

Design and Deployment of Industrial Sensor Networks: Experiences from a Semiconductor Plant and the North Sea

Lakshman Krishnamurthy[†], Robert Adler[†], Phil Buonadonna[‡], Jasmeet Chhabra[†],
Mick Flanigan[†], Nandakishore Kushalnagar[†], Lama Nachman[‡], Mark Yarvis^{†1}

[†] Intel Corporation
2200 Mission College Blvd.
Santa Clara, CA 95052, USA

[‡] Arched Rock
2168 Shattuck Ave.
Berkeley, CA 94704, USA

ABSTRACT

Sensing technology is a cornerstone for many industrial applications. Manufacturing plants and engineering facilities, such as shipboard engine rooms, require sensors to ensure product quality and efficient and safe operation. We focus on one representative application, preventative equipment maintenance, in which vibration signatures are gathered to predict equipment failure. Based on application requirements and site surveys, we develop a general architecture for this class of industrial applications. This architecture meets the application's data fidelity needs through careful state preservation and over-sampling. We describe the impact of implementing the architecture on two sensing platforms with differing processor and communication capabilities. We present a systematic performance comparison between these platforms in the context of the application. We also describe our experience and lessons learned in two settings: in a semiconductor fabrication plant and onboard an oil tanker in the North Sea. Finally, we establish design guidelines for an ideal platform and architecture for industrial applications. This paper includes several unique contributions: a study of the impact of platform on architecture, a comparison of two deployments in the same application class, and a demonstration of application return on investment.

Categories and Subject Descriptors

C.3 [Special-Purpose and Application-Based Systems]: *real-time and embedded systems, microprocessor/microcomputer applications, signal processing systems.*

General Terms

Performance, Design, Economics, Experimentation.

Keywords

Industrial applications of sensor networks, embedded hardware design.

1. INTRODUCTION

Sensing technology is a cornerstone for many industrial applica-

tions. Manufacturing plants and general engineering facilities, such as shipboard engine rooms, use sensors to ensure product quality, as well as efficient and safe operation. Predictive maintenance is one key application that improves efficiency and productivity. Predictive Maintenance (PdM) is the general term applied to a family of technologies used to monitor and assess the health status of a piece of equipment (e.g., a motor, chiller, or cooler) that is in service. PdM technologies allow the user to detect most impending failures well in advance, as long as analysis is performed with sufficient frequency. PdM is an important and relevant example of the class of industrial sensor networking applications that provide measurable value in real deployments.

We have chosen one form of PdM, vibration analysis, to drive our investigation of industrial wireless sensor networks. We formulate the requirements and develop hardware and software architectures for this application. We then evaluate our design in two industrial environments. The first is a central utility support building (CUB) at a semiconductor fabrication plant, which we will refer to as *FabApp*. The CUB houses machinery to produce pure water, handle gases, and process waste water for the fabrication line and spans indoor & outdoor locations. For this scenario, we also consider two sensing platforms, one based on the Mica 2 [3] and another based on the Intel Mote [7], each of which possesses very different hardware features. Using these two platforms in the same deployment, we identify the impact of individual platform features (i.e., processor, radio, memory) on the overall hardware and software architecture, as well as the overall performance of the application. The results can be extrapolated to the capabilities of other hardware platforms, and provide direction for both hardware and software architectures for industrial sensing applications.

The other environment we consider is shipboard PdM aboard an operating oil tanker. The chosen oil tanker sails in the North Sea and represents one of the roughest environments for industrial sensor networks. The basic requirements of the application are similar. However, the oil tanker's aft engineering spaces are constructed of steel floors and bulkheads and are subdivided into three major watertight compartments with hatchways in between. The hatches may be periodically open and shut. The sensor network was expected to work despite the periodically disconnected nature of these watertight compartments. While this scenario was

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SenSys'05, November 2–4, 2005, San Diego, California, USA.
Copyright 2005 ACM 1-59593-054-X/05/0011...\$5.00.

¹ Authors may be contacted at {lakshman.krishnamurthy, robert.p.adler, jasmeet.chhabra, mick.j.flanigan, nandakishore.kushalnagar, lama.nachman, mark.d.yarvis}@intel.com and pbonadonna@archedrock.com

* Other names and brands may be claimed as the property of others.

similar to FabApp, it provided new challenges to hardware, network architecture, data gathering, and analysis. We report the results from a four month run of the sensor network.

Our focus on industrial PdM sensor networks is motivated by specific goals that are the main contributions of this paper:

1. Validation of requirements for industrial environments
2. Evaluation of the effect of the deployment environment on sensor network architecture, including characteristics such as fault tolerance.
3. Assessment of the impact of platform characteristics (e.g., processor speed, network bandwidth) on architecture and performance of real deployments.
4. A set of techniques for production functions such as quality assurance and qualification for deployable wireless sensor networks.
5. Lessons learned from running the network for extended periods of time in a production environment.

We make these contributions in the context of real deployed applications. Using this foundation, we hope to develop simple architectures that are broadly applicable.

The remainder of this paper is organized as follows. Section 2 develops the background of PdM and the motivation for applying wireless sensor networks in this space. Section 3 describes the particular application space of our deployment. Section 4 discusses related work. Section 5 describes site pre-planning and requirements of industrial sensor networks. In Sections 6, 7, and 8 we develop and evaluate the hardware and software architectures in the context of FabApp. In Section 9, we discuss the shipboard deployment. Finally, in Section 10 we summarize lessons learned and present our conclusions.

2. BACKGROUND AND MOTIVATION

PdM technologies include vibration signature analysis, oil analysis, infrared thermography, and ultrasonic frequency detection. These tools can either act alone or together to provide a detailed picture of defect sources and current life expectancy of the test subject. Each of these technologies utilizes specific sensing and data analysis techniques:

Vibration Analysis: Time domain and frequency domain waveform analysis identifies changes in amplitude and frequency patterns, suggesting repair or replacement. This technology presumes that source vibration frequencies can be identified and assigned to specific components of the test subject.

Oil Analysis: In depth analysis of wear particles, viscosity, acidity, and raw elements. By capturing a small sample of oil from a source and comparing to baseline samples, potential problems can be seen well in advance of a failure.

Infrared Thermography: Specialized cameras and detection probes sense heat at frequencies just below visible light. Operators can detect abnormal heat sources and compare to baseline data for temperature changes. Because infrared cameras detect relative heat, they are also useful in detecting cold areas, liquid levels in vessels, and escaping gases.

Ultrasonic Detection: Ultrasonic frequencies are captured to detect wall thickness, corrosion and blistering, erosion, flow dynamics, and wear patterns. By comparing data to standards, change rates can be measured and lifetimes projected.

The popularity of predictive maintenance with corporations and governments can be attributed to four primary objectives:

- Reduction in catastrophic equipment failures and the associated repair and replacement costs
- The desire to change the business model from calendar-based maintenance to indicator-driven maintenance
- The ability to quantify the quality of a new system within the warranty period
- Meeting factory uptime and reliability requirements

Because all of these factors eventually lead to a reduced bottom line, there is a drive to implement cost effective solutions to capture, trend, track, and alarm the data from these tools.

There are two primary models for data acquisition in industrial environments today: manual “sneaker net” data collection and fully integrated online surveillance. The manual model utilizes hand-held instruments and remote-installed or hand-carried sensors to which instruments are connected. Data is captured locally in the hand-held device for transport back to a central repository for analysis. These instruments also provide powerful field analysis capabilities, and have added functionalities such as balancing, frequency response testing, and multi-channel analysis. Online surveillance utilizes sensors that are hard wired to a data acquisition unit which processes the data and delivers it across a wired network to a central repository.

Both manual and online systems are in place across a variety of applications and industries; however, they are not always a good fit. Manual data collection is insufficient in many applications due to the potential for user error, the high cost to train and keep experts, and the manpower required for frequent data collection. Online systems allow more reliable and frequent data collection. However, the cost of purchasing and deploying the modules and the network and power infrastructure can be prohibitive. Online solutions are appropriate for equipment and systems with a potential cost impact greater than \$250K. For the majority of equipment in a typical industrial deployment, an online system provides an insufficient return on investment. An industry cross section shows that online system penetration into the market is less than 10%, primarily due to cost. In the remaining 90% of the market, 20% use manual data collection, and most are not happy with the level of prediction and correlation they provide. *Finding a solution to address this market and tap into the remaining 70% may represent a killer application for wireless sensor networks.*

For a typical factory deployment, a cost analysis of the three technologies is shown in Table 1. While most of the inputs are extrapolated from the actual cost breakdown of previous deployments, some costs (e.g., the cost of contracted labor) were estimated. This data suggests that wireless sensor networks can be less expensive than an online system, and yet provide the repeatable, frequent data collection not seen in a manual system.

3. AN APPLICATION OF VIBRATION ANALYSIS

In our particular application of PdM, vibration analysis is used to monitor the health of equipment in a central utility support building (CUB) at a semiconductor fabrication plant. The CUB houses machinery to produce pure water, handle gases, and process waste water for the fabrication line. The same sensor network was also

deployed to monitor machinery onboard an oil tanker. The system was designed to work with standard off the shelf accelerometers, and interfaced with an off-the-shelf software application which provided post processing of the raw waveform data transported by the wireless sensor network.

Wilcoxon model 786A sensors with Integrated Circuit Piezo (ICP) accelerometers were used. Each was calibrated by the manufacturer to 100mV/G with 5% calibration sensitivity at 25°C. The dynamic range of the accelerometers is 80g's peak with a maximum frequency range of 30 kHz. The accelerometer uses a ceramic-shear type piezo, which is hermetically sealed and interfaced to a 2-wire lead via a Mil-Std 2-pin connection. All sensors in this trial were stud mounted to the machinery (Figure 1).

For our initial deployment, we developed a sensor board with a sampling rate of 19.2 kHz, allowing for a frequency range of 9.6 kHz. While this sensor board allowed us to demonstrate a proof of concept by obtaining 3000 data points per measurement, it was insufficient to entirely replace the existing PdM capability. In particular, the number of data points obtained provided insufficient resolution for analysis. After factoring in averaging and overlap, a frequency resolution of +/- 12 Hz was obtained, 1/24th the density of current offerings in the handheld market. Resolution will be increased in subsequent deployments by obtaining a larger number of data points per measurement.

The sensor network was integrated with an off-the-shelf software application which provided long term data storage, trend analysis, and fault alarms. The raw waveform signals, each representing a fraction of a second of collected data, are transformed into the frequency domain for analysis. Frequency peaks are associated with specific defect characteristics such as bearing failures, gear-box defects, electrical and other mechanical issues. To obtain a frequency resolution of 0.5 Hz with a maximum input signal frequency of 5 KHz, it is necessary to sample at a rate of 40 KHz. Averaging is used to reduce contamination of the signal from transient noise. Window functions (e.g., Hanning, COS², Kaiser Bessel, and rectangular) are applied to minimize signal discontinuities across averages.

4. RELATED WORK

There have been numerous published efforts related to deployment of sensor networks. This section provides brief overview of this work and clarifies our contributions relative to these efforts.

Wireless PdM technology has been available for several years from traditional manufactures [8]. Typically these solutions are targeted as a simple wire replacement between sensors and collections points. These solutions do not fully reduce the cost of the deployment and limit the use of wireless with fixed 1:1 relationships. A multi-hop sensor network removes this 1:1 requirement and provides better fault tolerance and resilience against propagation effects, but also increases software complexity. Recently the industry has started moving towards multi-hop wireless solutions [15] [16]. Some research efforts have embarked on designing wireless sensor networks for PdM [19]. Our contributions are complementary to these efforts focusing more on the experiences relating to platform architecture impact and deployments.

There have been a few reported efforts involved in gathering vibration data for various applications. Work at USC and CENS [9] [21] has focused on gathering vibration to monitor structures for

Table 1. Cost breakdown of three approaches to PdM.

	Manual Collection	Online System	Wireless Data / Wired Power
# Wired APs	0	450	35
# Wireless APs	0	0	875
# Analyzers	8	1	1
Hardware Costs			
Sensors (installed)	\$1,260,000	\$1,260,000	\$1,260,000
Wired APs	\$0	\$2,250,000	\$17,500
Wireless APs	\$0	\$0	\$262,500
Analyzers	\$144,000	\$18,000	\$18,000
Installation Costs			
Wired APs	\$0	\$3,375,000	\$262,500
Wireless APs	\$0	\$0	\$1,726,974
Labor (Collection Costs)	\$168,000	\$3,360	\$3,360
Total Costs	\$1,572,000	\$6,906,360	\$3,550,834
Total Costs w/o Sensors	\$312,000	\$5,646,360	\$2,290,834



Figure 1. Sensor nodes deployed in the FabApp.

devices within and for outdoor devices (through the CUB exterior wall). The 916MHz Mica2 Motes exhibited some RF shadows at certain points both in the indoor and outdoor locations (433 MHz Mica2 Motes were not used for the site survey or the deployment). Moving the sensor nodes to alternate points alleviated the shadowing effect and stable network formation with 90%+ packet reliability was achieved. The Intel Motes were able to form a stable network topology and no shadowing was observed. A spectrum analysis test was not performed at the CUB site because most potential interference sources were known to site engineers.

While 802.11b connectivity between Stargate Gateways in the CUB was initially excellent, we found that connectivity changed over the course of days. During the deployment, it was necessary to repeat the survey and redeploy the gateway nodes several times. In the future, a mesh network may be used to alleviate this issue.

The shipboard tests demonstrated excellent sensor node and gateway connectivity in the engineering spaces. The network discovery mechanisms were able to form a network within a few seconds and maintain connectivity throughout each test. For the Mica2 Motes, the topology was consistently single hop and test packet end-to-end yields were greater than 99%, regardless of location within a watertight compartment. The only exception was the 433 MHz Mica2 Motes, which were not able to communicate through a non-watertight bulkhead/doorway. For the Intel Motes, each trial formed a stable scatternet with no observed shadows. For the gateways, 802.11b connectivity was excellent between the test points and the access point.

The shipboard spectrum analysis test showed no adverse interference conflicts for the motes, the gateways, or the ship. The engine room ambient noise did not show any significant spikes at the mote or gateway transmission frequencies, even when the ship's radar was functioning. To ensure that the sensor network would not interfere with the ship, we obtained a list of critical radio navigation frequencies in use. The only devices that would generate potential interference were the 433 MHz devices.

The site surveys lead to a few important conclusions. First, fewer gateway nodes were required to achieve adequate coverage than initially anticipated at each site. Second, at the CUB site, coverage of outdoor nodes from an indoor gateway was shown to be feasible. Similarly, aboard the ship, an entire compartment could be covered by a single gateway node. Thus, a few well placed gateway nodes could be utilized at each site, reducing requirements for wired power and network connectivity, as well as overall cost.

The results of the surveys also tended to favor the use of higher RF frequency devices in the industrial environment. Aboard the ship, the 916 MHz and Bluetooth devices exhibited excellent connectivity properties, whereas the 433 MHz devices could not penetrate some barriers. We attribute the RF performance to the steel materials found at this site. Unlike an office or outdoor environment bulk-heads and machinery tend to reflect rather than attenuate RF energy, thus promoting connectivity pathways that might not be observed in other environments.

5.2 Requirements

The site surveys and interaction with field personal and corporate IT led to specific requirements that drove network design:

Fault tolerance and reliability - There are two aspects to reliability of our system that were stressed by our customers. First, data from a sensor node must be accurate and both acquired and delivered in a timely manner. While some failures may be tolerated, failure recovery is required. Second, the network must be robust to the extreme temperature, humidity, and vibration in the environment. Proper engineering of electronics and enclosures can accommodate such environments, but this is outside the scope of this paper. Rather, we address how the system can recover from faults caused by such external influences.

Long-lived battery powered operation - Power management is just as critical in the industrial environment as in other deployment scenarios. Initially, this may seem counterintuitive since most machinery plants have ample power supplies and distribution systems. However, operating and safety regulations typically call for each piece of equipment to have a dedicated power circuit, requiring separate power connections for sensor nodes. To reduce installation costs, the sensor network must either be battery powered (and provide aggressive power management) or make use of "trickle" sources such as solar or energy harvesting.

Maintainable - Management and diagnostics are key identifying hardware failures or the cause of errant data. The sensor network management system must provide a simple user interface for maintenance personnel that enables continuous and rapid diagnostics to detect problems and enable repair. While network health consoles were deployed, they are outside the scope of this paper.

Seamless integration into existing application - A sensor network PdM solution must integrate into existing PdM applications, to provide the same end-user interface and analysis tools. In our deployments, an automated data import capability was required.

Security - Ensuring data integrity, confidentiality, and authenticity is necessary in nearly all industrial sensor networks. Modified or falsified data from the sensing infrastructure can have crippling or even dangerous effects in industrial environments. Operationally sensitive data must also be protected. However, in our deployments, very little security was provided within the sensor network itself. Due to the physically secure environment, the low value of individual sensor results, and the open-loop nature of the system, the risk of an open system was deemed to be relatively low. In future deployments, additional security features may be added.

6. HARDWARE ARCHITECTURE AND COMPARISON

One of the goals of the FabApp deployment was to compare different sensing platforms in the context of this application. We start with a comparison of hardware platforms and a description of their impact on the design of the overall system. Table 2 highlights the main differences of the two platforms.

We deployed two platforms in the fabrication facility: one based on the Mica2 Mote and the second on the Intel Mote. The Mica2 Mote has an 8-bit microcontroller (ATmega 128L) running at 8MHz, 4 kB of RAM, 128 kB of flash and a 916 MHz radio with 38.4 kB/s maximum theoretical bit rate [3]. The Intel Mote on the other hand has a 32-bit ARM7TDMI processor running at 12 MHz, 64 kB of RAM, 512 kB of flash and a Bluetooth radio with 750 kb/s maximum theoretical bit rate [7]. While the use of a Bluetooth radio introduces a very different communication model,

Table 2. Comparison of Intel Mote and Mica2 design features.

	<i>Mica2</i>	<i>Intel Mote</i>
<i>Processor</i>	8-bit microcontroller	Processing power not explicitly exploited.
<i>Volatile Storage</i>	Required additional volatile storage on the sensor board, managed by an additional microcontroller.	Simplified sensor board design enabled by on-chip storage of ADC samples.
<i>Radio</i>	Simpler network stack, better theoretical receive sensitivity.	10x throughput. Point-to-point link model required custom network stack. Frequency hopping less impacted by interference.
<i>I/O Interfaces</i>	Direct SPI connection from sensor board controller to mote processor.	No H/W SPI port; sensor board implements UART-SPI bridge logic.

requiring different routing protocols, the availability of a processor, radio, and memory in an integrated package [24] greatly reduced the total cost of the system.

Block diagrams of the sensor boards designed for these mote platforms are shown in Figure 3 and Figure 4. The Intel Mote sensor board was much simpler than the Mica2 sensor board, mainly due to the Intel Mote’s larger RAM, which allowed internal storage of vibration samples. In contrast, the Mica2’s limited internal SRAM required external RAM and a processor on the sensor board, thus increasing the cost, complexity, and power consumption of the sensor board. The use of a microcontroller was one design choice. If a CPLD had been used instead, it would have been necessary to implement the control logic for the SPI port, the SRAM, and the sensor board in firmware.

The Intel Mote’s direct streaming feature resulted in a higher strain on its I/O interfaces. In addition, its lack of a SPI interface required that the sensor board bridge the SPI output of the A/D to the UART interface supported by the mote. Fortunately, the Intel Mote’s fast UART (up to 960 kb/s) was more than adequate to support required sampling rate of 16 bit data at 19.2 kHz.

Despite the Intel Mote’s more capable processor, in-network data processing was not implemented for four reasons. First, trend analysis requires that all data be delivered and stored at a central location. Second, we wanted to enable direct comparison of measured data against the manual system. Third, many of the algorithms implemented in the back-end software to predict equipment failures are proprietary. Finally, a fair comparison of network performance in Section 8 required the same load on networks utilizing each platform. The only data processing performed on the mote was DC offset removal. In the future, the extra processing capacity of Intel Mote may be utilized to implement data compression or complete in-node data analysis, hence reducing the amount of data to be transferred over the radio.

7. NETWORK ARCHITECTURE AND COMPARISON

The network architecture for the FabApp includes a hierarchical communication structure, a cluster-based power management protocol, and a reliable bulk transport. These elements work together to coordinate periodic data collection across a large number of sensing points while maximizing sleep time. We will first describe the components of the software architecture designed for the Mica2 hardware, followed by a description of the changes

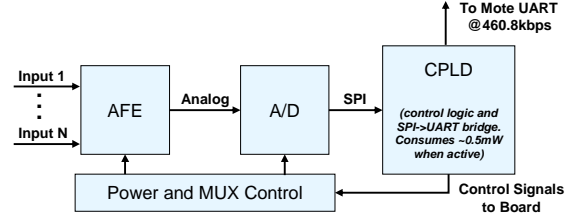


Figure 3. Intel Mote sensor board.

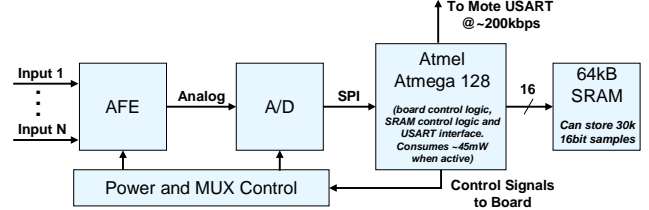


Figure 4. Mica2 sensor board.

made to leverage features of the Intel Mote. In Section 9, we will also consider the changes required for the shipboard deployment.

Figure 5 shows the FabApp’s high level architecture, and Table 3 lists the network components. Each sensor node is connected to a multi-channel sensor board (not shown). On each sensor board, a subset of channels is connected to vibration sensors. Each sample consists of 6 kB of time-series data from the vibration sensor, for a total of 36 kB per node. The goal is to capture data from all sensors at regular intervals and deliver it to a server for analysis.

To allow the network to scale to thousands of sensing points (4000 in a typical fabrication facility) we use a hierarchical network [23], with an 802.11 network providing a high-speed, highly-reliable backbone. Stargate nodes, each with a mote radio and an 802.11 radio, act as gateways between the two networks. Data flows from a specific sensor, across the sensor mesh to a Stargate gateway, across the 802.11 backbone to the network edge, where it is delivered to the data analysis server. We examine this hierarchical architecture in detail in the next subsection.




To meet the battery lifetime requirements (Section 5.2), we use a cluster-based sleep/wakeup protocol. Sensor nodes form clusters around gateway nodes. Each cluster wakes at regular intervals to capture and send data to the backend server. The sleep schedule of each cluster is coordinated by a cluster-head node, which is connected to the gateway via a serial port. Sleep schedules are independent, and no inter-cluster coordination is required. The cluster sleep/wakeup protocol is described in more detail in Section 7.3.

Each collection period, the cluster-head schedules data capture/transfer for every sensor connected to each node in the cluster. When a node is scheduled to send data, it initiates a connection to the gateway application (which resides on the Stargate) using a reliable transport protocol described in Section 7.4.

Once the data has been transferred to the Stargate gateway, it is time-stamped, and a file is created for each collection of a sensor channel. The Stargate gateway periodically copies data files to the root Stargate using a secure transport over the 802.11b radio.

The root Stargate node transfers this data across a serial cable using Kermit protocols to a bridge Stargate node. The serial cable is used to isolate the wireless network from the corporate network

Table 3. FabApp network components.

Platform	Description
	Mica2 Sensor Node: Atmel AtMega128L, Chipcon 900 Mhz radio, Battery powered.
	Intel Mote Sensor Node: ARM Core, Zeevo Bluetooth radio, Battery Powered
	Stargate Gateway Node: Intel XScale® processor (PXA255), 802.11b radio, serially-connected Mica2/Intel Mote, wall powered.

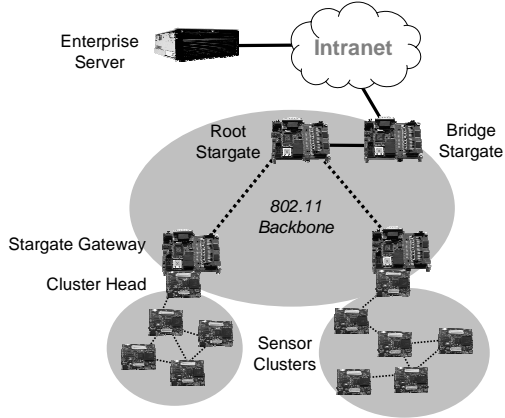


Figure 5. FabApp network architecture.

for security reasons. The bridge Stargate node is connected to the corporate intranet and transfers the data to the server using a secure transport. On the server, the data is converted to a format that can be understood by the analysis tool and imported into a database. At this point the data can be accessed and analyzed by the end user with backend analysis tools.

7.1 Hierarchical Network Structure

The FabApp uses a hierarchical architecture to increase network scalability and to exploit resources available in the environment. Use of a hierarchical network structure is not new [2] [18]. In an industrial setting, a hierarchical architecture exploits heterogeneity within the sensor network. The FabApp architecture may be broken down into three logical hierarchies. The lowest layer, Tier 1, includes networks of sensor nodes. This tier is expected to have the lowest compute capability and significant limitations on radio bandwidth and battery capacity. Sensor nodes are subdivided into groups called *clusters*. Each cluster has one or more gateways that provide the interface to the next tier. A node may be pre-assigned to a particular cluster, or it may select a cluster dynamically based on the quality of available routes to cluster heads.

The middle level, Tier 2, forms the sensor network backbone. It is composed of individual cluster gateways linked by a robust communication medium. The nodes in this tier have significantly more compute, communication and power capacity than those in Tier 1. These nodes offload the burden of communication and computation from the lower tier. This tier also acts as a convergence point for data from clusters of different types of nodes at Tier 1.

Tier 3 is the sensor network's interface to the enterprise. Devices at this tier may be gateways or servers that receive data from Tier 2 and export it as required by the application. It abstracts the spe-

cific needs of the application away from the sensor network, thus minimizing the amount of custom software and preprocessing necessary within the sensor network itself. This tier also provides the management and diagnostic interface to the sensor network. The operator may interact with this interface directly, or it may generate external alarms to indicate noteworthy events.

7.2 Bulk Transfer Protocol

To transfer each large chunk of sensor data reliably across the sensor network, we implemented an end-to-end reliable bulk transfer protocol. When a sensor node is scheduled to transmit captured data, it sends a connection request to the gateway. The connection request contains a set of connection parameters tuned to the capabilities of the node, including fragment size, data size, and transfer rate. These parameters allow the network to support multiple sensor platforms (including Mica2 and Intel Mote). If the request is accepted by the destination, data is transferred in multiple fragments using a standard NACK based sliding window protocol [20], described in detail in [10].

7.3 Mica2 Power Management Protocol

Power management is achieved using a centralized protocol. Members of each cluster wake and sleep in a synchronized manner, under the direction of the cluster head. The cluster head uses application-level sampling requirements to schedule sleep periods, similar to the approach taken in [10].

The nodes in the cluster know when to wake up based on a parameter communicated by the cluster head at the end of the previous period. At boot time (e.g. on initial install), nodes are awake and the protocol proceeds as described below.

Once the cluster is awake, the cluster head initiates metric-based single-destination-DSDV routing [22] to allow all nodes to find a path to the cluster head. Next, each node sends periodic "trace-route" packets to the cluster head, allowing the cluster head to discover the nodes in its clusters. The cluster head waits a pre-defined period, to allow all nodes to report.

Once discovery is complete, the cluster head sends a data capture and transfer request to each node. The resulting data is transferred using the bulk transfer protocol. Once data collection is complete, the cluster head sends beacons indicating a start time and duration of the sleep phase. The sensor nodes then go to sleep for the requested duration, waking once again in unison.

Since the nodes may sleep for long periods of time (e.g., days or weeks) the clock drift may be quite large, causing nodes to wake at different times. To ensure that all nodes are awake after a sleep period, the cluster head waits for a "guard period" greater than the maximum possible clock drift before initiating communication.

7.4 Fault Tolerance

Industrial sensor networks must operate unattended in potentially harsh environments for long periods of time. Fault tolerant design is required to prevent individual failures from shortening network lifetime. Four major design features increased fault tolerance.

First, multiple watchdog timers were used to recover from any non operational state. Each node tracked the time since the last packet reception (in the wake state). If no packets were received for a predetermined period, the node would automatically reset

itself. Other watchdog timers were used to catch hardware errors during data transfer from the sensor board or detect radio lockups. Hardware resets were also triggered by unexpected protocol states, such as the receipt of a new data capture/send request before the previous one was finished. In each case, recovery was accomplished either by a hardware reset or a state re-initialization.

The second major design feature was storage of the core network states in cluster heads. Because sensor node protocol state was soft, nodes could return to normal operation after being reset. In many cases, only a small performance degradation would result.

The third feature was intentional re-initialization of sensor nodes after each collection cycle. Since data collection was controlled by the cluster head, sensor nodes did not need to maintain state beyond the current cycle. Consequently, sensor nodes could start each cycle with fresh state, preventing problems in one collection cycle from affecting the next cycle. This feature was added only after system testing, as it has a tendency to conceal bugs.

The final feature was non-volatile storage of critical state at the cluster head after every collection. Thus, the cluster head could also be reset immediately prior to each wake period, removing any stale state in the operating system. While brute force, this technique was very effective in meeting the overall application fidelity and continuous operation of the network

7.5 Intel Mote Network Architecture

The broadcast nature of the Mica2 platform radio allows a traditional approach to topology discovery and optimization, based on DSDV. Due to the connection-oriented nature of the Bluetooth radio, the Intel Mote platform required a very different approach.

We implemented a scatternet formation algorithm (described in [7]) to grow a network beyond the limits of a Bluetooth piconet. This algorithm creates a tree topology with a predefined root node. All intermediate nodes in the tree are slaves in their parents' piconets and masters in their children's piconets. The root node has a master role only, while leaf nodes have slave only roles.

We also implemented a network low power mode (also described in [7]). This mode maintains all Bluetooth links in a low power state using the Bluetooth hold mode, which allows for a very low network latency. We implemented a protocol to enable this low power mode in periods of low activity, and wake the network before a data collection phase is started.

The same reliability protocol was used for both the Mica2 Mote and the Intel Mote, but the parameters were adjusted to leverage the high bit rate Bluetooth radio and available RAM: sender throttling was reduced, the sliding window size was increased, and the fragment size was increased to take advantage of the larger MTU.

Although the routing and sleep protocols were different for the Intel Mote, we enabled clusters of different platforms to coexist by terminating these protocols at the cluster head. Cluster heads translated data packets and route updates at the cluster boundary.

8. MICA2 AND INTEL MOTE PERFORMANCE COMPARISON

We deployed three Intel Mote clusters and three Mica2 Mote clusters in the fabrication facility. We analyzed the performance of both systems and summarize the performance results below.

8.1 Data Transfer

The Intel Mote had a roughly 10x greater data transfer rate, due to radio's higher throughput and larger MTU. We also observed more consistent performance in Intel Mote clusters, both across nodes and across collection cycles. Details are presented in [7].

We found the Mica2 throughput to be much lower than expected. This performance gap can be attributed to heavy throttling of the reliable transport protocol, which was configured very conservatively. These results suggest that a dynamic throttling implementation would have been more appropriate for this protocol.

8.2 Performance Across Clusters

Figure 6 shows the average transfer time of all clusters as well as the average duration of the collection cycles. The number of nodes in each cluster is shown in brackets. We expected the collection cycle duration to increase linearly with the number of nodes per cluster, given our simple power save protocol and use of sequential data collection. The transfer time on the other hand is calculated per mote, and should not be largely affected by the small variations in the cluster sizes chosen in this experiment.

As shown in Figure 6, the Intel Mote performance was consistent across the different clusters. The change in average transfer time of all 3 clusters is within 6%, and is less than 4% of the collection cycle duration after adjusting for the cluster size. In Mica2 clusters, the variation in the average data transfer time across clusters is in the range of 50% to 160%. Closer examination of the data showed that one Mica2 cluster (cluster 21) suffered from very high RF interference from nearby power line equipment. The interference resulted in a much higher average transfer time and many data transfer timeouts. In contrast, independent testing of Intel Motes in the same location revealed no degradation in the data transfer times or dropped packets. The Intel Mote's consistent performance was likely due to the operating frequency range, frequency hopping, and link layer reliability.

8.3 Power Consumption

At a high level, the power save protocols used in the Intel Mote and Mica2 clusters were very similar. In both cases, the entire cluster was awake during data collection and otherwise asleep. We divide the power consumption analysis into two parts: power consumption during an active collection cycle and power consumption during the sleep phase.

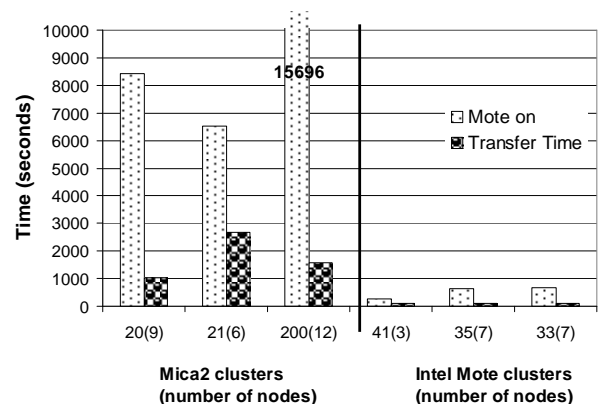


Figure 6. Average transfer time on Mica2 and Intel Mote.

Table 4. A comparison of current consumption across Intel Mote and Mica2 platforms.

Power Mode	Intel Mote		Mica2 mote	
	Current (mA)	Time (seconds)	Current (mA)	Time (seconds)
Sleep	0.7	Variable	0.05	Variable
Full Power	20.7	591	17	6432
Full power + sensor on	54.6	76	63	693
Full power + sensor on + A/D on	58.8	1.5	66.4	60

The power consumed per collection cycle for Intel Motes was less than one tenth that of Mica2. It can be seen from Table 4 that while the power consumption across platforms in each active mode is similar, Mica2 Motes spent more time in each mode due to the slower radio.

In sleep mode, the Intel Mote platform consumed 3 mA, compared to 0.05 mA on the Mica2 platform. This high power consumption is a result of the connected sleep mode implemented on the Intel Mote to reduce the network response time. By maintaining connections while in sleep mode, a network response time of about one minute is maintained. With the Mica2 platform, the network is completely inaccessible during the sleep phase. The second factor is the higher deep sleep power of the Intel Mote compared to the Mica2 mote, which is dictated by the processor.

In this application, since the sleep durations are large, disconnecting the network and putting the nodes to sleep will result in lower power consumption. We measured the current consumption of the Intel Mote in a disconnected network to be 0.7 mA. We added the overhead of network formation to the collection cycle durations and calculated the total power consumption and resultant lifetime in disconnected mode as a function of the sleep duration. The results are shown in Figure 7 for both platforms (marked as existing H/W). As expected, at lower sleep durations, the Intel Mote cluster consumes less power, since the total power consumption is dominated by the active power. At longer sleep durations (1 week or longer), sleep power is the dominating factor, resulting in longer lifetime in the Mica2 case.

Further power reduction would require the addition of an external real time clock (RTC) that allows the system to be completely turned off. This approach has been previously used on the XYZ platform [6]. We calculated the total power consumption assuming this external RTC solution (50 uA sleep current) for both platforms. The lifetime with this solution is shown in Figure 7 (marked as 50 uA sleep). Note that given the long cycle time of the Mica2 cluster, even with a sleep duration of one month, the active current still has a considerable effect on the total power consumption. In the case of the Intel Mote, the effect of the active current is negligible with 1 month sleep duration.

9. SHIPBOARD DEPLOYMENT

9.1 Unique Characteristics

Although based on the same application as the FabApp, the shipboard deployment presented several key differences. First, the sensor network needed to accommodate disconnections in the 802.11 network, primarily between the gateway nodes and the root node. Typically a sensor cluster inside of a watertight com-

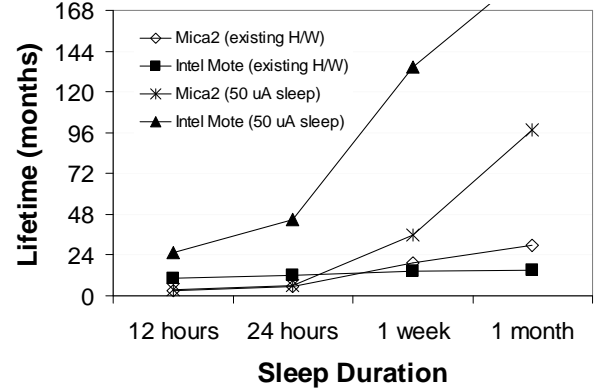


Figure 7. Lifetime of Mica2 and Intel Mote platforms.

partment did not have 802.11 connectivity unless a hatchway was open. Disconnection tolerance was enabled with simple check in the cluster head gateway secure copy routine to check for connectivity prior to transferring data. Data was buffered on the gateway until a connection could be established. The hatchways were typically open, except at night and for particular ship operations.

A watchdog and automated reset feature was also added to the gateway nodes to monitor gateway performance. The gateway was rebooted if the application software hung (i.e. did not periodically reset the watchdog). In addition, after each collection cycle, the gateway would automatically reboot itself. Adding these features reduced the risk of unforeseen problems in the gateway software that would require manual intervention by an operator.

Automated trace data collection and backup features were added to the server. Since access to the ship was only possible in port, this feature was needed to allow the operation of the sensor network to be periodically analyzed, as described in Section 9.3.

Finally, the root and bridge gateway functionality were merged into a single bridge gateway with no serial-link. The bridge acted as a basic 802.11 access point and router between the cluster head gateways and the backend infrastructure. This change was acceptable since external threats to the sensor network were not anticipated due to its location inside the hull.

9.2 Installation

The hardware used for the trial included 150 accelerometers, 26 sensor nodes, 4 Stargates, and 1 PC. Three of the Stargates were used as cluster head nodes and one as the root node. The PC was installed in the Ship's office and was connected to the sensor network via the ship's wired Ethernet. The PC and the root node were assigned static IP addresses from the ship's 24-bit subnet.

The sensor network itself was divided among two compartments in the engineering spaces: starboard 2nd deck and floor (the level beneath the 2nd deck), and center 2nd deck. The starboard compartment had easy access to the ship's wired Ethernet via an unused port in the parts stowage space on the 1st deck. The starboard deployment was further subdivided into two clusters for diagnostic capabilities. The bridge node was installed just outside this compartment on the 1st deck gallery. The center compartment was included in the trial because it could be isolated via a watertight hatch/bulkhead. This allowed evaluation of the sensor network in

an occasionally connected environment. The gateway nodes were placed as in the RF survey, adjacent to available power outlets.

For the trial sensor network software/hardware combination, the sample period is determined by the cluster sleep interval and the number of nodes in the cluster. The user specifies the sleep duration. Data collection time is variable and depends on cluster size and network performance. An entire sample period is the sum of the sleep duration and the data transfer time.

We chose the starboard deployment to exhibit the longest possible sensor network lifetime, and the center deployment to be driven to early failure. Lifetime estimates were made using the baseline energy data reported in Section 8.3. The starboard deployment was set to a sleep duration of 18 hours, the maximum permitted by the current software. At this rate, the sensor network was expected to produce good data for over 82 days with a sample period of ~20.5 hours. The center deployment was set to a sleep duration of 5 hours which was calculated to yield at least 21 days of good results with a sample period of ~7 hours.

The PC software included dynamic web pages, diagnostic collection scripts, and backup script. The web pages provided 'at-a-glance' indication of sensor network health. Each sensor node was listed, along with the time of the last successful data collection. Depending on the time elapsed since the last successful collection, the node was highlighted in red (more than 2 periods overdue), yellow (one period overdue), or green (not overdue). Other pages overlaid sensor locations on a diagram of the engine room. The same color scheme was used to indicate missed collections.

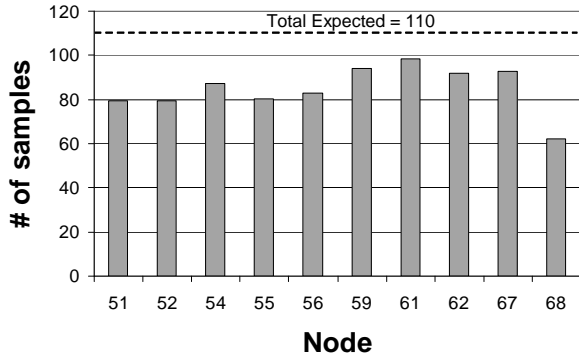


Figure 8. Histogram of total number of samples received/node from the center deployment.

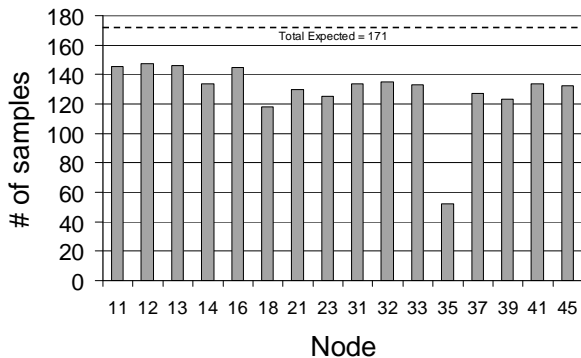


Figure 9. Histogram of total number of samples received/node from the starboard deployment of a 19 week period.

The diagnostic collection scripts collect continuous network packet trace data from the sensor networks in the Starboard compartment. This trace data assists in sensor network diagnosis. Trace data could not be collected from the center compartment because connectivity there was intermittent, and there was insufficient storage capacity on the Stargate for the trace data. Finally, a backup script provided a simple means of copying the trace data, converter logs and the back-end database to a CD which could be delivered ashore for analysis.

9.3 Performance

Sensor network performance was evaluated on two criteria: 1) the ability to collect and deliver data to the backend PC, and 2) the ability to recover from loss or errors. The center deployment produced data for 6 weeks before batteries were exhausted. In the starboard deployment, data collection continued for 19 weeks.

Figures 8 and 9 plot the histogram of vibration samples received from each node in the starboard and center deployments, respectively. Time-series plots (Figure 10 and Figure 11) of received samples provide a global view of sensor network dynamics of the sensor network during the trial period. These results show that the majority of the nodes successfully delivered results at least 80%

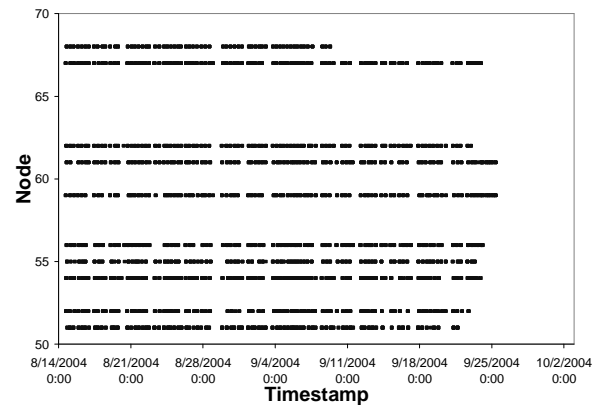


Figure 10. Time-sequence plot of received results for the center deployment. A dot represents a sample received from a particular node. Spacing along the Y axis has no meaning.

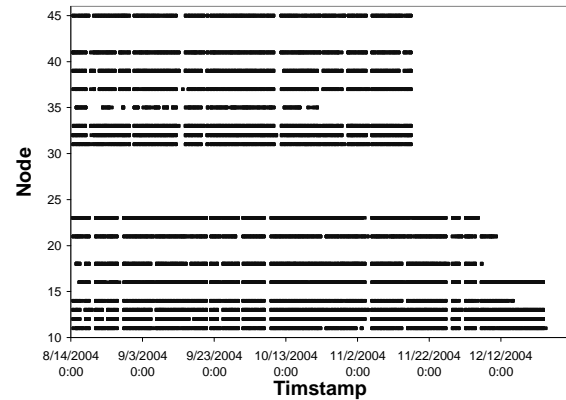


Figure 11. Time-sequence plot of received results for the starboard deployment. A dot represents a sample received from a particular node. Spacing along the Y axis has no meaning.

of the time. Furthermore, most nodes maintained a regular reporting schedule and were able to recover from errors in a previous sampling round. Nodes 35 and 68 were notable exceptions.

Using packet trace data from the starboard deployment, we analyzed some of the specific causes of failures in the sensor network. We focused on reporting gaps, which seemed to occur simultaneously on multiple nodes in a cluster. Table 5 summarizes major findings from the first 11 weeks (a representative sample of the entire run) and suggests possible causes. Similar analysis for the center deployment was not possible because trace data could not be collected. However, we believe that observations from the starboard deployment capture common failure modes.

A key observation from the trace data was that failures were highly correlated within a particular sensor network cluster. In a common case, the cluster would begin a round of data collection normally, but during data collection, the entire sensor network cluster would go silent and no further data collected for the remainder of the period. The cluster head would attempt collection from other nodes without success. However, at the next period, the cluster would recover, with all nodes functioning normally.

Some failures were traced to specific sources. One was a faulty serial port on one of the gateway nodes. While this failure would prevent data collection during the rest of the period, the automated reset would clear the malfunction for the next sample period. Others failures were caused by a ship-wide power failure that resulted in a shutdown of the gateways. Once power was restored, the gateways were able to resynchronize with the sensor network.

The root cause of other failures was more difficult to determine. Failures within a particular cluster could not be easily correlated to similar failures in the other cluster. It is possible that transient RF noise not observed during the original site survey might have interfered with data collection. While a software bug is possible, the correlated nature of failures suggests an external cause.

Node 35 appeared to be a rogue node during the course of the trial, with the lowest yield of all nodes in its cluster. Analysis of the trace data suggests this node may have had a hardware problem which caused it to behave erratically. Node 68 had lower overall reporting yield because it was the first to exhaust its battery supply. While other nodes lasted longer, its lifetime was well beyond the 3 weeks predicted for nodes in the center deployment.

10. CONCLUSIONS

Through two trial deployments, we have shown that predictive maintenance is a viable application of wireless sensor networks. Our cost analysis of various solutions demonstrates that sensor networks can provide high quality data at a relatively low investment in installation and operation. As a result, wireless sensor networks have broad applicability in industrial predictive maintenance, which may represent a killer application of the technology.

In the context of this application, we evaluated two sensing platforms to evaluate the impact of hardware architecture on overall network architecture. We found that providing more capabilities in the sensing platform enabled a simpler and more effective overall system design. Sufficient RAM and I/O bandwidth eliminated the need for external intelligence and buffering to the sensor board, hence reducing complexity, cost, and total energy. In addition, while the Intel Mote platform had a higher active mode

Table 5. Observed failures in the starboard deployment.

Date Time	Nodes	Observations	Possible Cause(s)
8/17 04:00	12,13, 18	All nodes respond at start of cycle. Contact lost during transfer from node 13 (ch 1)	RF interference
8/18 19:55	32-41	Cluster head 4 local serial link to mote failed.	Hardware failure
8/19 17:30	11-12, 14-23	All nodes respond at start of cycle. Contact lost during transfer from node 18 (ch 1)	RF Interference
8/21 06:30	35-39	Affected nodes did not respond at start of cycle.	RF Interference / Loss of sync
8/26 23:00	35-41	Affected nodes did not respond at start of cycle	RF Interference / Loss of sync
8/27 13:30	11-23	All nodes respond at start of cycle. Contact lost during transfer from node 16 (ch 2)	RF Interference
8/31 23:30	31-45	Node 35 flooding network with discovery beacons causing cycle to miss.	Hardware failure
9/5 22:45	12,13, 18	All nodes respond at start of cycle. Contact lost during transfer from node 11 (ch 4)	RF interference
9/13 01:30	31,32, 35,37, 45	All nodes respond at start of cycle. Contact lost during transfer from node 33 (ch 3)	RF Interference
9/14 05:50	31-35, 39-45	Root node 4 local serial link to mote failed.	Hardware failure
9/20 00:30	31,32, 35,39	All nodes (except 35) respond at start of cycle. Contact lost just after node 41 (ch 5)	RF Interference
9/21 02:00	11-23	Root node lost cluster state	Hardware or Power failure
9/30 10:00	11-23	Service discovery not started. Cluster sleeps immediately.	Software Fault
10/7	11-23	Root node was apparently restarted inadvertently causing loss of sync with the network.	Hardware or Power failure
10/8 05:20	11-23	Affected nodes did not respond at start of cycle	Loss of sync due to previous outage
10/10 07:47	31, 33-45	Affected nodes did not respond at start of cycle	RF Interference / Loss of sync
10/11 01:00	31-45	Root node 4 local serial link to mote failed.	Hardware failure

power, the total power consumed in a collection cycle was much lower, due to the lower processing and communication times. Thus, a high-performance sensing platform provides a lower energy per bit cost in this relatively high-bandwidth application.

In our deployments, we found that both the physical environment and equipment layouts in industrial applications lent themselves to sparse clusters of sensors. In this environment, multi-hop is uncommon, but enables rapid deployment and tolerance to changing RF conditions. Building code regulations forced battery-powered networking despite power availability. The resulting naturally tiered architectures allowed centralized data collection and power management protocols.

Finally we demonstrated the use of a sensor network to meet a four month continuous operation requirement. Several techniques, including the offloading of critical state from sensor nodes to cluster heads, watchdogs, and periodic resets of system state enabled sufficient reliability for completely unattended operation.

Overall, the data collected in this trial was of sufficient quantity to demonstrate proof of concept for the application. In the next phase of the project, we will make operational use of a sensor network for PdM decision making, to demonstrate its ability to replace manual data collection. We will also utilize the processing abilities of the platform by pushing the FFT computation to the edge of the network. Edge processing will increase network lifetime and enable greater collection frequencies.

Predictive maintenance is just one of many possible applications in this environment. This deployment has allowed us to create a reality check for commercial use of sensor networks in industrial applications. In the end, these settings provide a rich environment for a wide spectrum of sensing applications.

11. REFERENCES

- [1] Bilstrup, U. and Wiberg, P.-A. Bluetooth in industrial environment. In *Proceedings of the 11th IEEE International Workshop on Factory Communication Systems, (WFCS 2000)* (Portugal, September 2000).
- [2] Cerpa, A., Elson, J., Estrin, D., Girod, L., Hamilton, M., and Zhao, J. Habitat monitoring: application driver for wireless communication technology. In *Proceedings of the ACM Sigcomm Workshop on Data Communication* (San Jose, Costa Rica, April 2001).
- [3] Crossbow, Inc., MICA2 mote, <http://www.xbow.com/Products/productsdetails.aspx?sid=72>
- [4] Estrin, D., Govindan, R., Heidemann, J. and Kumar, S. In *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCOM '99)* (Seattle, WA, August 1999).
- [5] Kjesbu, S. and Brunsvik, T. Radiowave propagation in industrial environments. In *Proceedings of the IEEE International Conference on Industrial Electronics, Control and Instrumentation (IECON 2000)* (Nagoya, Japan, October 2000).
- [6] Lymberopoulos, D., Hseush, J., Mascia, J., Tully, S., Barton-Sweeny, A., Lindsey, Q., Kelly, M., and Savvides, A. Light-based localization and XYZ node architecture. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN 2005)* (Los Angeles, CA, April 2005).
- [7] Nachman, L., Kling, R., Adler, R., Huang, J., and Hummel, V. The Intel mote platform: a bluetooth-based sensor network for industrial monitoring. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN 2005)* (Los Angeles, CA, April 2005).
- [8] Olney, D. Vibration monitoring with wireless networks. http://www.qualitymag.com/CDA/ArticleInformation/features/BNP_Features_Item/0,6425,98976,00.html
- [9] Paek, J., Caffrey, K.C., Govindan, R., Masri, S. A wireless sensor network for structural health monitoring: performance and experience. In *Proceedings of the Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II)* (Sydney, Australia, May 2005).
- [10] Ramanathan, N., Yarvis, M., Chhabra, J., Kushalnagar, N., Krishnamurthy, L., and Estrin, D. A stream-oriented power management protocol for low duty cycle sensor network applications. In *Proceedings of the Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II)* (Sydney, Australia, May 2005).
- [11] Rappaport, T. Indoor radio communications for factories of the future, *IEEE Communications Magazine*, 27, 5, (May 1989), 15-24.
- [12] Rappaport, T. Characterization of UHF multipath radio channels in factory buildings. *IEEE Transactions on Antennas and Propagation*, 37 (Aug. 1989) 1058-1069.
- [13] Rappaport, T. and McGillem, C. UHF fading in factories. *IEEE JSAC*, 7, 1 (Jan 1989), 40-48.
- [14] Sadler, C, Zhang, P., Martonosi, M., and Lyon, S. Hardware design experiences in ZebraNet. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems* (Batimore, MD, Nov. 2004).
- [15] Sencicast Press Release. http://www.sencicast.com/news_events/news_events-detail.081004.php
- [16] Sensor Network Magazine. Condition based monitoring, <http://www.sensorsmag.com/articles/0602/14/main.shtml>
- [17] Staub, O., Zurcher, J.-F., Morel, P., and Croisier, A. Indoor propagation and electromagnetic pollution in an industrial plant. In *Proceedings of the IEEE International Conference on Industrial Electronics, Control and Instrumentation (IECON)* (New Orleans, LA, Nov. 1997).
- [18] Szewczyk, R., Polastre, J., Mainwaring, A., Anderson, J., and Culler, D. An analysis of a large scale habitat monitoring application. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems* (Batimore, MD, Nov. 2004).
- [19] Tiwari, A., Lewis, F.L., and Ge, S.S. Wireless sensor networks for machine condition based monitoring. In *Proceedings of the Int. Conf. Control, Automation, Robotics, and Vision* (Kunming, China, Dec 2004).
- [20] Wan, C.-Y., Campbell, A.T., and Krishnamurthy, L. Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks. *IEEE Journal on Selected Areas in Communications*, 23, 4, (April 2005) 862-872.
- [21] Xu, N., Rangwala, S., Chintalapudi, K., Ganesan, D., Broad, A., Govindan, R., and Estrin, D. A wireless sensor network for structural monitoring, In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems* (Batimore, MD, Nov. 2004).
- [22] Yarvis, M.D., Conner, W.S., Krishnamurthy, L., Chhabra, J., Elliott, B., and Mainwaring, A. Real-world experiences with an interactive ad hoc sensor network. In *Proceedings of the International Workshop on Ad Hoc Networking (IWAHN 2002)* (Vancouver, B.C., Canada, August 2002), 143-151.
- [23] Yarvis, M., Kushalnagar, N., Singh, H., Rangarajan, A., Liu, Y., and Singh, S. Exploiting heterogeneity in sensor networks. In *Proceedings of the IEEE International Conference on Computer Communication (INFOCOM 2005)* (Miami, FL, March 2005).
- [24] Zeevo, Inc, TC2001P Product Brief, http://www.zeevo.com/pdf_files/TC2001P_HCI_prodbrief_v1.2.pdf