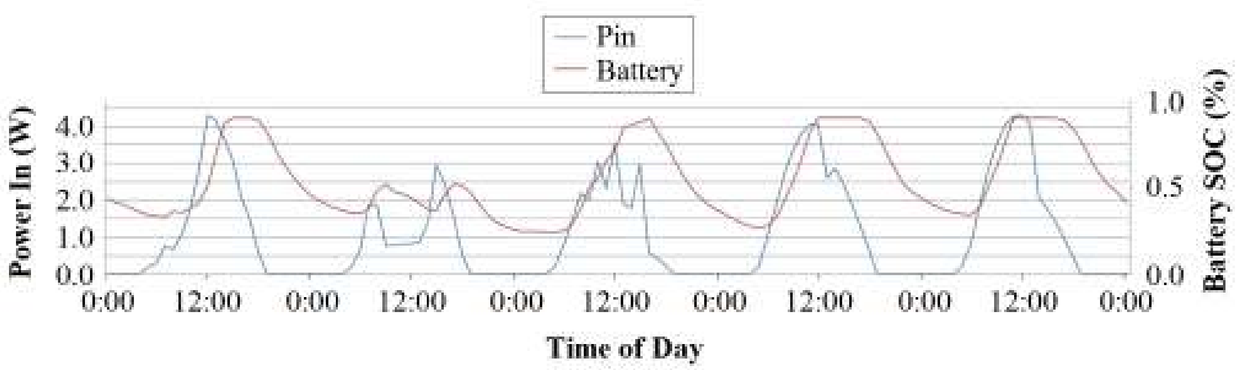


Figure 12: Power generated by the solar panel and residual state-of-charge on the Li-ion batteries, over a period of five days.

Figure 13 (a) presents the average number of clients served by the node when the mean arrival rate of clients is varied. The averages are calculated for the summer and winter season over the entire year, and the errors bars correspond to standard deviation across days. The result shows that the percentage of survivors served during a day is lower during the winter when the amount of energy harvested is low. While the percentage of users served is reduced with an increase in the client arrival rate, we note that in a real deployment, several *PERPETUU* nodes would be deployed in close proximity and the task of serving clients would be distributed across nodes. The system can, however, serve nearly 100% of the users if there are 20 clients or fewer at a node every hour.

We next evaluate the value of our prediction algorithm used to predict the amount of energy that will be harvested in the next hour. Recall that the wakeup algorithm is based on the WCMA algorithm and uses the prediction information to determine whether the Gumstix module should be woken up to serve users. We can see the efficacy of using prediction in the context of the entire system from Figure 13 (b), which shows the number of users served with and without prediction. We only use the number of users served during the day time for this result, when the predictions are useful. The figure shows that using prediction helps improve the number of users served when the number of clients increase, which can be attributed to the fact that more accurate prediction leads to more accurate determination of wakeup intervals for the low-power Wi-Fi module and Gumstix subsystem.

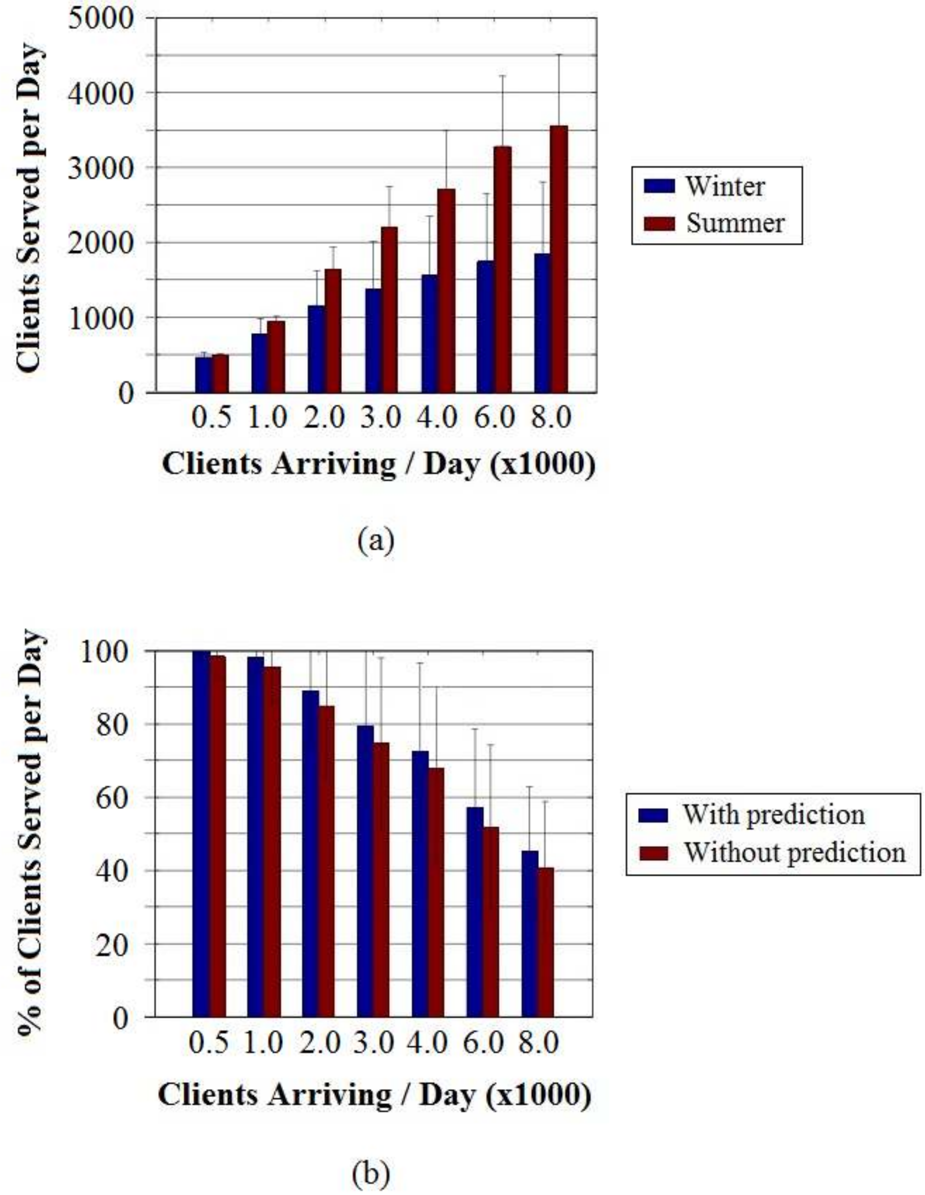


Figure 13: (a) Percentage of clients served during the summer and winter seasons for different client arrival rates. (b) Improvement in the number of clients served when energy prediction is used.

We next evaluate the effect of varying the Gumstix maximum wait time ($T_{WaitMax}$) has on the number of clients served. As described in Chapter IV, here $T_{WaitMax}$ refers to the maximum time the Gumstix can remain on after serving a client in order to wait for more clients to arrive.

We evaluate the effect of varying $T_{WaitMax}$ in Figure 14. The three charts in Figure 14 convey much information regarding the effect of varying $T_{WaitMax}$ has on clients served. In this figure a different day and night wait time is used for each chart. For each chart, the number of incoming clients is varied while showing the number of clients served and the power breakdown for the different Wi-Fi subsystem components.

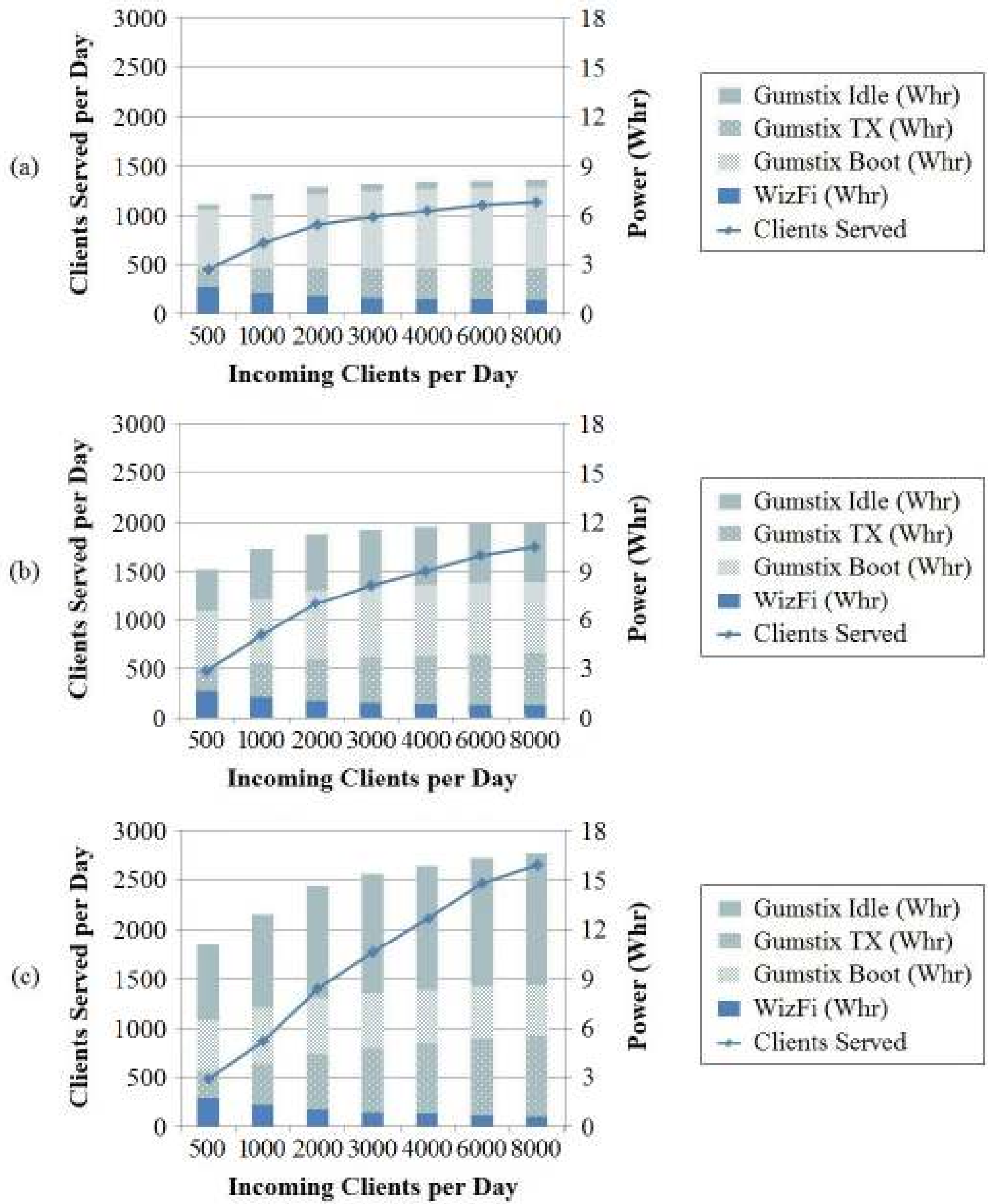


Figure 14: Number of clients per day for three different sets of both night and day values of $T_{WaitMax}$. (a) day = 0 seconds, night = 0 seconds (b) day = 60 seconds, night = 120 seconds (c) day = 90 seconds, night = 180 seconds.

A tradeoff can be extrapolated from the information in Figure 14 between the number of clients served and varying the maximum Gumstix wait times (and thereby varying the power consumption). The ideal night and day $T_{WaitMax}$ values are those that serve the maximum number of clients while consuming the minimum amount of energy. These ideal values for $T_{WaitMax}$ depend on the number of incoming clients per day. For example, at 500 incoming clients per day (or less), day and night $T_{WaitMax}$ should be zero, since this consumes a minimal amount of energy and serves almost as many clients as large values for $T_{WaitMax}$. However, in a scenario where we have 8000 incoming clients per day, using longer wait times (such as 90 and 180 seconds) is justified; although it consumes twice as much energy as using no wait time, it serves twice as many clients. This is a favorable tradeoff, since serving as many users as possible is the primary goal, and prolonging the long-run life of the node is secondary to serving users when an emergency occurs. The predicted number of incoming clients, and therefore the value of $T_{WaitMax}$, could be estimated based on the population density in the area surrounding each mesh node. This set of experiments varying $T_{WaitMax}$ serves as an illustration of how choosing an ideal value for $T_{WaitMax}$ based on population can serve to help maximize system throughput. Figure 14 also shows that the useful energy spent by the system (Gumstix TX) to disseminate maps increases with the number of users served — this shows that the system optimizations help in increasing the amount of useful work performed by the system when the workload increases.

The optimization of parameters such as $T_{WaitMax}$ (and others described in Chapter IV) could increase the system performance of *PERPETUU* even further. There are a countless number of different ways to consider the different parameters and their relationships to one another. Constant or variable weights could be assigned to different parameters, each parameter in equation (10) for example, as could their relationship be modified in order to possibly improve overall performance. Given the vast number of different ways to consider these variables, a near optimal solution could emerge as a result of an *evolutionary programming* approach to configuring these parameters.

VII. FUTURE WORK

Our ultimate objective for this project is to save lives, by developing the self-sufficient infrastructure needed to provide survivors with proper post-disaster relief information. Such a problem requires an extensive knowledge base, innovation, and expertise in many different areas of computing. Given the potential societal impact of the project, the problem is worthy of continuation. Due to the broad scope of the problem(s), some of the barriers standing in between our current system and an ideal emergency mesh have yet to be fully worked out. The foreseeable future work in this area seems be trending toward making every smart device an actual mesh node, as part of a giant "disaster proof" mesh, as is the goal of the Serval Mesh Project developers. Virtually all of the research presented in this paper, as well as future work proposed in this chapter, can be applied to such a crowd-based mesh. In the future, we will still need solar-driven low-power nodes, both stationary and mobile. In the remainder of this chapter we discuss some important future work, that both may need to be fulfilled as part of a continuation of *PERPETUU* as well as applied to other related applications.

Node Placement

Terrain and demographically aware node placement is critical to the performance and overall reliability of the *PERPETUU* mesh. Node placement must ensure redundancy and fault tolerance, in order for self-stabilization to even be possible. Node placement within a network topology has been a popular area of study for many years. Most contemporary methods use a graph topology generation algorithm that optimizes connectivity and reliability metrics [65], [66], and subsequently use a heuristic to approximate the problem of realizing a graph in a two-dimensional plane – an NP-Hard problem [45].

Topology design algorithms do not consider important geospatial elements such as terrain and elevation which have a substantial impact on a solar-powered mesh performance. However, the amount of energy scavenged and the amount of energy expended (in radio transmissions) depend heavily on the elevation and terrain. Algorithms that ignore these geospatial properties may yield node placement that is ineffective. Population demographics should also be considered in a node placement algorithm as well. Heavily populated areas should require more nodes per unit area in order to distribute the workload among the nodes evenly, enhance availability, and prolong the expected life of nodes.

Self-Stabilizing Data Distribution

In order to ensure acceptable levels of availability to survivors and energy-efficiency, a well thought-out data distribution and management scheme must be in place. Here, we will address three major data management issues within our system: (1) initial static data distribution, i.e. what information do nodes have pre-programmed before deployment, (2) data redistribution, i.e. how is data automatically redistributed in the event of node failures, and (3) alert percolation, i.e. how alerts are distributed to mesh nodes *from* emergency personnel. To address all three of these problems, we propose a multi-resolution data distribution and redistribution scheme.

The static data distribution plan is to store on each node the high-resolution data for the grid to which it belongs, as well as its eight adjacent grids. While each node lacks the memory to store high-definition data for the entire mesh area, this amount of redundancy should be sufficient for self-sustainability. Redistributing data in the event of a node failure involves two steps: fault detection and data replication. Periodically, each node wakes up to transmit a "heartbeat" packet to its neighboring nodes using the XBee radio. If a node does not receive its neighbor's heartbeat for two consecutive intervals, it assumes its neighbor is dead. At this point, the data on the failed node is redistributed among the surviving adjacent nodes, using an energy-aware process that selects nodes with the highest energy level to participate in the transmission. Alert information percolation must be handled differently, as it may often be delay-intolerant information such as a warning of impending disaster. To alleviate this problem, if an alert is large (requiring subsystem-2), the node with the highest amount of energy within the grid should download the alert and distribute it to the other nodes in the grid when they have energy available.

Solar Tracking and Concentration

Another method for increasing the power throughput of a solar panel, often confused with MPPT, is called *solar tracking*. Solar tracking is accomplished by physically moving a solar panel so that the *angle-of-incidence* of light acting upon the surface of the panel is 0°, that is, it is always facing in the direction of the light source. Solar tracking alone can increase the efficiency of a solar panel by up to 30% [40], and should be considered an option for increasing the output power for a solar-powered system. A solar tracking system may be infeasible for a *PERPETUU* node, as it is usually reserved for larger solar panels and solar panel grids. If we want to increase the power from a solar panel by 30% we can just buy a solar panel that is 30% larger, since it is hard to imagine a tracking system that would cost less than the difference in price of a 30% larger panel.

However, in solar tracking or non-solar tracking systems, the solar panels can also be either *concentrating* or *non-concentrating*. A concentrated solar panel uses mirrors to reflect additional light onto the surface of the solar cells, and can increase the amount of light incident upon the panel by several or even hundreds fold. However, the increased light also increases the temperature significantly, which lowers the efficiency and can shorten the life of the panel. Concentrated solar panels require a cooling system when concentration is high, which can be very costly. Although if the mesh is only used for a brief time (during an emergency), the life of the solar panel may not be much of an issue, especially if concentration is not extremely high.

Solar tracking and concentration essentially offer the benefit of saving money, by allowing for the use of smaller less expensive solar panels. Solar tracking, and concentration, present a trade-off between their cost, the energy they gain, and the energy they expend. The cost of solar tracking, as well as the cost of using concentrated panels, should cost less than the price of simply using a larger solar panel which would have the same effect – gaining power.

Low-Power Hardware Specialization

While this method may not be conducive to rapid prototyping, hardware specialization can dramatically reduce overhead and power consumption in later stages of development. A WSN node architecture based on micro-tasking, as in [38], can allow for ultra-low power consumption. Here, the authors have developed an architecture based on *micro-tasks* that are activated on an event-driven basis, where each micro-task is dedicated to a specific system task (sensing, MAC, data processing, etc.). This hardware specialization, which reduces dynamic power consumption, is used in conjunction with *power gating*, which also reduces static power consumption.

Messaging Services

While providing relief information to survivors is the purpose of the emergency mesh, as it could save lives, there may be other uses for the mesh as well. Designers in [54] recently described a do-it-yourself project building a lightweight version of a Wikipedia server and messaging service – all solar powered. This server hosts a stripped-down version of Wikipedia, having roughly 1/1000th the information of Wikipedia, as well as messaging service that allows users to leave messages for one another. Our emergency mesh could be used to host a simple text-based messaging tool, similar to [54], where users can leave each other messages such as "I am stranded at this location: …", or "John, meet us at Dad's house" (it almost does this now).

After receiving potentially life-saving information after a disaster, the next most important thing to most people intuitively would be correspondence with their family and friends, which the *PERPETUU* node could in theory provide. An emergency text-message service such as this would need to be as simple as possible in order to conserve energy, creating many types of complications. Security and privacy concerns may be a high priority in such a system – issues such as how to authenticate users for viewing their personal messages. Then again, privacy may not be a high priority during an emergency situation, as we have seen with the Patriot Act for instance, post-911.

Third World Countries

In many third world countries, or countries lacking infrastructure and internet access in many locations, may have many uses for a solar-powered mesh such as *PERPETUU*. For example, in such areas, disease often appears seemingly out of nowhere and spreads rapidly. Without any known infection signs to look for, or notification that a virus is spreading at all, innocent people often fall victim to innocuous diseases under such conditions [88]. A system such as *PERPETUU* could provide people in these regions a means of communication, acting as a warning system for people within a community or even between communities. Such a system could also assist doctors who are treating spreading illnesses, by allowing the doctors to communicate with each other regarding signs of infections, preventative measures, and treatments. Again, these areas have no developed telephone or internet infrastructure – something often taken for granted in the United States. A system such as *PERPETUU* has potential to be of great use in such places, and possible could save many lives.

VIII. CONCLUSION

We have presented *PERPETUU*, a solar-powered GIS micro-server mesh node that uses a multi-tiered hardware and software architecture for serving information-rich maps to survivors of disasters. *PERPETUU* combines a computationally weak but low-power micro-controller device with a computationally capable Gumstix platform to explore tradeoffs between energy consumed, system availability, and computational capability. Through the use of energy prediction, power profiling capabilities, and an intelligent wakeup controller, the system can balance energy harvested from solar panels with energy consumed by different subsystems. Our evaluation of *PERPETUU* shows the system can last *near* perpetually while serving a large number of survivors during an emergency.

Research Contributions Made

As stated in greater detail in Chapter II, our research into the design, implementation and evaluation of *PERPETUU* has led to four notable contributions:

(1) A multi-tiered, multi-radio hardware architecture for a unique solar-panel driven system, capable of balancing energy harvested with energy expended in order to provide high availability to clients at a relatively low power cost.

(2) A set of several intelligent software decision-making techniques which virtualize system resources at the micro-controller.

(3) A fully functional and evaluated *PERPETUU* node prototype implementation of the multi-tiered architecture.

(4) An improvement on an existing energy prediction algorithm based on *Exponentially Weighted Moving Average*.

Experience and Skills Acquired

In addition to the contributions made, the author of this paper has gained a tremendous amount of experience and honed or acquired many useful technical skills in conducting the research described. Listed below are some of the more predominant skills gained, all of which should prove to be valuable in a computer engineering career. The list is incomplete however, since many of the skills that were gained are intangible and cannot be specified.

- Design and layout techniques for printed circuit boards using CadSoft EAGLE™ PCB design software
- Generate and analyze Gerber files, and submit them to PCB fabricators for production
- Use hand-soldering methods for assembling and debugging PCB's
- Read and interpret datasheets for electronic devices of all kinds
- Proficient usage of power supplies, multi-meters, oscilloscopes, logic analyzers, and data loggers
- Embedded programming and debugging techniques for micro-controllers
- Knowledge of battery chemistries, physics, behaviors, and care
- Knowledge of solar panel compositions and behaviors
- Experience programming wireless modules using AT commands
- Basic knowledge of power converters
- Design of hardware simulators in software
- Linux programming and development skills

REFERENCES

[1] Agarwal, Y., Hodges, S., Chandra, R, Scott, J., Bahl, P., and Gupta, R. Somniloquy: Augmenting Network Interfaces to Reduce PC Energy Usage. NSDI'09, pages 365–380, Berkeley, CA, USA, 2009. USENIX Association.

[2] Ali, M. I., Al-Hashimi, B. M., Recas, J., and Atienza, D. Evaluation and Design Exploration of Solar Harvested-Energy Prediction Algorithm. In *Design, Automation & Test in Europe Conference &Exhibition* (DATE 2010).

[3] Anand, M., Nightingale, E. B., and Flinn, J. Self-Tuning Wireless Network Power Management. In *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking* (MobiCom 2003).

[4] Andrea, D. *Battery Management Systems*. Artech House. 2010.

[5] Armstrong, S., and Hurley, W.G. Investigating the Effectiveness of Maximum Power Point Tracking for a Solar System. In *IEEE Electronics Specialists Conference* (2005).

[6] http://www.articlesnatch.com/Article/Satellite-Phone-Communications-During-A-Natural-Disaster/366336

[7] Banerjee, N., Corner, M. D., and Levine, B. N. An Energy-Efficient Architecture for DTN Throwboxes. In *IEEE Infocom* (May 2007).

[8] Banerjee, N., Sorber, J., Corner, M. D., Rollins, S., and Ganesan, D. Triage: Balancing Energy and Quality of Service in a Microserver. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services* (MobiSys 2007).

[9] http://batterydata.com/

[10] http://batteryuniversity.com.

[11] Becerra, O., Johnson, N., Meier, P., Restrepo, J., and Spagat, M. Natural Disasters, Casualties and Power Laws: A Comparative Analysis with Armed Conflict. In *Annual Meeting of the American Political Science Association* (2006).

[12] Beeby, S., and White, N. *Energy Harvesting for Autonomous Systems*. Artech House. 2010.

[13] Bergonzini, C., Brunelli, D., Benini, L. Algorithms for Harvested Energy Prediction in Batteryless Wireless Sensor Networks. In *3rd International Workshop on Advances in Sensors and Interfaces* (2009).

[14] Chiu, C. –S., and Ouyang, Y. –L. Robust Maximum Power Point Tracking Control of Uncertain Photovoltaic Systems: A Unified T-S Fuzzy Model-Based Approach. In *IEEE Transactions on Control Systems Technology* (2011).

[15] Chou, P. H., Li, D., and Kim, S. Maximizing Efficiency of Solar-Powered Systems by Load Matching. In *Proceedings of the 2004 International Symposium on Low Power Electronics and Design* (ISLPED 2004).

[16] Davis, M. W., Fanney, A. H., and Dougherty, B. P. Measured Versus Predicted Performance of Building Integrated Photovoltaics. http://www.scribd.com/doc/63972888/Davis-Fanney-Dougherty-2003-Measured-Versus-Predicted-Performance-of-Building-Integrated-Photovoltaics (2003).

[17] Dilmaghani, R. B., Rao, R. R. Hybrid Communication Infrastructure and Social Implications for Disaster Management. In *Proceedings of the 40th Hawii International Conference on System Sciences* (HICSS 2007).

[18] http://www.dtic.mil/dtic/tr/fulltext/u2/a488621.pdf

[19] http://eetweb.com/solar.

[20] http://www.electronics-lab.com/articles/Li_Ion_reconstruct/

[21] http://en.wikipedia.org/wiki/2011_T%C5%8Dhoku_earthquake_and_tsunami

[22] http://en.wikipedia.org/wiki/Hurricane_Katrina

[23] http://en.wikipedia.org/wiki/Hurricane_Sandy

[24] Fanney, A. H., Davis, M. W., Dougherty, B. P., King, D. L., Boyson, W. E., and Kratochvil, J. A. Comparison of Photovoltaic Module Performance Measurements. In *Journal of Solar Energy Engineering* (May 2006).

[25] http://www.gumstix.com/

[26] Huang, C. –K., Sakamoto, J. S., Wolfenstine, J., and Surampudi, S. The Limits of Low-Temperature Performance of Li-ion Cells. In *Journal of the Electrochemical Society* (2000).

[27] Hussein, K. H., Muta, I., Hoshino, T., and Osakada, M. Maximum Photovoltaic Power Tracking: An Algorithm for Rapidly Changing Atmospheric Conditions. In *IEEE Proceedings- Generation, Transmission, and Distribution* (1995).

[28] Iqdour, R., Zeroual, A. A Rule-Based Fuzzy Model for the Prediction of Daily Solar Radiation. In *IEEE International Conference on Industrial Technology* (2004).

[29] Jiang, Z., Jin, X., and Zhang, Y. A Weather-Condition Prediction Algorithm for Solar-Powered Wireless Sensor Nodes. In *6th International Conference on Wireless Communications Networking and Mobile Computing* (WiCOM 2010).

[30] Kansal, A., Hsu, J., Zahedi, S., and Srivastava, B. Power Management in Energy Harvesting Sensor Networks. In *ACM Transactions on Embedded Computing Systems* (TECS 2007).

[31] Kassem, A., and Hamad, M. A Microcontroller-Based Multi-Function Solar Tracking System. In *2011 IEEE International Systems Conference* (SysCon 2011).

[32] King, D. L., Kratochvil, J. A., and Boyson, W. E. Measuring Solar Spectral and Angle-of-Incidence Effects on Photovoltaic Modules and Solar Irradiance Sensors. In *26th IEEE Photovoltaic Specialists Conference* (1997).

[33] Koutroulis, E., Kalaitzakis, K., and Voulgaris, N. C. Development of a Microcontroller-Based Photovoltaic Maximum Power Point Tracking Control System. In *IEEE Transactions on Power Electronics* (2011).

[34] Krishna, H. A., Misra, N. K., and Suresh, M. S. Solar Cell as a Capacitive Temperature Sensor. In *IEEE Transactions on Aerospace and Electronic Systems* (2011).

[35] Lee, S. J., Kim, J. H., Lee, J. M., Cho, B. H. The State and Parameter Estimation of a Li-ion Battery Using New OCV-SOC Concept. In *Power Electronics Specialists Conference* (2007).

[36] Liu, B., Duan, S., Liu, F., and Xu, P. Analysis and Improvement of Maximum Power Point Tracking Algorithm Based on Incremental Conductance Method for Photovoltaic Array. In *7th International Conference on Power Electronics and Drive Systems* (PEDS 2007).

[37] http://www.mapnik.org/

[38] Pasha, M.A., Derrien, S., Sentieys, O. A Complete Design-Flow for the Generation of Ultra Low-Power WSN Node Architectures Based on Micro-Tasking. In *ACM Proceedings of the 47th Design Automation Conference* (2010).

[39] http://www.physicscentral.com/explore/action/lithium-1.cfm

[40] Piorno, J. R., Bergonzini, C., Atienza, D., and Rosing, T. S. Prediction and Management in Energy Harvested Wireless Sensor Nodes. In *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology* (Wireless VITAE 2009).

[41] http://www.pveducation.org/

[42] Reicosky, D.C., Winkelman, L.J., Baker, J.M., Baker, D.G. Accuracy of Hourly Air Temperatures Calculated from Daily Minima and Maxima. In *Agricultural and Forest Meteorology* (1989).

[43] Rizk, J., and Chaiko, Y. Solar Tracking System: More Efficient Use of Solar Panels. In *World Academy of Science, Engineering and Technology* (2008).

[44] http://rredc.nrel.gov/solar/old_data/nsrdb/1991-2010/SolarAnywhere/#table.

[45] Rusterholz, T. On the Approximation on Unit Disk Graph Coordinates. Tech. rep., 2003.

[46] http://www.scribd.com/doc/49581110/Battery-Management-Systems-Accurate-State-of-Charge-Indication-for-Battery-Powered-Applications-Philips-Research-Book-Series

[47] Seong-jun Lee, Jong-hoon Kim, Jae-moon Lee, Bo-hyung Cho. The Estimation of the SOC and Capacity for the Li-ion Battery Using Kalman Filter. School of Electrical Engineering and Computer Science, Seoul National Univ.

[48] Sorber, J., Balasubramanian, A., Corner, M.D., Enner, J.R., Qualls, C. TULA: Balancing Energy for Sensing and Communication in a Perpetual Mobile System. In *IEEE Transactions on Mobile Computing* (2013).

[49] Sorber, J., Banerjee, N., Corner, M.D., Rollins, S. Turducken: Hierarchical Power Management for Mobile Devices. Seattle, WA, June 2005.

[50] Sorber, J., Kostadinov, A., Garber, M., Brennan, M., Corner, M. D., and Berger, E. D. Eon: A Language and Runtime System for Perpetual Systems. *In Proc. ACM SenSys* (Syndey, Australia, November 2007).

[51] Spagnol, P., Rossi, S., Savaresi, S.M. Kalman Filter SoC Estimation for Li-ion Batteries. In *IEEE International Conference on Control Applications* (2011).

[52] http://www.sunrisesunset.com/usa/

[53] Tang, X., Mao, X., Lin, J., and Koch, B. Li-ion Battery Parameter Estimation for State of Charge. In *American Control Conference* (ACC 2011).

[54] http://www.techthefuture.com/technology/redditor-builds-solar-powered-wikipedia-server/

[55] Tikhonov, K., and Koch, V. R. Li-ion Battery Electrolytes Designed for a Wide Temperature Range. http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA507154.

[56] Ullah, K.W., Nurminen, J.K. Applicability of Different Models of Burstiness to Energy Consumption Estimation. In *International Symposium on Communication Systems, Networks, and Digital Signal Processing* (CSNDSP 2012).

[57] http://www.wunderground.com/history/?cm_ven=PPCgoogle&cm_cat=genweather&cm_-pla=genweath&cm_ite=historybirthday.

[58] Xiao, W., Elnosh, A., Khadkikar, V., and Zeineldin, H. Overview of Maximum Power Point Tracking Technologies for Photovoltaic Power Systems. In *37th Annual Conference on IEEE Industrial Electronics Society* (IECON 2011).

[59] Xiao, W., Ozog, N., and Dunford, W. G. Topology Study of Photovoltaic Interface for Maximum Power Point Tracking. In *IEEE Transactions on Industrial Engineering* (2007).

[60] Xu, X., Liu, Q., and Zou, Y. A Study on All-Weather Flexible Auto-Tracking Control Strategy of High-Efficiency Solar Concentrating Photovoltaic Power Generation System. In *2nd WRI Global Congress on Intelligent Systems* (GCIS 2010).

[61] Young, S., Balluz, L., and Malilay, J. Natural and Technological Hazardous Material Releases During and After Natural Disasters: A Review. In *Science of Total Environment* (2004).

[62] Yuan, W., and Nahrstedt, K. Energy-Efficient Soft Real-Time CPU Scheduling for Mobile Multimedia Systems. In *Proceedings of the Symposium on Operating Systems Principles* (Bolton Landing, NY, October 2003).

[63] Zhang, H., Salajegheh, M., Fu, K., and Sorber, J. Ekho: Bridging the Gap Between Simulation and Reality in Tiny Energy-Harvesting Sensors. http://sigops.org/sosp/sosp11/workshops/hotpower/09-zhang.pdf.

[64] Zhihang Chen, Shiqi Qiu, Masrur, M.A., Murphey, Y.L. Battery State of Charge Estimation Based on a Combined Model of Extended Kalman Filter and Neural Networks. In *International Joint Conferenceon Digital Object Identifier* (2011).

[65] C, R. J. M. Optimal Networks, Congestion and Braess Paradox. In *2006 Workshop on Interdisciplinary Systems Approach in Performance Evaluation and Design of Computer & Communications Systems* (2006).

[66] Donetti, L., Neri, F., and Munoz, M. A. Optimal network topologies: expanders, cages, Ramanujan Graphs, Entangled Networks and All that. In *Journal of Statistical Mechanics* (2006).

[67] Safie, F.M. Probabilistic Modeling of Solar Power Systems. In *Proceedings of Reliability and Maintainability Symposium*, pp. 425-430 (1989).

[68] Sayegh, A.A., Ghosh, S.C., Todd, T.D. Optimal Node Placement in Hybrid Solar Powered WLAN Mesh Networks. In *Wireless Communications and Networking Conference* (2008).

[69] Vargas, E.J., Sayegh, A.A., Todd, T.D. Shared Infrastructure Power Saving for Solar Powered IEEE 802.11 WLAN Mesh Networks. In *IEEE International Conference on Communications* (2007).

[70] IEEE Computer Society LAN MAN Standards Committee. *IEEE 802.11: Wireless LAN Medium Access Control and Physical Layer Specifications*. IEEE Press (1999).

[71] Badawy, G.H., Sayegh, A.A., Todd, T.D. Energy Aware Provisioning in Solar Powered WLAN Mesh Networks. In *Proceedings of 17th International Conference on Computer Communications and Networks* (2008).

[72] Cai, L.X., Yongkang Liu, Luan, T.H., Xuemin Shen, Mark, J.W., Poor, H.V. Adaptive Resource Management in Sustainable Energy Powered Wireless Mesh Networks. In *Global Telecommunications Conference* (2011).

[73] Velayos, H., Aleo, V., Karlsson, G. Load Balancing in Overlapping Wireless Cells. In *IEEE Conerence on Communications* (2004).

[74] Zefre, M., Todd, T.D. Energy Provisioning in Green Mesh Networks Using Positional Awareness. In *IEEE Transactions on Vehicular Technology* (2012).

[75] http://servalproject.org/

[76] http://www.huffingtonpost.com/2013/07/16/serval-mesh-mobile-network_n_3600560.html

[78] http://www.makeuseof.com/tag/how-to-use-serval-mesh-to-chat-to-other-mobile-phones-without-a-phone-network-android/

[79] http://www.stuff.co.nz/dominion-post/business/8993109/Smartphone-app-could-allow-contact-in-natural-disaster/

[80] http://developer.servalproject.org/dokuwiki/doku.php?id=content:meshextender:-main_page/

[81] http://developer.servalproject.org/dokuwiki/doku.php?id=content:tech:serval_-mesh_for_iphone/

[82] Wenxian Li, Yanmin Zhu, Tian He. WiBee: Building Wi-Fi Radio Map with ZigBee Sensor Networks. In *2012 Proceedings of IEEE INFOCOM* (2012).

[83] Raiss-el-fenni, M., El-Azouzi, R., Menasche, D.S. Optimal Sensing Policies for Smartphones in Hybrid Networks: A POMDP Approach. In *2012 6th International Conference on Performance Evaluation Methodologies and Tools* (2012).

[84] http://www.ushahidi.com/

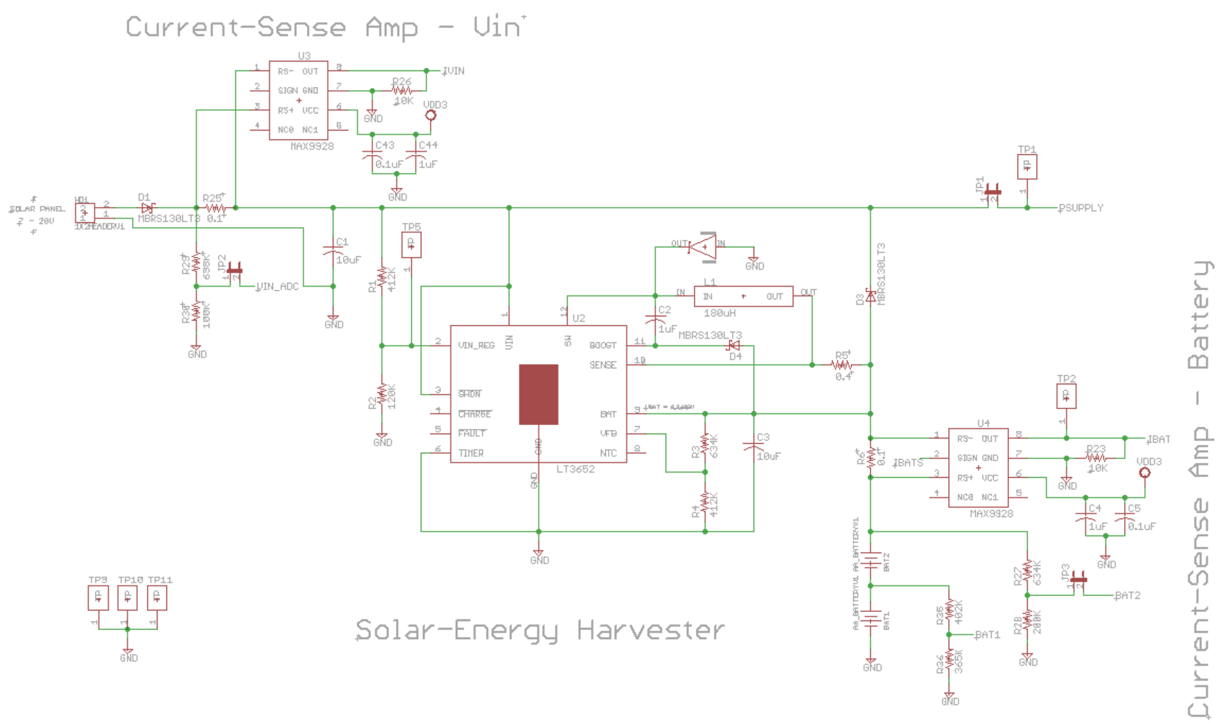
[85] http://news.cnet.com/8301-1035_3-57589546-94/at-t-adds-wireless-emergency-alerts-update-to-iphones/

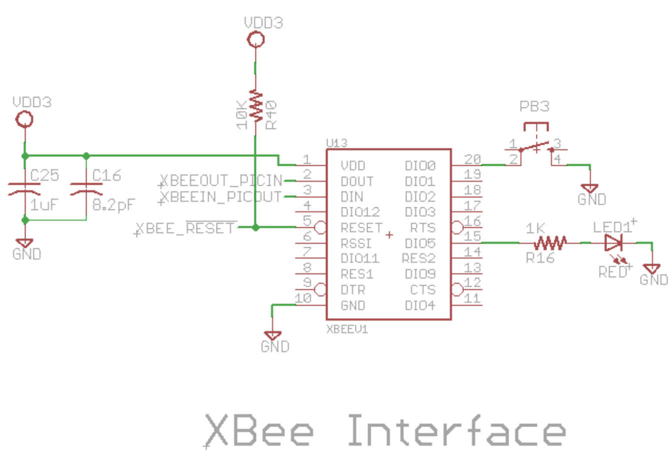
[86] http://www.govtrack.us/congress/bills/109/hr5556/

[87] http://www.iphondejd.com/iphone_jd/2013/07/wireless-emergency-alerts-iphone.html/

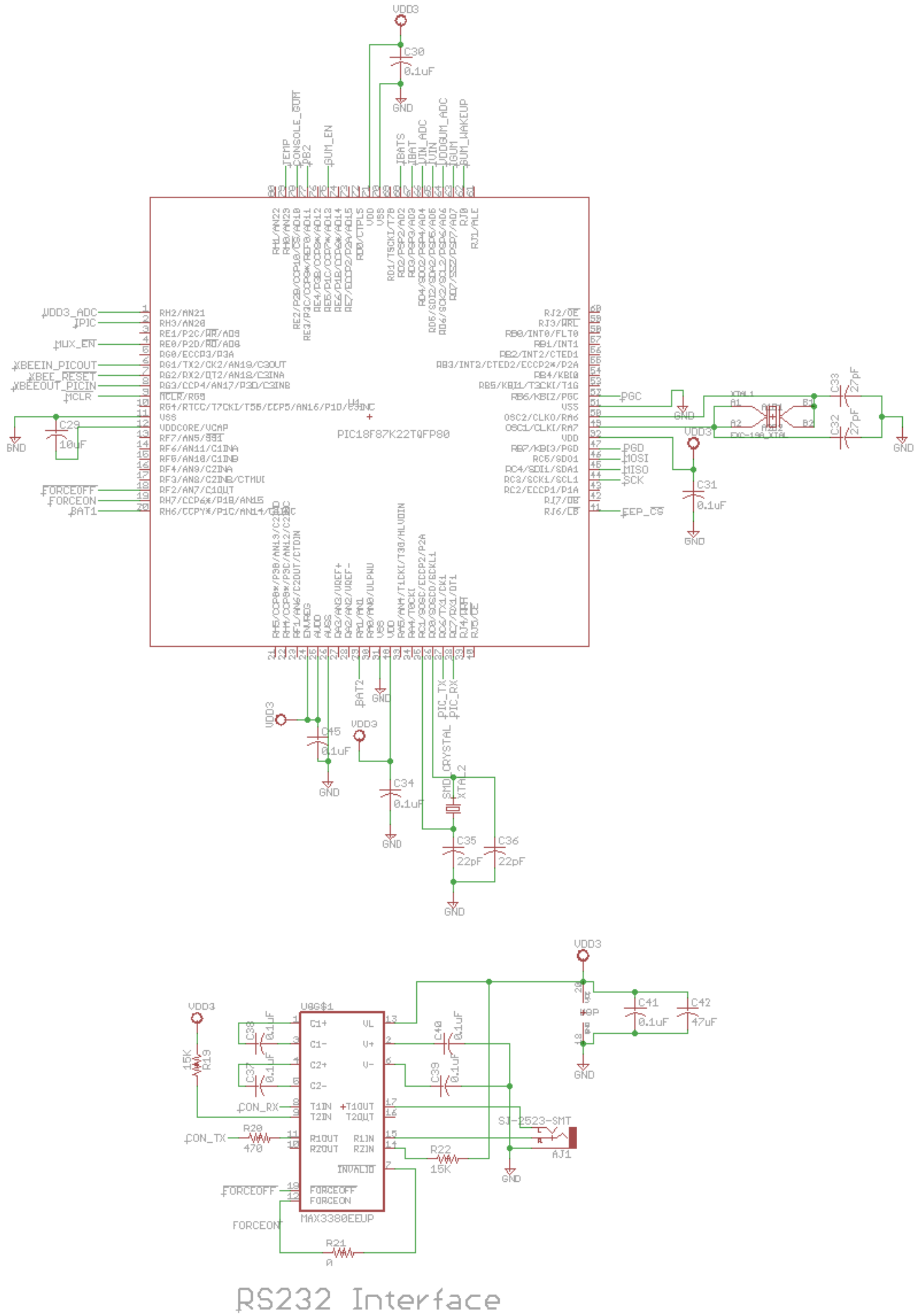
[88] http://www.globalissues.org/article/218/diseases-ignored-global-killers/

Appendix A – Supply board schematic (1/5)

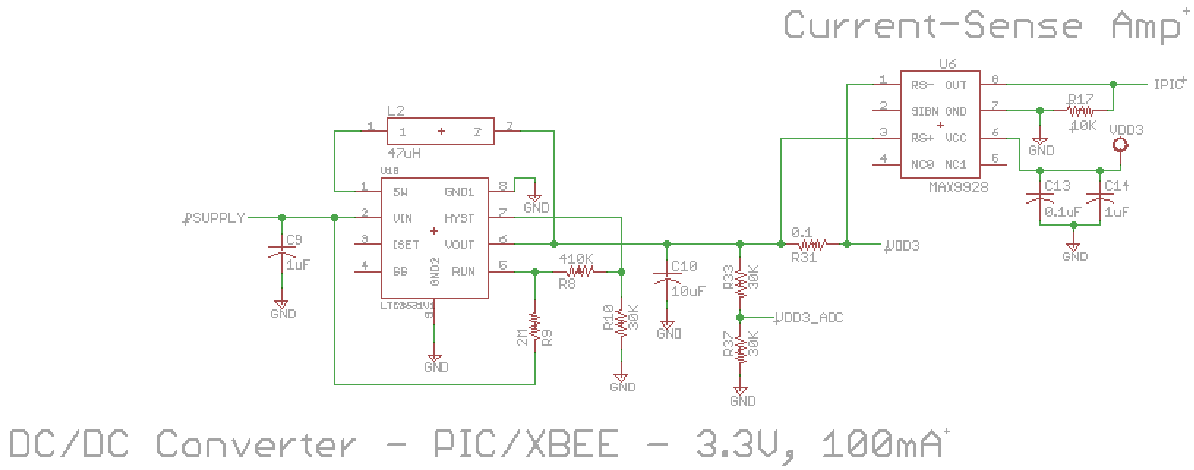


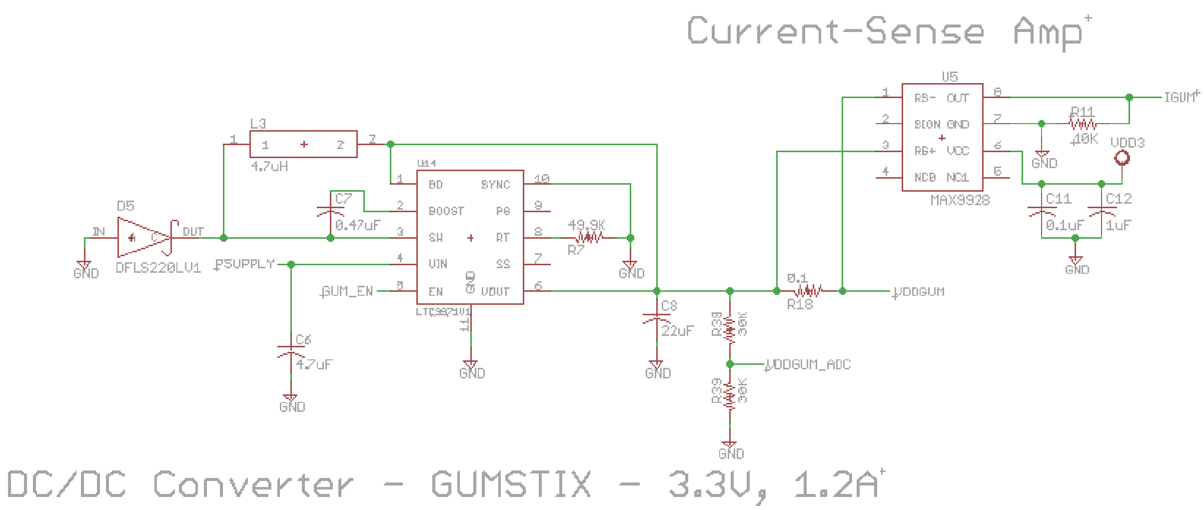


Appendix A – Supply board schematic (2/5)

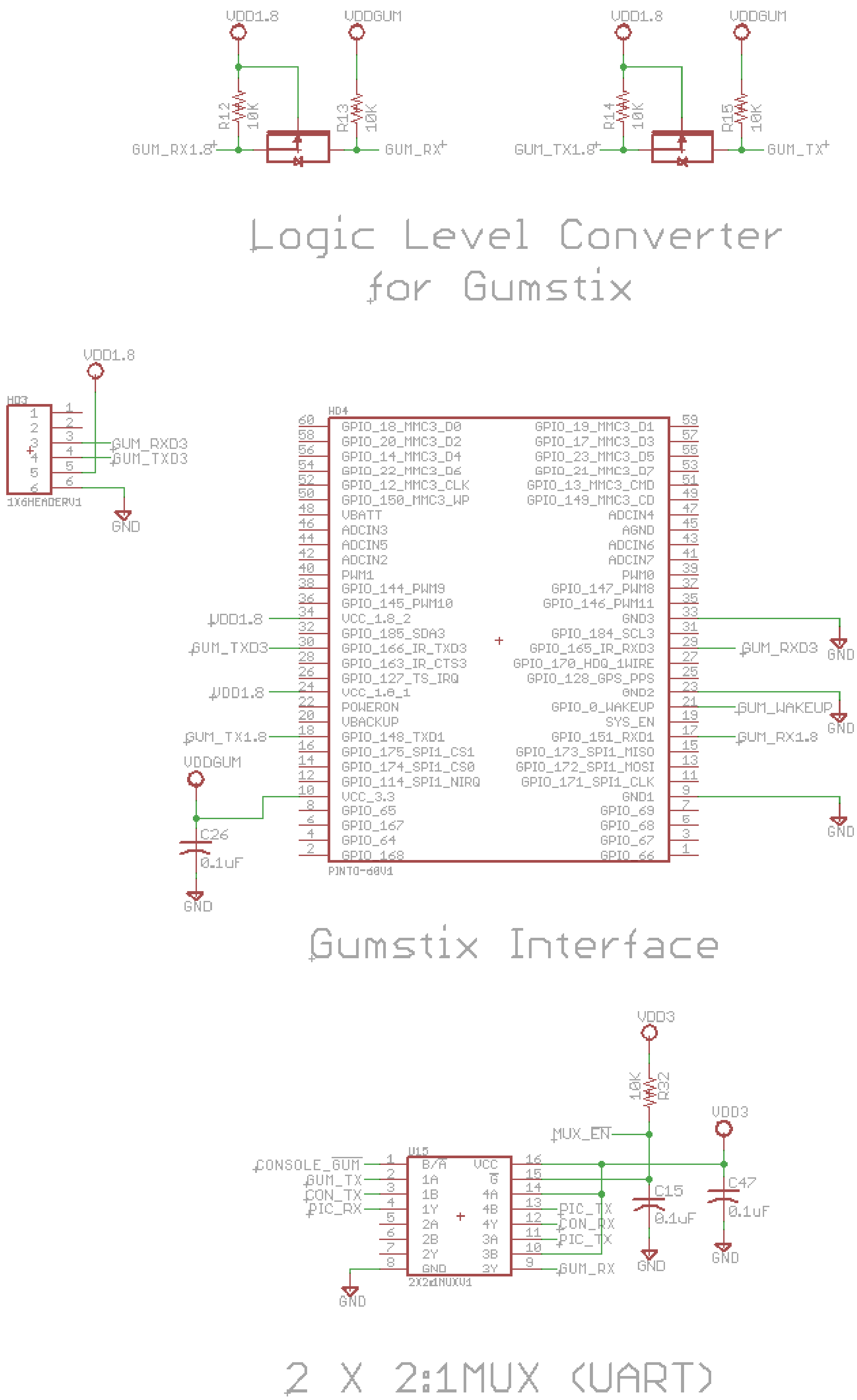


Appendix A – Supply board schematic (3/5)

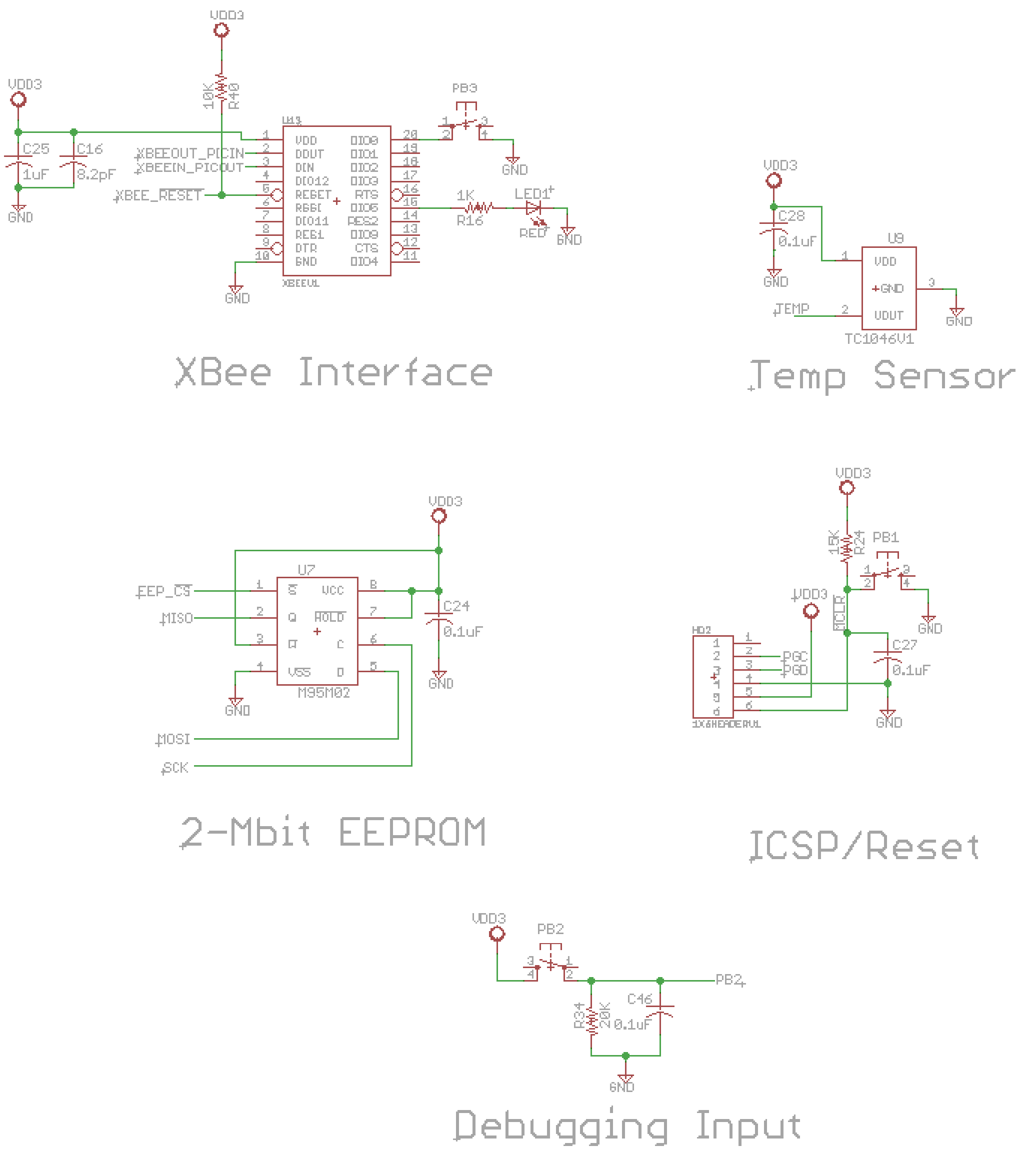




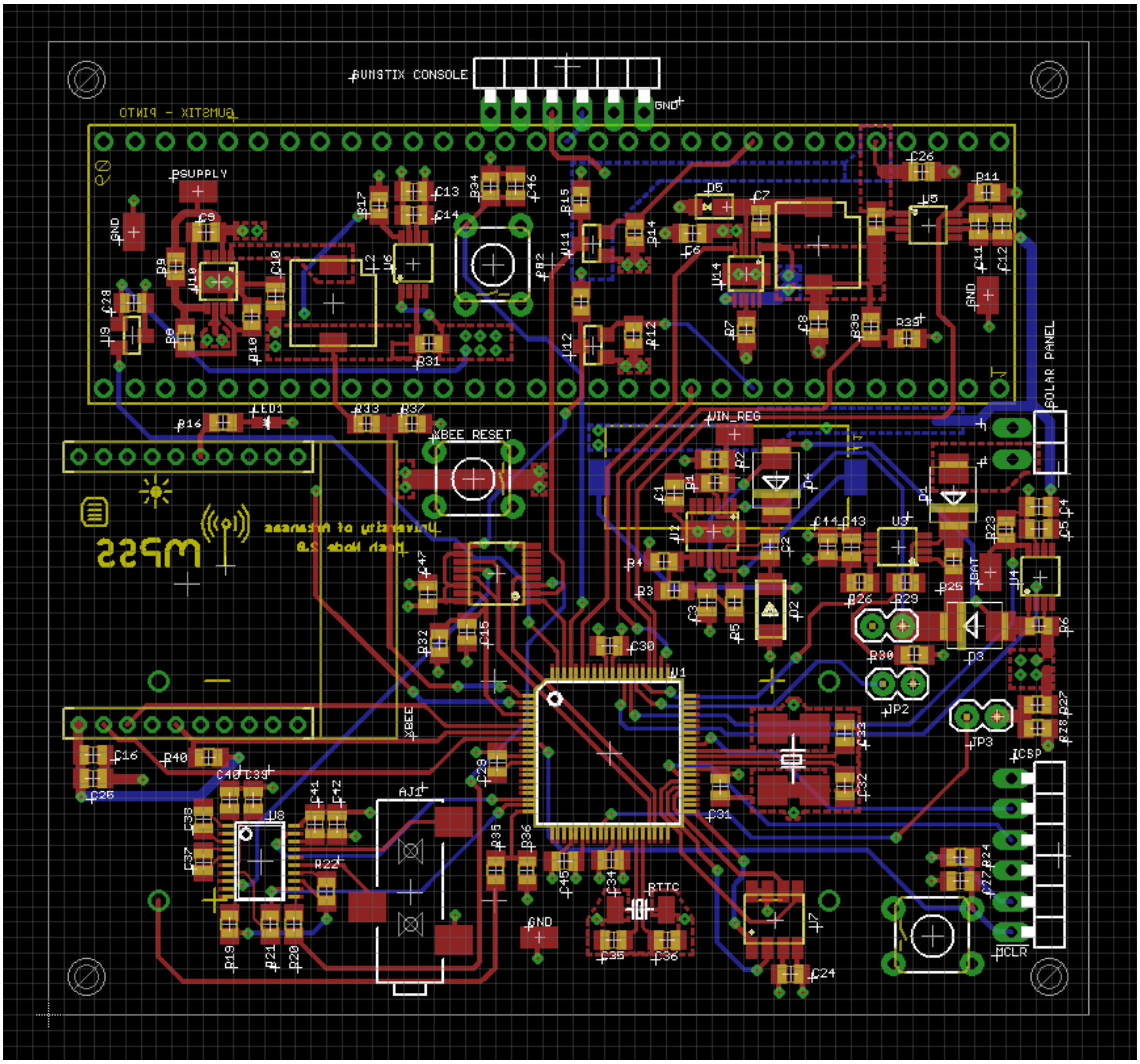
Appendix A – Supply board schematic (4/5)



Appendix A – Supply board schematic (5/5)



Appendix B – Supply board – PCB layout



Appendix C – Permission for use of image in Figure 3 – email from William Johnston (creator)

from: **William Johnston** <wgj@cast.uark.edu>
to: Adam Matthews <amatthe@email.uark.edu>
date: Tue, Nov 26, 2013 at 1:49 PM
subject: Re: Written permission
mailed-by: cast.uark.edu

Adam Matthews,

You have my permission to publish in your dissertation a picture of the WiFi-user interface for the software I designed for the emergency mesh node.

William Johnston