Experiences in Measuring a Human Contact Network for Epidemiology Research

Maria A. Kazandjieva*, Jung Woo Lee*, Marcel Salathé†
Marcus W. Feldman†, James H. Jones‡, and Philip Levis*
*Computer Science Department, †Department of Biological Sciences, ‡Department of Anthropology
Stanford University, Stanford, CA, USA
mariakaz, jungwoo.lee, salathe, jhj1@stanford.edu, marc@charles.stanford.edu, pal@cs.stanford.edu

Abstract

This paper discusses our experience in designing and deploying a 994-node sensor network to measure the social contact network of a high school over one typical day. The system aims to capture interactions of human subjects for the study of infectious disease spread. We describe unique challenges posed by a large-scale network that is heavily affected by humans. We present techniques to address challenges such as frequent node reboots and global timestamps. The end result of the deployment is a dataset of 792 traces which can be used to calculate the school population's contact network and the rough location where interactions occurred.

1 Introduction

Epidemiology studies the spread and control of infectious diseases. Many airborne diseases, such as the flu, spread through social contact. Therefore, contact network epidemiology studies the properties of social graphs and their implications for disease transmission.

The traditional method for gathering contact data is asking subjects to report with whom they have had proximity contact. This approach does not result in a high fidelity contact graph, but rather in a rough estimate of the graph's general properties, such as degree distribution. Furthermore, the methodology has many problems, such as cognitive bias towards more recent events and random population sampling.

Instead of a pen-and-paper approach, this work presents a sensor network deployment to measure a high school contact network. We use 824 TelosB motes, one per participant, to detect proximity between subjects by measuring the signal strength of periodic beacon packets. In addition, we deploy 170 nodes in fixed locations throughout the school campus.

The deployment monitored all students, faculty, and staff for a typical school day, between 7am and 4 pm. After collecting the motes, we downloaded a total of 3 million records of contacts between participants and another 3 million records indicating locations. These records provide a complete, fine-grained data set of social contacts, which can be used to compute the contact network of the high school population. This is the first data set of such precision and magnitude, therefore computing and analyzing the contact graph is a separate research problem that we plan to present in an epidemiology publication in the near future.

This paper discusses our experiences in designing and deploying a sensor network heavily affected by human subjects and the unique challenges it introduces. The work makes the following contributions. We investigate how received signal strength (RSSI) can be used to define a contact between two people. We also describe the necessary post-deployment data processing so that data are usable. The main problem we encounter is a high number of node reboots – 1500 over 500 nodes, with some nodes having as many as 45 reboots. We propose several techniques for reconstructing the local node time at which beacons were received. Lastly, we use local times to synchronize all data traces to a relative global clock in order to place all events on a common timeline.

2 Contact Networks

The spread of an infectious disease in a host population is a diffusion process in a network. The network consist of hosts (the nodes of the network) and their interactions through which a disease can be transmitted (the edges of the network). Defining what constitutes an edge in the contact network is important because diseases spread through different routes, for example through airborne droplets (influenza), sexual contacts (HIV), or insect vectors (malaria.)

In this deployment, we were interested in measuring the contact network relevant for the spread of influenza-like diseases such as seasonal influenza, H1N1, and the common cold. Such diseases are thought to be transmitted mainly through airborne droplets exhaled by an infected subject sneezing or coughing in proximity to a healthy one. Our goal was to capture contacts between all students, teachers and staff who were within 9 feet of each other. The structure of a contact network has been shown to have a strong effect on diffusion processes, and knowing this structure will help make better predictions for the dynamics of disease spread. In addition, data on contact networks can inform mathemati-

HotEmNets'10, June 28 -29, 2010, Killarney, Ireland.

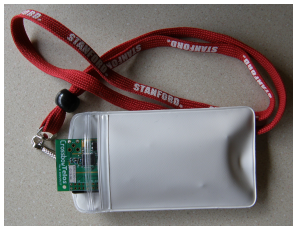

Figure 1: Each participant in the deployment received a TelosB mote inside a white pouch. Subjects wore the motes around their necks using a lanyard.

cal models designed to determine the best intervention strategies for controlling disease spread. The rest of this paper uses contact graph and contact network interchangeably.

3 Deployment Overview

The goal of the deployment is to measure the contact graph of an entire school and to collect information about participants' locations throughout a school day. Localization is key in testing the hypothesis that frequently visited places, such as bathrooms, have an effect on disease spread.

The entire school population consists of about 850 students, teachers, and staff. Participation was voluntary and participants' nodes were given out randomly. We agreed to keep the name of the school and any other sensitive information confidential. Analysis of the campus map suggested the use of an additional 170 nodes for location information. We purchased 1000 Crossbow TelosB motes; the first 850 of them were placed in pouches and attached on lanyards so people could carry them around their necks (Figure 1.) The remaining 150 motes, plus an additional 20, were locator motes. Again, location node IDs were eassigned randomly.

The deployment proceeded in two stages. First, we placed 170 stationary motes in the five high school buildings and a few auxiliary structures (a bookstore and locker rooms). These were put in place the night before the data collection day. We covered all classrooms and bathrooms, the dining hall and common spaces such as open study areas. For privacy reasons, we do not include a map of the deployment. The second stage of the deployment took place between the hours of 7 am and 4 pm the next day. Each person received a mote with one battery inserted, a pouch and lanyard, a loose battery, and an assent form. After completing the form, participants started the motes by inserting the second battery and noted down the start time in hours and minutes.

We set up a help desk in one of the buildings. Throughout the day, students and staff came with questions and concerns. For example, several stationary motes had fallen and the batteries had come out. Some people were seeking explanation on how exactly the data collection worked, while others were more creative, asking what happens if you rub two motes together or swing them around on the lanyards.

Towards the end of the school day at 3:30 pm, we started receiving motes back. Participants had been instructed to take one battery out to stop the program from executing. After collecting all motes, we made another round through the school, removing all locator motes. At about 8pm, after 13 hours at the school, the deployment was completed.

The total number of participants was 824, which excluded a few students and teachers who were not at school that day. From the 824 motes 2 were lost, 792 collected data, and 30 had empty flash logs. We suspect that while some people did sign the assent form and picked up a mote, they opted out of the experiment by never starting the motes. This, together with several nodes that had the batteries inserted the wrong way, accounted for the 30 empty motes. Therefore, the deployment yielded 792 individual data traces.

4 Design Considerations

This section describes in more detail the code running on the participants' and locator motes, our choice of parameters, and the reasoning behind these design considerations.

4.1 Code Description

Participants' Motes. Each participant's node was programmed to broadcast beacons at -16.9 dBm (power level 6 on the CC2420 chip) at a regular 20-second interval; the packet includes the senders local sequence number. Upon receiving a beacon, a node checks the RSSI value of the packet. If the signal strength is lower than -80 dBm, the packet is discarded. Otherwise, the receiver creates a contact entry consisting of the sender's ID and beacon sequence number, as well as the local node's sequence number and the RSSI value of the packet. The contact entries are stored in a twenty-entry buffer in volatile memory, and once the buffer is full, it is written to flash. Participants' motes also receive beacons from locator motes whose IDs are greater than 10000; these beacons are not subject to RSSI filtering.

Locator Motes. The code on these motes has the sole purpose of sending a beacon every 20 seconds. These beacons were used for identifying participants' locations throughout the high school campus. Since the stationary nodes were deployed around the school the night before the experiment, the code included a 12-hour timer after which nodes would begin beaconing. It is worth noting that since batteries were inserted anywhere between 4:15 and 4:30 pm the previous day, locator motes were not synchronized. Packets originating from these nodes were sent at -11 dBm.

4.2 Design Choices

A number of program parameters were consciously chosen during the code development process. The transmission and signal strength threshold were key to whether a contact would be recorded when two people came within a 9-foot distance of each other. We defer the discussion on these two parameters to Section 5.

The beaconing interval we chose had to accommodate several constraints. First, all contact and location entries received over the 9-hour duration had to fit in the 1 MB flash memory. Though the size of each entry was fixed at 7-bytes, we were not certain what the density of neighbors would be. The team was unanimous that we would rather overestimate the number of neighboring motes than cause data to be lost due to full memory. Calculations showed that with 20-second beacons we could support about 60 neighbors at all times. We decided that 20 seconds was conservative enough on the storage-side, while still providing high data granularity. Furthermore, from an epidemiological perspective, the sampling rate was fast enough.

To validate the choice of beaconing interval, we analyzed the amount of flash memory that was used by each node dur-

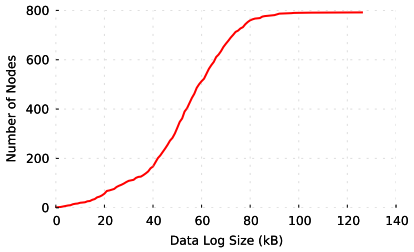

Figure 2: A CDF of used flash memory shows that about half of all participants' nodes used under 60 kB. Future deployments measuring a similar environment could sample faster, gathering even more fine-grained data.

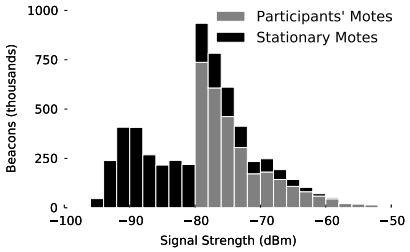

Figure 3: Data collected during the school deployment exhibit a range of RSSI values. For participants' nodes, most values are close to the threshold of -80 dBm we selected.

ing the deployment. Figure 2 show a CDF with size of the flash log on the x-axis. The amount of data each node collected was much less than what our original calculations assumed, and the longest data trace was a little under 130 KB. If we were to repeat the deployment, we would allow for faster beaconing and longer packet sizes if necessary.

4.3 The Human Factor

By far, human participation had the largest effect on the code development and the deployment outcome. A major concern from the beginning was the ability for subjects to opt out of the project due to regulations from the Institutional Review Board. From a technical standpoint, we considered two options: suspending participation by taking out one battery or by pressing the user button on the TelosB mote.

The original application design took advantage of the user button. While the mote was active, it sent, received, and logged beacons, and a green LED indicated its active state. Pressing the user button turned off the LED light and disabled both sending and receiving on the mote, rendering it inactive. We chose to set the default state as inactive; we could then insert the batteries the night before and ask participants to press the button to start them up in the morning.

We first conducted a test deployment with 20 students and teachers. The data were collected over a school day in December and resulted in very few log entries of proximity contacts. We discovered that many data traces had been cut short because of mote reboots or user button presses. The possible causes for reboots included loose batteries, pushed reset button, or a sudden shock (some students banged motes against hard surfaces.) Since the motes were in solid white pouches, it took students some time to realize that the LED light was off and the motes were inactive.

This experience indicated that the mote state should be active by default, so a mote could quickly recover from unexpected resets. More importantly, we decided that the user button was not a good solution given that there was no way to recover from an accidental opt-out, unless the participant looked at the mote and detected that the light was off. We proceeded to use battery removal as an opt-out mechanism.

The problem of multiple unexpected reboots remained in the final deployment. In anticipation of this, we logged the local sequence number associated with every logged beacon entry. Section 6 describes on what we observed during the deployment day as well as how the data traces were processed to restore sequence numbers where resets occurred.

5 Transmission Power and RSSI Cutoff

Many sensor network applications and deployments are configured to transmit at the highest power in order to have larger coverage and higher delivery rates. From an epidemiological standpoint, communications within the contact distance of 7 to 9 feet are most important. Therefore, we chose a power level that was lower but did not compromise the packet reception rate. Additionally, we chose a signal strength threshold that would filter faraway packets given a known transmission power.

To select these two parameters, we conducted an experiment with two students, each wearing a mote. The independent variables were transmission power, distance, and relative position of the nodes; we measured packet reception ratio (PRR) and received signal strength (RSSI). The experiment was not limited to face-to-face orientation of the two motes, because obstacles and body effect attenuate the wireless signal [2], affecting PRR. Rather, we had the two subjects face in the same direction and also change the relative degree between each other. For each 3-tuple of (power, distance, orientation) one mote sent 900 packets at 50 millisecond intervals, while the other logged received packets' RSSI.

Our experiments verified the results of previous work [2, 3], indicating that both RSSI and PRR decrease as distance increases, depend on the orientation of the two transceivers, and are affected by the body effect. For example, the RSSI difference between 3 and 13 feet was more than 10 dBm on average if two motes were not completely blocked by bodies.

After examining the PRR data we decided that transmission power of -16.9 dBm was enough to provide over 95% reception of packets over contact distances and different orientations, without unnecessarily extending the range of the beacons. The stationary motes used to provide location identifiers were programmed to beacon at the higher transmission power of -11 dBm to cover larger spaces.

We chose -80 dBm as the threshold beyond which beacons would be discarded. This decision was based on the experimental data showing that when subjects were facing each other, packets within 9 feet had RSSI of roughly -80 or above. Packets sent when one subject was facing the other persons back had a lower RSSI. While signal strength measurements are not a perfect indicator of distance, the propagation of the wireless signal is similar to that of airborne diseases in that it weakens with distance and obstacles, and has an uncertain behavior.

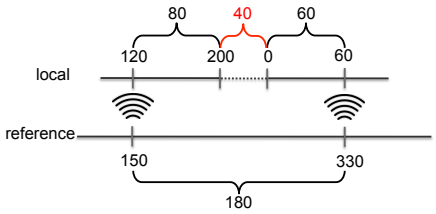

Figure 4: Loose batteries or a participant's opt out caused reset sequence numbers and gaps in the data. Using two beacons from the same sender as a reference, we could determine the length of the gap and restore the trace.

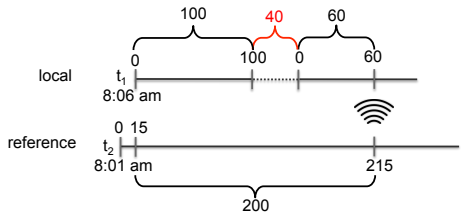

Figure 5: Some data traces did not have two beacons to use as reference points. We used the start times of motes to obtain information about the real time elapsed between the beginning of the experiment and the reset.

Figure 3 shows the distribution of actual RSSI values as logged by the participants' motes during the deployment day. Most packets coming from neighboring participants have RSSI values between -80 and -70 dBm, indicating that we chose a reasonable threshold.

6 Node Reboots

After data were retrieved from the participants' motes, we discovered that many nodes experienced reboots. We identified reboots by checking the logs for entries in which the local sequence number was not monotonically increasing. We found two reasons that caused these reboots – hardware resets, causing the node to quickly turn off and then back on, and loss of power, when one or two of the batteries were disconnected for an unidentified period of time.

Hardware resets were caused either by a subject pressing the reset button on the TelosB or shaking/hitting the mote, causing a loose connection. Both of these actions could have occurred accidentally or intentionally. In at least two cases, students acknowledged that they were playing with the reset button. Batteries could also have been disconnected with or without the knowledge of participants. Some participants willingly opted out by temporarily taking a battery out, while in other cases the batteries came loose.

The raw data retrieved after the deployment included 272 traces from nodes that did not experience reboots. The remaining 520 nodes had anywhere between one and 45 reboots, for a total of over 1500. It was important to restore accurate sequence numbers because they were acting as a local clock for each mote. Section 7 shows how these local timestamps were later used to create a global time that all logged contacts were relative to.

Hardware Resets. This type of node reboot did not create gaps in the data because resets were instantaneous; they simply caused the local sequence number to revert to zero. We identified these cases by finding two consecutive beacons from the same sender logged right before and after the local sequence number was reset to zero.

We reconstructed the data by incrementing the last valid local sequence number by one and replacing the zero with it. This technique recovered the sequence numbers up to the next reset. Applying it iteratively, fixed all hardware reset, yielding a total of 701 data traces with uniformly increasing local time, second row of Table 1.

Multiple beacon gaps. The remaining 71 motes experienced battery disconnections that left the motes turned off for an unknown period of time before rebooting. This caused nodes to potentially miss multiple beacons.

We extended the technique for fixing hardware resets to traces with gaps in the sequence number. Instead of looking at the beacons immediately before and after the sequence reset to zero, we scanned the entire trace before the reset and after it for a reference node. Knowing how much time had elapsed between two beacons from the same sender and comparing that with the time elapsed on the local clock let us calculate the length of a gap during which the mote was off.

Figure 4 shows an example application of this approach. The local node received beacon 150 from the reference node when its own sequence number was 120, and then received beacon 330 when the local sequence number was 60, after the reset. This information, together with the last known sequence number before the reset, are enough to compute the gap. In the example in the Figure 4 the gap was 40 sequence number; from that is is easy to translate the reset 0 to 340 and adjust all following sequence numbers by this offset.

This technique relied on using information from a node's neighbor, the reference node, so we could only use neighbors that never experienced a reset themselves. Otherwise, there would be no guarantee that the ticks counted between two received beacons from the same sender are accurate. When processing the data, we only used one of the 271 'safe' motes that originally had no reboots or the locator motes.

Table 1 shows that the processing step left only 25 data traces with incorrect sequence numbers. These traces had no matching beacons from safe nodes, so it was impossible to know how long the node was turned off.

Start Time Offset. To fix the the remaining data traces we had to create a second point of reference in the data. We used the mote start time as denoted on the assent forms. For each trace, we searched the entries after the sequence number reset. Once a beacon from a 'safe' mote was found, we used its sequence number and the start time of both the sender and receiver to calibrate the receiver's sequence number.

For example, Figure 5 shows two nodes that started 5 minutes or 15 beacons apart. When the reference node sent beacon 215, the local node should have had a local sequence number of 200. Instead, its sequence number increased from 0 to 100, then there was a gap in the data, and then it started increasing again from 0 to 60. As with the previous approach, this data is enough to calculate the gap length, 40, and recover all sequence numbers with an offset of 140.

The accuracy of this technique is lower than that of the previous approaches because it relies on wall-clock times provided by the participants. However, it did help reconstruct another 19 data traces, as shown in Table 1.

Processing Stage	Usable Traces	Remaining Resets
Raw data	34.4% (272)	1559
Hardware reset, no gap	88.5% (701)	214
Multiple beacon gap	97.0% (768)	119
Start time offset	99.4% (787)	34
Global time offset	100% (792)	0

Table 1: Reconstructing local sequence numbers was necessary due to the high number of mote resets. By applying several data processing techniques we were able to create consistent data traces.

The last five traces did not have enough data after the reboot to fix the local sequence number. This was the case for traces which had no entries from 'safe' motes, as well as traces which only had location beacons.

7 Global Time

Reconstructing the full contact network measured during the deployment required a global timestamp, relative to which all human interactions happened. We could not use time synchronization techniques because nodes in the network were mobile and often disconnected. Since the local sequence numbers in each data trace acted as clocks, we decided to treat them as offsets from one node – the master clock providing global time.

The location motes did not suffer any node reboots, so the sequence numbers in the beacons were reliable. In addition, the beacons were transmitted at a higher power and were not subject to the RSSI filtering at the receiver. Therefore, these motes were good candidates for the master clock. We wanted to choose a node that maximized the number of nodes that heard a master beacon. After reviewing a physical map of the high school, we chose a node from the dining hall area. A quick query on all data confirmed that 742 of the 792 motes had received one or more beacons from stationary mote 10055, located in the middle of the student cafeteria.

For nodes that had received a master beacon, we calculated the offset between the master and the local sequence numbers. An additional outcome was the creation of a table of offsets; this lookup table included participants' nodes as well as other stationary nodes. To processes data traces from nodes that did not hear directly from node 10055, we used the offsets table to transitively compute a timestamp from another node that already had its global time.

The techniques in Section 6 left five data traces without consistent local sequence numbers. To fix those and deliver a complete, timestamped data set, we treated each trace as several sub-traces delineated by node reboots. Each sub trace had enough beacons from location motes that we could use the offsets lookup table to generate global time. After processing each sub-trace, we concatenated them back together, delivering the original trace plus the global timestamp. This was the last step, as shown in Table 1, to removing all node reboots and providing global network tim.

8 Related Work

In addition to manual pen-and-paper techniques for measuring contact networks, cellphones and RSSI-based methods have been proposed in the literature. In a study by Eagle et al. [1], 94 participants were given Nokia 6600 phones with Bluetooth used to periodically scan for nearby devices. Collected data was either sent over the cell network or stored in the local flash memory. The cellphones did not filter Bluetooth signals; therefore, the possible range of contact distances is much higher (16-32 feet) than with RSSI filtering.

Olguin et al. [4] studied high-level human behavior with sociometric badges that use RSSI to detect physical proximity. The sociometric badges were tested in several environments, including a set of 60 nurses in a Boston hospital. The focus of this research was to identify individual and group behavior using different sensing modalities including audio and acceleration in addition to physical proximity data.

The ZebraNet [5] deployment observed that the addition of sensors affected subject behavior. In the case of the zebras, biologists observed additional head shakes for the first day. However, this behavior did not affect the data negatively. In our deployment, playing with the motes resulted in some unexpected reboots.

9 Conclusion

Our deployment measured the proximity contacts between 792 students, teachers, and staff and their locations over a typical high-school day. The success of the final deployment hinged largely on a number of lab and field tests. The first helped discover code bugs and determine parameters, such as RSSI and power level. However, working with human subjects had its own unpredictable effects, and small-scale pre-deployments at the school were invaluable.

We collected 6 million log entries, of which 50% were proximity contact beacons. The remaining 3 million were from stationary motes across the school campus indicating location. The design choices and techniques in this study ensured that we could produce complete and accurate data that can be used in the study of infectious disease spread.

Acknowledgements

This research was supported by a National Science Foundation award BCS-0947132, DE-AR-0000018, grants #0615308 and #0846014, a Branco Weiss fellowship to M.S., NICH award 1K01HD051494 to J.H.J., and in part by National Institutes of Health grant GM28016 to M.W.F. We also acknowledge generous gifts from DoCoMo Capital, Foundation Capital, and Microsoft Research. We thank Ignacio Cancino, Elena V Jordan, Alison Brown and Rahel Salathé and members of the Feldman and Levis groups for their help.

10 References

[1] N. Eagle, A. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *PNAS*, 106(36):15274–15278, 2009.

[2] J.-H. Hauer, V. Handziski, and A. Wolisz. Experimental study of the impact of wlan interference on ieee 802.15.4 body area networks. In *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN)*, 2009.

[3] D. Lymberopoulos, Q. Lindsey, and A. Savvides. An empirical characterization of radio signal strength variability in 3-d ieee 802.15.4. In *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN)*, 2005.

[4] D. O. Olguin, P. A. Gloor, and A. Pentland. Capturing individual and group behavior with wearable sensors. In *AAAI Spring Symposium on Human Behavior Modeling*, 2009.

[5] P. Zhang, C. Sadler, S. Lyon, and M. Martonosi. Hardware design experiences with zebranet. In *Proceedings of the ACM Conference on Embedded Networked Systems (SenSys'04)*, Nov 2004.