Smart Structures and Systems, Vol. 10, No. 3 (2012) 271-298
DOI: http://dx.doi.org/10.12989/sss.2012.10.3.271

WiSeMote: a novel high fidelity wireless sensor network for structural health monitoring

Davis P. Hoover[1], Argenis Bilbao[1] and Jennifer A. Rice*[2]

[1]Department of Electrical and Computer Engineering, Texas Tech University, Lubbock, Texas, USA
[2]Engineering School of Sustainable Infrastructure and Environment, University of Florida, Gainesville, Florida, USA

(Received December 16, 2011, Revised July 20, 2012, Accepted July 22, 2012)

Abstract. Researchers have made significant progress in recent years towards realizing effective structural health monitoring (SHM) utilizing wireless smart sensor networks (WSSNs). These efforts have focused on improving the performance and robustness of such networks to achieve high quality data acquisition and distributed, in-network processing. One of the primary challenges still facing the use of smart sensors for long-term monitoring deployments is their limited power resources. Periodically accessing the sensor nodes to change batteries is not feasible or economical in many deployment cases. While energy harvesting techniques show promise for prolonging unattended network life, low power design and operation are still critically important. This research presents the WiSeMote: a new, fully integrated ultra-low power wireless smart sensor node and a flexible base station, both designed for long-term SHM deployments. The power consumption of the sensor nodes and base station has been minimized through careful hardware selection and the implementation of power-aware network software, without sacrificing flexibility and functionality.

Keywords: structural health monitoring (SHM); wireless smart sensors; mesh networks; ultra-low power

1. Introduction

Structural health monitoring (SHM) is a rapidly growing research area of increasing importance for the maintenance and reliability of civil infrastructure. While wired sensor networks are costly to install and maintain wireless sensor networks can reduce these costs while harnessing the latest low power and energy harvesting technologies. Such networks can be of great benefit in preserving government investment in infrastructure and improving public safety. By providing a powerful communication framework with which to access data as well as software tools for network configuration, these networks will allow access to a user-configurable system that can provide a wealth of structural information.

Wireless smart sensor networks (WSSNs) are composed of two main subsystems: 1) the sensor nodes that comprise the network and 2) a centralized base station that controls the network and acts as a gateway to the outside world. The first subsystem consists of an ensemble of wireless and usually battery powered data collection nodes that are distributed within an environment being

*Corresponding author, Assistant Professor, E-mail: jrice@ce.ufl.edu

monitored. The nodes are responsible for acquiring, processing (to various degrees) and communicating data and information within the network. Each node consists of a low power processor, onboard or externally connected sensor(s), a radio transceiver capable of communicating with other nodes and the base station, and other onboard or peripheral devices such as memory, power management chips, etc. WSSNs used for SHM applications require low power nodes that allow for low maintenance cost and long-term field deployments. Of the tasks typically carried out on the sensor nodes, communication is often energy intensive, and in the case of the WiSeMote the radio largely dominates the level of power consumption. It is, therefore, important to avoid sending large amounts of raw data within the network, which can be achieved by processing the data on the node prior to communication. While some existing networks (Polastre *et al*. 2005 and Whelan and Janoyan 2009) employ ultra-low power consumption components, many lack the memory and processing capabilities to collect large data records and perform larger amounts of in-network data processing. This paper presents a new node with faster processor speeds and external parallel SRAM for needed storage and processing of which other systems are not capable. High-fidelity data (that is tightly synchronized in time and has low measurement noise) is required for vibration-based SHM applications (Nagayama *et al*. 2007). The nodes developed in this research include a unique analog front-end that provides a stable voltage source and buffers the data collected from an onboard accelerometer in order to minimize measurement noise. Additionally, hardware and software measures have been implemented to ensure data from each of the nodes in the network is highly synchronized. For the system presented in this paper, a robust, highly functional wireless network has been developed that allows for a larger network that maintains low power consumption. The node radios use the DigiMesh protocol, a reliable and self-healing proprietary communication protocol with increased routing capabilities.

The next major subsystem of WSSNs is the base station. The roles and responsibilities of the base station, as well as the included hardware, vary from system to system. Base station responsibilities typically include commanding the network, processing and analyzing data, logging data, and then communicating data with the outside world. To perform these tasks, many WSSN deployments use a PC interfaced with a gateway radio transceiver for communication with the wireless nodes. The deployment achieved by Jang *et al*. (2010) includes a PC loaded with Windows XP Professional OS and appropriate software to communicate through a gateway node. While a PC provides a quick, adequate solution for performing processing in a centralized fashion, significant installation and setup are required to establish such a network and the cost is unnecessarily high. The WiSeMote system presented in this paper provides a dedicated hardware solution for centralized network control and data storage. Unlike a PC, all of the base station's processing resources are available to the networking task at hand. This dedicated hardware module will also be out-of-the-box installation ready, as opposed to PC-based base station approaches. Further configuration such as establishing default sampling settings, configuring sample timers, and setup for communication with the outside world may still be necessary. The base station unit described herein includes similar radio and processor hardware as the wireless sensor nodes as well as extra peripherals to communicate with remote hosts (hosts at a location other than the network installation site). A GPRS module is included for wireless communication with a remote host and an Ethernet/web server module for similar wired communication. The GPRS module allows for WSSN installation in environments where wired base station accessibility is not preferred or feasible.

This paper outlines the successful design, build and testing of the WiSeMote: a wireless sensor network utilizing a dedicated hardware base station and low power node that is intended for long-term structural health monitoring. The hardware and software design will be discussed as well as experimental validation of power consumption and data acquisition consistency. Overall network performance, including time synchronization of data acquisition and reliable communication will also be evaluated. The target application for the WiSeMote is long-term vibration based structural monitoring that utilizes in-network data processing to communicate timely and pertinent information on the health of the structure over an extended deployment period. The WiSeMote hardware allows for larger amounts of in-network data processing than many other WSSNs while maintaining reasonable power consumption for long-term deployment. The WiSeMote also allows for the potential of real-time data transfer for single-hop communities within larger multi-hop networks. Sampling rates achieved by the WiSeMote include typical vibration responses below a few hundred Hertz.

2. Related work

Many wireless network sensing technologies have emerged in recent years. Telos (Polastre *et al*. 2005) used by Whelan and Janoyan (2009), Narada nodes (Swartz *et al*. 2005), Imote2 (MEMSIC, Inc. 2007) and Waspmote (Libelum 2010) are just a few whose aim towards low-power structural health monitoring are comparable to the goals of the WiSeMote. Though there are many other comparable platforms, the capabilities of these few systems along with the WiSeMote are given in Table 1 for comparison.

Telos was the last of a line of Berkeley "motes" developed from 1998 to 2005. An updated version, the TMote Sky, was deployed in 2009 on an in-service highway bridge (Whelan and Janoyan 2009). Telos included a 16-bit low power Texas Instruments MSP430 microcontroller with 8 MHz maximum clock speed and eight 12-bit ADCs with non-simultaneous sampling. Optional onboard humidity and

Solar radiation sensors were also available. Telos was intended for low power, high throughput data acquisition, leveraging a lower power processor and a 1MB external flash chip for extra data storage. However, reading and writing from flash consumes a significant amount of current and can be slow, affecting the node battery lifetime and limiting the amount of data that can be saved or processed on the nodes.

The Narada is an academic wireless sensor prototype designed by researchers at the University of Michigan (Kurata *et al*. 2010). The Narada is designed for low power, high fidelity data acquisition with a communication protocol that allows for accurate time synchronization and network scalability. Included in this platform is an 8-bit ATmega128 processor with max 8 MHz clock, a 4-channel, 100 kHz Texas Instruments ADS8341 16-bit ADC, 128 KB of external SRAM and a Chipcon CC2420 radio. The Narada's extended onboard memory, high-resolution ADC, and low-noise four-layer PCB make it well suited for acquisition of extended, high-fidelity data records (Swartz *et al*. 2005). The ATmega processor speed, however, does limit in-network processing capabilities. Lower sleep current consumption has also been achieved on other platforms.

Libelium recently introduced the Waspmote, an all-in-one system for wireless sensing. It includes an ATmega1281 processor running at 8 MHz, with 128 KB flash, 8 KB RAM, and eight 10-bit ADCs. It offers the choice of several XBee radios with various protocols including 802.15.4, ZigBee, DigiMesh, and others, all of which are discussed further in Section Wireless protocol. The Waspmote includes connectivity for several peripheral sensor boards, which can measure

Table 1 WSSN platform comparison

	Telos (2005)	Narada (2005)	Imote2 (2007)	Waspmote (2010)	WiSeMote (2011)
System current draw					
Active*	≤ 15 mA	30 mA	46 mA	9 mA	≤ 5 mA
Sleep**	54.5 μA	10 mA	N/A	62 μA	150 μA
Hibernate***	5.1 μA	N/A	500 μA	0.7 μA	N/A
Microcontroller					
Type	16-bit TI MSP430F1611	8-bit ATmega128	32-bit X-Scale PXA271	8-bit ATmega1281	16-bit TI MSP430F47187
Max clock speed	8 MHz	8 MHz	416 MHz	8 MHz	16 MHz
RAM size	10 KB	4 KB	256 KB SRAM 32 MB SDRAM	8 KB	8 KB
Flash size	48 KB	128 KB	32 MB	128 KB	120 KB
ADC channels	8 12-bit (non-simul.)	4 16-bit (simul.)	N/A (only digital inputs)	8 10-bit	7 ΣΔ 16-bit (simul.)
Max ADC Sampling rate	200 kHz	10 kHz	N/A (external sensor board)	15 kHz	32.768 kHz
Radio					
Type	Chipcon CC2420	Chipcon CC2420	Chipcon CC2420	XBee****	XBeeDigiMesh
RX current	19.7 mA	19.7 mA	19.7 mA	40 mA****	50 mA
TX current	17.4 mA	17.4 mA	17.4 mA	40 mA****	45 mA
Sleep current	0.3 μA	0.3 μA	0.3 μA	< 1 μA****	< 50 μA
Wireless Protocol	802.15.4 only	802.15.4 only	802.15.4 only	ZigBee/ 802.15.4****	DigiMesh/ 802.15.4
RF data rate	250 kbps	250 kbps	250 kbps	24-250 kbps	250 kbps
External modules					
RAM size	--	128 KB	--	--	256 KB
Flash size	1 MB	--	--	--	4 MB
SD card	--	--	--	nodes	base station
GPS	--	--	--	nodes	base station
GPRS	--	--	--	nodes	base station
USB	nodes	--	nodes	nodes	base station

*MCU fully awake, worst case consumption for peripheral modules, radio asleep
**All modules asleep, MCU in idle state
***All modules asleep, some modules completely electronically disconnected. Module cannot quickly wake up from sleep
****XBee-802.15.4, 802.15.4-Pro, ZB, ZB-Pro, 868, 900, or XSC Available. Attributes vary

acceleration, luminosity, soil moisture, and temperature. GPS and GPRS modules are also optionally included. The Waspmote is best suited for centralized data processing at the base station because of its lack of onboard memory and processing speed on the nodes. A single Waspmote connected to a local PC acts as the network base station.

The Imote2 (MEMSIC 2007) has been widely investigated for use in SHM applications (Jang *et al*. 2010, Rice *et al*. 2010, Nagayama *et al*. 2010). Its scalable processor speed and relatively large amount of onboard memory make it particularly well-suited to higher sampling rate and processing intensive applications such as SHM. The primary drawback of the Imote2 is its high power consumption when used with the ISM400 (SHM-A) accelerometer sensor board. During active sensing, the Imote2/ISM400 consumes more than 160 mA (Rice *et al*. 2010), an order of magnitude higher than the WiSeMote and other platforms described in this section. Despite its popularity in academic applications, the Imote2 is no longer in production.

The WiSeMote is geared towards high data rate applications and distributed data processing. For this reason, this system slightly trades off power consumption for processing capabilities. The increased active power consumption of the WiSeMote in Table 1 is largely due to the radios and their DigiMesh protocol, which allow for increased functionality. More information on DigiMesh and other wireless protocols is given in Section Wireless protocol. Instead of using the typical SPI flash memories that the other sensor nodes are using, this system uses a low power parallel SRAM chip. The SRAM enables the node to access data stored externally from the microcontroller very efficiently in just a few clock cycles. Since this SRAM chip is always powered, both the microcontroller and SRAM chip can go into sleep mode and retain the data.

With a CPU that is able to run at 16 MHz, twice as fast as the Telos and WaspMote, simple compression algorithms (that enable significant savings in communication time) can practically be implemented as well as data processing and analysis within each node. These are features not commonly available in wireless sensor nodes. When the nodes do not require peak performance, the processor speed can be reduced to 1.0 MHz to save energy.

3. Platform description

3.1 Sensor node

The motivations in the development of the nodes were to provide improvement in the network reliability, measurement noise mitigation, power consumption, and signal acquisition and throughput capabilities of the currently available remote SHM systems. To achieve these goals, careful component selection was exercised, paying particular attention to their individual current consumptions. The dimensions of the data collection node are $5.1 \times 6.4 \times 1.9$ cm (2.00” × 2.50” × 0.75”). A WiSeMote node is seen in Fig. 1. Contained in the node PCB are five major components: 1) a Texas Instruments MSP430F47187 microcontroller (MCU), 2) a parallel SRAM chip external to the microcontroller, 3) an external serial flash memory chip, 4) an XBee DigiMesh radio transceiver module, and 5) an onboard three-axis accelerometer. Additional sensors may be utilized through an expansion connector on the bottom PCB layer. The node PCBs are housed in a plastic enclosure with an external connector for the radio antenna and an on/off switch accessible on the outside of the enclosure. A 1.76 Ah Lithium-Ion battery is adhered to the enclosure lid using double sided tape. The battery operates at 3.7 V and from -20°C to 60°C. It is rechargeable, but requires opening the lid for charging with the existing enclosure. Information about the Lithium-Ion battery life is given in Section Node current consumption. Note that this battery is only intended for development purposes. Four PCB offsets are used to secure the node to the enclosure with screws. Four magnetic mounts are attached to each node. A Neodymium magnet with a countersunk screw hole is attached

Fig. 1 Sensor node circuit board with enclosure, enclosure lid with Li-Ion battery, and neodymium magnetic mounts

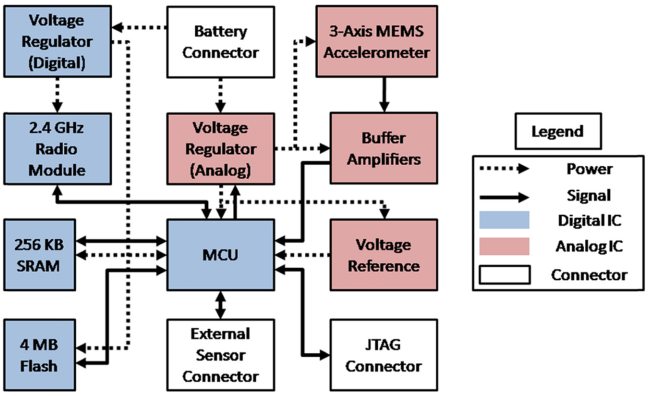

Fig. 2 Node block diagram

to the bottom of each of the four PCB standoffs on the outside of the enclosure for easy installation to metallic structures. The magnetic mounts can be seen in Fig. 1.

3.1.1 Digital circuitry

The node MSP430 microcontroller handles and coordinates all operations occurring in the node at all times, with a maximum clock speed of 16 MHz, 120 KB of flash, 8 KB of RAM, and seven Sigma-Delta 16-bit ADCs that can be simultaneously sampled. A fast, low power, parallel 256 KB SRAM chip is populated on the nodes to provide additional storage for data while it is being analyzed. The SRAM chip typically draws 1.6 mA while active and 1 μA while in standby. Using an 8-bit data port, the SRAM chip can support read data transfer rates of up to 100 KB/s and write speeds of 66 KB/s. This bandwidth enables the node to acquire data from all seven ADCs simultaneously at sampling rates greater than 1500 Hz. Our usable sampling rates are limited, however, to 1500 Hz because of the analog anti-aliasing filter. Additionally the 4 MB external flash chip can be used to store data for reuse after power cycling the nodes. This also provides the opportunity for the node to externally store a different firmware for the node microcontroller should the application needs change after deployment. Other potential uses for the external flash include storage of long sine/cosine lookup tables for internal calculations, image storage for camera applications, or data storage for extended data records. The onboard XBee DigiMesh 2.4 GHz radio transceiver is used to offload the raw or processed data gathered by the node to the base station or to other nodes when data sharing between nodes is required. The radio selection criteria and more information about the XBee radios are given in Section 3.3.

3.1.2 Analog front-end circuitry

The nodes have a three-axis accelerometer mounted on the PCB. Because short signal paths reduce track impedances and provide greater noise immunity, the accelerometer is mounted as close as possible to the microcontroller (less than 3 cm away). The accelerometer chosen is the LIS344ALH inertial MEMS sensor manufactured by ST Microelectronics. With two software-selectable measuring ranges, ±2 g and ±6 g, the accelerometer can be used in a wide range of applications by simply selecting the appropriate range. The sensor is able to operate in temperatures ranging from -40°C to +85°C with a typical temperature sensitivity change of ±0.01%/°C. Maximum current consumption is 850 μA while active. The maximum power-down current is 5 μA.

The accelerometer has three analog outputs, each with voltage proportional to the acceleration along that particular Cartesian axis. These outputs have a high impedance of 110 kΩ ±20%. A buffer amplifier is included for each axis. The use of buffer amplifiers lowers the accelerometer's effective output impedance and reduces the output resistance tolerance. The next stage consists of a low pass RC filter on each of the three accelerometer outputs. Each filter scales the amplitude of the analog voltage to the voltage range required by the MSP430 ADC as well as low pass filters the signal with a cut-off frequency of 750 Hz to prevent aliasing by the ADC. This allows for sampling data at rates up to 1500 Hz. The MSP430 user's manual recommended cutoff is below 10 kHz for the ADCs but we use a lower cutoff because structural vibration responses are often in the frequency range well below that. Note that the analog anti-aliasing filters introduce a group delay to the sampling process given in seconds by

$$\tau = \frac{1}{1+[\tan^{-1}(-0.5\,\omega(1\cdot 10^{3}))(22\cdot 10^{-9})]^{2}}, \tag{1}$$

where ω is the frequency of the analog signal in radians and the group delay is given as the negative derivative of the analog filter's phase response.

The Sigma-Delta ADC uses oversampling and has an internal digital comb filter whose cutoff frequency is adjustable. This digital filter acts as an anti-aliasing filter with a cutoff frequency that is changed according to the requested sampling rate. If the accelerometer outputs were connected directly to the external analog filters (without an intermediary buffer amplifier) the cutoff frequency of the analog filters would vary significantly due to the high tolerance that the accelerometer's output resistors have (±20%). This could potentially create significant phase shifts between the signals collected from different nodes, thus undermining data synchronization between different nodes. The buffer amplifier alleviates this problem.

3.1.3 Power supply circuitry & low power operation

The analog front end includes two voltage references and a dedicated low dropout regulator. The regulator is used to supply and power down the entire analog front-end, thus eliminating the front-end power consumption when the node is asleep waiting for commands. One of the voltage references is used to power the accelerometer. Voltage references are similar to voltage regulators but include temperature compensation and typically provide less power. This provides the accelerometer with low noise, temperature compensated, stable power, which is critical since the output voltage of the accelerometer is ratiometric to its power supply. The other voltage reference is used as a regular reference input for the MSP430 ADC.

A second voltage regulator provides power only to the digital circuitry and it is active at all times.

Table 2 Active clocks during low power modes

Mode	CPU	MCLK	SMCLK	ACLK	FLL	Internal Reference
Active	On	On	On	On	On	On
LPM0	Off	Off	On	On	On	On
LPM1	Off	Off	On	On	Off	On
LPM2	Off	Off	Off	On	Off	On
LPM3	Off	Off	Off	On	Off	Off
LPM4	Off	Off	Off	Off	Off	Off

The analog and digital circuitry each have separate regulators in order to reduce the presence of noise from the digital circuitry in the analog circuitry, which will show up in the accelerometer output. A block diagram of the power supply circuitry is shown in Fig. 2.

The MSP430 supports five low power modes, denoted as LPM0 through LPM4. All low power modes turn the CPU off but keep different internal hardware modules (and their supplied clocks) enabled. This flexibility is useful when a task only requires a particular module to be active, but saving as much power as possible is required. Table 2 shows the different clock lines that continue to be active once a low power mode is activated. MCLK, SMCLK and ACLK (master clock, sub-master clock, and auxiliary clock, respectively) are clock lines internal to the MSP430. These clock lines can be configured to provide clock signals to the MSP430's internal peripherals such as timers, ADCs, etc.

To implement regularly scheduled data collection, a sampling timer is needed. For this reason, not all of the clock signals can be off when the sensor node is sampling data. SMCLK, sourced by the frequency-locked loop (FLL), is the clock used to source the sampling timer and the ADC module. Once the first sample from each enabled ADC channel is taken, the node goes into LPM0 waiting for an interrupt from the timer. The timer interrupt wakes up the CPU and repeats this process until all samples are taken. The data samples from each channel are then placed in the external SRAM chip. When all the samples are collected, the SRAM chip is disabled, the analog front-end is turned off and the CPU goes back into LPM3, waiting for the base station to request the data just acquired. Table 3 shows the measured current consumption in every network state.

From the experimental data shown in Table 3, it is clear that the WiSeMote battery life is almost

Table 3 Node measured current consumption

State	Current	Active Blocks
Idle	150 μA	None
Sampling @ 20 Hz	3.68 mA	Analog front-end, SRAM, CPU
Sampling @ 40 Hz	3.69 mA	Analog front-end, SRAM, CPU
Sampling @ 100 Hz	3.82 mA	Analog front-end, SRAM, CPU
Sampling @ 250 Hz	4.13 mA	Analog front-end, SRAM, CPU
Sampling @ 400 Hz	4.44 mA	Analog front-end, SRAM, CPU
Sampling @ 500 Hz	4.55 mA	Analog front-end, SRAM, CPU
Compressing Data	8.93 mA	SRAM, CPU
Transmitting Data	45 mA	SRAM, CPU, Radio
Receiving Data (Commands)	50 mA	CPU, Radio

entirely governed by the duration of radio wake time since the radio consumes much more current than any other circuitry. For this reason, average radio duty cycles of one percent (i.e., the radio is only awake one percent of the time) and below are advisable. Such low duty cycle operation is common in WSSN protocols such as ZigBee (Ergen 2004). The base station can conveniently change the node's duty cycle wirelessly if desired.

3.1.4 Wireless communication protocol

The following is a description of the upper level wireless communication protocol for data acquisition. The upper level protocol consists of two layers that are programmed on the MSP430: the application programming interface (API) layer that formats packets for XBee transmission and

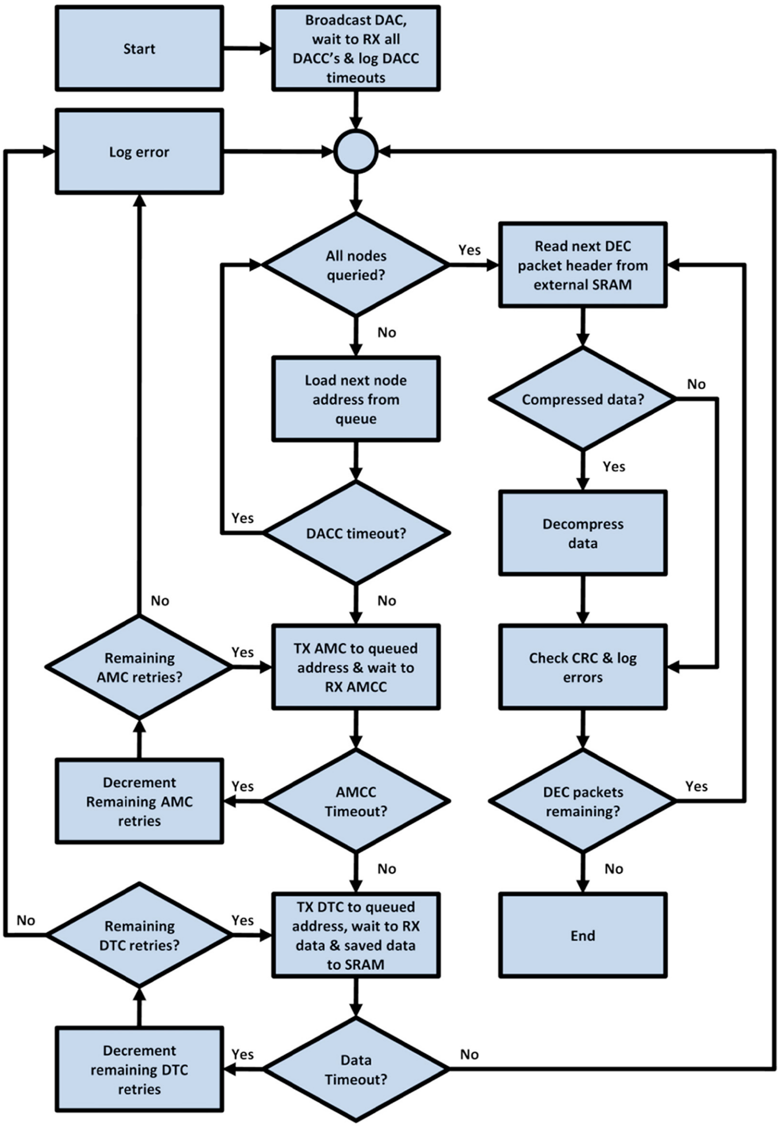

Fig. 3 Base station communication protocol flowchart

the decompression and error checking (DEC) layer that controls data decompression and cyclic redundancy checks (CRCs). The DEC protocol was designed to allow for simultaneous sampling and queued data transmission on the WiSeMote nodes with added retries, timeouts, and error checking for increased fault tolerance redundancy. The DEC protocol is intended for queuing data in external RAM as it a sampled and transmitted that data to the base station in a sequential node-by-node fashion once all samples for all nodes are taken. Lower level network DigiMesh protocols that control routing, hopping, radio synchronization and sleep/wake cycling are handled by the radio module and are abstract to the microcontroller. The DEC protocol is slow and not resource efficient, however it allows for proof-of-concept data acquisition. Future work will enable more efficient data acquisition implementations such as real-time data processing distributed among the nodes. The DEC protocol maintains a master/slave relationship between the base station and nodes during data acquisition.

First the network wide data acquisition process is described and then the low power operation of the nodes is explained in the context of communications and data acquisition. The flowchart for the communication protocol for a data acquisition on the base station is shown in Fig. 3. A data acquisition can be initiated by a scheduled real-time clock interrupt on the base station or an immediate command from the WiSeMote Command Utility GUI via the base station USB. The Data Acquisition Command (DAC) is broadcast to the nodes commanding them to start sampling data. Each node responds with a Data Acquisition Completion Command (DACC) once it is finishes sampling data. The base station then subsequently queries each node. During a query, the base station sends an Awake Mode Command (AMC) to set the queried node radio to be awake all the time, then sends a Data Transmit Command (DTC) commanding the queried node to send its data record. The queried node's radio is set to be awake so that the data transmission occurs all at once instead of in small bursts. The node responds to the AMC and DTC with an Awake Mode Command Confirmation (AMCC) and the entire requested data record, respectively. The AMC and DTC both include timeouts and retries (on the base station side) based upon the AMCC and data record responses. These uppermost level timeouts and retries attempt to provide reliable network communication. The initial DAC includes only a single timeout for the DACC response. If timeouts occur and/or retries expire for any of the replies (DACC, AMCC, or the data record), the base station logs the error and moves on to query the next node. The node includes retries for each of its confirmation responses. Once all nodes are queried, the node returns its radio to sleep mode and the DEC protocol decompresses each data record as needed and checks the CRCs. Note that each node's data record is sent as multiple DEC packets.

Next we outline the upper level wireless communication protocol at the node level in the context of its low power operation as illustrated in Fig. 4. At all times, the node maximizes the time it spends in low power modes. The node hardware has been designed to properly operate most of the time in LPM3. This is the case when there are no commands being executed (idling).The nodes are accessible via radio in low power modes LPM0 through LPM3. Most commands contain simple network status messages that, once received, are decoded and handled. Once the command is executed the nodes return to LPM3. For commands like the DAC, the process is slightly different. For example: if the node gets a DAC command requesting to sample the accelerometer at 10 Hz for 200 s, the node wakes up, processes the command and sets up its internal registers to start the sampling operation. This includes enabling the analog front-end, enabling the external SRAM chip and setting up the ADC and timer interrupts. Next, the node samples at the requested rate for the requested amount of time, processing the other base station commands as they are received. Once

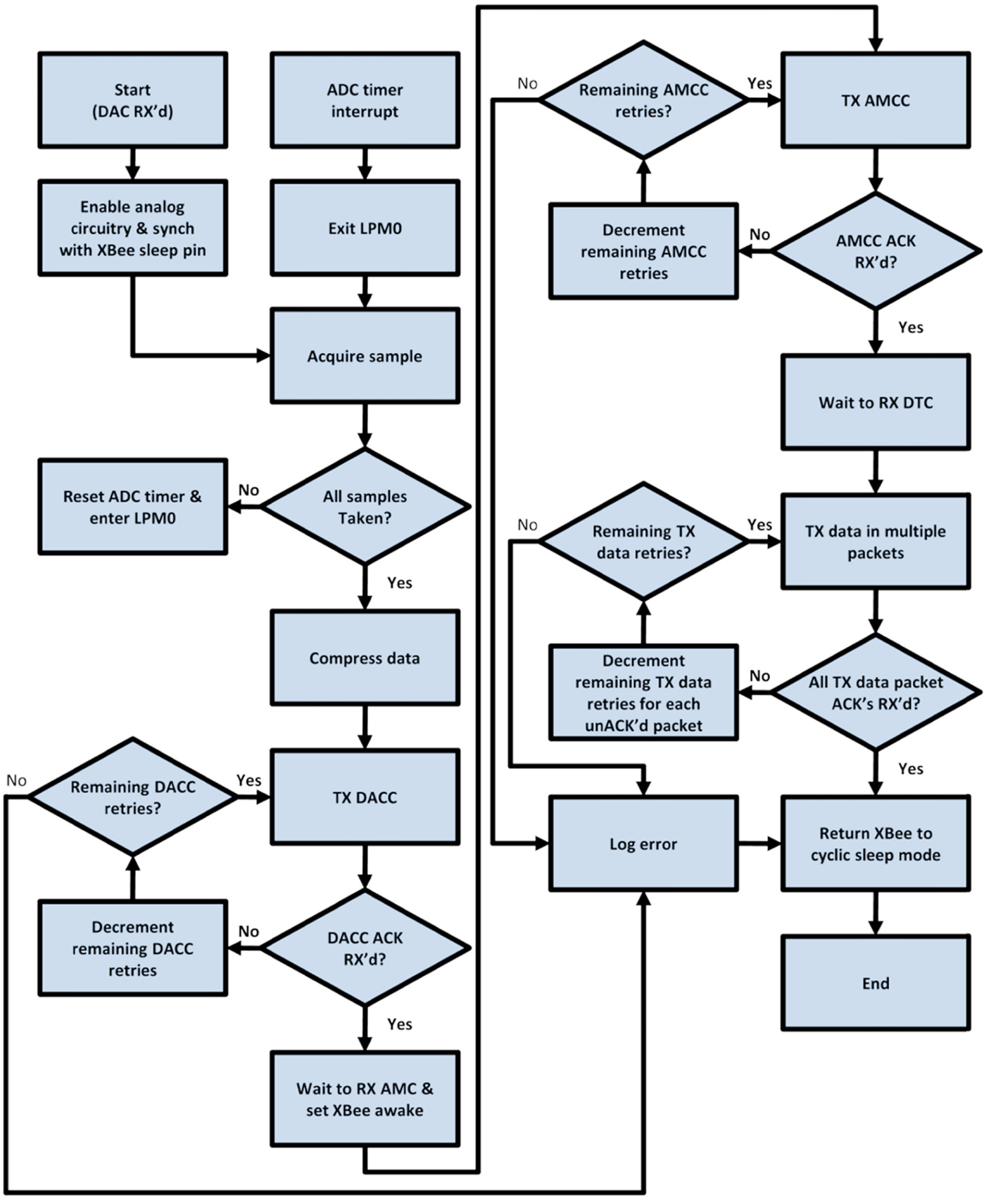

Fig. 4 Node communication flowchart and low power operation

all samples are taken the node compresses the data, sends the DACC, waits for the AMC, sends the AMCC and wakes up the radio, waits for the DTC, transmits the data once the DTC is received and then returns the radio to sleep and the microcontroller to LPM3.

3.2 Base station

The base station serves as the centralized control data concentration unit for the network. It is responsible for commanding and querying remote nodes as well as acting as a server that can share

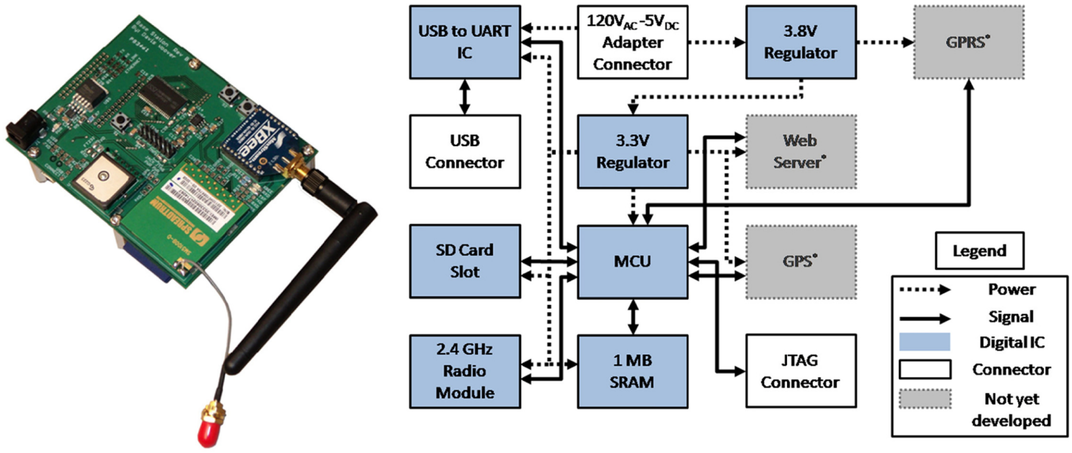

Fig. 5 Base station (left) and block diagram (right)

collected data with clients in the outside world. The nodes collect and send data back to the base station, which then logs the calculations locally on an SD card. The base station PCB dimensions are 10 × 8 × 1.9 cm (3.94” × 3.15” × 0.75”).

The base station, shown in Fig. 5, is powered by a 120 V to 5 V power adapter which is regulated down to 3.3 V and 3.8 V by voltage regulators for its various hardware modules. Processing is provided by a Texas Instruments MSP430F5438 microcontroller. The MSP430F5438 has up to 18 MHz processor clock speed, 256 KB flash and 16 KB RAM. A 1 MB external parallel SRAM chip is included for larger amounts of data processing. The SD card slot is also included for providing permanent data storage. SD cards with capacities of a few GB are now widely available and inexpensive. An FT232RL chip provides USB to RS232 interface on the microcontroller. The base station data acquisition settings can be configured by a local PC via the USB port. The Spreadtrum SM5100-B GPRS module offers wireless remote access to data. The SUP500F GPS module provides the option for saving global location and time information along with each recorded data record or event. The GPS module is not required for our application, however. The GPS, GPRS and Ethernet modules are not currently operational on the base station but are planned for future development. The base station has an XBee-Pro DigiMesh 2.4 GHz RF Module for node mesh communication onboard. It has the same DigiMesh communication protocol as the node radios but has extended communication range (see Tabel 4), allowing broadcast commands to potentially be received in fewer hops.

3.2.1 Base station software

The software implemented on the base station serves to control network operations such as communication and synchronization, as well as compute commonly-used data processing algorithms. The base station uses a radix-2 Fast Fourier Transform (FFT) algorithm for computing the Discrete Fourier Transform (DFT) of sampled data. Either the complex or magnitude DFT can be computed. The base station also uses a windowed frequency domain approach to compute the cross correlation of two simultaneously sampled data records from separate nodes. This computation, when performed at the node level, helps network data aggregation and provides the input for many widely used modal analysis and SHM algorithms implemented on WSSNs (Lynch *et al*. 2005, Zimmerman

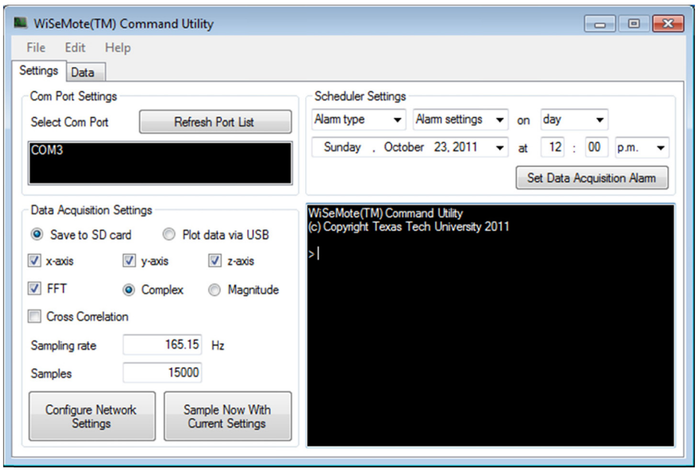

Fig. 6 Base station command utility GUI

2008, Nagayama and Spencer 2010, Sim *et al*. 2010).

The base station can be configured through USB via a command utility GUI shown in Fig. 6. The utility allows the user to configure system settings such as sample rate, number of samples and which axes to sample from. The user may also specify which signal processing algorithms to execute such as FFT and cross correlation. The GUI can also be used to configure the base station to set up a sampling schedule, such as sample every day at 4:00 p.m., once a week, etc. When a sampling schedule is configured, the real time clock is configured on the base station microcontroller to wake up at the specified time. Upon power up, the base station initiates all peripheral modules and begins running in an idle state waiting for an initiation of a data acquisition. Either an immediate external command from the command utility GUI or an interrupt from the real time clock can initiate a data acquisition.

3.3 Wireless protocol

The protocol used by the wireless network has many implications on the performance and power consumption of the overall system. The unique needs of wireless sensor networks must be met in a way that keeps power consumption low enough for long-lasting battery use while maintaining reasonable transmission bandwidth. A summary of possible network protocols is shown in Table 4.

WSSNs typically operate with low power consumption, low duty cycles ($<$ 0.1% to 1%) and in many cases do not require large radio transmission bandwidths (ZigBee Alliance 2009). Low power operation allows wireless nodes to run on batteries longer, reducing cost and maintenance. Wireless sensor networks are typically low duty cycle since many SHM applications only require the collection of five to fifteen minutes of data a day. Take for example a three-axis accelerometer collecting 15,000 16-bit samples. This corresponds to 30 s of data sampled at 500 Hz, which is a high sampling rate for most structural monitoring applications. This data stream of length $3 \times 15{,}000 \times 16 = 720{,}000$ bits would only take 72 s to transmit if the effective data throughput was 10 kbps (ignoring any overhead in the protocol). Note that the data throughput on the WiSeMote is about 16.8 kbps. There is, however, a growing requirement for an increase in transmission bandwidths in sensor network deployments with large networks that require aggregation of many

Table 4 Network protocol comparison

Protocol	DigiMesh/ 802.15.4	ZigBee/ 802.15.4	Bluetooth/ 802.15.1	WiFi/ 802.11b	GSM/GPRS
Focus application	Monitoring & Control	Monitoring & Control	Cable Replacement	High-Speed Internet	Wide Area Voice/Data
Battery life(Days)	100-1000+	100-1000+	1-7	0.5-5	1-7
Range (meters)	1-100+	1-100+	1-10+	1-100	1000+
RF data rate	250 kbps	250 kbps	720 kbps	11 Mbps+	128 kbps+
Nodes per network	264	264	7	32	1
Software stack size	4-32 kB	4-32 kB	250 kB+	1 MB+	16 MB+
Mesh enabled (ability to hop)	Yes	Yes	No	Single hop	TCP/IP Connectivity
Sleeping routers	Yes	No	N/A	N/A	N/A
Homogenous Routing capabilities	Yes	No	N/A	N/A	N/A
Advantages	Robustness, Reliability	Reliability, Low Power & Cost	Convenience, Low Cost	Speed, Flexibility	Existing Infrastructure

data records. With an ongoing tradeoff between power consumption and RF data throughput, an optimum compromise must be chosen. Data compression is one feature which can help improve effective data throughput.

WSSNs must also be easily installable and scalable, i.e., they must accommodate large structures with many sensor locations and various installation topologies. This makes wireless mesh topologies particularly well suited for sensor networks. Wireless mesh networks are networks in which devices communicate with one other by hopping through some subset of intermediary devices that are immediately in range. A mesh topology is seen in Fig. 7. The use of hopping allows mesh-enabled devices to save power by limiting their communication range while still allowing messages to be sent over long distances. Mesh networks also usually include a protocol that adaptively configures routing settings, which allows for arbitrary installation topologies.

ZigBee/802.15.4 is a standardized mesh protocol that is considered the industry standard for

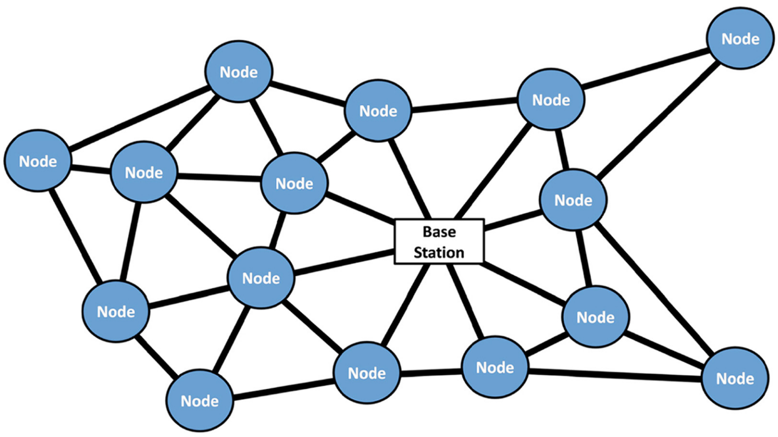

Fig 7 Mesh network block diagram

wireless sensor applications in general. The basis of the ZigBee protocol is that a radio is sleeping most of the time to extend battery life. When a radio is being used it quickly wakes up, transmits the data, and quickly goes back to sleep. This cyclic sleep operation is made possible through periodic synchronization among the radios in the network, which is implemented in the 802.15.4 MAC layer (Ergen 2004). Note, however, that SHM requires unique network capabilities different than other wireless sensor applications.

A ZigBee network consists of a coordinator, routers and end devices. Coordinators are responsible for establishing and maintaining the network, must always be on or awake and are usually grid powered. Routers allow hopping in a ZigBee network. One drawback of ZigBee is that every end device must communicate or hop through a router or coordinator even when another end device is in range, i.e., ZigBee networks are not homogenous. This is an inefficient use of system resources. ZigBee routers are not allowed to sleep (Digi International 2008). Consequently, routers are not intended for battery operation and usually need to be grid powered. This is another major drawback since the idea of a low power WSSN is to have a single grid-powered base station (coordinator) and multiple battery-powered nodes (end devices). End devices are the most common devices in a ZigBee network and are intended to sleep to extend battery life when not in use. ZigBee is preferable if compliance with other ZigBee-enabled devices is needed or if low cost nodes with reduced functionality are a sufficient solution (Digi International 2008).

DigiMesh improves upon the ZigBee protocol in its routing capabilities and device functionality. Any DigiMesh device in this network may communicate with any other device within range without restriction. This is made possible by the fact that all DigiMesh devices have the same functionality. All DigiMesh devices are all capable of establishing networking connections, routing and sleeping, creating a homogenous network. DigiMesh also has self-healing capabilities, meaning that routing connections can be re-established if there is a communication failure somewhere in the current hopping path. This also allows radio time synchronization to be maintained in the event of a failed communication link regardless of the location of the failure. DigiMesh should be used for systems requiring more robust networks at the cost of reduced interoperability with other systems. DigiMesh is a proprietary protocol and thus can only communicate with other DigiMesh enabled devices.

The DigiMesh protocol is better suited than ZigBee for SHM applications in particular due to its increased device functionality. This does come at the cost of high transmission/reception current consumption than other simpler 802.15.4 radios, but this is partially overcome due to the sleeping

Table 5 XBee DigiMesh 2.4 GHz radio specifications

	XBee (on nodes)	XBee-Pro (on base station)
Indoor/Urban range	Up to 100' (30 m)	Up to 300' (100 m)
Outdoor/Line of sight range	Up to 300' (100 m)	Up to 1 mile (1500 m)
Transmit power	1 mW (0 dBm)	100 mW (20 dBm)
Receiver sensitivity	-92 dBm	-100 dBm
TX peak current	45 mA	210 mA
RX current	50 mA	80 mA
Power-down current	< 50 μA	< 50 μA
Networking & Security	Retries &acknowledgements Self-routing, self-healing DigiMesh networking DSSS (Direct Sequence Spread Spectrum) modulation	

router capabilities. The WiSeMote nodes use the XBee DigiMesh 2.4 GHz RF modules and the base station uses the XBee-Pro DigiMesh module. The base station and nodes together form a low power mesh network suitable for arbitrary deployment topologies. Table 5 summarizes the XBee DigiMesh specifications.

3.4 Performance validation

3.4.1 Communication range tests

A series of line-of-sight communication tests were performed between the base station and a single node with the XBee-Pro and XBee radios, respectively. The relationship between packet yield and communication distance is dependent on many factors, including the intricacies of the DigiMesh protocol, the test environment, the network topology, radio mobility patterns, and others (Zhao 2003). This test is intended to characterize the performance of two stationary XBee radios for single-hop radio transmission in an urban, line-of-sight environment. Each radio had a 2.4 GHz half-wave dipole antenna with a gain of 2.1 dBi. The test was performed in an urban outdoor environment with various buildings, cars, etc. in the vicinity but not obstructing the line-of-sight between radios. For each test a group of packets was transmitted from base station to node and then echoed back to the base station. Each echo test contained 500 packets, each with a 70 byte payload. Every packet transmission was retried up to three times on the base station and up to three times on the node in addition to the retries built into the radio modules on the MAC level. These retries are similar to the timeouts/retries implemented in the DEC protocol mentioned earlier. The echo test was performed in communication range increments of 10 meters up to 240 meters. Transmit power levels were the highest settings available [1.42 dBm on XBee (node), 18 dBm on XBee-Pro (base station)]. Note that the performance for an echo test will be dominated by the lowest transmitter

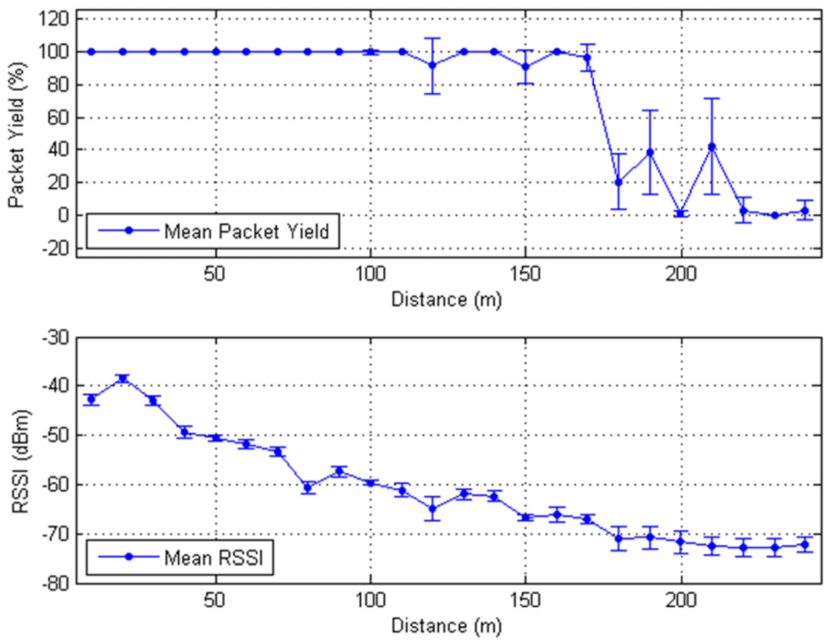

Fig. 8 Communication range test results

power to receiver sensitivity ratio. Fig. 8 summarizes the ratio of total successes to total transmission attempts, or packet yield, as well as the received signal strength indicator (RSSI) for each test. RSSI is a measure of received signal power measured in dBm. The average lines are plotted with error bars one standard deviation above or below. Packet yield was above 90% up to 170 meters, which is a significant improvement over the rated 100 m distance for XBee to XBee outdoor/line of sight communication (Digi 2011). This improvement is due in part to the upper level retries on the base station and node. Another factor is that the 100 m distance rating is for XBee to XBee communication where each receiver has a sensitivity of -92 dBm. For our test, the XBee-Pro has an increased sensitivity of -100 dBm (8 dB improvement). The free space path loss for a 2.4 GHz signal at an 8 dB improvement more than accounts for the 170 m communication range.

3.4.2 Node current consumption

To assess the power consumption of the nodes, current draw was measured before, during, and after a cycle of the accelerometer sampling and data communication operations on a single node. Average current draw divided by battery capacity in Amp-hours represents the theoretical operation time on a single battery. The nodes were commanded to sample 15,000 samples at 250 Hz sampling for 60 s. Current draw from the battery was measured and sampled at 1.706 kHz by a National Instruments CompactDAQ data acquisition system. Fig. 9 shows the measured current across time. Various system states can be easily identified in Fig. 9 and are denoted by the shaded regions labeled one through five. During region one (0 s to about 20 s) the node is idling waiting for a DAC command from the base station. The current pulse in region one corresponds to the duration that the radio wakes up, realizes there is no data transmission pending, and returns to sleep. The radio is set to be awake for 100 ms and asleep for 9.9 s (1% duty cycle with 10 s sleep period). The DAC command is received at the beginning of the network wake-up pulse in region two, when the node starts sampling the data from the ADC for 60 s. The current draw is slightly higher here since the microcontroller is switching between awake mode and LPM0. The node sends a DACC to the base station at the next radio wake period in region three, notifying the base station that the data is ready to be transmitted. At this point the node returns to LPM4 waiting to be queried. At the beginning of region four the node receives the AMC and the radio is reconfigured to exit sleep mode. This allows all the data to be transmitted in a single burst and greatly minimizes the data transmission time when the network wake duty cycles are very low. Soon after receiving the AMC, the DTC is received and data transmission begins. The data transmission burst lasting about 35 s can be clearly identified as the series of spikes in region four.

The average current draw is 631 μA while the node is in an idle state waiting for a command from the base station and 13.07 μA for the duration of the data acquisition (from the radio wake period at the beginning of region two to the end of the radio wake period at the end of region four). Assuming that the nodes are acquiring data 1% of the time (14.4 minutes a day on average), the nodes draw an average current of 755.4 μA. Taking into account the discharge curve of the battery, the useful service lifetime of the nodes is approximated. Estimated service life is compared to that of the Imote2, which used three Industrial D batteries (EN95) for service life estimation (Rice 2009). If three Industrial D batteries (EN95) are used, the estimated service life is given by (Rice 2009)

$$T_{service,est} = \frac{V_{range,device}}{N \cdot V_{range,battery}} 41960\ \text{mAh} \cdot (I_{discharge,avg}\text{mA})^{-1.25} \tag{6}$$

Given a useful node voltage supply range of 3 V to 4.5 V (3 D batteries and useful battery voltage

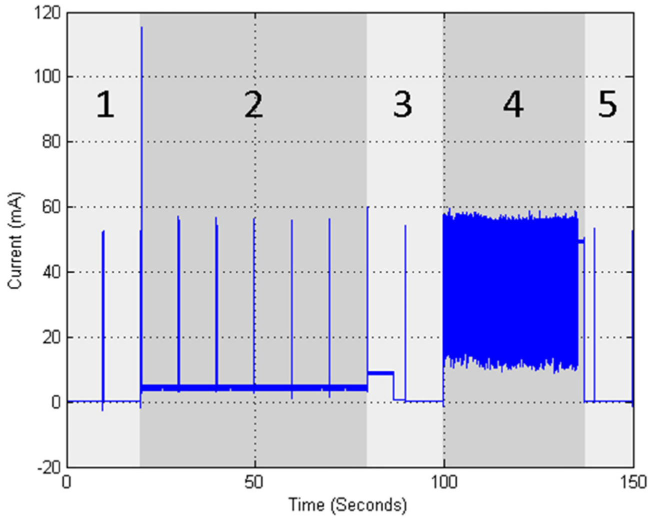

Fig. 9 Current consumption during sample cycle of single node

range of 0.8 V to 1.5 V for an alkaline cell), the nodes can operate for about 58 months.

$$T_{service,est} = \frac{V_{range,device}}{3 \cdot (1.5-0.8)V} 41960 \text{ mAh} \cdot (0.7554 \text{ mA})^{-1.25} \tag{7}$$

$$T_{service,est} = 42,558 \text{ hours} \approx 58 \text{ month} \tag{8}$$

Note this does not include power consumption due to post processing damage detection algorithms. It can be shown that the Lithium-Ion battery used only for development will yield a service life of about 100 days using the same calculations. Network deployments using this battery might be possible if energy harvesting components were used for battery recharging.

3.4.3 Time synchronization

Time synchronization of data acquisition is an important need for WSSNs and has been widely researched and implemented (Hill 2002, Mechitov *et al*. 2004, Maroti *et al*. 2004, Sundararaman 2005, Nagayama and Spencer 2007). Here we discuss the sources of synchronization error, the implications of synchronization on structural algorithms, and techniques for achieving synchronization in other WSSNs as well as the in the WiSeMote.

Because each wireless node independently samples data, there exists differences in the times when the data acquisitions are initiated as well as the rate at which the different ADCs are sampled. Differences in data acquisition initiations are due to random radio transmission delays and random delays before the execution of interrupt service routines on the node microcontrollers. In mesh networks, random transmission delays exist due to MAC synchronization errors and other variables resulting from non-uniform network topologies, such as varying number of hops for different nodes. These transmission delays cause a command that is broadcast by a base station to be received on the serial interface of each node microcontroller at different times. The delay before the execution of interrupt service routines is usually much less than network transmission delays. For the WiSeMote node microcontrollers, typical wake up from sleep is less than 6 µs. Differences in ADC sampling rates are due sampling clock jitter. Sampling clock jitter occurs due to ADC aperture jitter as well as jitter from the ADC clock source (Neu 2010). ADC aperture jitter is sampling jitter due to variations during the ADC sampling conversion process. The ADC clock source also introduces sampling jitter due to thermal noise. The WiSeMote node ADC clock reference is a frequency-locked loop (FLL) that is sourced with a 32.768 kHz crystal. The crystal has a frequency tolerance of ±20 ppm. For FFTs of typical length (less than 4096) in the typical frequency range of civil structures (less than 500 Hz), the spacing between discrete FFT frequencies is often much higher than the sampling rate error.

The exact impact of data acquisition synchronization is application-dependent, however we will discuss it here in the context of commonly measured modal parameters, i.e., natural frequencies, damping ratios, and mode shapes. Nagayama (2007) explored the effects of time synchronization on various methods for estimating modal parameters from multi-degree-of-freedom structures, including the Natural Excitation Technique (James *et al*. 1992), Eigensystem Realization Algorithm (Juang and Pappa 1985), and estimation from transfer functions. The modal parameters can be estimated in terms of the transfer function between system inputs and outputs. For a multi input multi output system, the transfer function between the m-th input time shifted by t_m^{in} and an n-th output time shifted by t_n^{out} is given as

$$\overline{H}_{nm}(\omega) = \frac{\overline{Y}_n(\omega)}{\overline{F}_m(\omega)} = \frac{e^{-j\omega t_n^{out}}Y_n(\omega)}{e^{-j\omega t_m^{in}}F_m(\omega)} = e^{j\omega(t_m^{in}-t_n^{out})}H_{nm}(\omega) \tag{6}$$

Where $F_m(\omega)$ and $Y_n(\omega)$ are the Fourier transforms of the input and output, respectively. The magnitude of the time shifted transfer function $\overline{H}_{nm}(\omega)$ is the same as the original transfer function $H_{nm}(\omega)$ but the phase has been shifted. The natural frequency, damping ratio, and mode shape magnitude remain the same for the time shifted (unsynchronized) data, but the mode shape phase has changed with the introduced time synchronization error $t_{err} = t_m^{in} - t_n^{out}$. The phase shift of the k-th mode is estimated by

$$\phi_k = 2\pi f_k t_{err} \tag{7}$$

From Eq. (7), a time synchronization error of 1 sms corresponds to 3.6 degrees of phase shift of a mode shape at 10 Hz and 36 degrees of phase shift of a mode shape at 100 Hz. For our system, we must ensure that phase shift remain insignificant at common structural frequencies (usually below 100 Hz).

Most researchers have attempted to solve synchronization by making measurements on the departure and/or arrival times of synch messages in order to estimate relative differences between local clocks. One example of this is the Reference Broadcast Synchronization protocol (Sundararaman 2005). Synchronization on a network of Imote2s was achieved by estimating global timestamps on collected data and resampling data after the fact (Nagayama and Spencer 2007). For WSSNs, one source of relative time information for the entire network is the sleep synchronization for the radio protocol, usually the 802.15.4 MAC protocol. Depending on the particular radio being used, radio sleep synchronization may not be accurate enough for the application or in some cases may not be available outside the radio chip at all. The WiSeMote system has the advantage of including a radio that provides information about network synchronization via an external pin. This synchronization is dependent, however, on certain radio settings. Because information about XBee synchronization accuracy is not available, experiments were conducted to determine accuracy. Table 6 summarizes the synchronization accuracy over various XBee settings. For this test, synchronization error was measured for 20 occurrences on 4 different nodes. Note that the 1 s sleep cycle in Table 6 is the setting that is actually used during data acquisitions. From these experiments it is clear that the level of synchronization accuracy among XBees is dependent upon the sleep cycle settings, i.e., the amount of time each cycle that the nodes are awake and sleeping. Lower sleep durations improve synchronization accuracy since the synchronization information is updated more often within the XBee. Low sleep durations, however, are undesirable for steady state operation in our case. This is due the fact that low sleep durations result on higher wake duty cycles, which ultimately increases power consumption. For this reason, the WiSeMote uses lower power settings for the idle state the majority of the time and higher power settings for a short time during the synchronization state before each data acquisition. While idle, the radio sleep settings are awake 9.9 s, asleep 100 ms. Before sampling, the base station changes the network-wide sleep settings to be awake for 890 ms, asleep 110 ms. The synchronization state's sleep cycle is as short as possible to allow for rapid synchronization. The lower limit to total sleep cycle duration is dependent upon numerous XBee settings such as number of hops, number of MAC retransmissions, and others. When initiating a data acquisition event, the base station broadcasts a command and changes the sleep settings. Each node, upon receiving the command, turns on its analog front-end and initializes its ADC. Each node waits three sleep cycles to ensure synchronization has occurred for the new XBee sleep settings and then takes the first ADC sample within an interrupt service routine triggered by the third XBee wake cycle. Waiting for three cycles (3 s) also allows more than ample time for the analog front end voltage references to settle after turning them on. Fig. 10 shows a histogram of the synchronization error using this process on 2 nodes for 20 tests. For these tests the absolute value of the synchronization errors was between 3 μs and 53 μs with an average of 26.2 μs. Note that a time delay of 53 μs corresponds to only 0.033 degrees of phase shift of a mode shape at 100 Hz.

To determine synchronization accuracy, not only discrepancies in the initiation of a data acquisition must be validated, but also the discrepancies in sampling rates and sampling rate linearity. Table 7 shows the mean errors in sampling rates on a single node for several different sampling rate settings. These measurements were determined by comparing an MSP430 pin sourced by the ADC clock source with a Tektronix AFG3022B function generator set to the same sampling frequency. Sampling linearity was also validated by measuring the times at which samples are taken for two nodes. The nodes sampled 60 samples at 32 Hz. The previously mentioned CompactDAQ sampled the voltage across a pin that went high once a sample was taken. The CompactDAQ sample rate was 51200 Hz, resulting in a sample time tolerance of ±19.5 μs. The sample voltage-time sequence, however, was upsampled by a factor of 20 to achieve more precise measurements. The error between measured sample times and a least-squares-fit line for the entire data series is shown in Fig. 11. The non-linearity is within ±2.5 μs. The WiSeMote data acquisition method has

Table 6 Synchronization error for various XBee sleep settings

XBee sleep period (s)	Minimum error (μs)	Maximum error (μs)	Mean error (μs)
1	3	53	26.2
2	15	90	40.4
5	60	150	98.0

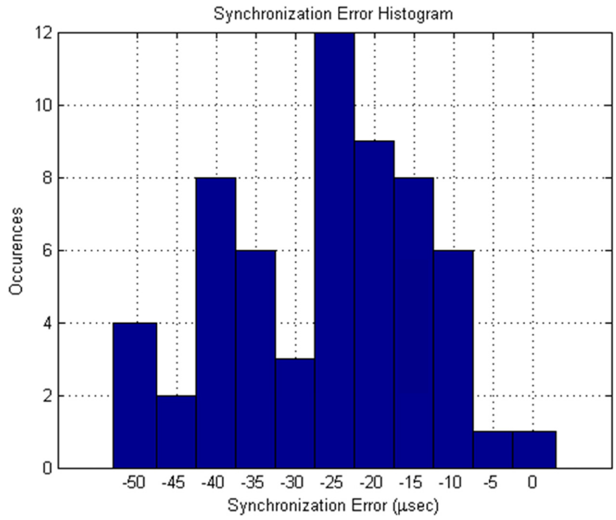

Fig. 10 Histogram of synchronization error between two nodes

Table 7 Node sampling rate error

Nominal sampling rate (Hz)	Measured sampling rate (Hz)	Error
40	40.0866	0.216%
100	100.2128	0.212%
250	250.5142	0.205%
400	400.8132	0.203%
500	500.969	0.193%

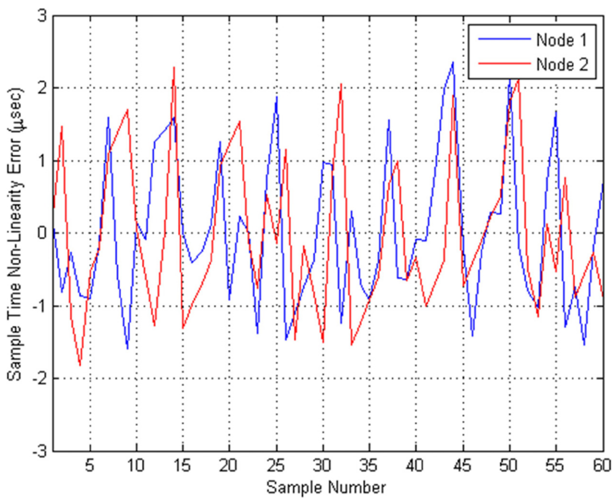

Fig. 11 Non-linearity error for two nodes sampling at 32 Hz

the advantage of achieving synchronization before the data acquisition starts, removing the need for further data processing after data has been sampled.

3.4.4 Low noise operation

The noise level of the nodes determines the smallest level of acceleration that can be sensed while maintaining an adequate signal-to-noise ratio. The majority of noise in the collection of acceleration data is attributed to thermal noise within the sensor, ADC quantization error, and ripples or spikes in the power supply voltage of the sensor and voltage reference of the ADC. The noise density of the LIS344ALH accelerometer is 50 $\mu g/\sqrt{Hz}$, although this is an extreme worst-case scenario for all three axes. A closer look at the histogram of the noise densities given in the accelerometer data sheet for all three axes shows that the mean value is closer to 25 $\mu g/\sqrt{Hz}$ for x/y and 40 $\mu g/\sqrt{Hz}$ for the z-axis. The ADC quantization error is ±33.5 μg/LSB. To measure the overall noise level of the node, data was sampled from two nodes at 40 Hz for about 40 s. Data was collected while the nodes were sitting on foam in a still room without any lights on or any electronics powered on. The RMS noise level for both nodes is shown in Table 8. The z-axis PSD for this test is also shown in Fig. 12. Note the difference in sensitivities between that of the x or y axes and the z axis. This is due to the difference in noise densities between the axes of the accelerometer chip. The WiSeMote RMS noise is appreciably lower than that of the Imote2 which has 0.3 mg RMS for the x/y axes and 0.7 mg RMS for the z-axis (typical).

3.4.5 Data quality

Acceleration data was collected from a magnetic shaker to validate the consistency of data acquisition on the nodes. In order to validate that the nodes are accurately capturing data, it must be

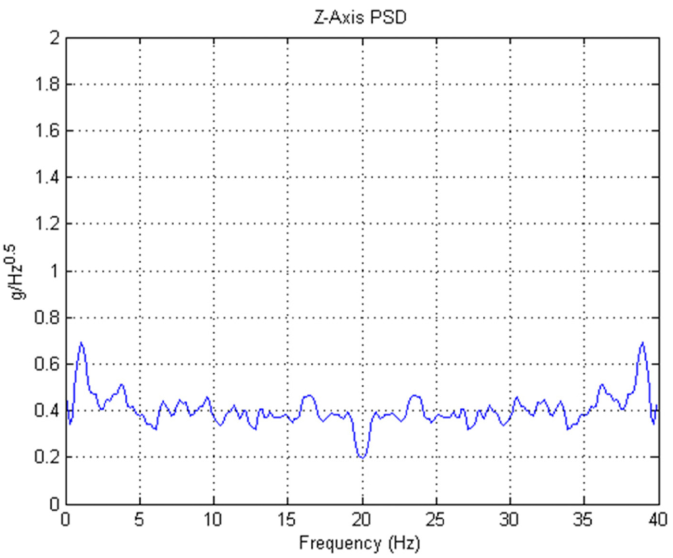

Fig. 12 Node PSD for z-axis noise test

Table 8 Node RMS noise measurements

	Node 1 RMS Noise (mg)	Node 2 RMS Noise (mg)
x-axis	0.137	0.124
y-axis	0.132	0.126
z-axis	0.188	0.176

confirmed that data is consistent between subsequent nodes as well as between the nodes and a known accurate reference. First the consistency between two subsequent nodes was validated. For this test the shaker was excited with a 125-Hz low pass filtered white noise signal and the nodes were sampling at 250 Hz. Two nodes were secured to the shaker where the shaker motion occurred along the vertical axis of the accelerometer. The auto power spectral densities of the data acquired from each node are computed as the square of the periodogram and are shown in Fig. 13. The low pass filtered white noise signal used to excite the shaker motion is further filtered due to the physical constraints of the shaker itself, explaining why the auto power spectral densities do not look like the data consisted of white noise. Due to these physical constraints the amplitude of shaker acceleration decreases rapidly as the frequency of the excitation signal approaches 0 Hz or more than about 15 or 20 Hz. In order to validate that the nodes are collecting the same data we estimate the transfer function $H(e^{j\omega})$ between the signals from node 1 ($y_1[u]$) and node 2 ($y_2[u]$). It holds that $H(e^{j\omega})$ can be determined from the relationship

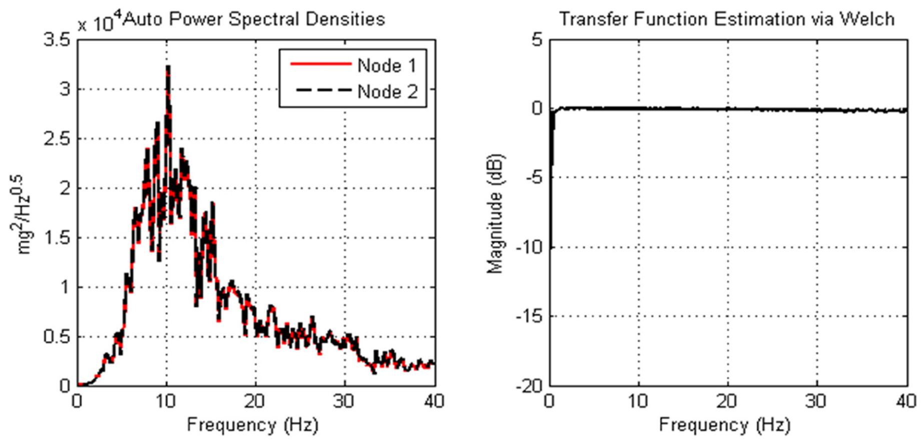

Fig. 13 Auto power spectral densities and transfer function of simultaneously sampled acceleration data from two different nodes

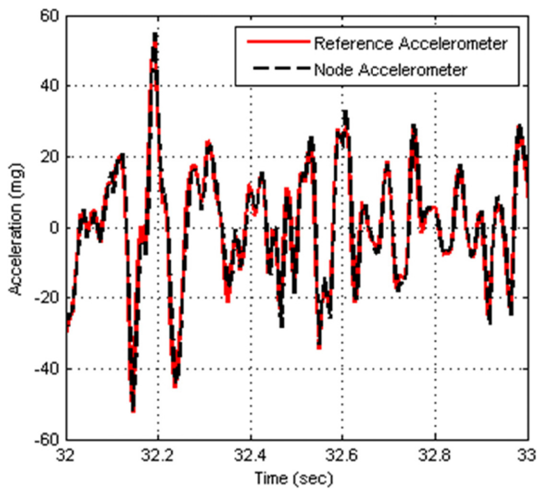

Fig. 14 Sampled acceleration-time data from node and reference accelerometer

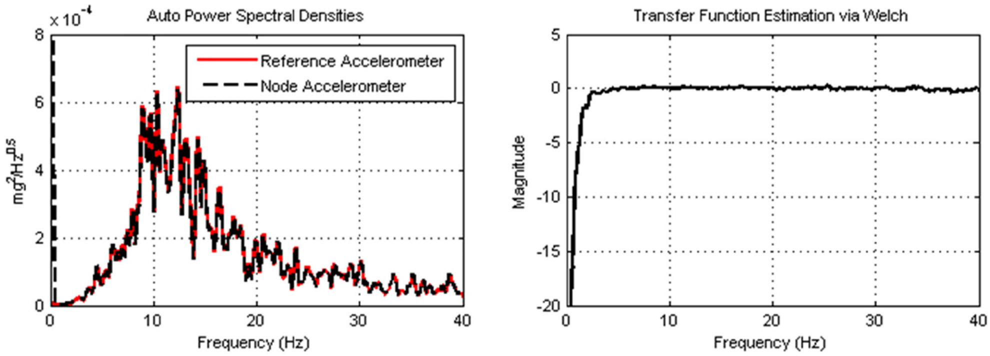

Fig. 15 Auto power spectral densities and transfer function of simultaneously sampled acceleration data from node and reference accelerometer

$$S_{y_2y_1}(e^{j\omega}) = H(e^{j\omega}) * S_{y_1y_1}(e^{j\omega}) \tag{8}$$

where $S_{y_ay_b}$ is the cross spectral density between time domain signals $y_a[u]$ and $y_b[u]$. For our computation $S_{y_ay_b}$ is computed as the element-wise multiplication of the periodograms of $y_a[u]$ and $y_b[u]$. The transfer function estimate $H(e^{j\omega})$ is shown in Fig. 13. The value $|H(e^{j\omega})|$ is less than 1 dB across all frequencies, confirming that the sensitivity of each node's accelerometer is approximately the same.

Next, data quality was compared between a node and a high-quality reference accelerometer (PCB333B50). A similar test setup was used, with the shaker excited with a 40-Hz low pass filtered white noise signal. The auto power spectral densities and transfer function estimate of the time domain data is shown in Fig. 15. The nodes accurately capture the time and frequency content of an acceleration data record, and their results have shown to be reasonably close to a high-end accelerometer.

3.4.6 Multi-node data acquisition validation

Finally, network-wide data acquisition was validated by simultaneously sampling acceleration data from four nodes installed on a model truss bridge model. As in the previous experiment, the data was compared against data collected from wired reference PCB333B50 accelerometers. The nodes were attached to joints 3, 7, 13 and 21 on the bridge model shown in Fig. 16 using magnetic

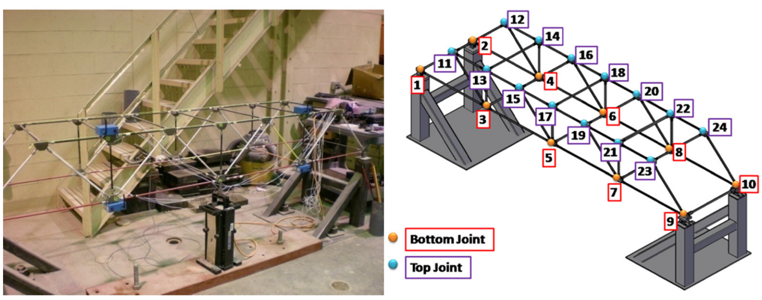

Fig. 16 Truss bridge network validation test setup with four nodes (left) and rendering of truss bridge with joint IDs used for network validation (right)

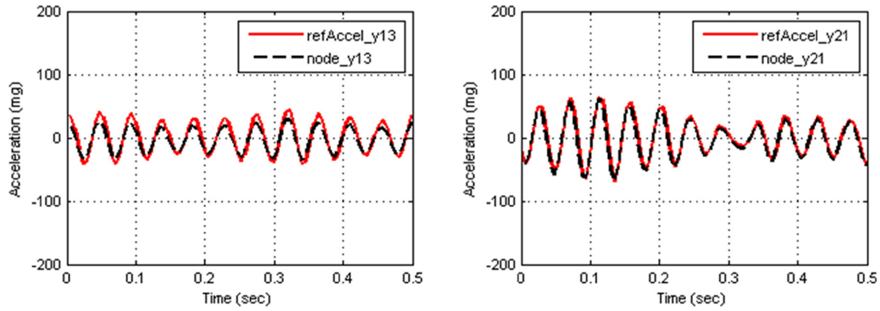

Fig. 17 Acceleration-time data collected from the transverse motion of the model bridge

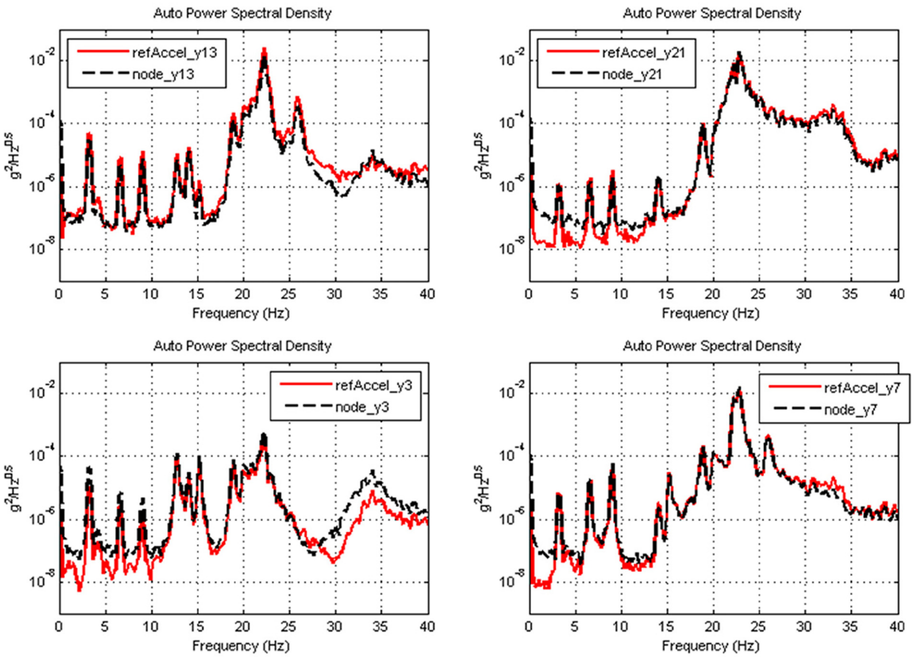

Fig. 18 Auto power spectral densities of acceleration data records collected from the model bridge

mounts. The magnetic shaker discussed previously was used to excite the bridge vertically at joint 5 with a 40-Hz low pass filtered white noise signal. Because most structures contain the majority of their vibrational energy below 40 Hz, exciting the bridge with a 40-Hz low pass filtered white noise signal represents data acquisition under typical real-world operating conditions. The nodes were commanded to sample 5,000 samples at 165.16 Hz resulting in a total of about 30 s of data. Acceleration from the PCB333B50 accelerometers was sampled using the CompactDAQ at 1651.66 Hz, then resampled by a factor of 1/10 and manually aligned in time with the node data. Manual alignment is needed since there is no time synchronization between the CompactDAQ and the nodes. Fig. 18 shows the auto power spectral densities for the acceleration records collected from the node and reference accelerometers for each of the four joint IDs. The auto PSDs were calculated with 39 windows each of length 220 with a 50% overlap, 4096-point FFTs, and using Hamming windows. The auto PSDs line up well over the excitation range from 0 to 40 Hz.

4. Future plans

The current WiSeMote platform has gone through significant development to enable tightly synchronized data acquisition while maintaining low power consumption suitable for extended deployment. The WiSeMote is capable of aggregating data through mesh communication from multiple nodes installed in an arbitrary fashion on a structure. Future plans include modifying existing algorithms to improve performance. In particular, there is room for improvement with the data compression algorithm. Compression performance may be improved by using auto-regressive models to obtain optimal predictor weights, as opposed to the current weights which cause the predicted value to be a linear extension of the previous two samples. The predictor order may also be increased by using more than two weights. Multi-stage compression may also improve performance by compressing the error signals multiple times. Another important feature that will be added is wireless remote reprogrammability of the base station and nodes. This will allow for a mobile agent framework in which new network functionality and SHM algorithms can be tested with the network in situ. Functionality for a GPRS will be added for communication capabilities with hosts located away from the installation site. A web server will also be added for similar remote host communication through Ethernet. These modules are important for remote access to structural data as well as network information such as battery voltages and data acquisition settings.

The effects of temperature on the MEMS accelerometer chip have been overcome in other systems such as REF, but have not been explored here. Future work will include temperature compensation for the accelerometer. This is made possible by a temperature sensor integrated in the MSP430. It is expected that the temperature of the adjacent MSP430 and accelerometer chips will be closely related.

With the majority of the basic operational capabilities developed, the foundation is laid to implement sophisticated damage detection algorithms for various SHM applications. Future work will include implementing the Damage Location Vector (DLV) method (Nagayama 2007) on our WiSeMote network. The DLV method along with a DLV distributed computation strategy has been well-developed in the context of sensor networks and has been implemented and tested on a network of Imote2s. We will further develop and optimize the DLV distributed computation strategy on the WiSeMote system in an effort to improve utilization of resources, reduce total computation time, and compute the algorithms for damage detection in a more parallel manner.

5. Conclusions

This paper has presented theWiSeMote: a newly developed, ultra-low power wireless smart sensor node and base station. The WiSeMote system is capable of low power operation (755.4 μA average at 1% radio duty cycle, 1% sampling rate duty cycle), collecting tightly synchronized data (≤ 53 μs), and aggregating the data to the base station that saves it to an SD card. A hardware and software description of the various operating conditions of the wireless nodes and the base station has been provided. We verified useful wireless communication range via an outdoor line-of-sight test. Power consumption was verified by sampling current draw from a node battery before, during, and after a data acquisition and communication event. Time synchronization of data acquisition was verified by observing the time instance of the first ADC sample taken from two different nodes. We confirmed the consistency of data quality from sampled acceleration data among subsequent nodes as well as between nodes and high quality reference accelerometers for acceleration data generated by a magnetic shaker as well as a lab-scale truss bridge model. Going forward, this system will provide a platform on which to develop sophisticated damage detection and other structural health monitoring algorithms as well as advanced network functionality and flexibility.

Acknowledgments

The authors gratefully acknowledge the support of this research by the U.S. Department of Energy under grant DE-FG36-06GO86092 (Scale Testing, Characterization, System Optimization and Demonstration of Grid Connected Wind Turbines and Wind Powered Water Desalination Project). Also, special thanks to Dr. Tanja Karp from Texas Tech Department of Electrical and Computer Engineering for advice on signal processing techniques.

References

Digi International, Inc. (2008), "Wireless mesh networking Zigbee vs. Digimesh", http://www.digi.com/pdf/wp_zigbeevsdigimesh.pdf.

Digi International, Inc. (2011), "XBee/XBee-PRO 2.4 DigiMeshRF Modules", ftp://ftp1.digi.com/support/documentation/90000991_D.pdf.

Ergen, S.C. (2004), "ZigBee/IEEE 802.15.4 Summary", http://staff.ustc.edu.cn/~ustcsse/papers/SR10.ZigBee.pdf.

Hill, J.L. and Culler, D.E. (2002), "Mica: a wireless platform for deeply embedded networks", *IEEE Micro*, **22**(6), 12-24.

James, G.H., Carne, T.G., Lauffer, J.P. and Nord, A.R. (1992), "Modal testing using natural excitation", *Proceedings of the 10th Int. Modal Analysis Conference*, San Diego, CA.

Jang, S., Jo, H., Cho, S., Mechitov, K., Rice, J., Sim, S., Jung, H., Yun, C., Spencer, B., and Agha, G. (2010), "Structural health monitoring of a cable-stayed bridge using smart sensor technology: deployment and evaluation", *Smart Struct. Syst.*, **6**(5-6), 439-459.

Juang, J.N. and Pappa, R.S. (1985), "An Eigensystem realization algorithm for modal parameter identification and model reduction", *J. Guid. Control Dynam.*, **8**(5), 620-627.

Kurata, M., Lynch, J., Galchev, T., Flynn, M., Hipley, V.J., van der Linden, G., Mortazawi, A., Najafi, K., Peterson, R.L., Sheng, L., Sylvester, D. and Thometz, E. (2010), "A two-tiered self-powered wireless monitoring system architecture for bridge health management", *Proceedings of the SPIE Smart Structures/NDE*, San Diego, CA.

Lee, C.S. (2006), *Bluetooth Security Protocol Analysis and Improvements*, Master°Øs thesis, San Jose State University.

Libelium, (2010), "Waspmote technical guide", http://www.libelium.com/documentation/waspmote/waspmotetechnical_guide_eng.pdf.

Lynch, J.P., Wang, Y., Law, K.H., Yi, J., Lee, C.G. and Yun, C.B. (2005), "Validation of a large-scale wireless structural monitoring system on the GeumdangBridge", *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Rome, Italy.

Maroti, M., Kusy, B., Simon, G. and Ledeczi, A. (2004), "The flooding time synchronization protocol", *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD.

Mechitov, K., Kim, W., Agha, G. and Nagayama, T. (2004), "High-frequency distributed sensing for structure monitoring", *Proceedings of the 1st Intl. Workshop on Networked Sensing Systems*, Tokyo, Japan.

MEMSIC Inc. (2007), "Imote2 hardware reference manual", http://www.memsic.com/support/documentation/wireless-sensor-networks/category/6-user manuals.html?download=56\%3Aimote2-hardware-reference-manual.

Nagayama, T., Sim, S.H., Miyamori, Y. and Spencer Jr., B.F. (2007), "Issues in structural health monitoring employing smart sensors", *Smart Struct. Syst*., **3**(3), 299-320.

Nagayama, T. and Spencer Jr., B.F. (2007), *Structural health monitoring using smart sensors*, NSEL Report Series, Report No.NSEL-001, 849-864.

Nagayama, T., Spencer Jr., B.F. and Rice, J.A. (2009), "Autonomous decentralized structural health monitoring using smart sensors", *Struct. Health Monit*., **16**(7-8), 842-859.

Polastre, J., Szewczyk, R. and Culler, D. (2005), "Telos: Enabling ultra-low power wireless research", *Proceedings of the* 4th *International Symposium on Information Processing in Sensor Networks (IPSN)*.

Rice, J.A. and Spencer, B.F. (2009), *Flexible smart sensor framework for autonomous full-scale structural health monitoring*, NSEL Report Series, No. 18, University of Illinois at Urbana-Champaign (http://hdl.handle.net/2142/13635)

Rice, J.A, Mechitov, K., Sim, S., Nagayama, T., Jang, S., Kim, R., Spencer Jr., B.F., Agha, G. and Fujino,Y. (2010), "Flexible smart sensor framework for autonomous structural health monitoring", *Smart Struct. Syst*., **6**(5-6), 423-438.

Rice, J.A., Mechitov, K.A. and Spencer Jr., B.F. (2010), "Autonomous smart sensor network for full-scale structural health monitoring", *Proceedings of the SPIE Smart Structures/NDE*, San Diego, CA.

Sim, S.H., Spencer, Jr., B.F., Zhang, M. and Xie, H. (2010), "Automated decentralized modal analysis using smart sensors", *Struct. Health Monit*., **17**(8) 872-894.

Sundararaman, B., Buy, U. and Kshemkalyani, A.D. (2005), "Clock synchronization for wireless sensor networks: a Survey", *Ad Hoc Networks*, **3**(3), 281-323.

Swartz, R.A., Jung, D., Lynch, J.P., Wang, Y., Shi, D. and Flynn, M.P. (2005), "Design of a wireless sensor for scalable distributed in-network computation in a structural health monitoring system", *Proceedings of the* 5th *International Workshop on Structural Health Monitoring*.

Neu, T.(2010), *Clock jitter analyzed in the time domain, Part 1*, Texas Instruments Inc.

Whelan, M. and Janoyan, K. (2009), "Design of a robust, high-rate wireless sensor network for static and dynamic structural monitoring", *J. Intel. Mat, Syst. Struct*., **20**(7), 849-864.

Zhao, J. and Govindan, R. (2003), "Understanding packet delivery performance in dense wireless sensor networks", *Proceedings of the SenSys*, Los Angeles, California, USA.

ZigBee Alliance (2009), "Zigbee overview", http://www.ie.uia.mx/tit/pr09/ib03avg/public_html/TESIS/Zigbee_Alliance_ Documents_files/ZigBee_Alliance_Overview.pdf.

Zimmerman, A.T., Shiraishi, M., Swartz, R.A. and Lynch, J.P. (2008), "Automated modal parameter estimation by parallel processing within wireless monitoring systems", *J. Infrastruct. Syst*., **14**(1), 102-113.

BS