SYSTEM DESIGN ISSUES IN SINGLE-HOP WIRELESS SENSOR NETWORKS

Sunayana Saha and Peter Bajcsy
National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign, Illinois, USA
ssaha@uiuc.edu, pbajcsy@ncsa.uiuc.edu

Abstract

We present an optimization study of a wireless sensor network with the two objectives of maximizing information content and minimizing the system's wireless information loss. The information content is represented by (a) a reference time point, (b) a time stamp of each sensor reading, (c) raw sensor readings and (d) calibrated sensor readings converted to engineering units. The wireless sensor network's information loss is measured as the number of sensor readings that were acquired but lost before they reached an information gathering place of the sensor network, such as, the base station. A single-hop network consisting of Crossbow Inc.'s MTS-101CA Mica sensors is described and optimized with respect to the two objectives. In contrast to many other research efforts, we focus on a network of sensors that continuously sense the environment and transmit data to the base station. Thus, our studied sensor network inherently generates much heavier traffic and is applicable to monitoring continuous variables for hazard aware environments.

Key Words
Wireless communication, sensor-network system design.

1. Introduction

Any application that requires continuous sensing, processing and wireless information delivery of ambient conditions like temperature, luminance, movement, sound etc, can exploit the power of smart micro electro-mechanical systems (MEMS) sensors, by deploying them in large quantities in the field of interest. Sensor data can be analyzed to make intelligent inferences about the environment in many applications including surveillance of battlefields, habitat monitoring [1] of interesting birds and animals, structural material health monitoring [2], earthquake resistant structural design, and home security or home power consumption control.

Although the recent MEMS sensor technological innovation has shown a significant promise in many application domains, it has also exposed several technical limitations that must be improved. In brief, the limitations include memory and energy constraints, broadcast range, available processing power (CPU), transmission rate, synchronization difficulties and robustness with respect to wireless information loss. Many of these technical limitations can be overcome by optimization of a wireless sensor network design.

Before utilizing smart MEMS sensors in any application, one needs to resolve many system design issues, such as, (a) deciding what is the most effective mechanism to receive information from the wireless sensor network, (b) synchronizing all sensors, (c) assigning time stamps to every sensor reading and (d) evaluating the application environment with respect to the number of deployed sensors, their spatial arrangement and interference from other wireless devices. This paper presents an optimization study of these four system design issues with the two objectives of maximizing the information content (synchronization and time stamps) and minimizing the system's wireless information loss (data acquisition mechanism and sensor arrangements). The novel contribution of this work lies in (a) the investigation of multiple sensor data collection mechanisms and (b) the optimization of multiple sensor network design issues based on the aforementioned design objectives.

The paper is structured as follows: We overview sensor network system design issues in Section 2 and discuss the related work in Section 3. Section 4 describes the sensor network hardware, followed by the proposed solutions for data collection, synchronization and evaluation of data losses in Section 5. Section 6 presents the experimental results and analysis. In Section 7 we draw our conclusions and list the benefits and limitations of our study.

2. Sensor Network System Design

We envision that many applications will use these MEMS sensors to 'continuously' sense an indoor environment. This differs from most other sensor applications where sensors are placed outdoors and data is acquired only under specific conditions like when an interesting event occurs or when prompted by the user. A simple application needing continuous sensing could be, for instance, that of monitoring and detecting gradual changes in the ambience conditions during the course of some experiment in a laboratory. Continuous sensing could also be required in an indoor hazard detection system where the environment must be continuously sensed, and the feedback given to the human inspector. Such monitoring systems would need to sense the environment continuously and send data periodically to a base station. We investigate the factors that affect effective data collection in such an indoor setting of continuously sensing sensor network.

MEMS sensors operate on a limited power supply provided by batteries. Hence, one must try to devise energy-efficient sensor network operations. It has been found that in most sensor networks, communication power is a significant component of the total power consumed. Data losses through collisions and useless transmissions imply loss of precious energy. We try to compare different design alternatives by evaluating the data loss in each. The best design is one that incurs the least data loss.

In this work, we propose, evaluate and optimize (1) a simple single-hop network setup of continuously sensing sensors, (2) an approach to synchronization using a *'RESET'* broadcast signal, (3) a time stamping technique using a CPU clock counter and the *'RESET'* signal, (4) two mechanisms for collecting sensor readings ('*autosend*' and '*query*' data gathering schemes), (5) several spatial sensor arrangements achieved by varying (i) the number of sensors (ii) geometrical arrangement; to provide quantitative results for a wireless smart MEMS system design with minimum information loss. We also determine the amount of data lost due to interference by other commonly used wireless devices.

Although there has been a lot of work on synchronization and communication problems in multi-hop wireless networks [3], [4], [5], [6], [7], we have conducted our study with single-hop networks in order to understand the less complicated networks. The quantitative results from our study can be viewed as lower bounds on information loss for the more complicated multi-hop networks.

3. Related Work

Sensor networks, like any other upcoming field, have many interesting challenges and have been the prime focus of many researchers. A lot of work has been done on transmission protocols for medium access control (MAC) [8], [9], [10] that take into consideration the limitations specific to sensor networks. However, instead of MAC protocols, we are more concerned with data communication protocols at the application level and assume that a reasonable MAC layer is already in place at the lower level. Ideas like 'data aggregation' [6] and 'directed diffusion' [7] are novel paradigms for efficient and energy-aware data communication and should be utilized for multi-hop networks. In a single hop network such as the one we have, data aggregation is not required. Each node transmits data independently to the central processing and logging unit, eliminating the need for complicated aggregation logic on sensor nodes. The aim of our study is to measure the amount of data lost in 'a single-hop, continuously sensing and transmitting network', in an indoor environment. Thus, our sensor network inherently generates heavy traffic.

We have not come across a study similar to ours that would address system design issues under heavy traffic. We have not elaborated some design issues like, time synchronization, in depth because many researchers have already devoted much effort on this topic [3].

4. Sensor Network Hardware

For our experiments, we used the MTS-101CA Mica sensors provided by Crossbow Inc [11]. These sensors are programmed with TinyOS [12], an open source code operating system developed by researchers at the University of California, Berkeley and actively supported by a large community of users. Each sensor is equipped with the following (1) A thermistor and photo sensor, (2) 4 MHz Atmega 128L processor, (3) 128K bytes Flash, 4K bytes SRAM and 4K bytes of EEPROM, (4) 916MHz radio transceiver with a maximum data rate of 40Kbits/sec, and (5) Attached AA (2) battery pack.

5. Proposed Sensor Network Design Solutions

For the purpose of our experiments we set up our sensor network to consist of MEMS sensor nodes and a base station. We now give a description of each sensor's software configuration and the techniques used for synchronization, time stamping, data collection and evaluation of data losses.

5.1 Sensor Software Configuration

The Mica sensors were placed in an indoor laboratory. Each sensor was programmed to record continuous values of temperature and luminance of the point where it was placed. After every 100 milliseconds, the sensor mote would store a temperature value in a local array. The luminance values were stored in a separate local array. Because of our concern for the limited memory on the mote, we fixed the size of each of the two local arrays to hold only 10 readings. Also, we preferred transmissions of smaller packet sizes. Bigger packets have more chances of collision and corruption. Depending on the scheme used for data collection (discussed later), a sensor would send a packet containing 10 readings to the base station. The base station would save this data to a file for interpretation and analysis later.

5.2 Sensor Node Synchronization

Synchronization of the sensors is a critical need of most applications and has several implications: (i) Temporal correlation of the readings of one sensor with another (ii) Energy savings due to sensing and transmitting only when the base station is interested in the data. (iii) Temporal correlation of the sensor readings with other devices, for example, cameras. We handled the issue of synchronization by instructing the sensors to wait for a *'RESET'* signal from the base station. Until they receive this signal, they do not start sensing the environment. The base station broadcasts the *'RESET'* message. Once this message is received, a sensor starts sensing the ambience.

Theoretically, the precision to which the nodes are synchronized with each other depends on the possible sources of synchronization message latency – send time, access time, propagation time and receive time [3]. If we only consider propagation time, then inter-node synchronization can be estimated as follows. Suppose the propagation delay of *'RESET'* message from the base station to a node is *'p'* time units. Then, if all sensors are placed at an equal distance from the base station, they will get the *'RESET'* message *'p'* time units after it is broadcasted. Thus they will all be perfectly synchronized. If the sensors are placed at varying distances from the base station, they would get the *'RESET'* message between p_{min} and p_{max} time units, where p_{min} is the propagation delay to the closest sensor and p_{max} is the propagation delay to the farthest sensor. In this case, the nodes would be synchronized within a bound of '$p_{max} - p_{min}$' time units.

Another hurdle for synchronization is clock-drift. The hardware clocks on different nodes may count time at slightly differing rates. With progressing time, a particular sensor may be far ahead or behind some other sensor. This factor, together with the slight propagation delay of the synchronization message (as described above) can lead to a significant offset between different sensor timers, thus making them out of sync. A simple solution to rectify this asynchrony is by specifying an upper bound 'E' for this asynchrony, and re-synchronizing the sensors at time 't', after which the error goes beyond 'E'. If we know the clock drift rate 'd', propagation delays p_{min} and p_{max} and the error bound 'E', calculating 't' is straightforward.

Let $t_{diff} = p_{max} - p_{min}$. In the beginning, a node may be $(t_{diff}(1 \pm d))$ time-units out of sync with another node. After t time units, the maximum time units a node may be out of sync with another node is $(t_{diff}(1 \pm d)) \pm (2td)$. Bounding this by E, we get $|(t_{diff} (1 \pm d)) \pm (2td)| < E$. Solving to get an upper bound, we get $t < (E-(t_{diff} (1+ d)))/2d$. This gives the maximum value for t after which the nodes become asynchronous and need to be re-synchronized.

5.3 Time Stamping Sensor Readings

Sensors do not have any notion of clock time, which is available on bigger computational devices. However, sensors do have a timer, which can be programmed to start at some instance and then repeat after a fixed interval. We start the timer on each sensor when we are synchronizing the sensors with the *'RESET'* message. The timer is set to repeat after every 100 milliseconds. Subsequently, we maintain a counter (called *'count'*) that is incremented by one, after every timer interval. This counter will give us the time relative to the timer start time. A counter value of 10 would thus mean that 10 * 100 = 1000 milliseconds have elapsed since the start of the timer. Moreover, since all the sensors start their timers at the same time (when they are synchronized) a counter value of *'c'* on two or more sensors would refer to the same time instance.

This counter value is copied into the packet sent to the base station. We realize that sending the counter value wirelessly to the base station consumes bandwidth, but it is important for correlating sets of readings. We try to save bandwidth by transmitting a single counter value for all the readings in a packet instead of sending a counter value for each reading in the packet.

5.4 Wireless Data Collection Schemes

Our goal was to find a simple data collection approach that would ensure minimum data loss in transmissions to the base station. We evaluated the following two different schemes for collecting data from the sensors:

- *'Autosend'* scheme: A sensor sends a packet to the base station as soon as it has 10 readings in its local array. Ideally, a sensor would transmit a packet for the temperature readings and a packet for the luminance readings after every one second (100 milliseconds/reading * 10 readings/packet = 1 second/packet).

- *'Query'* scheme: The base station queries each sensor mote in a round robin fashion to send the fixed number of readings. When a mote receives the query, it checks to see if a packet with 10 readings can be sent. If yes, it is sent immediately. Otherwise, it simply sets a local flag indicating that the base station query is pending. When 10 readings have been collected and the base station query is still pending, then a packet with the readings is sent.

There are reasons why we choose the above two schemes. The *'autosend'* scheme is appealing because of its simplicity. Each node works as an independent unit and transmits a packet to the base station whenever a packet worth of data is ready. The base station too has no other responsibility than to collect and log data flowing towards it. However, since there is no control on any node's transmissions, there are bound to be collisions. This is different in the *'query'* scheme, where we can control a node's transmission to a certain extent.

5.5 Evaluation approach

In this section we outline our approach in estimating the data losses and consider wireless sensor network variables.

5.5.1 Determining the amount of data lost

We compare two or more sensor network setups on the basis of the data loss in each of them. All data losses are calculated at the base station. The base station starts tracking the losses as soon as it sends the *'RESET'* message to synchronize and start the sensor nodes. Data loss is measured in terms of the total number of missing readings, from all the sensor nodes. Detecting losses is based on the following observation:

If, at the base station, the previous packet from mote *'a'* had a counter value of 'x', then the next packet from *'a'* should have a counter value of *'x + 10'*, since the previous packet contained 10 readings. If however, the next packet counter *'y'*, is greater than *'x + 10'*, then the readings for counter values between 'y' and 'x + 10' are missing i.e., 'y – (x + 10)' readings are considered lost.

At the end of the experiment, the data loss is calculated as a percentage of missing readings, which is the ratio of the total number of missing readings from all the motes, to the total number of (missing + correct) readings from all motes.

5.5.2 Wireless sensor network variables

In order to infer the best network design for any application, there are several variables to consider and test the data losses against. Our experiments measure the data losses as a function of the following:

a. Number of sensor nodes – the number of nodes in the network is increased from 1 to 7 nodes.

b. Spatial Arrangement of the sensors. There were three mote arrangements that we tried to compare and evaluate. They were: (1) Nodes arranged in a *'straight-line'*, within 10 to 15 inches from the base station, 3 inches from each other. (2) Motes arranged in a *'circular'* fashion around the base station with a radial distance of 10 inches between the base station and a mote. (3) Nodes scattered in a *'random'* fashion in the laboratory – at a distance anywhere from 10 inches to 150 inches from the base station.

c. Data collection technique – '*autosend*' versus '*query*' scheme of data collection.

d. Interference from other wireless devices – the sensor node's transmission was tested in the presence of other devices that could possibly interfere with the sensor network transmissions.

At this stage, we have only pointed out collisions as the reason for data losses, common to the two schemes of data collection. There may be a few data loss factors specific to a particular scheme, but we will elucidate them while analyzing the results for each scheme.

6. Experimental Results and Analysis

We will first analyze the two data collection schemes individually against increasing number of nodes and the three sensor node spatial arrangements we mentioned above. The experiments to test interference with other wireless devices are only conducted with the best data collection scheme. All sensor nodes were provided with fresh batteries at the start of the experiment, and thus have similar battery power. This is important to mention, as battery power determines the strength of data transmissions and receptions.

6.1 '*Autosend*' Vs. '*Query*' Scheme of Data Collection

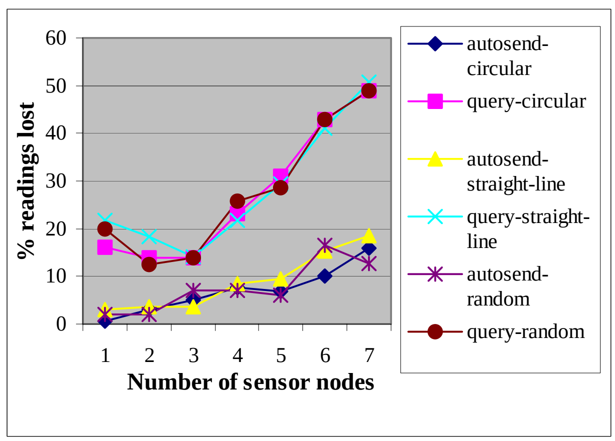

Figure 1 Percentage of readings lost in *'autosend'* and *'query'* data collection schemes

The graph in Figure 1 shows the percentage of readings lost as a function of increasing number of motes for each of the three mote spatial arrangements, for *'autosend'* and *'query'* schemes of data collection.

We note that, in general, in all the network spatial arrangements, the percentage of readings lost increases as the number of motes increases. The increase in the percentage of readings lost is gradual in the beginning but spikes drastically as we go on increasing the number of motes. This is because, the number of transmissions increases linearly with the number of motes. The percentage of readings lost in *'query'* scheme is at least 10% higher than the readings lost in *'autosend'* scheme.

As the number of nodes increases, the losses in '*query'* scheme shoot to about 50% while those in '*autosend'* stay around 15%. Thus, it is obvious that in terms of the number of readings lost, the '*autosend'* scheme is far superior to the *'query'* scheme.

Readings are mostly lost in '*autosend'* scheme due to collisions or initial startup stabilization. However, in the *'query'* scheme, apart from readings lost due to collisions and initial stabilization, there are other latent reasons that could be causing the large number of reading losses.

For example, imagine a situation in the '*query'* scheme where a node has a packet with the required number of readings ready, but has not being queried by the base station. Since we have limited the amount of data stored in the local array on a node to 10 readings only, in the next timer interval, even though there is a reading to be stored, it is dropped. Readings are dropped until this mote is queried and the local array is set to blank again. The more the number of motes, the longer a particular mote will have to wait before it is queried, and thus more and more readings will be dropped. This would be a problem in any *'query'* scheme operating in a continuous sensing application, unless we remove the limitation on the amount of data stored on a mote.

Another problem with *'query'* scheme occurs when the base station queries a node when it has just started filling its local array. This node will not have 10 readings in the 300 milliseconds for which the base station waits, before it queries another mote. In this case, this node will send the packet later, when is has 10 readings, increasing a possibility of collision with another mote's transmission.

Another problem specific to the '*query'* scheme is if a node dies out in between. Since nodes are queried in a round robin fashion, the transmission slot of the dead node goes wasted. Not only that, at this time other motes may start loosing readings too. Such a failure has no affect on the '*autosend'* scheme, but adversely affects the *'query'* scheme.

One might wonder why the data loss in '*autosend'* scheme is not 100% since nodes are synchronized in the beginning and will send a packet with 10 readings simultaneously, every second. This could be credited to the CSMA [13], [14], a medium access control (MAC) protocol, which is implemented in the lower levels of the TinyOS and is responsible for transmissions. It could also be due to the loss of sync between the sensor nodes (described earlier).

We concluded based on the experimental results that the *'autosend'* scheme is better than the *'query*' scheme.

6.2 Spatial Arrangements of Sensors

For the '*autosend'* scheme, the circular arrangement seems to give minimum data losses in almost all cases. The random arrangement gives low data losses in some cases, but in other cases gives the highest losses.

In '*query'* scheme, there does not seem to be a clear winner among the three different mote arrangements. It appears that the spatial arrangements do not affect the performance of the '*query'* scheme as much as other reasons like wait-time before a mote is queried and collisions due to out-of-turn transmissions.

6.3 Interference from other devices

The Mica motes use a radio frequency of 916 MHz for wireless transmissions. In this experiment we tried to determine the extent of interference by the following commonly used indoor devices that use radio frequencies for their operations:

1. Telephones and wireless video transmitters: Many powerful cordless telephones and wireless video transmitters use frequencies in the 900 MHz range. We tested the performance of our sensor network in the presence of such a telephone (EnGenius SN920) and a video transmitter (accompanying CCTV-900 receivers). The results were disheartening because our sensor network had almost 100% data loss the moment these devices were switched on. These losses could not be reduced even when we tried to increase the distance between the transmitting devices (telephone and camera) and the sensors, or decrease the distance between the base station and our sensors.

2. Wireless LAN (802.11b): 802.11b wireless LANs are commonly deployed in offices. These networks use a frequency of 2.4 GHz and do not interfere with our sensor network.

3. Wireless audio transmitters: we operated our network in the presence of audio-technica's ATW-3110D audio transmitters that use frequencies between 655-680 MHz for transmitting sound information to their corresponding receivers. They did not interference with our network.

4. Other independently existing sensor motes: it is possible that there is another sensor network close to ours, which could interfere with our sensors' transmissions. We simulated such an independent network by placing some sensors in the laboratory, which would broadcast meaningless data after every 150 milliseconds. The purpose of this setup was to determine how, similarly powered devices, using the same 916 MHz frequency but working independently (asynchronously), could hinder our network performance.

Figure 2 shows that the data losses in our network increased significantly as the number of sensor motes in the 'other' independent sensor network increased.

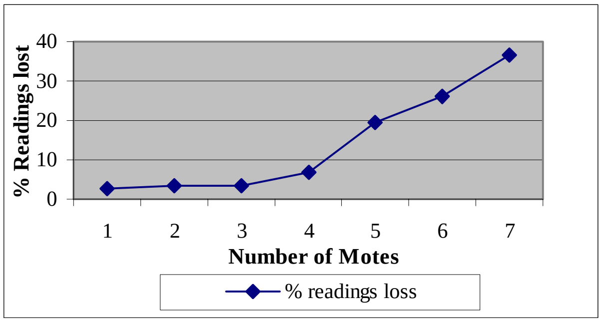

Figure 2. Percentage of readings lost as a function of number of motes in 'other' network

The 'other' network is totally out-of-sync with our network and since its nodes are transmitting at a much faster rate than our node in *'autosend'* scheme, maybe our node's transmissions are suffering more collisions and lesser useful data is being received at the base station.

The experiments in this category have shown that MEMS sensors' transmissions are susceptible to interference from other devices that use the same frequency range for transmissions. We have realized the Mica motes are relatively low powered and, in the presence of more powerful devices like the wireless camera and telephones, a sensor network is prone to total failure. In the presence of similarly powered devices, operating asynchronously with our network, the loss increases linearly as the number of nodes in the 'other' network goes up.

7. Summary

In this study we have discussed many design issues that concern a sensor application. We have tried to evaluate different sensor network setups by estimating the effectiveness of each. We conducted experiments and analyzed the results to find the best method to deploy sensors as well as to gather data from them in a single hop indoor sensor network.

Between the two data collection techniques – *'autosend'* and *'query'*, *'autosend'* gave lesser data losses. With the *'query'* scheme there are bound to be losses unless the base station and the sensor nodes are perfectly synchronized and, nodes do not fail in between. An application that requires constant monitoring and sensing of the environment will need to transmit data continuously to the base station to avoid memory shortage on the sensors. According to our experiments, such an application should be designed to use the *'autosend'* scheme for data collection. Our experiments have also helped us realize the potential problems that may arise due to the presence of other devices that can interfere with sensor network's radio transmissions.

This study investigated smart MEMS network system design issues. The results of this study will guide our future application designs and help in building more robust wireless sensor networks.

References

[1] A. Mainwaring, J. Polastre, R. Szewczyk & D. Culler, Wireless Sensor Networks for Habitat Monitoring, *Intel Research, IRB-TR-02-006*, June 2002.

[2]NIST's SHIELD Project - http://jazz.nist.gov/atpcf/prjbriefs/prjbrief.cfm?ProjectNumber=00-00-5306

[3] J. Elson & D. Estrin, Time Synchronization for Wireless Sensor Networks, *Proc. 15th International Parallel and Distributed Processing Symposium*, San Francisco, CA, 2001.

[4] D.A. Maltz, J. Broch & D.B. Johnson, Experiences Designing and Building a Multi-Hop Wireless Ad Hoc Network Testbed, *CMU Scholl of Computer Science Technical Report CMU-CS-99-116*. March 1999.

[5] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin & S. Wicker, Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks, *Technical Report CSD-TR 02-0013, UCLA*, February 2002.

[6] B. Krishnamachari, D. Estrin & S. Wicker, Modelling Data-Centric Routing in Wireless Sensor Networks, *Proc. IEEE INFOCOM*, New York, NY 2002.

[7] C. Intanagonwiwat, R. Govindan & D. Estrin, Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, *Proc. 6th ACM/IEEE International Conf. on Mobile Computing and Networks*, Boston, MA, 2000. 56-67.

[8] C.S. Raghavendra & S. Singh, PAMAS: Power Aware Multi-Access protocol with signaling for ad hoc networks, *ACM Computer Communication Review*, *28*(3), 1998, 5-26.

[9] A. Woo & D. Culler, A transmission control scheme for media access in sensor networks, *Proc. 7th ACM/IEEE International Conf. on Mobile Computing and Networking*, Rome, Italy, 2001, 221-235.

[10] W. Ye, J. Heidemann & D. Estrin, An Enery-Efficient MAC protocol for Wireless Sensor Networks, *Proc. IEEE INFOCOM*, New York, NY 2002. 1567-1576.

[11] Crossbow Inc. http://www.xbow.com, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA.pdf

[12] TinyOS official website: http://webs.cs.berkeley.edu/tos

[13] L. Kleinrock & R. Tobagi. Packet switching in radio channels: Part 1-Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics, *IEEE Transactions on Communications, 23*(12), 1975, 1400-1416.

[14] S.S. Lam, A carrier sense multiple access protocol for local networks, *Computer Networks, 4*(1),1980, 21-32.