

WRENSys: Large-Scale, Rapid Deployable Mobile Sensing System

Kyeong T. Min, Andrzej Forys, Anh Luong, Enoch Lee, Jon Davies, Thomas Schmid

Department of Computer and Electrical Engineering

University of Utah

Salt Lake City, Utah 84112

{kyeong.min, andrzej.forys, anh.n.luong, enoch.lee, jon.davies, thomas.schmid}@utah.edu

Abstract—We present *WRENSys*, a system that allows for a low cost, rapidly deployable, large-scale and easy to maintain wireless mobile network, with minimal subject interference, to study contact networks of a population. We discuss our experiences with several hardware designs, including our new *WREN* sensor, base stations for charging and programming 2000 motes in parallel, software applications, and our deployment experiences and results. We deployed the system at 26 different locations with an average of 500 participants per school, the largest having more than 1,500 sensors, resulting in 35 million contact data points. On average, our system required only 30 minutes of preparation time and 30 minutes of deployment time with one graduate student and two additional personnel. Our system allowed for contact networks with coverages at approximately 80%. This shows a sustainable platform for studying human contact networks.

Index Terms—Wireless Sensor Networks, Low Cost, Low Maintenance, Large-Scale, Rapid Deployments

I. INTRODUCTION

The influenza virus affects hundreds of children and elderly people yearly in deadly manner. While over 100 children died from the flu in 2013, during the epidemic of 2009-2010, about 350 children lost their lives due to the flu [1]. Experts believe the flu virus spreads mainly by expelled droplets when infected individuals cough, sneeze, or talk. In “How Flu Spreads” [2], the Centers for Disease Control and Prevention (CDC) of the United States reports the flu can potentially infect others at a distance of up to two meters. Therefore, we focus on studying the human contact network of school-aged children, how long they are in proximity with whom, and how these contacts mix between the different classes and grade levels during recess, lunch breaks, extracurricular activities, and homes. Measuring human contact networks by equipping people with contact sensors allows researchers to understand and study how airborne diseases spread by using real life data sets instead of relying on computer population models. Also, the capabilities of large-scale, rapidly deployable systems enable researchers to include larger populations and gain statistically meaningful data and models. Figure 1 shows a human contact network graph depicting the complex data and contact patterns between students throughout a school day with *WRENSys*.

Contact among Utah School-age Population (*CUSP*) aims to extract real-life human contact networks in schools using wireless sensor for the purpose of modeling the transmission of airborne diseases. Performing this study is challenging,

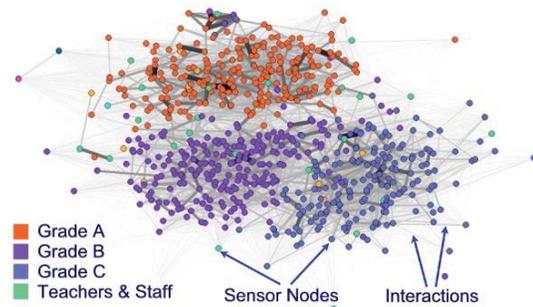


Fig. 1. Human contact network graph showing data and complex contact patterns between students during a typical school day. The *WREN* platform collected this data sample. Circles represent sensor nodes and lines represent their interactions. Different colors represent the various grade levels as well as teachers and staff while line thickness represents contact intensity. Epidemiologists access this data to study the spread of airborne diseases.

however, as every study subject has to be equipped with a sensor. Our deployments did not only include high schools, where we could expect large concentrations of smartphones, but also middle and elementary schools, as well as kindergartens, all of which smartphone usage could be prohibited. In addition to the large age-span of study subjects, our plan was to deploy the system at different schools on a weekly basis with minimal interference to daily school activities. Therefore, the system must be prepared, configured, programmed, and deployed within a 30 minute window while scaling to 2000 sensors. This leads to special requirements and design approaches for various challenges such as sensor maintenance, system-wide programming, data collection, and decreases in unit cost for large-scale rapidly deployable sensing systems.

The primary requirements for the *CUSP* project are:

- accurately detect contacts within a 1-3 meter range
- enable high density coverage ($\sim 2,000$ sensors)
- unobtrusive, small, comfortable, and safe to wear
- easy maintenance, charging, faulty node detection
- parallel programming and data downloading
- repeatable, rapid deployments
- minimal development schedule

II. BACKGROUND

The advent of smartphones and their large penetration in society make these devices the ideal tool to study large-scale human populations for data such as: localization, movement



Fig. 2. Wireless nodes representing the three hardware iterations. From left to right: TelosB (Memsic), Irene (Moteware), and WREN (our design). They iteratively improve on size, weight, cost, and usability. Enclosures (b). Left: custom laser-cut acrylic case for the TelosB (3.55 cm x 2.54 cm x 7.11 cm). We used tape to hold the case together and to protect the wearer from sharp edges. Middle: standard Irene case. Right: the injection molded WREN case, making it smaller, safer, and more durable (3.3 cm x 1.52 cm x 5.33 cm).

patters, or determine what places are “hot” right now [3]. Although smartphones have been gaining popularity in studying human contact studies, how do we deal with the heterogeneity inherent in smartphone hardware, reliable deployments, application upkeep, and various types of users?

In [4], Ahmad et al. studied various usage models of the iPhone and its dependence on the socioeconomic status (SES) of a person. They find that different SES groups use the iPhone very diversely, and suggest various design implications to support studies across SES levels. Furthermore, Nathan et al. indicate in [5] that while the Reality Mining study supplied 100 identical mobile phones as wearable sensors, their localization was limited to an accuracy of 10 meters. This would be unusable in human contact studies for influenza that require the two meter range recommended by the CDC. Furthermore, not all students, especially kindergartners, have smartphones and policies can limit access to phones during the day.

While wireless system with similar goals have been deployed before in schools [6], it became evident to us after experimenting and deploying commercially available nodes that they do not meet our requirements. We found severe drawbacks and limitations with those solutions, such as high-cost, non-scalability, and maintainability issues which motivated the design of a new platform. For example, the wireless nodes used in [6], the TelosB, are powered by two AA batteries and lack a recharging mechanism that does not involve replacing all the batteries. Thus, after our initial deployments, we began developing our own platform while substituting the TelosB nodes with Irenes, which unfortunately presented other drawbacks discussed later.

WRENSys addresses the aforementioned issues by re-engineering the whole Wireless Sensor Network (WSN) system stack, from hardware including wireless nodes (sensing, physical design, weight, maintenance, battery life, see Figure 2) and base stations (maintenance, recharging, reprogramming) to supporting applications. We show our results of *WRENSys* and discuss the comparison to existing hardware platforms such as the TelosB and Irene in Section VII. With *WRENSys*, one graduate student can provision and manage up to 2000 nodes, all within 30 minutes, truly allowing for rapid, successful deployments.

Node	TelosB	Irene	WREN
Cost (\$)	72	200	35
Size (cm)	3.5 x 2.5 x 7.1	5.0 x 1.7 x 6.8	3.3 x 1.5 x 5.3
Weight (g)	83	33	30

TABLE I
SUMMARY OF THE COMPARISON BETWEEN TELOS B, IRENE AND WREN PLATFORM. WE CAN SEE THAT WREN PLATFORM STANDS OUT IN COST, SIZE AND WEIGHT.



Fig. 3. TelosB nodes with base station and case. The TelosB was worn around the neck using an easy-break safety lanyard, though study subjects often complained about the total weight. A large 50-port USB base station connects the TelosB to a computer for programming and data download. The TelosB were dropped from our deployments due to complications in programming (only around 150 nodes of 200 actually enumerated, even using multiple USB buses), weight of nodes and base stations, and the non-rechargeable batteries.

Our previously published work on this topic includes a demo [7] and data mining workshop [8]. We demonstrated our proof-of-concept hardware design at [7] to show a functional system using the WREN nodes. Since then, we have completed the majority of our deployments and experienced new insights to sustain such a system. While a brief description of *WREN-Sys* occurs in [8] to give the reader an understanding of the platform, that paper focuses on actual deployment challenges, data collection and processing, and network validation. Given the amount of raw data, we needed to manage it correctly during and post deployment, and develop methods for better node synchronization as detailed in [8].

Our key contributions are, first of all, to meet all aforementioned system requirements and to provide a design and implementation of a rapid, large-scale, low-maintenance wireless sensing system with design approaches that set the ground work for further research. Future applications are not limited to social contact network study, but to other applications where the goal of rapid and large-scale deployment is paramount to the system requirements. Importantly, the scope of this paper does not delve into the epidemiological contact data analysis, which is planned to be analyzed by *CUSP*. All hardware schematics, base station construction, TinyOS *WREN* platform and application software code are open source and publicly available at [9].

III. RELATED WORK

In [6], Kazandjieva et al. used 994 TelosB sensors inside a conference badge with a lanyard for a one-day, one-time deployment in a high school in California to measure its social contact network. Deploying a large number of TelosB sensors requires considerable effort. Kazandjieva reported issues with hardware resets, battery disconnects, and uncomfortable lan-



Fig. 4. Irene nodes with base station and case. After study subjects’ disapproval of lanyards, we designed wrist bands for the Irene nodes. The base station is the same as the TelosB with Micro USB instead of Female USB plugs. The Irene has a rechargeable Li+ battery, removing the need for replacing batteries. This made maintenance easier, but the high price and mechanical reliability issues didn’t allow us to scale beyond 200 nodes. Almost every USB connector on the charging circuits had to be epoxied due to breakage after several insertions.

yards that would swing around the neck of the study subjects. Yet the collected contact data was extremely insightful, and was published in [10]. As we will describe later, we attempted using TelosB nodes, but found that even 200 nodes is impractical for a rapidly deployable low-maintenance system. *WREN* nodes utilize a light-weight, rechargeable Li-Ion Battery and are designed to fit in a small, two-part plastic enclosure.

The OpenBeacon Project [11] developed a low-cost active RFID tag for similar contact studies as the *WREN* targets. The main difference between the OpenBeacon tag and our developed *WREN* is the required infrastructure of the OpenBeacon. Having this infrastructure simplifies the tags significantly, since no data must be stored locally on the node. The OpenBeacon tag measures the RSS of an incoming message, then switches to the infrastructure channel and sends a message to a server over the installed wireless base stations, recording this information. Installing the infrastructure is possible, for example, during a conference where Ethernet plugs and Internet connectivity is readily available. However, our targeted schools do not have such an infrastructure, and deploying the wireless base stations adds significant deployment overhead. Thus, we decided to include additional storage capabilities on our *WREN* nodes, such that any infrastructure at deployment sites is unnecessary.

OpenBeacon’s sister project, an EU funded project called SocioPatterns [12], used a similar system in a primary school to gather their high-resolution contact network [13]. The system was deployed for 2 days and collected 77,602 contacts between 242 individuals. In contrast, *WREN* has now been deployed in over 26 different locations, averaging 2 days deployments, and collected over 35 million contacts so far.

IV. COMMERCIAL PLATFORMS

We developed three major hardware iterations (Figure 2), learning from every design to improve the next. The original plan was to use commercially available wireless sensor nodes in order to quickly start studying contact networks. Our positive and negative experiences throughout each iteration of sensors, base stations, and software influenced our subsequent platform design decisions.

A. TelosB Platform

The first deployment involved a network of 200 TelosB nodes (Figure 2, Figure 3). While these nodes are excellent for small networks, they lack an efficient and scalable mass programming option. This limit is imposed by the nature of the USB host communicating with UART bootloaders on the TelosB nodes. The TelosB nodes also require new batteries after a few days of deployment. Simply replacing the batteries required several hours of work by over five people to prepare for each deployment.

We designed a case to hold the node and protect it from the adolescents that carried them (see Figure 2). After multiple enclosure iterations, we decided on a custom, laser-cut acrylic design reinforced with packing tape. While the case avoided unexpected resets due to button pushes as reported in [6], it made battery replacement even more tedious as the case had to be taken apart to access the batteries. We also noticed the battery holder and heavy AA batteries allowed for a battery disconnect when the case was shaken during bumps or running activities. This introduced severe problems with time synchronization and algorithms, necessitating heavy data post-processing to fix time discontinuities.

The TelosB uses USB for programming; thus, we developed four 50-port USB base stations (Figure 3) allowing us to connect all 200 nodes to one computer using two independent USB buses. We found that our Linux operating system (Ubuntu 12.04) was not able to enumerate all the connected devices, forcing us to use two laptops to program the whole network of 200 nodes. While we considered wireless programming, the amount of debugging required for this type of system when programming failures occur would be too much for the planned crew of non-technical deployment personnel. Thus, we abandoned the idea of scaling this system up to the required 2,000 nodes, as it would have been unmanageable [14].

B. Irene Platform

Due to the limitations of TelosB platform, our second network deployment comprised of 200 Irene nodes (Figure 2, Figure 4). The Irene node is based on the Epic hardware module [15], and contains the same microcontroller and radio chip as the TelosB nodes. Different from the TelosB, Irene nodes come with a rechargeable battery and small, light-weight case which greatly reduced our maintenance time and increased wearable comfort. Unfortunately the same USB programming limit experienced with the TelosB nodes plagues the Irene nodes. The battery charging circuit header needed to be epoxied on every node due to breakage after repeatable cable insertions. The Irene nodes also cost significantly more, approximately \$200 each, which prohibits minimal budgets from scaling to 2,000 nodes. Although several TelosB drawbacks were circumvented, those encountered using the Irene platform were enough to influence our decision to build a new node from the ground up, which is explained in detail in Section V.



Fig. 5. Base stations with WREN nodes plugged in. We designed the base stations specifically for ease of use, storage, and maintenance in mind. The base stations can be stacked to save space. LEDs indicate the proper function of each blade and their status. The WREN features an RGB LED for debugging and as a status indicator for charging/fully charged while plugged in. These LEDs make it simple to identify broken nodes, thus reducing the danger of re-deploying non-functioning systems.

C. TelosB and Irene Base Station

We built four base stations for the TelosB and Irene platforms. They were heavy (8.4 kg each, without nodes), bulky, and contain a clutter of 50 USB cables for up to 50 nodes per base station shown in Figure 3. During the transition from TelosB to Irene nodes, we rerouted and replaced each USB Type A cable for the corresponding USB Mini Type B cable used with the Irene.

Finding nodes that failed to program was a challenge due to the generally random orientation of nodes. The primary drawback was the inability to program over 150 nodes from our laptop using USB. Although deployments succeeded, sensors took over one hour to distribute and needed a substantial labor force to complete. These interface limitations proved unsustainable in deployments of several hundred nodes. We developed a new type of base station to accompany our new WREN nodes which improved upon the initial drawbacks and allowed for a consistently rapid-deployable system.

D. Initial Enclosure

We carefully considered the enclosure for the TelosB and Irene nodes for safety, low-cost, sensor protection, and user-friendliness/wearability for study subjects. Figures 2 and 4 show the clear acrylic TelosB and Irene cases, respectively. We ground down the sharp corners and covered the edges with clear tape for the students' safety when using the TelosB cases. To avoid breaking the case, lanyards were attached through the battery connector on the node as stress relief. We found this case unfit for continuous deployment due to the high maintenance and the size and weight of the entire unit. Less bulky cases are provided with Irene nodes, but instead of using lanyards we decided to make custom Velcro bands such that the students wear them on their wrists. However, this led to problems such as accounting for variable wrist sizes and bands easily becoming dirty or blocking the charging port. While our previous enclosures were carefully considered and used for several deployments, the case design for our WREN sensors minimized all the drawbacks we experienced using the TelosB and Irene case designs.

V. WREN PLATFORM

In this section we describe our new platform, architecture, and details of WRENSys. The WREN hardware architecture



(a) Top

(b) Bottom

Fig. 6. Top side of WREN showing radio, MCU, accelerometer, and flash memory. Bottom side showing RTC with integrated 32 kHz TCXO, switch-mode power regulator and battery charger, battery connector, and 10-pin high-reliability Hirose connector.

consists of our node, base station, and custom boot strap loader (BSL) based on the I2C serial bus.

A. Wireless Ranging Enabled Node (WREN)

An important design consideration is to decide whether to adhere to the same hardware families as the TelosB and Irene, or to switch to a new architecture. The advantage of staying is an easier port of already existing software, such as the software applications explained in the Section VI designed for TelosB and Irene nodes. However, the hardware used on the TelosB is almost 10 years old, and significant improvements have been made in microcontroller and radio designs since then. Therefore, we decided to redo the design from the ground up, while keeping some software compatibility, as the advantages of cost cutting choices, feature improvements, and lower power draws outweighed the disadvantages.

1) *Microcontroller*: The entire node design is based around the choice of the microcontroller. The venerable TI MSP430F1611 is the heart of both the TelosB and Irene nodes. This microcontroller, while well supported, has a low feature to cost ratio (\$10 @1K units). Since price was an important metric, we considered a more modern and cost optimized microcontroller. Table I compares the three platforms. We chose the TI MSP430F5342 (\$3 @1K units, Figure 6(a)) for several reasons:

Code compatibility: The MSP430 5 series run the same Instruction Set Architecture (ISA) as the 1 series. This allows a swift transition from the old software development platform.

Higher clock frequency: With a top clock frequency of 25 MHz versus 8 MHz, in the event of more processor intensive tasks, the processing power is available.

Lower power consumption: The CPU core in the 5 series draws 290 $\mu\text{A}/\text{MHz}$ at 3 V while the 1 series draws 500 $\mu\text{A}/\text{MHz}$ at 3 V.

High storage capacity: 10k of RAM and 128k of ROM provides enough storage for even complex applications.

2) *Radio*: We chose the Atmel AT86RF233 because of its IEEE 802.15.4 compliant radio as well as several key features

	Atmel AT86RF233	TI CC2420
RX	11.8 mA -101 dBm	19.7 mA -94 dBm
TX	13.8 mA 4 dBm	17.4 mA 0 dBm
Max data rate	2 Mbps	250 kbps

TABLE II
COMPARISON OF RADIO ICs.

that significantly improve over the older TI CC2420 used in the Irene and TelosB. We use the measure of radio signal strength indicators (RSSI) between nodes to determine the distance. Our unique design employs an antenna fabricated within the PCB operating at 2.4 GHz, which reduces cost and component space.

Transmit power and receiver sensitivity: The AT86RF233 has a maximum transmit power of +4 dBm (vs. 0 dBm), and a receiver sensitivity of -101 dBm (vs. -94 dBm). This increase in link budget of 11 dB allows a considerably longer range than the CC2420. Our own measurements of our final node show a range in excess of 100 meters on an open field at full power.

Lower power draw: In addition to the overall power reduction afforded by continuous operation in both receive and transmit (see Table II), receive power consumption can be further reduced by up to 50 percent using Smart Receiving Techniques implemented in receive mode.

Higher data rate: Bypassing the chip mode of the IEEE 802.15.4 radio, the AT86RF233 achieves data rates as high as 2 Mbps. One power saving technique employed in the integrated circuit (IC) is to desensitize the receiver. This reduces power, at the cost of not detecting messages with a very low RSSI. Desensitizing the receiver was necessary in order to reduce the maximum receiver range to the required 2-3 meters at the lowest transmit power, thus reducing the total number of overheard messages from a larger neighborhood. Figure 7 shows the RSSI given several range tests.

Another reason for choosing the AT86RF233 was its potential for RF ranging. It integrates a phase measurement unit, giving the possibility of a phase-based ranging systems. However, this capability has not been explored in the current software architecture.

3) *Power System:* The power system employed on the WREN revolves around the TI BQ25010, a lithium ion/polymer charging IC that includes a high efficiency switching regulator that is used as the primary voltage source for the system. This regulator provides power duty cycling that enables high efficiency at a wide range of system loads by changing the switching frequency depending on power demand. For example, during system sleep, the power supply reduces its switching frequency to reduce the quiescent current. On average the WREN draws 5.2 mW while executing our application (see Figure 8). During deep sleep it draws 60 μ W, of which 50 μ W accounts for the power supply's quiescent power. Radio transmit draws 42 mW. These power saving features can sustain

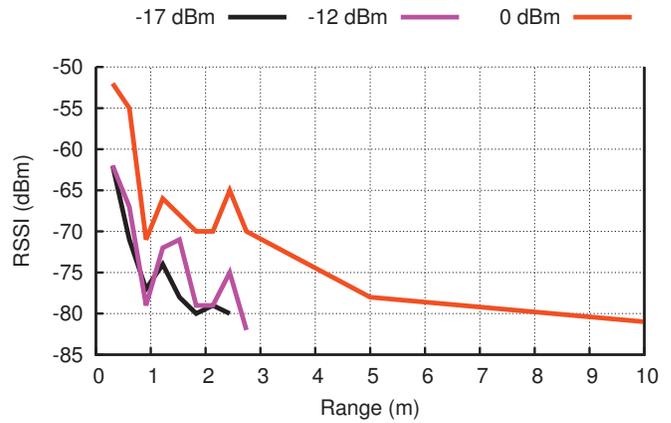


Fig. 7. Range vs RSSI. We desensitized the receiver to reduce power draw and range. Without these modifications, the minimum distance was 17 meters. Desensitizing the receiver reduced this range to match the requirements for our deployments.

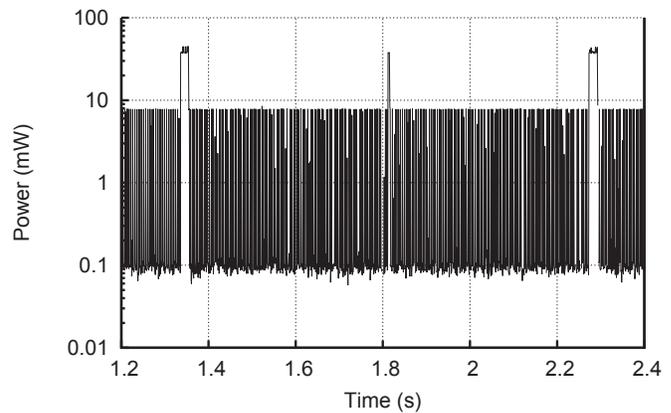


Fig. 8. WREN power draw for a 0.5 s sleep interval. The average measured power is 5.2 mW, sleep at 60 μ W, standard operation at 8 mW, radio actively transmitting 42 mW. The power supply quiescent power is 50 μ W. This corresponds to the expected power draws given by the component datasheets.

deployments for up to 1 work week on a single, 180 mAh battery included in the WREN.

B. WREN Base Station

We designed a base station capable of supporting large-scale sensor networks with solutions for charging, programming, debugging, and rapid-deployment in mind.

We considered the following design requirements for the base station.

- simple interface for power and programming
- charge/program of at least 100 nodes each
- easy access to connect/monitor nodes
- contain status indicators for debugging
- modularity for construction and repair
- minimal size and weight

1) *Implementation:* The main concept for the base station was to eliminate the bulky USB cabling. We designed a PCB backbone allowing each node to vertically plug into the base station. To ensure proper connections and handling, a sturdy 10-pin Hirose connector was used which is rated for up to 10,000

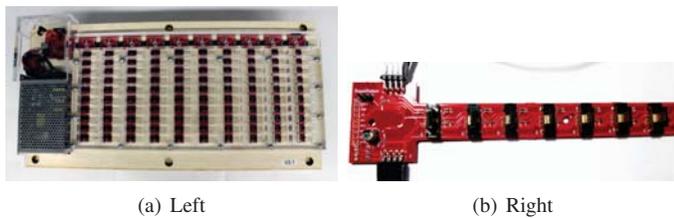


Fig. 9. Left: Base station structure showing 10 blades, power supply, and status LEDs. Using I2C in extended addressing mode, we could have connected up to 1000 nodes on one station, but power limits for charging, and signal integrity over multiple stations prevented us to go much larger than 100 nodes. Right: A single blade of WREN base station containing the high reliability 10-pin male Hirose connectors. The modular blades allow us to easily chain blades together for a variable node-sized base station. We were eventually limited by the power supply and trace widths for current, as well as maintaining proper signal integrity for the parallel I2C bus.

insertions. We designed PCBs to be modular for construction and debugging purposes. Each PCB module, which we call a "blade," supports up to 10 nodes and routes signals from the previous blade to the next (see Figure 9). A single base station has 10 interconnected blades, supporting 100 nodes. The programming interface was updated (see Section V-C) to support programming multiple base stations from a single bus. Debugging LEDs on each blade show power (5 V and 3.3 V) and reset status of the nodes. See Figure 9 for a base station with a clear, acrylic support structure and blade design.

We designed the support structure for the PCBs and nodes using layers of thin, light-weight laser cut wood. Each base station can be stacked on top of another to decrease size (see Figure 5). Even when fully populated, our new base stations only weigh 5.6 kg, which is significantly less than even the unpopulated base stations used for the TelosB and Irene nodes.

2) *Power Testing*: We monitored power usage to determine how long it takes a fully populated base station to finish charging when preparing for deployments. As seen in Figure 10, all 100 discharged nodes are fully charged within 2.5 hrs. Once charged, the base station draws approximately 249 mA, which accounts for 100 nodes programmed to sleep and a status LED on each node indicating full charge.

C. Inter-Integrated Circuit (I2C) Parallel Programming

WRENSys requires support from robust programming to minimize node failure during deployment. It supports fast, reliable, and parallel programming of thousands of nodes. We designed the programming interface around a more scalable shared I2C bus versus USB due to the maximum USB device shortcomings. We briefly considered a wireless programming system, but rejected it due to the large possibility of programming failures, and the existing requirement of a physical charging station.

1) *Bootstrap Loader (BSL) Design*: The BSL on the MSP430F1611 was fixed in ROM, and changing it was impossible. The new MSP430F5342 contains a special flash region for the BSL, allowing to rewrite its communication logic. We wrote a custom BSL for the WREN based on a shared I2C programming interface. The WREN enters the BSL only if it detects an active I2C connection (SCL and SDA pulled high)

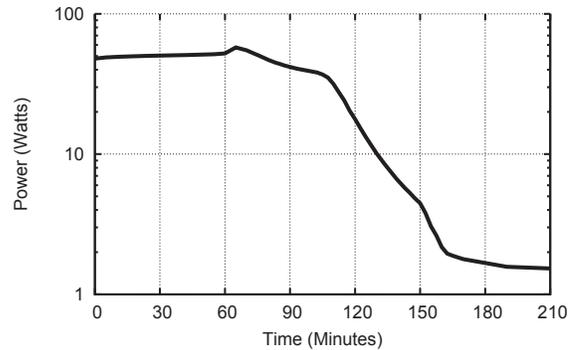


Fig. 10. Power usage graph of fully populated base station. The base station charges 100 nodes within 2.5 hours using 50 W. Each WREN draws around 100 mA at 5 V during charging. Thus, our power supply and blades had to carry up to 10 A of current on the power traces.

after a hardware reset. This prevents erroneous reboots from getting stuck in the boot sequence.

The MSP430's I2C accepts general broadcast instead of just single point-to-point communication. This allows simultaneous sending of programming packets to all the nodes. The response packets from the nodes are queued individually along with their unique addresses, providing a constant time scaling on the programming bus as we increase the number of nodes.

2) *Addressing*: I2C requires a 7 or 10 bit address for master-slave communication. Generating a per-base unique address can be challenging. The MSP430 provides an internal unique 32-bit chip ID, but hashing this address down to 7 bits is prone to many address collisions due to the large number of total nodes. Our scalable approach creates a large voltage divider on the base station, which is used in conjunction with the MSP430's Analog to Digital Converter (ADC) to create 100 distinct addresses for the I2C bus protocol. The voltage divider contains series resistors, one per node, powered by 3.3 V. Each node reads the voltage at its resistor using the ADC with 12-bits resolution. The 7 most significant bits will be the node's unique I2C address. This addressing scheme allows communication between the programmer and individual nodes for programming, testing, and debugging. This design gives us a flexible system where placement of the node on the base station does not affect the programming protocol.

D. WREN Enclosure

We approached the WREN enclosure selection differently from those of the TelosB and Irene nodes because we were able to design the hardware for the case. The following are the major design considerations that we took into account: minimal size and weight, debug light, easy connection to base station, and above all: *safety*. We use the Polycase AB-30 series case with minor modifications. This selection reduced the size and weight of the WREN node. Figure 2 shows our final WREN case that fits our specifications. We added a passive RFID chip for node identification in case they don't reply to regular RF communication, and to give deployment staff an option to quickly query the node using a

NFC equipped phone.

VI. SOFTWARE APPLICATIONS

To meet the requirements for *WRENSys*, we used Python for user interface applications, and TinyOS and nesC to program all sensor applications such that device drivers for each TI MSP430 processors for three hardware iterations would be compatible. The supporting software applications are mainly grouped into two categories: Stationary and Mobile applications. The stationary applications are placed at the deployment center and executed to prepare for and to collect data after deployments. Specifically, they consist of a User Commander, a Command Controller, and a Wireless Downloader. The mobile applications include a Mote Client and a Mote Sleeper application.

A. Stationary Applications

1) *User Commander*: For the front-end user interaction with sensors, we developed the user commander application (UCA) in Python, which allows actions such as start/stop sensing, get status, blink LEDs, run test routines, start wireless download, erase flash, and find certain nodes. The UCA provides a command prompt interface for accepting commands from the user and forwarding them to the command controller application.

2) *Command Controller*: The command controller application (CCA) is installed on a single commanding node, a TelosB node, which in turn broadcasts commands to client nodes using a predefined radio channel. The reason for using a TelosB instead of a WREN is for the convenience of a direct USB interface for the wireless broadcast; therefore we did not have to use a base station for USB to I2C conversion, or design a stand-alone board to handle the data transfer for the WREN. When a user enters a command on the UCA, the CCA receives the user command and wirelessly broadcasts it to all client nodes, which then process the command.

3) *Wireless Downloader*: The wireless downloader application (WDA) is installed on fifteen dedicated downloader nodes using 15 separate channels in the 2.4 GHz ISM band for parallel data download. Each WDA pairs and communicates with the mote client application installed on each *WREN*. Once a WDA and a client are paired, the WDA starts transferring all the data stored on the clients. For reliable in-order delivery of data, the WDA uses a packet-based data transmission protocol (the sliding window protocol) which waits for lost packets before moving onto the next window for data transmission. More detailed information on data collection, data download, and data processing for *WRENSys* can be found in [8]. The code for the download process is also open source and publicly available at [9].

B. Mobile Applications

1) *Mote Client*: The mote client program (MCP) is installed on the *WREN* nodes for detecting contacts among sensors within a two meter range. When a *WREN* receives a radio signal from other sensors, it measures the RSSI value of the signal and stores the value in flash memory. The flash storage

of *WREN* can store up to 129,000 contact records, which can contain data from deployments lasting more than one week with a duty cycle of 20 seconds, depending on average neighborhood size. Another important functionality of the client sensor is self-test and status indication. Although an LED might not be best suited for all situations, because our deployments are prepared for indoor use most of time, we embedded test code to utilize an RGB LED for determining its status to easily run testing routines before deploying sensors. For example, the red LED means a problem with the sensor and green means it is ready for deployment.

2) *Mote Sleeper*: After we realized that carrying base stations to every deployment site required many personnel and deploying sensors within 30 minutes was unrealistic, we created the mote sleeper program (MSP). We program the sensors before leaving for a deployment site to make the handout process simpler. MSP is installed on an extra mote, "sleeper node", used prior to and during the beginning of a deployment which prevents the client sensors from collecting data during transportation by sending small radio packets to the clients. Once the sleeper nodes are removed, the client sensors start collecting contact data. This allows deployment personnel to carry bags of sensors to the site and deploy them quickly without carrying base stations.

VII. RESULTS AND DISCUSSION

We discussed the *WREN* hardware and software design, and covered the implementation for a unique study of human contact networks. The study required large-scale, rapid, and frequent deployments to a dynamic environment. In addition, we designed the system to be used by non-technical people such that epidemiologists could easily maintain the system and manage deployments. In this section, we describe our results and findings from our deployments, and discuss their implications for future research in this space.

A. Overall Usage

Over the period of 10 months, from April 2012 to February 2013, we collected about 35 million contacts from 26 different deployments. Most deployments happened at schools, but some were completed during summer camps and various activities.

B. TelosB Deployments

Our first deployment at a school started in April 2012 with 200 TelosB sensors attached to four base stations as depicted in Figure 3. We deployed 138 sensors at that time and collected about 300,000 contact records. Before the second deployment, to make sure the nodes had enough battery life, we decided to replace all 400 batteries of 200 TelosB nodes. This involved opening up the case for each sensor, removing old batteries, putting the new ones in, and then finally closing the lid with a new tape. The whole process took 2 full working days for 2 people to finish. We used the TelosB in 3 total deployments before we were ready with a better solution, the Irene nodes, that would reduce the battery hassle and decrease node maintenance.

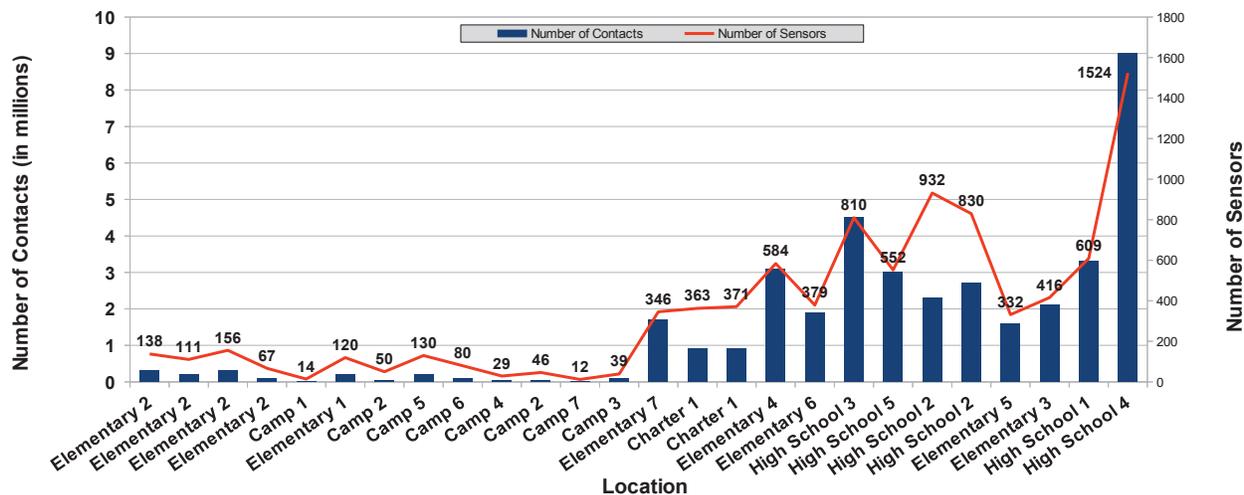


Fig. 11. Deployment results showing the number of sensors deployed and the number of contacts measured in school and camp deployments from 2012 to 2013. The 13 deployments with less than 200 sensors were performed using TelosB and Irene nodes. The larger deployments of more than 200 nodes were done using the WREN sensors. This was possible due to the reduced cost of the nodes and lower maintenance and programming overhead.

C. Irene Deployments

After the first 3 deployments, we switched to 200 Irene nodes. The new platform necessitated only a few software changes thanks to the similarity of the TelosB and Irene nodes, and the modularization of the code components in TinyOS. The Irene sensors are different in many aspects such as size, weight, and clip design as shown in Figure 4. The main difference between the TelosB and Irene is the charging capability, which helps to alleviate the maintenance concerns. The Irene uses a Lithium button cell battery with a custom designed charger that fits into the Irene case. The disadvantage of the Irene nodes is their price, totaling \$200 when including the case and additional charging mechanism.

Irenes were used for several deployments and summer camps as the case was significantly more robust than the custom acrylic case of the TelosB. However, the Irene sensors showed several drawbacks throughout the course of the deployments. Several charging ports and the USB connectors on the base stations broke after each deployment. We reinforced the connectors with epoxy glue, but still found that 10 out of 200 sensors occasionally broke. Part of the problem came from the Mini USB connector used on the charging circuit, which was rated to only 500 insertions, and was a surface mounted part. The other problem came from the mechanical stress the cable exerted on the case during plugin/plugout events and transport of the nodes. Clearly, a new node had to be designed addressing these issues in order to scale to the thousands of nodes we planned to deploy during the coming winter period.

D. WREN Deployments

After several Irene deployments, we were finally ready to use our new WREN sensors with a total of 20 base stations, each holding 100 individual sensors for charging and programming. As explained in Section V-A, the WRENs improves

significantly on ease of maintenance, weight of base stations and nodes, as well as ease of programming the whole system.

The ease of maintenance had a direct impact on the number of personnel needed to deploy the system. The previous TelosB and Irene deployments needed four or more people to deploy just 200 sensors over nearly an hour. This dropped to one engineering graduate student and two assistants easily being able to deploy 400+ sensors in only 30 minutes. Figure 11 shows deployment statistics with the approximate number of contact data collected during each deployment. So far, we made 13 deployments with the WREN platform, including the largest deployment of 1524 nodes in February 2013. Participation ranged from 60-100% through our opt-out program, with high-schools having the lowest participation rates, while most schools ranged in the 80%.

With WRENs we showed that large-scale, rapid deployments are possible with up to 1500 nodes, and are easily managed by just one graduate student. We believe that the system could scale to a few thousand nodes more, but after that we hit the limits of practicality. The bottleneck of the system is the process of plugging in each node for recharging and programming. While it takes only 80 seconds to plug in 50 WRENs, compared to 5 minutes for the Irene, it is too long once we scale to very large numbers. Wireless programming has been a large topic of research [16] and can be robust and secure. Thus, the largest issue is recharging the battery. Energy harvesting mechanisms could be employed, but the rapid changing environment does not ensure that enough energy can be harvested. We can imagine that a wireless charging base station based on either inductive charging or far-field RF systems could remove the necessity to have a wired base station connection. The wireless charging base station would be designed like a box where nodes can be dropped in to charge regardless of orientation. This would truly make the system easy to maintain.

VIII. CONCLUSIONS

In this paper, we described the design and implementation of *WRENSys*, consisting of the new *WREN* nodes, base stations and application software. Different from regular wireless sensor network deployments, we targets rapid, large-scale, mobile, and dynamic systems where nodes move around and infrastructure is non-existent. We showed successful deployment results for the last two years. The ease of deployment and maintenance of *WRENSys* are key to a fast turn-around time of node collection, data download, and redeployment. We used *WRENSys* to study the social networks from elementary to high schools and deployed at over 26 locations, with over 1500 nodes on one location, collecting a total of 35 million contacts between nodes. Our key contributions are, first of all, to meet system requirements for the large-scale human contact network study and to provide a rapid, large-scale, low-maintenance sensing system with design approaches that set the ground work for further research. The possible future applications are not limited to social contact network study, but to other applications where the goal of rapid and large-scale deployment is paramount to the system requirements.

REFERENCES

- [1] CDC, "Cdc reports about 90 percent of children who died from flu this season not vaccinated," <http://www.cdc.gov/flu/spotlights/children-flu-deaths.htm>, March 2013.
- [2] "How flu spreads," <http://www.cdc.gov/flu/about/disease/spread.htm>, September 2013.
- [3] S. Thompson, "Localization," *Journal of Comparative and Physiological Psychology*, vol. 69, no. 4, pp. 1–29, 1969.
- [4] A. Rahmati *et al.*, "Exploring iphone usage: the influence of socio-economic differences on smartphone adoption, usage and usability," in *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*. ACM, 2012, pp. 11–20.
- [5] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and ubiquitous computing*, vol. 10, no. 4, pp. 255–268, 2006.
- [6] M. A. Kazandjieva *et al.*, "Experiences in measuring a human contact network for epidemiology research," in *Proceedings of the 6th Workshop on Hot Topics in Embedded Networked Sensors*. ACM, 2010, p. 7.
- [7] A. Forsys *et al.*, "Rapid deployable system for human contact network research," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 383–384.
- [8] A. Forsys, K. Min, T. Schmid, W. Pettey, D. Toth, and M. Leecaster, "Wrenmining: Large-scale data collection for human contact network research," in *Proceedings of First International Workshop on Sensing and Big Data Mining*. ACM, 2013, pp. 1–6.
- [9] "Utah-wiesel github," https://github.com/Utah-WiESEL/wren_wsn.
- [10] M. Salathé *et al.*, "A high-resolution human contact network for infectious disease transmission," *Proceedings of the National Academy of Sciences*, vol. 107, no. 51, pp. 22 020–22 025, 2010.
- [11] "OpenBeacon," <http://www.openbeacon.org>, March 2013.
- [12] "SocioPatterns," <http://www.sociopatterns.org>, March 2013.
- [13] J. Stehlé *et al.*, "High-resolution measurements of face-to-face contact patterns in a primary school," *PloS one*, vol. 6, no. 8, p. e23176, 2011.
- [14] Y. Liu *et al.*, "Long-term large-scale sensing in the forest: recent advances and future directions of greenorbs," *Frontiers of Computer Science in China*, vol. 4, no. 3, pp. 334–338, 2010.
- [15] P. Dutta *et al.*, "A building block approach to sensor network systems," in *SenSys'08: Proceedings of the Sixth ACM Conference on Embedded Networked Sensor Systems*, nov 2008.
- [16] L. Mottola and G. P. Picco, "Programming wireless sensor networks: Fundamental concepts and state of the art," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 19, 2011.