EJSE Special Issue: Wireless Sensor Networks and Practical Applications (2010)

The Development of an On-line Structural Health Monitoring System based on Wireless Sensor Networks

C. Oliver Yang*#
Max Yen^
Ning Weng#
**Materials Technology Center, #Department of Electrical and Computer Engineering*
Southern Illinois University at Carbondale, Carbondale, IL, USA
^College of Engineering, Technology, and Computer Science
Indiana University-Purdue University Fort Wayne, IN, USA

Email: {ccyang, nweng}@siu.edu; yens@ipfw.edu

ABSTRACT: Monitoring structural conditions of the large number of aging bridges in the national transportation infrastructure has become an important issue that directly impacts public safety. Traditional manual inspection not only requires extremely high cost but also cannot address this problem in a timely manner. The wireless sensor networks have been accepted as a promising solution to the above problems. In this paper we have presented an on-line structural health monitoring system that consists of wireless sensor networks for real-time sensing and data sever for on-line analyzing. We designed each component, integrated them to the whole system, and deployed it to one of our campus bridges. Our laboratory simulation and prototype experimental results have verified the functionality and correctness of our system.

KEYWORDS: SHM, wireless sensor network, deployment

1 INTRODUCTION

Maintaining healthy civil infrastructures, such as bridges, is important to a nation's economic growth as well as public safety. According to the 2007 report of the Federal Highway Administration's National Bridge Inventory, over 72,500 bridges are structurally deficient [1]. Traditional visual inspections used to monitor almost all of the nation's bridges can lead to undetected damage or false damage detection which results in unnecessary retrofits. Due to the deteriorated state of bridges, engineers have worked to find more effective methods to monitor the health of structures. For this reason, various SHM systems that incorporate sensors and data acquisition techniques have been developed.

In the U.S., the cost of installing a monitoring system for buildings or bridges can exceed two thousand dollars per sensing channel, excluding the cost of the monitoring system and the sensor [2]. Due to the high cost and duration of installation for traditional wired SHM systems and rapidly evolving wireless communication technologies, researchers at various institutions mainly focus on the development of wireless SHM systems. For example, researchers at the University of Southern California designed a wireless system, called Wisden, to deliver structure vibration data [3]. Spencer et al. [4] proposed a decentralized structural health monitoring platform and implemented a damage detection algorithm using smart sensors. Chen and Wang [5] presented a high computation power wireless sensor node with both passive and active sensing capabilities. Kim et al. [6] implemented their system on the Golden Gate Bridge and successfully collected reliable and calibrated ambient structural vibration data for analysis. Most aforementioned researches pay attention to the short-term dynamic structural response to determine the damage to a structure by collecting structural vibration response. The focus of this research, on the other hand, is the study of the long-term continuous static behavior of a structure. By doing the long-term monitoring, the history of structural deterioration and degradation can be well defined. When checking the structure continuously, damage is able to be identified at the time it occurs. In order to perform the task, a system that consists of wireless sensor nodes and a data server including web interface are required.

In this paper, we present a complete wireless structural health monitoring system that includes both hardware and software components. This monitoring system is to continuously collect structural response data. It aims to provide users with up-to-date data at a location that is a considerable distance from the structure being monitored, even on the other side of the earth. The system consists of sensors, wireless sensor nodes, router(s), and a data repository server, as depicted in Figure 1. Every sensor node is a centralized data acquisition subsystem with sensors directly connected to each node. The collected sensor data is processed and sent to a nearby wireless router connected to the Internet. The data is immediately transmitted via the Internet to the data acquisition and repository system. Here, the data is further processed and stored in a database and made available to users through the Internet. Users can connect to the monitoring server to see the near real-time data using an Internet browser without the need for any special software. Using an Internet browser, the real-time data can be viewed in numerical and graphical formats and stored data can be retrieved. Because the data is broadcast to the Internet directly from the sensor nodes, there is no need to maintain a base station at the site of the monitored structure. While many wireless SHM systems exhibit sophisticated architecture and exceptional scalability of a decentralized topology, our system adopts a centralized approach for its simplicity, security, and ease of management. Rodríguez Peralta et al. (2009) [7] also implemented similar approach as all collected data was centralized in a PC. The whole system has been deployed and validated on a bridge located at a rural area of the university campus.

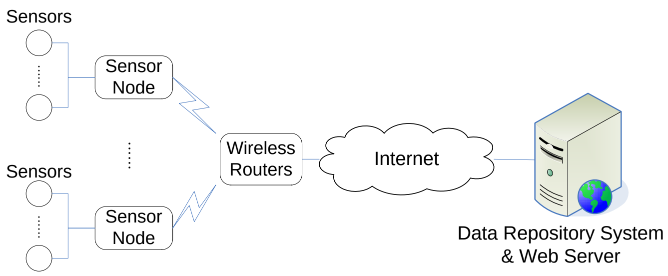

Figure 1: On-line SHM system architecture

The remainder of this paper is structured as follows. Section II describes the related work in the area of WSNs applied to the structural health monitoring. Section III details the design and development procedure of the wireless sensor node. Section IV presents the development of the data acquisition and repository system and the user interface. Section V verifies the system and demonstrates the result of implementing the system on a campus bridge. The accuracy of the strain measurement is examined. The instrumentation and test procedure on the bridge are described in detail. Section VI presents the conclusion of this research. Recommendations for future improvement on the system including hardware and software are given in Section VII.

2 RELATED WORK

In the area of structural health monitoring, both academic prototypes and commercial products have been proposed. In 1998, Straser and Kiremidjian proposed the design of a low-cost wireless modular monitoring system (WiMMS) for civil structures [8]. The design aims were to provide information for extreme-event (e.g. seismic) monitoring and identifying initial damage in structures experiencing long-term deterioration. The 8-bit Motorola 68HC11 microcontroller along with external 32 kB of RAM and 16 kB of ROM were adopted as the computational core. An eight-channel 16-bit analog-to-digital-converter was provided to connect to multiple sensors at a time. A Proxim Proxlink MSU2 wireless modem that operates in the industrial, scientific, and medical (ISM) band between 902 and 928 MHz is used as the wireless transceiver. This prototype represents the first major step by the structural engineering community towards decentralized data processing and wireless structural health monitoring [9]. By 2007, Wang and Law had developed a wireless structural sensing system [10]. This integrated hardware and software system was designed and implemented using low-coast, off-the-shelf components. Various computational algorithms are also embedded into the wireless node. One of the key functions of this system is the support of feedback control on the structure. This extends the capabilities of the system beyond collecting sensor data to include commanding structural actuators. The system uses an 8-bit Atmel ATmega 128 microcontroller with external 128 kB SRAM to support the execution of embedded data interrogation algorithms. Two different wireless transceivers are used on the system to provide the flexibility of use in different countries, communication ranges, data transfer rates, and power consumptions. One is the MaxStream 9XCite operating on the 900 MHz band, and the other is the MaxStream 24XStream operating at 2.4 GHz. Both Straser's and Wang's systems are examples of academic prototypes. The researchers acquired the commercial off-the-shelf (COTS) parts and made the sensor nodes from scratch. Similar approaches include, but are not limited to, Lynch (2002) [11], Mitchell et al. (2002) [12], and Sazonov et al. (2004) [13].

Rather than build their own prototypes, many researchers chose commercial wireless sensor platforms as their sensor nodes. The first wireless sensor platform unveiled was the Rene Mote from Crossbow. It drew the attention of the public because of its open source hardware and software (TinyOS) [9]. Its predecessor was the Berkeley Mote under development during the late 1990s and commercialized as the Rene Mote by Crossbow. Crossbow subsequently developed a series of wireless sensor modules including MICA, MICA2/MICAz, IRIS, and Imote2. Other companies that produce wireless sensor platforms that have been used for structural health monitoring also include Microstrain and Gumstix. In 2004, researchers at the University of Southern California and the University of California at Los Angles designed a wireless system to continuously connect structural response data from a multi-hop network of sensor nodes, to display, and to store the data at a base station [14]. The platform they adopted is MICA2. Kijewski-Correa et al. [15] also used MICA2 to achieve a multi-scale network. Their system combines both strain and acceleration data to enhance damage detection capabilities. Spencer et al. [4] adopted Imote2 as their smart sensor platform. Smart sensors are wireless sensors with a relatively low-cost that have onboard computational ability and can store sensed data, numerical algorithms, and process instructions in onboard memory [16]. The key to their system is that the roles of their nodes are interchangeable to increase network flexibility. The neighboring nodes can take over the roles of non-functioning nodes to prevent system failure. Chen and Wang [5] presented a high computation power wireless sensor node with both passive and active sensing capabilities using connex 400xm-bt from Gumstix for control, computation, and communication. The use of acceleration and strain response from passive sensing can achieve a more accurate damage detection result [17]. This could improve a structural health monitoring system's ability to quantify structural damage [18]. The combination of both passive and active sensing preserves the advantages of both methods.

Most mentioned researches focus on the development of damage detection algorithms to find out the damaged structure. This research, on the other hand, emphasizes on the development of a real-time wireless system that can continuously collect structural response and is accessible to users via the Internet. The damage detection algorithms can later be applied to the sensor node or the data server.

The goals of this research are to wirelessly deliver the structural data to the users and to have a database and management server to store and manipulate the data. By sending the data to the Internet directly from the sensor node, it eliminates the need to maintain a base station at the monitoring site. A database and management server allows real-time and post data analysis. The current system also includes a webpage for users to access to the monitoring data.

3 WIRELESS SENSOR NODE

For the majority of the wireless structural monitoring applications, the fundamental unit of the system is the sensor node, also called the mote in many systems. In general, the wireless sensor node includes the sensing interface, computational core, and wireless transceiver. For some applications that require active sensing, the node also incorporates actuation interface. The sensing interface is usually responsible for connecting to the sensors, performing signal conditioning and amplification (if needed), and converting the analog signal of the sensors into digital representation. The computational core is in charge of processing the digital data from the sensing interface, storing the data, and communicating with other sensor nodes or the base station (server) through the wireless transceiver.

3.1 *Design of wireless sensor node*

The sensor node developed in this research is an Internet-based wireless sensor platform that is simple yet flexible enough for the various demands of structural health monitoring applications. It is designed to run on a constant power source such as DC power line or solar power instead of batteries alone. The hardware architecture of the sensor node is depicted in Figure 2. For the analog sensors, the signal goes through necessary conditioning processes and amplification (if needed) before being converted into digital format by the analog-to-digital-converter (ADC). The digital representation of the analog signal and the data from the digital counterpart are processed in the microcontroller. After processing, the data is packaged and sent to the Internet module for transmission to the Internet. The voltage regulator is responsible for supplying clean power to every component and steady voltage for analog reference.

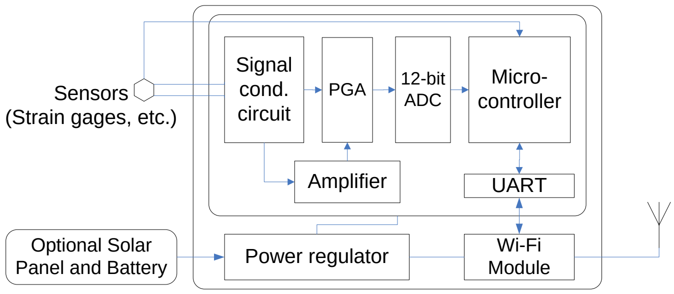

Figure 2: Hardware architecture of wireless sensor node

The sensor node is further divided into a main board and a sensor board. The main board houses a microcontroller, voltage regulating circuits, the Internet module, and a 5-pin header for In-System Serial Programming (ISSP). The sensor board is designed to meet the requirements of specific sensors. Strain gauges are chosen as a major sensor in this research. They are inexpensive, and vast knowledge in interpreting strain data has been accumulated [19]. For strain gauges, a signal conditioning circuit including offset balancing and filtering is necessary. The separation of the main board and sensor board increases the flexibility for various sensors.

Of all the components being chosen, the selection of the microcontroller might be the most important. It is the key element in the wireless sensor node. Not only is it responsible for data collecting and processing but it also communicates with other devices and executes the commands if required. A powerful microcontroller reduces the number of peripherals needed to achieve the same functions and, thus, also trims the total cost. The final selection for this computational core is a Cypress PSoC1® CY8C29466 [20]. PSoC is a programmable system-on-chip that integrates configurable analog and digital peripheral functions, memory, and a microcontroller on a single chip. The biggest difference between PSoC and other conventional microcontrollers is the flexibility. Most microcontrollers provide fixed functions such as the number of ADCs and digital communication types/interfaces. Since the functions are fixed, a change in requirements of the sensor node may inevitably lead to the change of the microcontroller. PSoC, conversely, includes configurable blocks of digital and analog logic, as well as programmable interconnect. By combining the digital block(s) and/or analog block(s), complex functions or peripherals can be created to match the requirements of each application. Therefore on PSoC, the limit is on its fixed number of analog and digital blocks, not on the functions. Even though the number of analog and digital blocks is fixed, the unique dynamic reconfiguration allows the reuse of the same block. Thus, it provides unlimited possibilities for how a microcontroller is being used. The use of PSoC allows the sensor node to connect to up to eight sensors simultaneously. It includes analog, digital, and mixed ones.

The whole sensor node is operating on 3.3 V which is provided by a set of voltage regulating circuits. A TI TPS54231 step down DC/DC converter is adopted to generate a stable 3.3 V. It provides cleaner (fewer ripples) voltage as well as higher efficiency (between 80% and 90%) than conventional voltage regulator. It thus significantly reduces the heat generated by the regulator when operating in higher load and/or higher input voltages. A 1.65 V voltage follower is to provide reference voltage for amplifiers and ADCs.

In order for the sensing data to be accessible from the Internet, the sensor node has to either transmit the data to a nearby base station with Internet connectivity, or connect to the Internet by itself. The former involves a computer device on each monitoring site to collect data from all sensor nodes and relay it to the server. The latter requires each sensor to have the ability to connect to the Internet. The second concept was adopted in this system to eliminate the need to maintain a computer in a remote monitoring site. The Internet module used in the sensor node is MatchPort™ b/g. It is an embedded wireless device server from Lantronix with two serial channels to communicate with the sensor node. The supported WPA2 certification provides the best security for wireless communication. A Buffalo WHR-HP-G54 wireless router is selected to relay the wireless signal from the sensor node to the Internet. Because all nodes are located behind the wireless router that holds a physical IP address, each node is distinguished by a specific port number. Port forwarding technique has been set up in the router. The monitoring server uses the IP address (on the router) along with the specific port number to locate the sensor node.

With the purpose of using strain gauges to collect structural response under stress, a strain sensor board is developed to meet this requirement. Following the signal flow from the strain gauge, a balance control circuit including a trimming potentiometer and resistor is connected to one of the output ends on the Wheatstone bridge to provide offset voltages to balance the bridge voltage. Followed by the balance control circuit is a TI INA126 instrumentation amplifier to provide the first stage amplification. The second stage amplification is a programmable gain amplifier (PGA) implemented in the PSoC. The hardware amplifier delivers higher gain while the software one increases the flexibility of gain adjustment. Before the analog signal is fed into the microcontroller, a Maxim MAX7414 Butterworth low pass filter serves as an anti-aliasing filter to provide a maximally flat passband response. The cutoff frequency used in this research is user adjustable from 1.25 Hz to 160 Hz. There are four channels in the developed strain sensor board to connect to four sets of strain gauges simultaneously.

Figure 3 shows the latest version of wireless sensor node in the weather-proof enclosure without the lid. The main board is in the bottom and the strain sensor board is on the top. The Internet module is mounted on the bottom layer of the main board.

Figure 3: Weather-proof wireless sensor node with lid off

3.2 *Calibration and validation of the sensor node*

In this research, full-bridge gauges were used to minimize the temperature effect applied to strain gauges. At the adopted arrangement, two gauges are aligned with the applied stress, and the other two are in the perpendicular direction to sense the Poisson strain. Shunt calibration was applied to assure the accuracy and linearity of the instrument. Figure 4 shows the linearity of the strain reading to the sensor node.

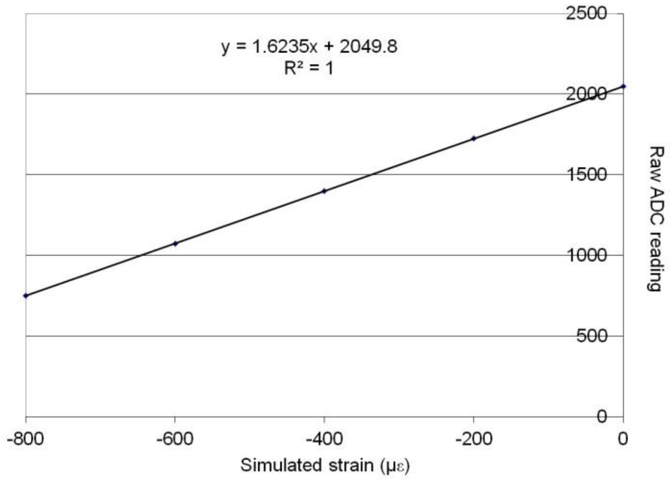

Figure 4: Linearity of reading after shunt calibration

Once calibration of the sensors has been completed, the accuracy of the system has to be validated as well. Two methods have been employed to verify the strain reading. One is to compare the data to data from a portable strain indicator P3500 from Vishay Micro-Measurement. The output of the monitoring system was compared directly to the result from the strain indicator when applying the same load on the same test sample. The difference between the readings is within 1%. The other verification method is to compare the data to that from a V-Link® Wireless Voltage Node from MicroStrain®. Two different deflections were applied to a deformation gauge that is a thin aluminum strip equipped with strain gauges on both sides at the same location. Strain gauges on one side were connected to the V-Link while gauges on the other side were connected to the wireless sensor node. Theoretically, the two gauges should detect the same strain value with opposite signs due to tension or compression on each side of the aluminum strip. The output of both nodes is shown in Figure 5. One of the output data has been reversed for comparison purpose.

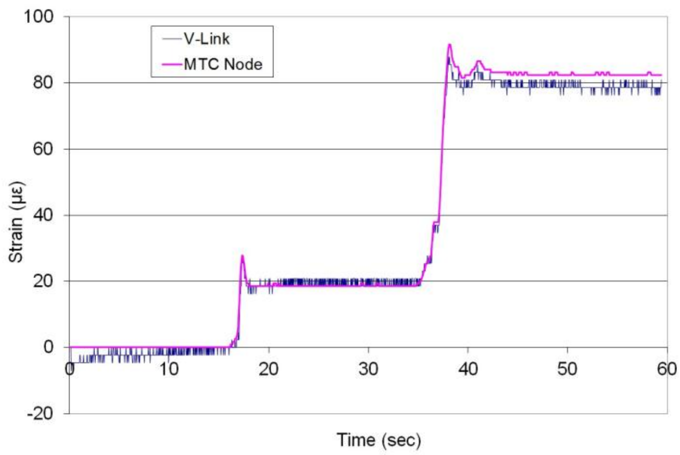

Figure 5: Strain reading accuracy verification with V-Link

Other than the strain gauges, the designed wireless sensor node has been applied to work with other sensors such as temperature sensor, inclinometer, deformation gauge, and PH indicator [21], [22]. This is to show the extensive capability of the developed node on both analog and digital sensors as well as different applications.

4 SERVER SYSTEM

Sensor data cannot be continuously collected from the sensor node without a server system to stay online and acquire data through the Internet. Such a system is called a Data Acquisition and Repository System (DARS). The Materials Technology Center (MTC) has proposed a distributed real-time sensor data management system for structural health monitoring [23]. It represents an original concept in the development of a monitoring system.

A DARS should have the capability to: acquire data from multiple sensor nodes, diagnose incoming data on the fly, display real-time data on a chart, retrieve data for further analysis, and offer users a web interface which they can connect to and use to perform all of these functions. Figure 6 shows the top level diagram of the DARS. The DARS consists of three subsystems: server, database, and webpage. All three subsystems can be placed in different machines and connected through protocols such as TCP/IP and RMI. Even the server subsystem can be broken down to different machines when the system grows bigger and a single computer cannot handle the load anymore. Thus the DARS is also a distributed system. The server contains most aforementioned modules. Once the server starts, the sensor node module will be created as a thread. The number of threads depends on the number of sensor nodes to be connected. The MySQL® database is adapted in the DARS to provide fast performance, high reliability, security, and an environment that is easy to use [24]. The user interface was built using Java applets and is accessible from most browsers. The web interface offers real-time data graphically for each sensor as well as stored data in both graphical and numerical formats.

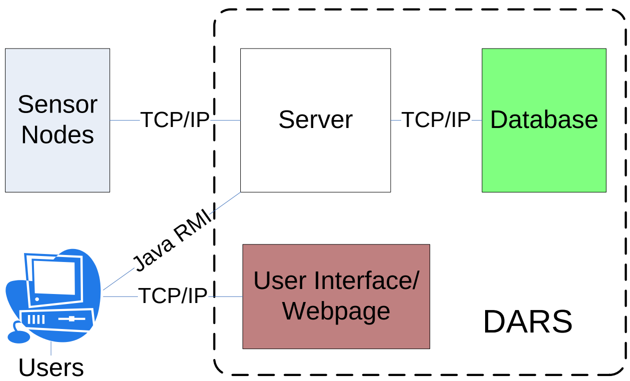

Figure 6: System diagram of data server (DARS)

4.1 *Software modules*

The majority of the DARS was developed in Java™. The key reasons for adopting Java as the platform for developing DARS are object-oriented, platform-independent, distributed and multithreaded [22]. There are three key modules: server, sensor node, and database.

The server module is the core of the DARS. First, it initializes the system. During the initialization process, the server creates sensor node threads. Each thread is assigned unique node identification. After that, the server binds ("binding" is the creation of a reference that can be used frequently later by the program.) its remote objects in the Java Remote Method Invocation (RMI) registry. This allows clients (users) to connect to the server. The remote objects are a series of methods (i.e. functions). Those methods provide interfaces for clients to access and manipulate the data on the server. Once the initialization is done, the server waits on the connections from the clients.

The sensor node module is responsible for all the jobs regarding the sensor node. It can be seen as the data acquisition part of the DARS. Once the server creates the sensor node thread, it also assigns two identifications to the thread. One is site identification, which is used to locate different monitoring sites (differentiated by IP addresses), and the other is node identification, which is used to locate the specific sensor node under each site. Each node is distinguished by its port number when the server makes connection to the node.

When the sensor node module starts, the first task is to establish communication with the database. Each node establishes its own communication tunnel with the database to avoid conflicts. After connecting to the database, the thread begins the process of building a connection with the sensor node. During the early stage in the development of the system, some computers were observed blocking the thread in its attempts to connect or reconnect. This caused the CPU usage to stay on a very high level, usually between 90% and 100%. To solve this issue, a feature known as non-blocking socket connection was introduced into the sensor node module. The idea is to put the building-socket-connection into a separate thread. When the socket connection is established, this thread closes. If the thread is still trying to build a connection, the calling method waits a period of time then rechecks the connection again.

Once the connection is established, the sensor node first clears the input buffer to prevent any unwanted data being used by the system. Then, the data is synchronized in order to find the header of the data packet. The length of the packet depends on the number of sensors connected to the sensor node. There is no checksum in the packet because the TCP protocol provides full reliability by handling the problems of loss, duplication, delay, and out-of-order delivery [25]. Every time a packet is received, the header is checked before receiving the next packet. Doing so ensures the consistency of the incoming data and prevents the wrong data from being used. A thirty-second timeout is set to monitor the receiving process. If the module does not receive any data for thirty seconds, it indicates that the DARS has lost connection with the sensor node. It will re-initialize the connection to build the socket connection with the sensor node. After the data is successfully received, the current server time along with the entire set of data (excluding the header) is stored into the data class. The data class is an object used to preserve the latest sensor data and its time information. This object is used by all other modules in the whole DARS. Since the current focus of the system is on a statistic structural response instead of a dynamic response, the latency between the data acquired on the sensor node and received on the server can be ignored. The accuracy of the time information is 1 millisecond.

The database module in the DARS is an object that is in charge of communicating with the database, MySQL. It provides the interfaces to create the tables, insert the data, and retrieve the data. The database module is instantiated (created) by the sensor node module. Each sensor node module creates its own database module, which simplifies the process of manipulating data. Each database module works independently, without interference with other database modules from other sensor nodes.

The data is stored to the database in its raw format. In this research, sensor data from 12-bit ADC occupies 12 bits in a 2-byte storage space with the leading 4 bits remaining zero. It can be expressed as 0x0### in hexadecimal. Data from inclinometer use the whole 2-byte space as 0x####. The advantage of keeping the raw data is that it occupies fixed storage space. Fixed length storage speeds up the access time for MySQL [26]. Furthermore, unless an ADC with more than 16-bit resolution or data larger than 65535 are required, a 2-byte storage space is enough for most sensors. Thus using raw data saves storage space and allows fast computation. The real value of the sensor data can be interpreted later when needed.

4.2 *User interface*

The user interface in the DARS was built using Java applets. A Java applet is a small application embedded in the web browser (popping out as a new window is also possible) to provide interactive features that cannot be provided by HTML. Since Java is platform-independent, Java applets can also be executed by browsers in various platforms such as Windows or Linux. One of the advantages of using Java applets is that an applet's code is transferred to the user's system and executed by the browser's Java Virtual Machine (JVM). This makes the web application more scalable with the increasing number of users/clients.

The user interface in the DARS provides two main panels for the users. One is real-time data charts and the other displays historical data in both graphical and numerical ways. In the real-time panel, users get to choose the sensors that they are interested in. If there is more than one sensor selected, they are placed together for better comparison. In the historical data panel, users can choose one sensor to retrieve at a time. The chart and the numerical data display below the panel. Both real-time and historical web interfaces are shown in Figure 7.

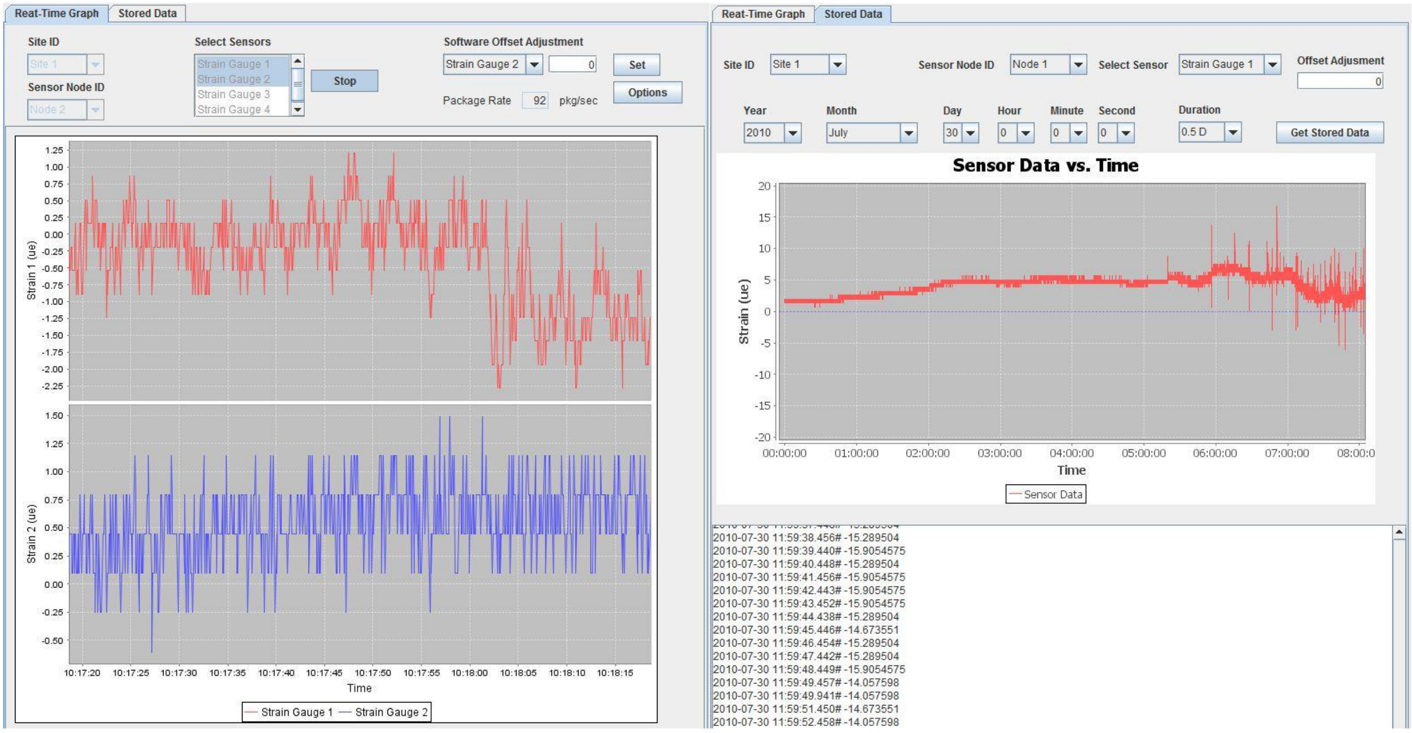

Figure 7: Snapshots of web interface: real-time (left) and historical (right) panels

When users connect to the monitoring website and enter the sensor-reading page, the applets code (interface module) is downloaded to the users' computer. This is how the users connect to the DARS. The interface module can be seen as a web application for users. This page can be password-protected, and people who have permission can access the DARS. When the interface module completes downloading, it starts automatically and connects to the DARS through Java RMI. The module is event driven most of the time. It does nothing but wait until the user operates the application. In the real-time mode, the module requests the data from the DARS 30 times a second, which also means the chart updates at the same rate. The higher update rate produces smoother charting but also a higher CPU load. Passively "request" (pull) from the client is adopted instead of actively "update" (push) from the server. It helps to reach users behind routers and consumes less bandwidth. In the historical data mode, the module gets one channel of data at a time. The data is encapsulated and transmitted using dual layers of Java ArrayList. The main reason for using array list is that it is expandable. The data in the ArrayList is also editable, so the raw data can be converted into engineering data before viewed. When retrieving data from the database, only the time tag and designated sensor data are retrieved. Each row of the data is placed into an inner array, and then this inner array is inserted into the outer array. Since the number of rows of data is unknown (data retrieving is based on time duration, not the amount of data), the size of the outer array keeps expanding until no more data is coming in.

5 CAMPUS BRIDGE DEPLOYMENT

A field test was performed to do further analysis of the stability, accuracy, and practicality of the monitoring system. A steel bridge located on the campus of Southern Illinois University Carbondale was chosen for the test due to its ideal location and easy accessibility. In addition, its design allows it to be easily analyzed; thus, the sensor data can be compared to theoretical results in order to determine the accuracy of the monitoring system. A support tower was built to support the antennas and a metal box houses the necessary equipment. Figure 9 shows the support tower and bridge in the back.

Because the chosen monitoring site is within an acceptable range for radio transmission to the nearby

SIUC Engineering Building, direct connection, which involves a pair of directional antennas and wireless outdoor Ethernet bridges, was adopted. The distance between two antennas is around 800 meters apart. The antenna is a 15 dBi Yagi directional antenna sitting on top of the tower, as clearly seen in Figure 8. The wireless bridge is an AW900x non-line-of-sight wireless Ethernet bridge from AvaLAN Wireless. Its 128-bit encrypted payload protection provides secure data delivery. The data received at the Engineering Building is then fed into the server and is accessible from the Internet. The aerial view of the experiment site is shown in Figure 8. The height of the tower is 3 meters and the antenna on top of the Engineering Building stands around 20 meters above the ground. The trees surrounding the tower are roughly 10 meters tall, as seen in the bottom left corner of Figure 8. Two tests have been performed in order to see the effects of the vegetation on the transmission. A laboratory test was also made for comparison purpose. The first test was conducted during early spring while most of the trees were still without leaves. The second test was carried out in early summer when the foliage of the trees was flourishing. The result is shown in Table 1. The result shows no significant difference due to the existence of the foliage. There is approximately a 4% reduction in the connection speed compared with the test in the laboratory due to the increased distance from 1 meter to approximately 800 meters.

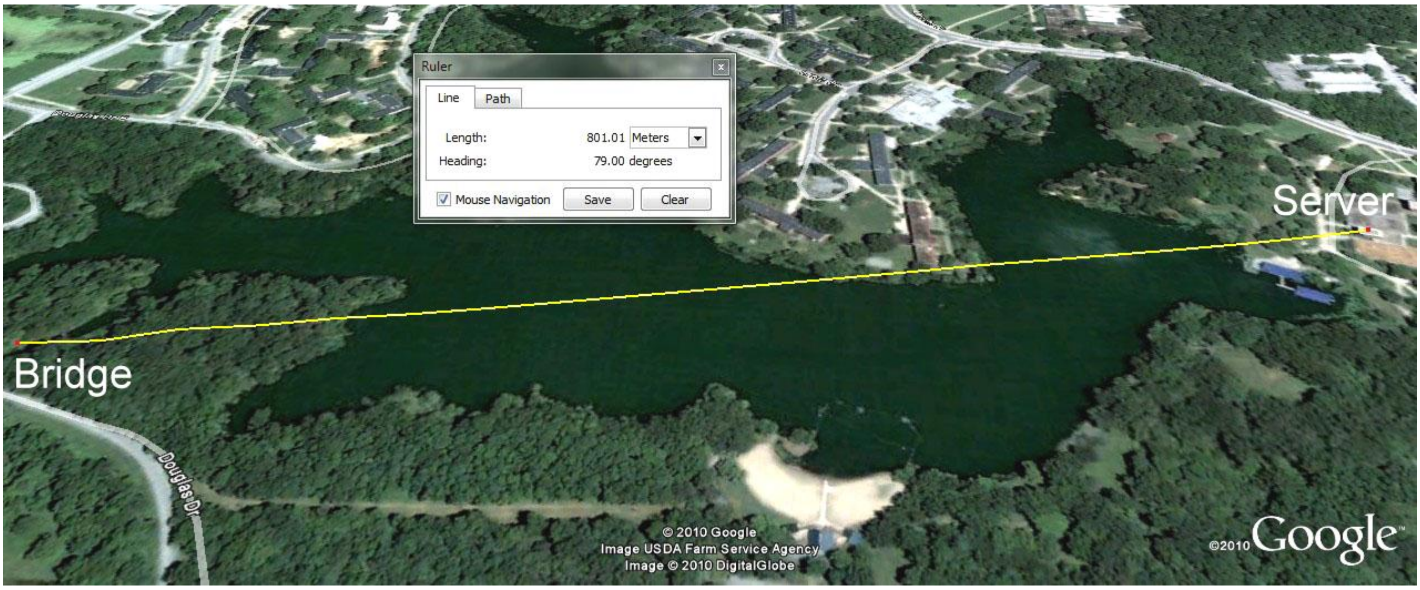

Figure 8: Aerial view of the experiment site

Figure 9: Support tower and the bridge

Table 1: Ethernet throughput of direct connection

	Distance between two antennas		
	1 m (laboratory)	800 m (w/o leaves)	800 m (w/ leaves)
Throughput	810 Kbit/sec	778 Kbit/sec	780 Kbit/sec

An experiment was conducted using 50-lb sandbags as the load. The load was increased 500 pounds at a time, up to 4000 pounds. The load was applied on the middle of the bridge and distributed evenly across the deck. The sensor node was attached on the bridge with two strain gauges connecting to the node. The strain gauges were installed on the bottom side of the upper truss, as #15 and #16 in Figure 10. The data from both strain gauges was recorded and charted in Figure 11. From the location of the strain gauges, the two gauges should yield identical strain response for the applied load. The result showed a positive trend of the structural response under loading condition. A full bridge 3-dimentioned model was built to conduct finite element analysis to compare with the test result. The analysis showed the strain value was -25.06 $\mu\varepsilon$, whereas the value conducted from the field test was -22.45 $\mu\varepsilon$.

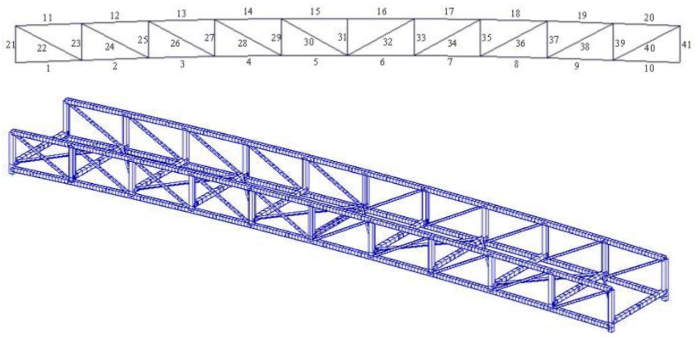

Figure 10: Bridge truss and its FEM model

After the field test, the system has been kept running on the bridge to continuously collect structural response. By doing so, a sufficient amount of data would be collected to construct the database for further data analysis. Meanwhile, running under all weather conditions helps to fine tune the performance of the entire system and reveal problems that have yet to be found in the laboratory. One particular event has been observed during the long-term data collection. When it starts to rain, the compressive strain value increases and then remains stable shortly after. The reason is that the bridge deck is made by wood, and the wood soaks in the water to increase the load to the structure.

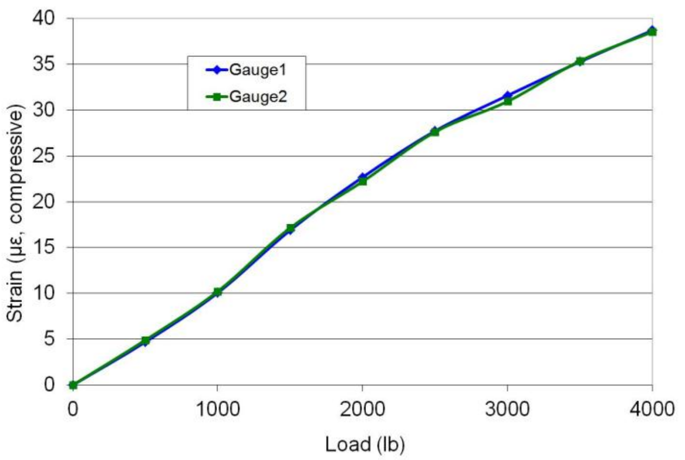

Figure 11: Strain reading at two symmetric gauges under varying loads

6 CONSLUSIONS

This paper presents an on-line, real time and complete structural health monitoring system consisting of sensor nodes, data server and users' web interface. Sensor nodes are centralized data acquisition subsystem which is able to connect to multiple and various sensors simultaneously while continuously working in harsh environments. The data server provides reliable data storage and fast data retrieval. The user web interface offers users a friendly and ease-of-use environment to monitor real-time and historical data in both graphical and numerical ways. Each individual subsystem and the integration of the whole system have been verified in the laboratory setting. The proposed SHM has been deployed into one of campus bridges. The accurate data signal from the sensor nodes was successfully received by data server a half mile away from the bridge. The experiment result showed the consistency with the theoretical value. The establishment of the data link realized the continuous and dependable structural monitoring.

The key feature of this system is to continuously monitor structural response and broadcast information in real-time to users via the Internet which clearly stands out above the others systems. We believe that our system poses a promising approach to addressing the complexities to monitoring structural conditions of the large number of aging bridges. It is conceivable that our sensor nodes, data server and user web interface will become part of hardware and software components of the future structural health monitoring systems.

7 FUTURE WORK

The system is planned to have a larger scale deployment to local bridges in the future. By doing so, long-term strain database on various types of bridges will be established.

Regarding to the system itself, a series of methods is being developed to improve the performance of the sensor node and detect the failure of the sensors. An auto-zero mechanism is developing to eliminate the manual zeroing after installing bridge sensors. An intelligent sensor readout circuit (ISROC) is to introduce to replace the conventional filters and amplifiers by detecting the sensor SNR and dynamically adjusting the sampling rate of the ADC. Self-testing mechanisms are going to be explored to ensure that the reading from the sensors is reliable. Data management subsystem is to be added to manipulate the data, perform damage detections, and remove redundant data.

8 ACKNOWLEDGEMENTS

The authors acknowledge the help from Christopher Williams, Bryant McDonnell, and Eric Miler who were undergraduate researchers at the Southern Illinois University Carbondale. This research is funded by Federal Highway Administration under Grant ITS-0517(109).

REFERENCES

1. "Deficient Bridges by State and Highway System (2007)," Federal Highway Administration (FHWA), http://www.fhwa.dot.gov/bridge/defbr07.cfm, 2008.
2. M. Çelebi, "Seismic Instrumentation of Buildings," Open-File Report 00-157, U.S. Geological Survey, 2000.
3. K. Chintalapudi, T. Fu, J. Paek, N. Kothari,S. Rangwala, J. Caffrey, R. Govindan, E. Johnson, and S. Masri, "Monitoring Civil Structures with a Wireless Sensor Network," Internet Computing, IEEE, Volume 10, Issue 2, pp. 26-34, 2006.
4. B.F. Spencer, T. Nagayama, J. A. Rice, "Decentralized structural health monitoring using smart sensors," Proc. of SPIE Sensors and Smart Structures Technology for Civil, Mechanical, and Aerospace Systems, Vol.6932, San Diego, CA, 2008.
5. B. Chen and J. Wang, "Design of a Multi-Modal and High Computation Power Wireless Sensor Node for Structural Health Monitoring," Proc. of IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications, pp.420-425, 2008.
6. S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks," Proc. of 6th International Symposium on Information Processing in Sensor Networks, 2007.
7. L.M. Rodríguez Peralta, L.M.P. Leão Brito, B.A. Teixeira Gouveia, "The WISE-MUSE Project: Environmental Monitoring and Controlling of Museums based on Wireless Sensors Networks," EJSE Special Issue: Sensor Network on Building Monitoring: from Theory to Real Application, pp.46-57, http://www.ejse.org/, 2009.
8. E.G. Straser and A.S. Kiremidjian, "A Modular, Wireless Damage Monitoring System for Structures," Technical Report 128, John A. Blume Earthquake Engineering Center, Stanford University, 1998.
9. J.P. Lynch and K. J. Loh, "A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring," The Shock and Vibration Digest, Vol. 38, No.2, pp.91-128, 2006.
10. Y. Wang and K.H. Law, "Wireless Senging and Decentralized Control for Civil Structures: Theory and Implementation," Technical Report 167, John A. Blume Earthquake Engineering Center, Stanford University, 2007.
11. J.P. Lynch, "Decentralization of Wireless Monitoring and Control Technologies for Smart Civil Structure," Ph.D. Thesis, Department of Civil and Environmental Engineering, Stanford University, 2002.
12. K. Mitchell, V.S. Rao, and H.J. Pottinger, "Lesson Learned About Wireless Technologies for Data Acquisition," Proc. of the SPIE, Vol.4700, pp.331-341, San Deigo, CA, 2002.
13. E. Sazonov, K. Janoyan, and R. Jha, "Wireless Intelligent Sensor Network for Autonomous Structural Health Monitoring," Proc. of the SPIE, Vol.5384, pp.305-314, San Deigo, CA, 2004.
14. N. Xu, S. Rangwala, K.K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A Wireless Sensor Network For Structural Monitoring," Proc. of the 2nd International Conference on Embedded Networked Sensor System, pp.13-24, Baltimore, MD, 2004.
15. T. Kijewski-Correa, M. Haenggi, and P. Antsaklis, "Multi-scale wireless sensor networks for structural health monitoring," Proc. of SHM-II'05, 2005.
16. J.A. Rice, B.F. Spencer, "Structural health monitoring sensor development for the Imote2 platform," Proc. of SPIE Smart Structures/NDE, 2008.
17. S.S. Law, X.Y. Li, X.Q. Zhu, and S.L. Chan, "Structural damage detection from wavelet packet sensitivity," Engineering Structures, vol. 27, pp. 1339-1348, 2005.
18. J.P. Lynch, "Design of a wireless active sensing unit for localized structural health monitoring," Structural Control & Health Monitoring, vol. 12, pp. 405-423, 2005.
19. H. Choi, S. Choi, and H. Cha, "Structural Health Monitoring System based on Strain Gauge Enabled Wireless Sensor Nodes," Proc. of 5th International Workshop on Networked Sensing Systems, pp.211-214, 2008.
20. PSoC®, http://www.cypress.com/psoc/.
21. C.O. Yang, C. Williams, and A. Miller, "Internet-Based, Wireless, Remote Sensing System and Its Practical Impacts," Creativity-In-Action Contest, Taiwan, 2007.
22. C.O. Yang, "Cyber-physical System: Real-time Internet-based Wireless Structural Health Monitoring System," Ph.D. Dissertation, Department of Electrical and Computer Engineering, Southern Illinois University Carbondale, 2009.
23. S. Zhang, C.F. Wang, and M. Yen, "Sensor Data Management in Structural Health Monitoring Systems," Proc. of 5th International Workshop on Structural Health Monitoring, Stanford, CA, 2005.
24. MySQL®, http://mysql.com/why-mysql/.
25. Douglas E. Comer, "Network Systems Design using Network Processors," pp. 20, ISBN 0-13-187286-9, 2006.
26. Luke Welling and Laura Thomson, "MySQL Tutorial," pp. 232, ISBN 0-672-32584-5, 2003.