Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks

Sukun Kim†, Shamim Pakzad‡, David Culler†, James Demmel†
Gregory Fenves‡, Steven Glaser‡, Martin Turon*
{binetude, culler, demmel}@eecs.berkeley.edu {shamimp, fenves, glaser}@ce.berkeley.edu mturon@xbow.com
†Electrical Engineering and Computer Sciences and ‡Civil and Environmental Engineering
University of California at Berkeley
Berkeley, CA 94720
*Crossbow Technology, Inc.
4145 N. First Street
San Jose, CA 95134

ABSTRACT

A Wireless Sensor Network (WSN) for Structural Health Monitoring (SHM) is designed, implemented, deployed and tested on the 4200ft long main span and the south tower of the Golden Gate Bridge (GGB). Ambient structural vibrations are reliably measured at a low cost and without interfering with the operation of the bridge. Requirements that SHM imposes on WSN are identified and new solutions to meet these requirements are proposed and implemented. In the GGB deployment, 64 nodes are distributed over the main span and the tower, collecting ambient vibrations synchronously at 1kHz rate, with less than 10μs jitter, and with an accuracy of 30μG. The sampled data is collected reliably over a 46-hop network, with a bandwidth of 441B/s at the 46th hop. The collected data agrees with theoretical models and previous studies of the bridge. The deployment is the largest WSN for SHM.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and Embedded Systems

General Terms

Experimentation, Reliability, Design

Keywords

Wireless Sensor Networks, Structural Health Monitoring, Deployment, Large-Scale

1. INTRODUCTION

Structural Health Monitoring (SHM) is a technology that allows the estimation of the structural state and detection of structural change that affects the performance of a structure. Two discriminating factors in SHM are the time-scale of the change (how quickly the state changes) and the severity of the change. These factors represent two major sources of system change: alarm warnings [27] (e.g. disaster notification for earthquake, explosion, etc.) and continuous health monitoring (e.g. from ambient vibrations, wind, etc.). The general approaches taken to SHM are either direct damage detection (visual inspection, x-ray, etc.) or indirect damage detection (detecting changes in structural properties or system behavior). This paper describes a platform for indirect detection of structural state through the measurement and interpretation of ambient vibrations and strong motion. The chosen test bed is the Golden Gate Bridge in San Francisco Bay. Performing SHM by the use of sensor networks is not a new concept [20, 9]. The traditional approach consists of conventional piezoelectric accelerometers hardwired to data acquisition boards residing in a PC. The drawbacks of such a system include (1) the high cost of installation and disturbance of the normal operation of the structure due to wires having to run all over the structure, (2) the high cost of equipment; and (3) cost of maintenance. Compared to the conventional methods, Wireless Sensor Networks (WSN) provide comparable functionality at a much lower price, which permits a higher spatial density of sensors. The prototype wireless system presented in this paper costs about $600 per node compared to thousands of dollars for a node with the same functionalities in a traditional PC-based wired network. Compared to the wired network, installation and maintenance are easy and inexpensive in a WSN, and disruption of the operation of the structure is minimal.

This work has three main contributions to WSN for SHM:

- It identifies requirements to obtain data of sufficient quality to have real scientific value to civil engineering researchers, and examines how to solve them.
- The system is scalable to a large number of nodes to allow dense sensor coverage of real-world structures. For example, a long-lived 46-hop network was implemented on the Golden Gate Bridge (Figure 1 and 2).
- It addresses a myriad of problems encountered in a real deployment in difficult conditions, rather than a simulation or laboratory test bed.

A WSN for SHM was deployed on the Golden Gate Bridge (GGB), see Figure 1. The 46-hop system consists of 64 nodes, which measure ambient vibrations with an accuracy of 30μG. The ambient vibrations were sampled at 1kHz with a time aperture less than 10μs. Figure 2 illustrates the bandwidth obtained by Straw, a new reliable data collection component written for this installation. The system provided high bandwidth data streaming of 441B/s from the 46th hop to the base station by implementing pipelining. This 46-hop wireless sensor network is the largest number-of-hop installation reported in the literature up to now.

This paper will explain how the system was designed and implemented to achieve this successful deployment on the bridge. The

IPSN'07, April 25-27, 2007, Cambridge, Massachusetts, USA.

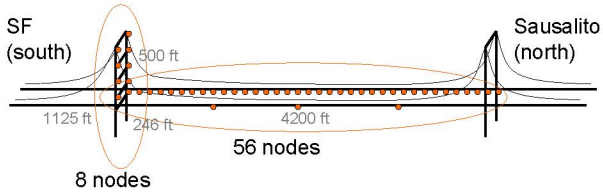


Figure 1: The Golden Gate Bridge and layout of nodes on the bridge. To cover this large bridge, long linear topology needs be used, and it brings challenges to the network.

later part of the paper will present an analysis of the collected acceleration data recorded along a linearly dense array. Six major requirements of SHM on WSN are identified here.

1. The data acquisition system had to be able to detect signals with peak amplitudes as low as 500μG [7]. The installation had to minimize sources of distortion such as the noise floor of the system (including accelerometer, amplifier, analog to digital converter, etc.), installation error, and temperature variation.
2. Because of structural interest in local modes of vibration, a sampling rate of 1kHz was chosen as the target rate. This rate and the need for 16-bit digitization accuracy require low jitter, i.e. low time uncertainty of the sampling intervals.
3. Time synchronization in sampling through the bridge is required to perform correlation analyses of the structural vibrations. This was particularly challenging due to the drift of independent clocks at each of the 64 nodes. An earlier reported solution to the time synchronization problem is FTSP [19].
4. The GGB installation required a large-scale multi-hop network due to the great length of the main span and the fact that the aggregator station could only be located in the base of the south tower. One existing solution to the collection network is MintRoute [26].
5. Commands had to reliably disseminate throughout the entire system so that all parts of the network could start on command, and insure against lost data or a blockage of hopping. Repeated Broadcast [2] can be one solution.
6. Data must be transferred reliably. Vibration data, in this case, is too valuable to be lost to communication error.

2. RELATED WORK

WSN applications can be divided into two categories. The first category is environmental monitoring; networks deployed in Great Duck Island [22] and a Redwood forest [24] are examples of this class. For this class of problem the focus is on networks with low duty-cycle and low power consumption. The second category of WSN applications consists of applications that require identification of a mechanical system through a measured system response. Health monitoring of mechanical machines [16], condition-based monitoring, volcano monitoring [25], earthquake monitoring [11], and structural health monitoring [21] belong to this class, which generally require high fidelity sampling. The focus of this paper is to address the requirements of the latter category. Related work on using WSN in SHM includes [10, 8, 14, 18]. However, these networks generally do not scale to a long enough multi-hop network needed to cover a large structure, and have not been implemented and tested in a harsh real-life environment. Wisden [27] and

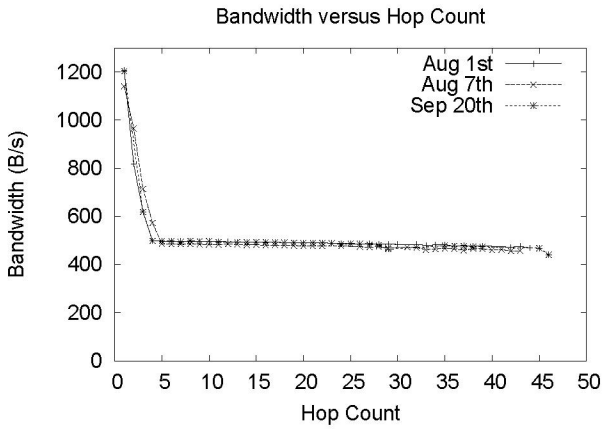


Figure 2: Bandwidth of Straw at the Golden Gate Bridge. It works over a 46-hop network. To sustain high bandwidth over a long path, pipelining is used avoiding interference.

Tenet [12] satisfy many requirements. They provide reliable data collection over multi-hop network. However, Wisden can sample only up to 160Hz, and Tenet demonstrated only 50Hz sampling, which is far below the threshold needed for structural health monitoring. Wisden and Tenet have not been analyzed for sampling jitter, which is needed in determining to what degree the resulting data has confidence for analysis in civil engineering. They are tested only in a small-scale indoor test bed. The most critical pitfall of these two systems is that they do not produce time-synchronized data. Wisden has a time stamp on each sample. However, the input for basic modal property analysis is a matrix of time-synchronized samples from multiple nodes. The data produced by Wisden and Tenet has no value for meaningful structural analysis.

3. OVERALL ARCHITECTURE

The wireless network is composed of multiple nodes and a base station. A node consists of a mote and a sensor board. The node measures vibration at two different orders of dynamic bandwidth, with the data communicated back to the base station through wireless communication. The base station is a server providing more computational power and larger storage than a mote node, and possibly a connection to the Internet. In the GGB deployment a laptop is used as a base station. The software architecture of the GGB nodes uses new components integrated into the TinyOS [13] infrastructure to satisfy the six requirements discussed above. Figure 3 illustrated the overall software structure. A low latency dissemination service is required so that the commands are not delivered after they are needed, therefore Broadcast [2] was used in place of Drip [23]. Drip provides dissemination service with an eventual reliability but has long latency. Even though Broadcast provides unreliable dissemination service, with repeated broadcast 100% even-

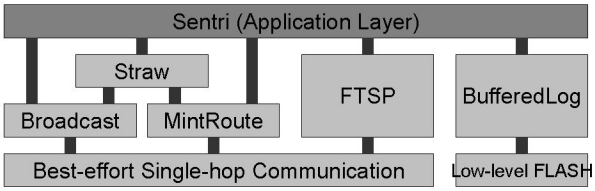


Figure 3: Overall Software Architecture

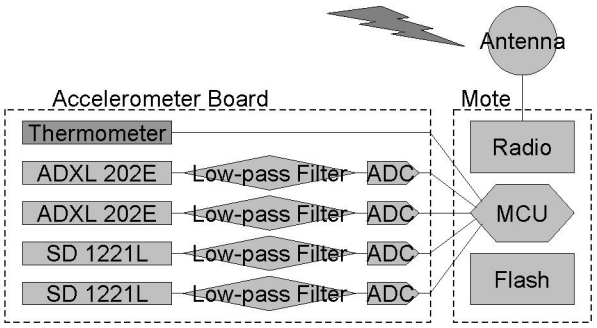


Figure 4: Hardware Block Diagram. Details of two accelerometers (ADXL 202E and SD 1221L) are in Table 1. A thermometer is used for temperature calibration.

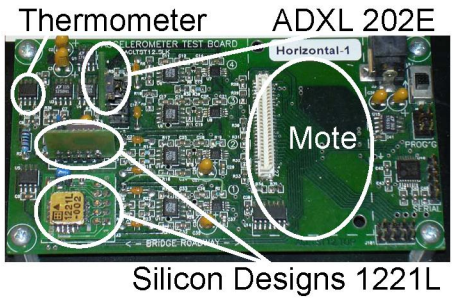


Figure 5: Accelerometer Board. ADXL 202E has two axis in a single chip. Either Mica2 or MicaZ can be used as a mote.

Table 1: Comparison of the Two Accelerometers. G means the acceleration of gravity.

	ADXL 202E	Silicon Designs 1221L
Type	MEMS	MEMS
Range of System	-2G to 2G	-0.1G to 0.1G
System noise floor	$200(\mu G/\sqrt{Hz})$	$32(\mu G/\sqrt{Hz})$
Price	$10	$150

tual reliability can be achieved in practice. MintRoute [26] was used for information reply since it provides a best-effort multi-hop convergence routing. Our new reliable data collection layer Straw lies above Broadcast and MintRoute. For time synchronization, FTSP [19] is used. BufferedLog [3] supports high frequency sampling with light-weight logging.

Structural hEalth moNiToRing toolkIt (Sentri) is an application layer program which drives all components. Instead of a standalone program, Sentri is structured like an RPC server: for every operation a command is sent from the base station to a node. In SHM, motes are heavily used and heavy traffic makes network bandwidth the bottleneck of the operation; therefore, additional processing and traffic overhead must be avoided. However, since the project is in the research stage in both system engineering and civil engineering and the operation sequences and parameters were changing frequently, the operational model was necessary. It allows us to figure out precisely which parts of the signals are more valuable, and to fine-tune the system parameters in an interactive process. This process was first tested and verified through a trial deployment of several nodes on a model steel bridge at UC Berkeley and then again on a footbridge over I-80 [21], both of which were used to determine system settings. Sentri provides 16 operations and the command for each operation is contained in a single packet. Operations like reset, erase flash, start sensing, and reading meta-data are examples of such high-level Sentri commands, which provide a great deal of flexibility and can be sent in sequence by the base station.

4. DATA ACQUISITION SYSTEM

Figure 4 shows an overview of the hardware as a block diagram. The data acquisition system performs three primary functions: sensing, signal processing and communication. Because of long experience with the product, Crossbow MicaZ [5] motes were used for control and communications. The analog signals output by the low-noise accelerometers pass through low-pass antialiasing filters on the way to a 16-bit analog-to-digital converter, before the data is first logged into the flash of the mote and then wirelessly transmitted.

4.1 ACCELEROMETER SENSOR BOARD

A new accelerometer board [5], shown in Figure 5, was designed for SHM applications. The board has four independent accelerometer channels monitoring two directions (vertical and transverse), and a thermometer to measure accelerometer temperature for compensation purposes. Low-amplitude ambient vibrations, due to wind loading and traffic, are resolved by a two-dimensional SiliconDesigns 1221L accelerometer. A low-cost ADXL202E two-dimensional accelerometer was used to monitor stronger shaking as might be expected from earthquake excitation. Because input battery power can vary between 6V and 12V, the sensor board contains a voltage regulator to provide a constant 3V output for the mote and a constant 5V output for the ratiometric accelerometers. Table 1 presents the characteristics of the two accelerometers used, and associated analog circuits. Two simple filters are used on the board. One is a hardware-implemented single-pole 6db low-pass filter with a cutoff frequency of 25Hz. Since the on-board ADC quantizes much faster than the target sampling frequency, this extra capacity allows on-the-fly digital filtering after a factor of $S_{over} = 10$ oversampling, and then averaging the samples before logging. Assuming a Gaussian distribution for the noise, oversampling by a factor of $S_{over} = 10$ reduces the noise level by a factor of $\sqrt{S_{over}} \simeq 3.16$.

Another key hardware consideration in WSN is power consumption. The high duty-cycle required by vibration SHM produces data sets that are between two to four orders of magnitude larger than that of an environmental monitoring application. In contrast to environmental monitoring, this application requires continuous operation. The gain from duty-cycling does not provide compelling savings compared to its complexity and overhead, so duty-cycling is not used. The higher data volume requires sophisticated onboard computation with a distributed system identification algorithm (which is expensive in terms of energy), or all the data needs to be transmitted to a base station for further processing (which is even more power-expensive). The use of batteries or other renewable sources of energy is justified for quick and temporary applications, or where a more permanent power source cannot be provided. An analysis of the power consumption of the boards was performed to determine the size of the batteries. In the deployment at the Golden Gate Bridge, 4 lantern batteries are used for each node. Table 2 shows the actual power consumption profile of a complete sensor unit, from which it is seen that the sensor board by itself consumes about twice the energy of the mote. The board design had a single power path for the mote, sensors, and ADC.

Table 2: Power Consumption in Various Operational Situations (9V input voltage). Idle is when both the sensor board and the mote are turned on, but are not performing any operation.

Situation	Consumption (mW)
Board Only	240.3
Mote Only	117.9
Idle	358.2
One LED On	383.4
Erasing Flash	672.3
Sampling	358.2
Transferring Data	388.8

Significantly lower energy consumption could be realized if only the mote is directly connected to the battery, so that all other components can be turned off when the unit is not collecting data.

4.2 CALIBRATION

The static noise floor of our accelerometer devices was quantified in the Berkeley Seismological Laboratory underground seismometer calibration vault. Testing showed that the SiliconDesigns 1221L devices have a noise floor of $32\mu G/\sqrt{Hz}$, which is small enough to allow resolution of the ambient vibrations of most structural systems. Examples of similar measurement systems in civil infrastructures can be found in [7]. Shaking table tests with patterns ranging from 0.5Hz to 8Hz were performed to study the dynamic behavior of the accelerometers. The accelerometers perform well within the expected dynamic range [21]. Each accelerometer channel was range-calibrated using a tilt test process. The boards were attached to a tilting machine [6], which has a rotational accuracy of 0.001 degree, and the digital output correlated to each angle tested. All four channels showed linear response, and the testing provided offset and scale factors. Prototype accelerometers were also tested in an oven to study the response of the devices to temperature. The tests showed that not only were the accelerometer chips sensitive to temperature, but also they are sensitive to the rate of temperature change as well demonstrating hysteretic response to external temperature fluctuations [15]. Calibration of each channel with respect to temperature is necessary for accurate results, especially when the temperature varies throughout the network.

5. HIGH FREQUENCY SAMPLING WITH LOW TIME UNCERTAINTY

The fundamental frequencies of most civil structures lie below 10Hz. Since the noise level is usually high in uncontrolled structural environments, over-sampling is generally performed to improve the signal-to-noise ratio by reducing the relative noise energy. A sampling rate of 200Hz was chosen as the target logged sampling rate for this study [21]. In order to allow on-the-fly digital filtering (smoothing), it was decided to digitize by a factor of five faster to a target-sampling rate of 1kHz. At this relatively high sampling rate, it is essential to cap the time uncertainty – jitter – in order to guarantee time synchronization in the node and across the network.

There are two primary sources for jitter: temporal jitter and spatial jitter (see Figure 6). Temporal jitter takes place inside a node because the software system cannot keep up with aggressive sampling and logging. Spatial jitter occurs between different nodes because of variation in mote oscillator crystals and imperfect time synchronization; internal clocks of different nodes in the network remain slightly untuned with each other even after the software time-sync

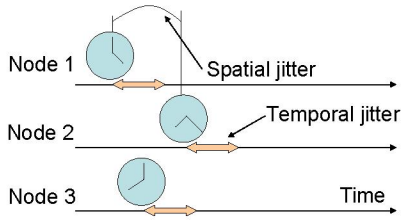


Figure 6: Sources of Jitter. Both temporal jitter and spatial jitter should be within a threshold for the data to have scientific value.

component declares them to be in sync. For a target-sampling rate of 200Hz, a total jitter of 250μs or 5% of the sampling interval was selected as the cap to total jitter. A study of the time synchronization component FTSP showed that it caps jitter at 67μs over a fifty-nine node eleven-hop network [19], so spatial jitter in this case is within the tolerance range. Temporal jitter can become larger than spatial jitter during periods of high-speed data collection, so this was studied in detail. In particular, we explored and modeled temporal jitter, and show that our model matches measured data. We will also show that jitter cannot be completely removed without adding another microcontroller.

5.1 TEMPORAL JITTER ANALYSIS

A statistical model may not catch every minute detail of the temporal jitter process, but it will provide understanding of the distribution of temporal jitter. The timer event for sampling ticks at uniform intervals is graphically presented in the upper portion of Figure 7. When the timer event fires, the CPU can be in the middle of servicing other tasks, such as writing data from RAM to flash. When the CPU is servicing an atomic section, the timer event is delayed, shown in the lower part of Figure 7.

Let N be the number of atomic sections and C be the context switch time when a timer event occurs while the CPU is executing a preemptible section. For modeling purposes, it is assumed that C is constant regardless of the code running. Furthermore let T_i be the length of atomic section i, P_i be the probability of atomic section i running on the CPU when a timer event occurs and $X(i)$ be a random variable which is the remaining execution time of atomic section i running on the CPU when a timer event happens. It is assumed that $X(i)$ is uniformly distributed in $[0, T_i]$. First, assume $N = 1$. The left graph in Figure 8 shows the distribution of jitter. The first peak at 0 indicates the case where no job is running on the CPU when a timer event occurs. The peak at C belongs to the case where a preemptible code is running when the timer event occurs. The constant portion above C is the case where an atomic section i is running on the CPU when a timer event occurs. The middle graph of Figure 8 shows the general case where $N > 1$.

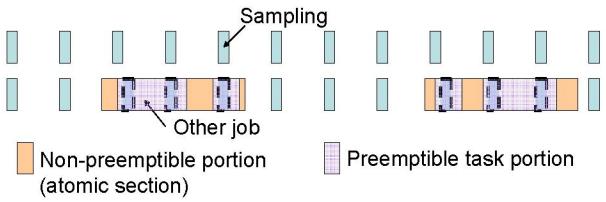


Figure 7: Causes of Temporal Jitter. The atomic section blocks and delays the task of sampling.

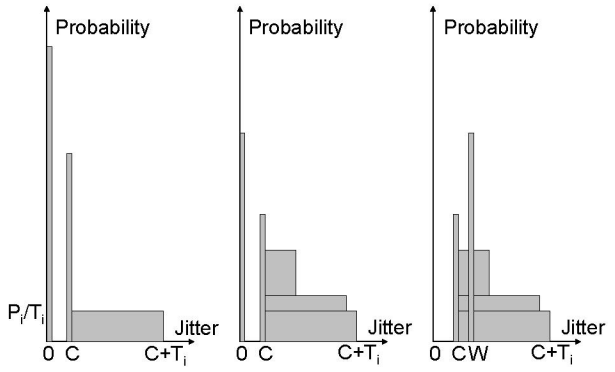


Figure 8: (Left) One Atomic Section, (Middle) Multiple Atomic Sections, (Right) Multiple Atomic Sections with CPU Sleep

The right graph of Figure 8 incorporates the effect of CPU sleep; the CPU goes into sleep mode when no job is running. Let W be the wakeup time; then the peak at 0 moves to W. In fact the entire graph can be moved to the left by C, because consistent jitter of C can be removed.

5.2 TEMPORAL JITTER CONTROL

For high-frequency timer events, the MicroTimer [1] is used instead of the Timer component [4]. The timer component of TinyOS can only trigger at 200Hz. While MicroTimer supports only a single trigger, it can trigger to at least 10kHz. The BufferedLog component [3] is used for light-weight flash writing at high frequency. It is clear from the jitter analysis in the previous subsection that the worst case of jitter is determined by the longest atomic section which can run on the CPU when the timer event occurs. This implies that the best way to reduce temporal jitter is to eliminate any chance that an unnecessary component's atomic section is running on the CPU by turning off every component except the flash during sampling.

A jitter test was performed by turning off all unnecessary components on the CPU. Figure 9 shows the time series of the jitter test. A 5μs jitter means the data is sampled 5μs later than it should be. Two sections are evident in these time series: a section where there is a significant variation in when the data is written (noisy) followed by a section when there is little variation (quiet). The noisy section is a result of writing the buffer to flash memory as a background task. The quiet sections are when sampling occurs without the interference of writing to the flash memory. The same test was performed for sampling rates of 1kHz, 2kHz, and 6.67kHz, with the jitter making up a higher proportion of the read cycle. At a 6.67kHz sampling rate, flash memory write affected most of the sampling period; this can be explained by the overhead of sampling itself. During the quiet sampling sections of the time-history plot, there is a constant delay for every sample. This delay is the time required to wake up the CPU between samples. When the CPU is idle, it enters a sleeping mode, and it takes five clock cycles to recover, including a function call to record the time, which adds to 625ns for the Mica2 and MicaZ. Figure 10 shows a histogram of sampling time uncertainty. There is an error peak at 625ns, which is wakeup time W, and another near 0s due to immediate context switching. The long but small tail shows that some sampling has a large time uncertainty. This result from the real experiment agrees with the theoretical model of the previous subsection, with jitter values limited to about 10μs, which is smaller than the target value of 250μs.

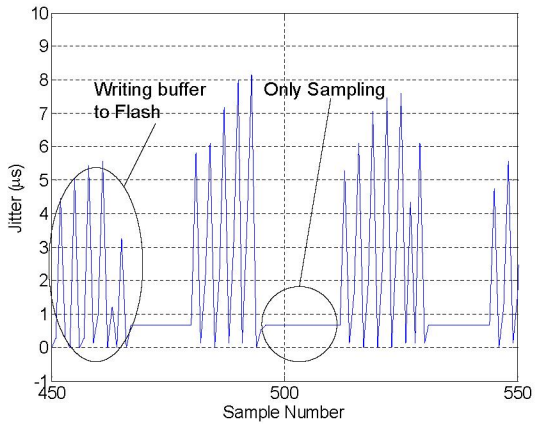


Figure 9: Time Series of Jitter at 5kHz Sampling Rate. Writing to flash interferes with the sampling.

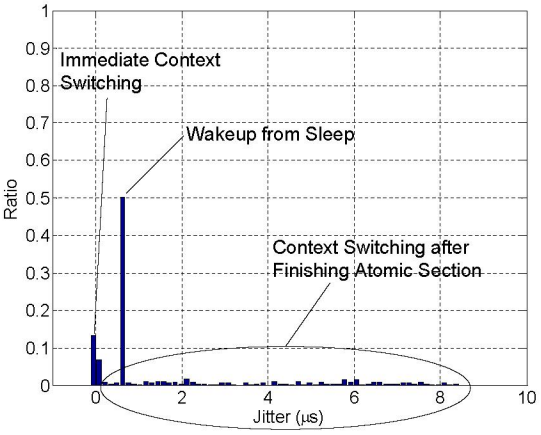


Figure 10: Histogram of Jitter at 5kHz Sampling Rate. Long and thin tail indicates that most samples experience small jitter, however a small number of samples still experience long jitter.

In WSN, microcontrollers are faster than sensors and flash. Many tasks in the operating system are delayed because time-consuming operations (like sampling) block other jobs (like computation or communication). Therefore, the operating system should be able to support multiple processes or threads to overcome this problem. To provide consistency in this case, a mechanism such as a lock, conditional variable, or atomic section is needed. When a microcontroller is running these components, sampling cannot be handled immediately after a timer interrupt. A real-time system cannot be guaranteed for any time-slicing or multi-threaded system, only the severity of sample time uncertainty can be capped. A device with a faster microcontroller or CPU (like a PDA) will have smaller jitter, but they still have the same problem. So using an expensive and power-hungry PDA for SHM or for other real-time applications is not justifiable, as long as the requirement for the worst jitter is satisfied with smaller and less expensive motes.

6. RELIABLE DATA COLLECTION

Maintaining reliable communications over an extended 46-hop path is in itself a challenging problem due to a large round trip time and a high loss rate. In SHM applications, an added imperative is

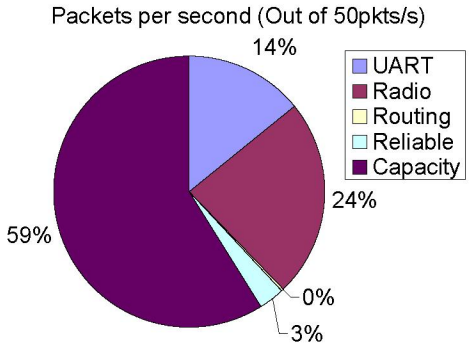


Figure 11: Packet Throughput is 29.4pkts/s with Mica2. The reliable transport layer constitutes only 3% of overhead in packet throughput.

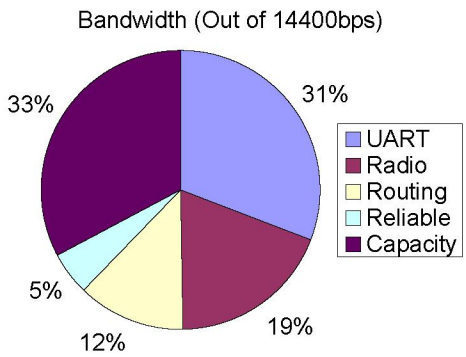


Figure 12: Effective Bandwidth is 588B/s with Mica2. Effect of GenericComm header is represented in UART. Compared to Figure 11, larger overhead is due to the high overhead of a header in a small packet.

that no data can be lost in the system since these events happen rarely and cannot be duplicated. The goal is to have reliable and lossless communications over a large network with minimal overhead for other network components. The two principal aspects of such a protocol are channel capacity, and scalability over a large-scale multi-hop network. It is also important to minimize usage of network resources, because wireless sensor nodes are limited in computational power, memory space, and energy. To this end we designed and implemented the Straw (Scalable Thin and Rapid Amassment Without loss) component, a reliable, highly scalable, data collection service. The following subsection explains its design and implementation.

6.1 PROTOCOL

Straw works over a multi-hop routing layer like MintRoute [26], with transfer initiated by the receiver. Since it is a collection protocol, the receiver is always a PC, and the sender a node. At a high-level, selective NACKs are used. In response to the request of the receiver, the sender sends the entire data once, the receiver identifies missing packets, and then sends a list of those packets (selective NACK) back to the sender. The sender resends those missing packets. At this point, the selective NACK is a single packet, so if there are missing packets some of them may not be reported. The receiver may send a selective NACK again, and this process repeats until all the packets are successfully received. The sender always decides at what interval to send consecutive packets. For WSN, there can be interference between two adjacent transfers, so the inter-packet interval should be large enough to prevent this from occurring. One possibility for the length of the inter-packet interval is the time taken for a transmitted packet to reach the receiver. This approach can work for a small network. However, as the number of hops in the path increases, this time interval becomes too large. In this case, a packet sent toward the receiver in the past can be far enough down the line that the next packet can be sent without interfering with the first. To maximize channel usage in a long multi-hop path, pipelining is used. For nearby nodes, the sender chooses the interval by looking at its depth in the communication tree – the needed delay interval is the time for a packet to arrive at the receiver. For nodes many hops from the receiver, the interval is forced to be at most five times the one-hop packet transfer time. The five hop threshold is empirically chosen from extensive field testing. The results for the MicaZ motes used for the GGB installation with link level retransmission is presented in Figure 2. The first few hops show sharp decreases in bandwidth to avoid direct interference. However, after the fourth hop, pipelining begins, and

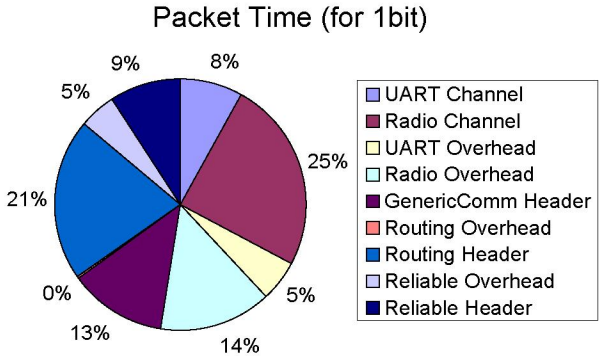


Figure 13: Time usage for 1-bit of data transfer with Mica2. Total time=212.7μs. Only 33% of the time is useful work. 43% of the time is the overhead due to the packet header.

the packet interval stays constant; the transmission bandwidth is constant from the fourth hop up to the 46th hop. The receiver initiates and keeps track of the transfer, so the complexity is confined to the receiver (base station), and the sender (mote) task is kept simple and light weight. A less empirical method for determining the inter-packet interval for a given topology with dynamically changing link quality and interference range remains for future work.

6.2 EVALUATION

Crossbow Mica2 motes without link-level retransmission were used to evaluate the performance of the Straw component. The target node is one hop away from the base station, which is a composition of one radio hop and one UART (serial) hop. Component overhead for a given packet throughput is shown in Figure 11. This figure outlines the limitations each layer contributes to limiting hardware channel capacity. Raw hardware allows a 50 packet per second transmission rate. For example, UART overhead decreases the capacity by 14%. Each layer also adds headers to the packet, which decreases the payload capacity. The multiple layers further reduce channel capacity so that Straw, providing 29.4 packets/s, is 94.8% of the routing layer's possible throughput. In total, after using Straw, 58.8% of packet throughput is available. The channel bandwidth is the product of the packet throughput and the payload capacity, so usable bandwidth after using Straw is only 32.7% of its nominal capacity (Figure 12). Figure 13 shows how much time is needed to send one bit of data. Since the UART and

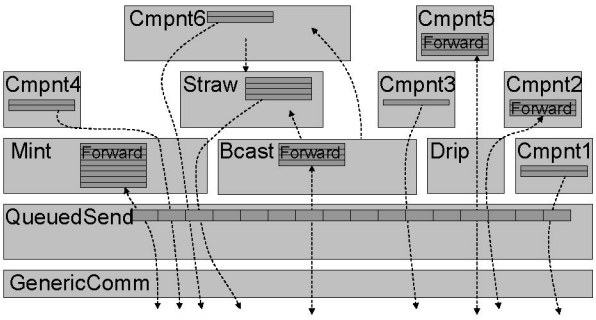


Figure 14: Usage of Packet Buffer. Forward queue space is kept separately in each component, and its size needs to be large to provide high reliability.

Radio channel are hardware components, the 33% value is a physical lower bound. The radio (preamble, MAC) and headers add a significant overhead (43%) to bandwidth, but the overhead from upper layer protocols are relatively small. This indicates that useful bandwidth can be increased by reducing the relative overhead of the header payload. Although the header size cannot itself be decreased in TinyOS, the packet size can be increased so the relative header overhead is decreased.

The utility of using larger packets was tested by doubling the packet size from 36 bytes per packet to 72 bytes. The payload increased from 20 bytes to 56 bytes. The payload increased by a factor of 2.8, but packet throughput decreased from 29.4 packets/s to 20.9 packets/s. The measured 71% throughput is slightly worse than the theoretical calculation of 75% (obtained by doubling UART channel and Radio channel time from Figure 13); the 4% decrease can be explained by additional overhead of the radio and the protocol. This combination increased the bandwidth by nearly 2 times (from 588B/s to 1172B/s). However, when the loss rate is high, a larger packet means a higher effective loss rate. Achieving doubled bandwidth is optimistic in the sense that the test environment has a high success rate (99.8%), but in a real deployment the success rate would be lower. The overall conclusion is that in most cases the benefit of a larger packet size exceeds the disadvantage of an increased loss.

6.3 FUTURE DESIGN ALTERNATIVES

Increasing packet size is an attractive way of increasing the bandwidth, but it has another problem which needs to be addressed. For the test code used in our testing, a 1-byte increase in packet size resulted in a 33-byte increase in RAM. When the packet size is doubled to 72 bytes, even basic services (e.g. time synchronization, broadcast, multi-hop routing, and reliable data collection) and a moderate application can use more than the 4KB of RAM available in the Mica2. The test program itself exceeded the 4KB limit, so the packet buffer size of the routing layer had to be reduced from 16 to 12. The need and usage of packet buffers in TinyOS is shown in Figure 14. In TinyOS, packet space is provided by the application layer, so even when a component rarely sends a packet, it still has to reserve packet space. The second usage is as a forwarding queue. There is a mismatch between the incoming and outgoing speeds. To avoid dropping packets, a forwarding queue is needed which is managed by the components that require this service. The size of the buffer is related to the reliability: for higher reliability, the buffer size must be larger, so to increase the reliability, the size of the forwarding queue in each component must be increased.

There is an alternative approach that would reduce packet buffer RAM consumption – actual buffer space would be provided by the lower layer at the sending queue, and the upper layer would store only the pointers. In this case the size of the sending queue determines the reliability of every forwarding queue. However, an unsafe sharing mechanism, like a direct manipulation of pointers, can often lead to bugs. It remains as future work to realize safe and controlled sharing of the packet buffer pool.

Figure 15: Board enclosure, antenna, and battery installed on the main span. The zip tie had to be put around the antenna to control wind vibration. Poor link quality was experienced with vibrating antenna under strong wind. Corrosion of C-clamp can be observed in the figure.

7. DEPLOYMENT AT THE GOLDEN GATE BRIDGE

The Golden Gate Bridge at the entrance to the San Francisco Bay is a compelling test bed for proving the usefulness of WSN for actual, difficult SHM installations. The cable-supported bridge was designed and constructed in the 1930s and opened to traffic in 1937. With a tower height of 746ft (227m) above sea level, and a 4200ft (1280m) long main span (see Figure 1), it was the longest suspension bridge in the world when it was completed. The extreme loading events for the bridge are expected to be from wind and earthquakes. The goal was to determine the response of the structure to both ambient and extreme conditions and compare actual behavior to design predictions. The network measured ambient structural accelerations from wind load at closely spaced locations, as well as strong shaking from a possible earthquake, all at low cost and without interfering with the operation of the bridge. For this deployment, 64 motes were deployed over the main span and southern tower (see Figure 1), and is the largest wireless sensor network ever installed for structural health monitoring purposes.

7.1 ENVIRONMENTAL CHALLENGES

The bridge is located in a difficult environment; gusty wind, strong fog, and rain present serious engineering challenges for deployment and maintenance of an electronic system. The combination of sea fog and strong wind results in quick condensation of salty water and fast oxidation of metallic components. C-clamps, metal supporting structures on the bridge tower, and even electrical connectors quickly gather rust. The enclosure for the boards is a waterproof plastic box that performed very well during the deployment, as shown in Figure 15. Due to strong wind, major items had to be secured to the bridge with C-clamps, and cables had to be tied or attached to the steel structure. When the wind was strong, the bidirectional antenna vibrated back and forth fiercely, resulting in poor link, so zip ties were used to fasten all antennas in order to reduce

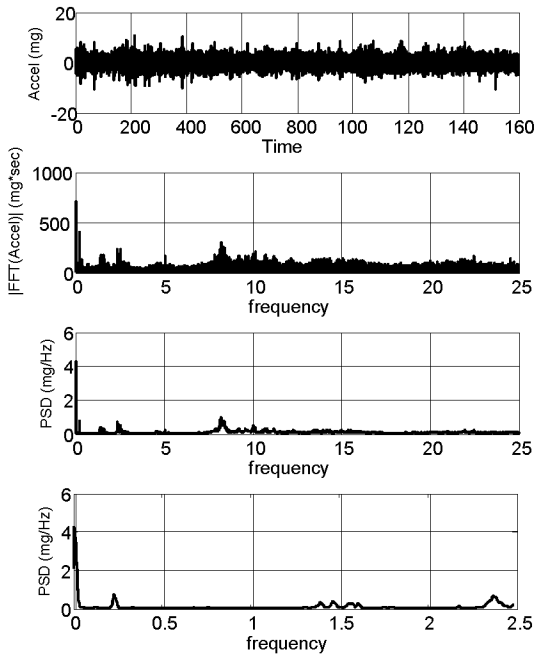


Figure 16: Time and Frequency Plots of Transverse (Horizontal) Sensor Located at Quarter span, 365m North of the South Tower. The data matches the fundamental frequency of the bridge in past studies [7].

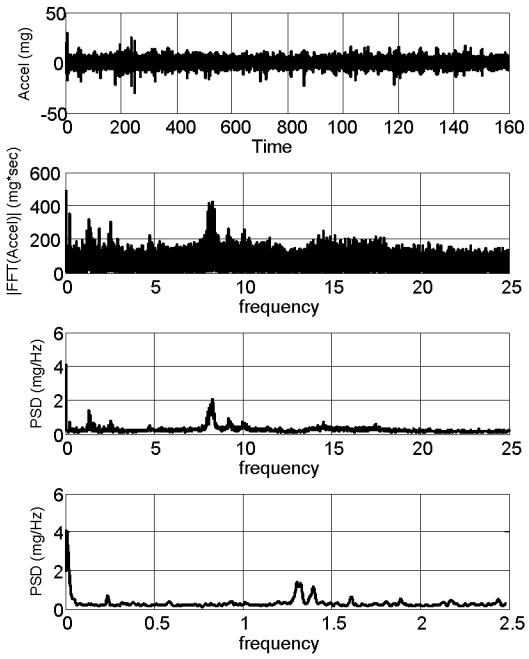


Figure 17: Transverse (Horizontal) Sensor, Mid-Span

the vibration. Since the bridge has a linear geometry, the radio signal had only to be bi-directional; therefore an external bidirectional patch antenna was used for communication, adding signal splitters when necessary to change direction. There is a very narrow passage along the side of the bridge which provides limited line of sight for the bidirectional antennas. This space is, of course, surrounded by steel components and reinforced concrete slabs, and at some places it is obstructed by tools and materials belonging to the maintenance crew. The range of the radio in that harsh environment is severely limited, with the functional range of the Crossbow MicaZ Mote used in this project being 50ft (15.24m) to 100ft (30.48m).

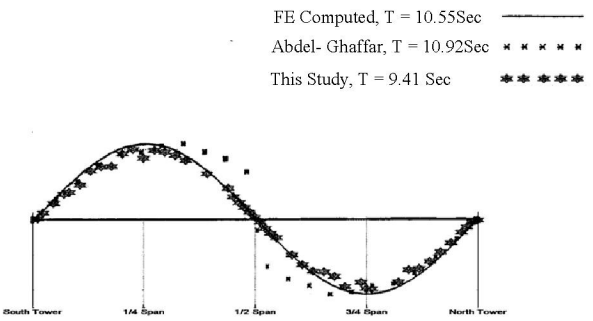
7.2 DEPLOYMENT PLAN

The bridge has suspension cables tying the stiffening longitudinal trusses to the main cables every 50ft (15.24m). The node mounting plates are attached to the gusset plate (or in a few cases to the top flange) on top of the plate-girders connecting the top flanges of the stiffening trusses. The deployment plan was initially designed based on radio tests on the bridge. MicaZ motes attached to the bidirectional antennas were deployed and the signal strength was measured. The tests showed that the signal weakens sharply after 175ft (53m), so 150ft (45.72m) was selected as the modular distance between the boards, hence twenty-nine boards were needed to cover each side of the main span. That nicely matched the distance between the suspension cables and floor beams as well. The actual deployment, however, required some readjustments since the second batch of MicaZ motes, purchased at a later time, proved to have weaker radio strength, by up to 7.5dBm, than the prototype devices. The new motes only yielded a reliable transmission range of 100ft (30.48m), and in some cases the inter-mote interval had to be reduced to 50ft (15.24m). Based on the adjusted deployment plan, a total of 64 sensor nodes were deployed: 53 on the west side of the main span, 3 on the east side of the main span, and 8 on both sides of the south tower. Figure 1 shows the overall layout of the nodes in the deployment. Nodes are deployed on the east side of the main span to provide information necessary for distinguishing between vertical and torsional modes of vibrations.

The high-level operation is executed as follows. At the trigger signal from the base station, every node starts sampling the vibration data. The sampling period is usually set to fill up the 512KB of flash memory on the MicaZ. Then sampled data is reliably collected from every node one by one. These constituted one cycle, and one cycle takes about 12 hours.

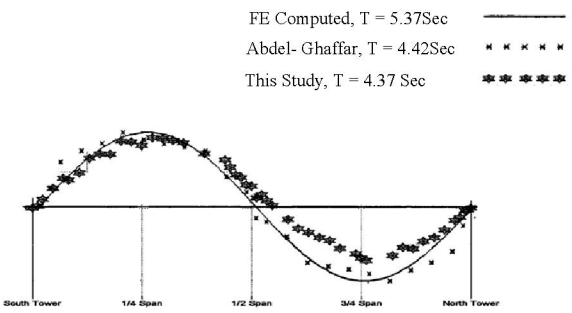
7.3 VIBRATION DATA

A sample of vibration data collected on the bridge at 6 PM on the 21st of September, 2006, is presented in Figures 16 and 17. They show acceleration time histories and frequency domain plots of the accelerations in transverse direction at two nodes: one located near the south quarter span of the bridge (about 365m north of the south tower) and one at the mid-span of the bridge. The accelerations were sampled at 1kHz, every twenty samples averaged and logged to flash. Each figure includes a zoom to a 20s interval of the acceleration time history. The Power Spectral Density (PSD) of the signals were computed using the Welch method [17]. The time histories show that the signal in both orientations have an average amplitude of about 5mg with peaks of approximately 10mg, which most likely corresponds to the passing of large cars or trucks. The frequency analyses show clearly defined peaks in the low frequencies, where the natural modes of vibrations of the bridge are expected to reside. For the vertical orientation, a peak at 0.11Hz matches the fundamental frequency of the bridge found by past studies [7]. Modal properties also match with the simulation model and the previous study; see Figures 18 and 19. Other resonant peaks of 0.17Hz, 0.22Hz, and 0.27Hz are consistently repeated in all the signals from the vertically oriented sensors, and are likely to be other fundamental modes of the bridge structure. More extensive analysis of the data will be presented in future publications.



Golden Gate Bridge, AntiSymmetric Vertical Mode 1

Figure 18: The vertical modal properties match among simulation model, previous study, and this study [7].



Golden Gate Bridge, AntiSymmetric Torsional Mode 1

Figure 19: Torsional modes also match [7].

8. CONCLUSION

This work has three major contributions to wireless sensor networks. First, requirements are identified to obtain data of sufficient quality to have real scientific value to civil engineering researchers for structural health monitoring. An accurate data acquisition system, high-frequency sampling with low jitter and time synchronized sampling were not provided by previous work like Wisden [27] and Tenet [12], but are crucial for data to be useful for structural health monitoring, and are provided in this work. Second, the system is designed to scale to a large number of nodes to allow dense sensor coverage of real world structures. This is verified on a 64-node, 46-hop deployment over the main span and a tower of the Golden Gate Bridge. Third, this network is deployed in a real world structure solving a myriad of problems encountered in a real deployment in difficult conditions. As a result, the network provided reliable and calibrated data for analysis, which was not possible in previous studies.

This work gives a few implications to WSN, which can be interesting research topics. It is found that a small packet size is a bottleneck for network data transmission bandwidth, but increasing packet size is not a good solution for the Mica motes due to the limited amount of available RAM; a limitation resulting from an unshared buffer pool. The routing layer and time synchronization protocol worked fine in laboratory tests, but revealed some problems in the deployment on the Golden Gate Bridge. One more difficulty confronted was that heavy traffic of Straw prevented MintRoute from estimating link quality correctly. Therefore, after some time of transmission, the routing layer broke down. The routing tree had to be frozen before each data collection. Our best guess is that heavy traffic increases the noise level so that the link estimator gets confused.

9. ACKNOWLEDGMENTS

Special thanks to the staff and management of the Golden Gate Bridge District, in particular Dennis Mulligan and Jerry Kao, for their close cooperation with us in every step of the project. Special thanks to Jorge Lee, without his extraordinary help this work would not have been possible. Many thanks to Tom Oberheim who helped in designing and developing the accelerometer board, and to Rob Szewczyk for his numerous suggestions with regards to many aspects of the project including the radio and jitter analysis.

This work is supported by the National Science Foundation under Grant No. EIA-0122599 and by the Center for Information Technology Research in the Interest of Society (CITRIS).

10. REFERENCES

[1] `http://cvs.sourceforge.net/viewcvs.py/tinyos/tinyos-1.x/apps/HighFrequencySampling/MicroTimerM.nc`.

[2] `http://cvs.sourceforge.net/viewcvs.py/tinyos/tinyos-1.x/tos/lib/Broadcast`.

[3] `http://cvs.sourceforge.net/viewcvs.py/tinyos/tinyos-1.x/tos/system/BufferedLog.nc`.

[4] `http://cvs.sourceforge.net/viewcvs.py/tinyos/tinyos-1.x/tos/system/TimerC.nc`.

[5] `http://www.tinyos.net/scoop/special/hardware`.

[6] `http://www.yuasa-intl.com/1axis.html`.

[7] A. M. Abdel-Ghaffar. Ambient vibration studies of golden gate bridge. *Journal of Engineering Mechanics*, 111(4):483–499, April 1985.

[8] J. M. Caicedo, J. Marulanda, P. Thomson, and S. J. Dyke. Monitoring of bridges to detect changes in structural health. *the Proceedings of the 2001 American Control Conference, Arlington, Virginia, June 2527, 2001.*

[9] P. Cheng, W. J. Shi, and W. Zheng. Large structure health dynamic monitoring using gps technology.

[10] J. M. Engel, L. Zhao, Z. Fan, J. Chen, and C. Liu. Smart brick - a low cost, modular wireless sensor for civil structure monitoring. *International Conference on Computing, Communications and Control Technologies (CCCT 2004), Austin, TX USA, August 14-17, 2004.*

[11] S. D. Glaser, M. Chen, and T. E. Oberheim. Terra-scope - a mems-based vertical seismic array. *Smart Structure & Systems*, 2(2):115–126, 2006.

[12] O. Gnawali, B. Greenstein, K.-Y. Jang, A. Joki, J. Paek, M. Vieira, D. Estrin, R. Govindan, and E. Kohler. The tenet architecture for tiered sensor networks. In *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (Sensys '06).* ACM Press, November 2006.

[13] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. In *The 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), Cambridge, MA*, November 2000.

[14] B. S. Jr., M. Ruiz-Sandoval, and N. Kurata. Smart sensing technology: Opportunities and challenges. *Journal of Structural Control and Health Monitoring, in press, 2004.*

[15] S. Kim. Wireless sensor networks for structural health monitoring. Master's thesis, University of California at Berkeley, May 2005.

[16] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (Sensys), San Diego*, November 2005.

[17] L. Ljung. Prentice Hall PTR, Upper Saddle River, N.J., 2nd edition, 1999.

[18] J. P. Lynch. Overview of wireless sensors for real-time health monitoring of civil structures. *Proceedings of the 4th International Workshop on Structural Control (4th IWSC), New York City, NY, June 10-11, 2004.*

[19] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi. The flooding time synchronization protocol. *the Proceedings of ACM Second International Conference on Embedded Networked Sensor Systems (SenSys 04), pp. 39-49, Baltimore, MD, November 3, 2004.*

[20] C. Ogaja, C. Rizos, J. Wang, and J. Brownjohn. Toward the implementation of on-line structural monitoring using rtk-gps and analysis of results using the wavelet transform.

[21] S. N. Pakzad, S. Kim, G. L. Fenves, S. D. Glaser, D. E. Culler, and J. W. Demmel. Multi-purpose wireless accelerometers for civil infrastructure monitoring. In *Proceedings of the 5th International Workshop on Structural Health Monitoring (IWSHM 2005)*, September 2005.

[22] R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler. An analysis of a large scale habitat monitoring application. *the Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys), November 3-5, 2004.*

[23] G. Tolle and D. Culler. Design of an application-cooperative management system for wireless sensor networks. In *the Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN 2005), Istanbul, Turkey*, January 2005.

[24] G. Tolle, J. Polastre, R. Szewczyk, N. Turner, K. Tu, P. Buonadonna, S. Burgess, D. Gay, W. Hong, T. Dawson, and D. Culler. A macroscope in the redwoods. In *the Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (Sensys 05), San Diego*. ACM Press, November 2005.

[25] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh. Monitoring volcanic eruptions with a wireless sensor network. In *the Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN 2005), Istanbul, Turkey*, January 2005.

[26] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multhop routing in sensor networks. *SenSys 2003 Los Angeles, California.*

[27] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. *the Proceedings of the ACM Conference on Embedded Networked Sensor Systems, November 2004.*