**2004 8th International Conference on**
**Control, Automation, Robotics and Vision**
**Kunming, China, 6-9th December 2004**

# Wireless Sensor Network for Machine Condition Based Maintenance[1]

A. Tiwari and F.L. Lewis
Automation and Robotics Research Institute
The University of Texas at Arlington
7300 Jack Newell Blvd. S
Ft. Worth, Texas 76118-7115
email lewis@uta.edu, http://arri.uta.edu/acs

Shuzhi Sam Ge
Department of Electrical Engineering
National University of Singapore
Singapore 117576
elegesz@nus.edu.sg
http://vlab.ee.nus.edu.sg/~sge

## Abstract

A new application architecture is designed for continuous, real-time, distributed wireless sensor networks. We develop a wireless sensor network for machinery condition-based maintenance (CBM) using commercially available products, including a hardware platform, networking architecture, and medium access communication protocol. We implement a single-hop sensor network to facilitate real-time monitoring and extensive data processing for machine monitoring. A LabVIEW graphical user interface is described that allows for signal processing, including FFT, various moments, and kurtosis. A wireless CBM sensor network implementation on a Heating & Air Conditioning Plant is presented as a case study.

## 1. Introduction

Remote sensing and measuring is becoming more important with accelerating advances in technology. The ability to process large amounts of data using the internet and advanced digital signal processing techniques means that data can be collected, processed, organized, and interpreted as never before in history. This opens up possibilities of detailed monitoring of the environment, wildlife habitats, complex industrial machinery, aerospace vehicle platforms, and consumer equipment and the home environment. Sensing has advanced from manual meter reading to centralized data acquisition systems to a new era of distributed wireless sensor networks (WSN). WSN can now provide an intelligent platform to gather and analyze data without human intervention. Typically, a sensor network consists of autonomous wireless sensing nodes that are organized to form a network. Each node is equipped with sensors, embedded processing unit, short-range radio communication module, and power supply, which is typically 9-volt battery. With recent innovations in MEMS sensor technology, WSN hold significant promise in many application domains. On the other hand WSN are also exposed to many technical limitations including energy and memory constraints, available processing power, transmission rate, synchronization rate, and robustness in operation [12]. To overcome these limitations one needs to have optimized WSN design.

With most research efforts targeting on applications like habitat monitoring [1], area monitoring [6], surveillance [4], etc, environmental sensing and processing remains the principle stimulant in the evolution of sensor networks. Most of the protocol architectures, viz. S-MAC [15], PAMAS [13], are thus designed for applications where data is acquired only when an interesting event occurs or when prompted by user. In contrast to this, we focus on applications requiring turn-wise, continuous, periodic, and real-time transmission of data from sensors. One such application is condition based maintenance (CBM) of machinery and equipment for reliability and health maintenance. Real-time monitoring and control increases equipment utilization and lifetime, and positively impacts system yield and throughput [5].

This paper develops a new application domain for continuous, real-time, distributed wireless sensor networks. It presents design requirements, limitations and guidelines for basic sensor network architectures for such applications. It describes a hardware platform, networking architecture, and medium access protocol for such networks. We implement a single-hop sensor network to facilitate real-time monitoring and extensive data processing for machine monitoring, using the commercially available MicroStrain wireless sensors. A LabVIEW graphical user interface is described that allows for signal processing, including FFT, various moments, and kurtosis. Time plots can be displayed in real time and alarm levels set by user. A wireless CBM sensor network implementation on a Heating & Air Conditioning Plant is presented as a case study.

## 2. Motivation for WSN in CBM

Distributed data acquisition and real-time data interpretation are two primary ingredients of an efficient CBM system. These two are mutually dependent on each other. Data interpretation algorithms are learning systems that mature with time. Distributed data acquisition should thus be adequate for both machine maintenance and learning by the monitoring system. In control theory terms, one needs both a component to control the machinery and a component to probe or identify the system. Wireless sensors are playing an important role in providing this capability. In wired systems, the installation of enough sensors is often limited by the cost of wiring, which runs between $10-$1000 per foot. Previously inaccessible locations, rotating machinery, hazardous or restricted areas, and

---

[1] This work was supported by ARO Grant DAAD 19-02-1-0366 and NSF Grant IIS-0326505

mobile assets can now be reached with wireless sensors. These can be easily moved, should a sensor need to be relocated.

Often, companies use manual techniques to calibrate, measure, and maintain equipment. In some cases, workers must physically connect PDAs to equipment to extract data for particular sensors, and then download the data to a PC [9]. This labour-intensive method not only increases the cost of maintenance but also makes the system prone to human errors. Especially in US Naval shipboard systems, reduced manning levels make it imperative to install automated maintenance monitoring systems. Wireless Sensor Networks, are highly flexible, unattended, self operative systems with low installation costs and minimal intrusion in the existing infrastructure. WSN are quick and easy to install, and require no configuration tools and limited technical expertise of the installer. WSN is also the best solution for temporary installation when troubleshooting or testing machines.

## 3. Design Requirements

In order to design an efficient architecture for WSN, it is important to understand the requirements that are relevant to the sensor applications [17]. We chalk out the following requirements for implementation of WSN for CBM and many such applications.

**Continuous Sensing.** Critical manufacturing processes and equipment must be continuously monitored for any variations or malfunctions. A slight shift in performance can adversely affect overall product quality or manufacturing equipment health.

**Periodic Data Transmission.** CBM systems rely on historical data for diagnosis of impending failures and defects. These are dynamic systems that continuously learn during their operation. Periodic data transmission thus helps update the historical data that in turn helps improve the overall efficacy of the system for both diagnosis and prognosis of system failures and computing remaining useful lifetime of equipments.

**User-Prompted Data Querying.** With a group of sensing nodes monitoring various manufacturing equipments and processes and transmitting data in periodic manner, situations may arise where the engineer might want to query data from some specific nodes to estimate current status of particular process or equipment. A provision for breaching the cycle of periodic transmission to address user-prompted querying is thus required.

**Emergency Addressing and Alarms.** There can be situations of unforeseen malfunctioning or variations beyond prescribed tolerance bands. A mechanism is hence required to define tolerance bands for each sensing module. When measurements at particular node exceed the tolerance, the node must breach the periodic cycle to send an alarm about the emergency.

**Adaptability.** CBM systems are adaptive learning systems, characterized by their evolutionary behavior over time. They should be capable of adapting to new situations and incorporating new knowledge into their own knowledgebase. This inherent adaptability of CBM systems demands a similar characteristic from the WSN architecture.

**Network Setup & Reconfigurability.** During the set-up phase of CBM system or during reconfiguration, the maintenance engineer may want to alter the functionality of individual nodes. This may include changes in sampling rate, number of data points transmitted during each transmission, the sequence in which nodes transmit, number of channels transmitted from each node, tolerance band for each sensor node etc. Such re-tasking provision should be built into the design of the WSN.

**Scalability.** Over the duration of operation, some sensing nodes may fail or their batteries may become depleted. Also, a need may arise for installation of more sensing nodes to monitor processes and equipments more closely and precisely. The WSN should be scalable to accommodate changes in number of nodes without affecting the entire system operation.

**Energy Efficiency.** Sensor nodes are autonomous devices that usually derive their power from a battery mounted on each node. It becomes necessary to have an inherent energy saving notion in every component of the WSN system to prolong the lifetime of each node in network. All layers of the architecture are thus required to have built-in power awareness.

**Feedback control.** To provide real-time control capabilities for certain dynamic processes, features might be added to allow breaches in normal network operation to transmit control signals back to the nodes. This could help in reducing manning levels by eliminating minor manual control or machine resetting operations.

## 4. MicroStrain Sensor Net Hardware

For our implementation, we used the X-link wireless measurement system purchased from MicroStrain, Inc. [http://www.microstrain.com/]. There are three types of sensor nodes- G-link (MEMS accelerometer - ±10G full scale range), SG-link (strain gauge), and V-link (supports any sensor generating voltage differences). Each node in itself is a complete wireless measurement system with a Microchip PIC 16f877A microcontroller as its heart. It has a RISC CPU with just 35 single word instructions, 8K bytes of flash program memory and 624 bytes of total data memory. External 2 MB data memory on the nodes is ATMEL serial flash memory. Nodes contain low power RF Monolithic transceiver model TR1000 [11] using on-off keyed (OOK) modulation of 916MHz carrier frequency and providing transmission rate of 19.2 Kbps up to 30 m of range. It draws input current of 3.1mA in receive mode, 12mA in transmit mode, and 0.7μA in the sleep mode of operation. Sensor nodes are multi-channel, with maximum of 8 sensors supported by a single wireless node. A single receiver (Base Station) addresses multiple nodes; maximum of $2^{16}$ nodes can be addressed as each node has 16-bit unique address. All nodes support a 9V external rechargeable battery. Baud rate on the serial RS-232 link between Base Station (BS) and terminal PC is 38400.

## 5. System Description

Having explicitly defined the requirements for the given application domain, we now look at the actual architecture, network topology and protocol design to address those requirements.

The intent of our WSN is to collect data from distributed sensors so that we can test-run various data analysis and decision making algorithms on the combined data from various sensors. We wish to compare these runs with a stored fault pattern library to diagnose faults or impending failures, and we wish to upgrade the existing fault pattern library. Finally, it is required to estimate remaining useful life (RUL) of the equipment and display results to maintenance personnel.

### 5.1. Single-Hop Topology

In consideration of our design requirements, we pose an adaptive and scalable data-gathering wireless sensor network with an event-driven emergency alarm tipster. A many-to-one paradigm is used whereby multiple sensor nodes transmit to a single sink for collective data analysis, decision-making and storage.

Keeping in view the energy constraints, latency requirements, required simplicity at the nodes, and to avoid all the control overheads, we developed a single-hop topology for our network. For the short-range radio used on nodes, energy consumption in the transmitter electronics is more than the energy required to generate RF output power. For e. g., for a low power transceiver available today [11], current consumption contributing to the RF output power of 1.5 dBm is only 0.45 mA out of 12 mA of total current consumption in transmitter section. Using multi-hop topology would be more energy exhausting, in this case, as a minimum of 11 mA current will be required by the transmitter section of each node for every transmission. Also, in the single-hop net, if a single node in the network fails, rest of the network remains unaffected.

## 6. UC-TDMA MAC Protocol

For CBM it is necessary for the user to explicitly define the sequence in which data will be collected. This will help in establishing relationships between two measurements and drawing conclusions. We here design a User Configured Time Division Multiple Accessing (UC-TDMA) based MAC protocol for our network. TDMA is intrinsically less energy consuming than contention protocols. Many researchers have focused their work on MAC protocols specifically for WSN [7], [10], [14], [15], [16]. Figure 2 gives the flowchart of our UC-TDMA protocol.

### 6.1. TDMA Slot Allocation

Even though TDMA based protocols offer natural collision avoidance and energy preservation, they are sometimes not preferred for memory constrained sensor networks. Both the traditional TDMA and the TDMA scheduler approach of TDMA implementation are not easy to implement for distributed networks. Maintaining a TDMA table at each node takes up a major chunk of its valuable memory. This could hamper other memory expecting operations like in-network data processing which, along with overheads involved in protocol, shares the same limited onboard memory of sensor nodes. Running a TDMA scheduler on each node is prohibitive.

We circumvent this major impediment by utilizing our central base station for maintenance of TDMA slot assignment table. Being a single hop network, all the nodes in our network communicate directly with base station. It is thus easier to maintain time slots for all nodes at base station alone and communicate with them accordingly. With this approach, nodes need not maintain any table or do any scheduling for time slots which saves both memory and complexity at nodes.

Using our GUI detailed in section 7, the user can define the sequence in which nodes will access the channel and also the time duration for their respective slots. Depending on application, nodes might access the channel more than once in a given time frame and also length of each slot can be different. Figure 1 shows the TDMA frame for our network. Note that, this is little different from usual TDMA method in a sense that we are trading off fairness at each node for meeting our application needs.

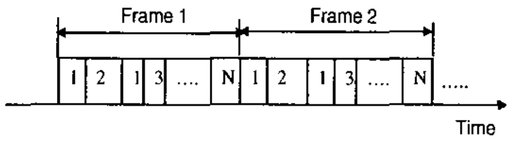

***Figure 1:* UC-TDMA frame showing time slots for N nodes in network**

### 6.2. Energy Model of RF Node

We now investigate the physical layer to build in an effective energy aware MAC protocol for wireless sensor networks. Using the radio model for power consumption given by Shih *et al.* [2] we developed a model for battery consumption of the radio used on our nodes.

$$\begin{aligned} AmpHrs/Hr = {} & N_{s-tx}\left[I_{tx}\left(T_{s-tx}+T_{tx}\right)+I_{txm}T_{tx}\right]+N_{rx-tx}\left[I_{tx}\left(T_{rx-tx}+T_{tx}\right)+I_{txm}T_{tx}\right] \\ & +N_{s-rx}\left[I_{rx}\left(T_{s-rx}+T_{rx}\right)\right]+N_{tx-rx}\left[I_{rx}\left(T_{tx-rx}+T_{rx}\right)\right] \\ & +N_{rx-s}\left[I_{s}\left(T_{rx-s}+T_{s}\right)\right]+N_{tx-s}\left[I_{s}\left(T_{tx-s}+T_{s}\right)\right]+I_{rx/tx}T_{turn-on} \end{aligned}$$

This formula models the per-hour battery consumption (Ampere-Hour) by radio on the sensor nodes. $N_{s/rx-tx}$ is number of times per hour that radio switches to transmit mode from either sleep or receive mode, similarly $N_{s/tx-rx}$ and $N_{rx/tx-s}$ are number of times per hour that radio switches to receive mode and sleep mode respectively from either of the remaining modes, $T_{s/rx-tx}$ is time taken by radio to switch to transmit mode from either sleep or receive mode, similarly $T_{s/tx-rx}$ and $T_{rx/tx-s}$ are time taken to switch to receive and sleep mode respectively from either of the remaining modes, $T_{tx/rx/s}$ is the actual time for which radio transmits, receives or sleeps after switching to respective modes. $I_{tx}$ is current drawn by transmitter electronics alone, $I_{txm}$ is the modulation current responsible to generate RF output power ($P_{rf} \propto$ f($I_{txm}$)). $I_{rx/s}$ is current drawn by radio in receive/sleep mode.

## 6.3. Sleep Scheduling

A typical radio for short-range transmission operates in either of the following modes: Transmit mode, in which it transmits some data on RF channel. Receive mode, in which it receives data transmitted for it on the RF channel. Idle/listen mode, in which it keeps on tracking the RF channel for any intended data. Sleep mode, during which the radio shuts off and sleeps. Neither can it receive nor can it transmit any data in this mode. Electric current consumption in idle mode and receive mode is almost similar. For commercially available radios, the ratio of current drawn in sleep mode to listen/idle mode is of the order of 1: 4500 or more [11]. As nodes are in idle state most of the time, except when they transmit or receive, listen/idle mode leads to substantial energy wastage. Many MAC protocols for WSN exploit this radio hardware feature by putting radio in sleep mode when it is neither transmitting nor receiving any data on the RF channel. In periodic sleep and listen scheme used by many protocols, radio sleeps for certain duration of time and then listen for certain time to see if anyone wants to talk to it. In doing this it switches back and forth in two modes. In order to keep the latency into tolerable limits, frequency of this switching is usually kept high so that a sender need not wait longer for a receiver to wakeup.

However, on considering our radio energy model, we quantified that the energy consumption in switching the modes periodically is more than that required in transmission of data. In steady phase of a sensor network, let's say radio switches its mode every 30 seconds from sleep to awake and awake to sleep mode. On the basis of the energy model given by equation (1), Current consumption per hour in just switching the modes is 0.059 mA-hour/hour. This much of current consumption is sufficient to transmit data for 17.7 seconds per hour. This is much more than the typical transmission time per hour for any sensor in the network. Given data is for commercially available radio [11]. Pertaining to application requirements, we thus seek to minimize the frequency of switching between the sleep and receive modes.

Given the sweep rate for each node, number of data points from each node, frequency at which each node transmits (every r hours) and the sequence in which they transmit data, the sleep duration for all the nodes in network can be calculated using following formulation:

$$T_p = U \div diag[S_r]$$

$$S_d = [T_p \times N_s^T - diag(N_s^T \times T_p)]^T \quad \textit{iff updating Rate not given} \, (2)$$

$$S_d = [3600 \times R_u^T - diag(N_s^T \times T_p)]^T \quad \textit{iff updating Rate given} \, (3)$$

Where $S_r^{1\times n}$ is matrix containing sweep rate for all the nodes in the sequence in which they transmit, $N_s^{1\times n}$ is matrix containing number of data points transmitted by respective nodes, $U^{1\times n}$ is a unit matrix, $R_U^{1\times n}$ is updating rate matrix which contains the rate (in every *r* hours) with which every node transmit its data, $T_p^{1\times n}$ is time period matrix (1/sweep-rate) for each node, $S_d^{1\times n}$ is matrix containing sleep duration (in seconds) calculated for each node in network.

## 6.4. Modes of Operation

Broadly, we have considered two modes of operation for our network. The first is *continuous mode.* This is useful for newly deployed networks where we are usually confronted with a question on how frequently data should be collected from various sensor nodes. In this mode we collect data continuously and sequentially from each node. Sleep durations given by equation (2) are used in this mode. Nodes transmit their data and then sleep for the time during which other nodes transmit. The other mode is *non-continuous mode* of network operation. After operating a newly installed network in continuous mode and through proper analysis of the data obtained, one can answer the question posed above. We can then obtain an updating rate matrix $R_u^{1\times n}$ and use equation (3) to obtain the sleep durations. Node transmits data after every r hours, where r is the rate specified by updating rate matrix for that particular node. All the nodes can have different updating rates. Here also node sleeps all the time except when it transmits on its turn. While calculating the sleep durations we have neglected the time taken by a node to setup link with the base station before transmitting the data. This provides us the margin to tackle with the clock drifts of nodes as each node wakes up a little before its turn to transmit.

## 6.5. Network Setup

To start-run the sensor network for machine monitoring, we first need to physically install sensors at proper locations on the machine. Each sensor contains a 16-bit node type associated with the physical quantity (like vibration, temperature, pressure, strain, etc.) it measure. Base station is connected through a serial port to some hand held computing device (like PDA, laptop) which runs the application program. At power-on all nodes in network are in receive mode. In the flow chart given in figure 2, first few blocks describes the setup of network. Through a user interface, user defines the functionality of various nodes in network. User can specify values of various parameters, individually for all the nodes. Node parameters include sweep rate, number of sweeps, node type, sequence number of node in TDMA frame and active channels. Base station keeps these parameters in different parameter arrays according to the sequence of nodes in TDMA time frame. After obtaining functional definition for all nodes in the network, base station checks for the availability of defined nodes. If any node is found to be missing then base station alarms the user about missing node and updates all of its parameter arrays. The sleep duration for each node is then calculated by using equation (2) or (3). The base station then configures all the nodes one by one with the specified parameters and calculated sleep durations.

## 6.6. Main Thread

After setting up the network and configuring all the nodes we are now ready for gathering data from distributed sensors. Data is collected in accordance with the UC-TDMA frame maintained by the base station. We seek to minimize two major sources of energy wastage viz. *collisions* and *protocol overhead* by using our modified version of RTS (Request To Send) and CTS (Clear To Send) mechanism used in IEEE 802.11.

We exploit our power plugged base station (BS) for affecting our modified RTS-CTS mechanism. From the TDMA schedule maintained, the BS knows which node in the network has access to channel at any particular instant. Instant any node acquires the channel according to its schedule; on behalf of that node the BS itself generates a virtual RTS after assuring that no other node is communicating with it. The node also wakes up at this instant and is ready to receive CTS signal from the BS before it transmits its data. The BS then sends a CTS signal with node address appended to it. On receiving this signal node transmits the predetermined number of data points to the BS. After successful reception of data points, the BS acknowledges the node with request-to-sleep. Similarly, data is collected sequentially from all the nodes in the UC-TDMA frame and then frame is repeated indefinitely to collect data from the network.

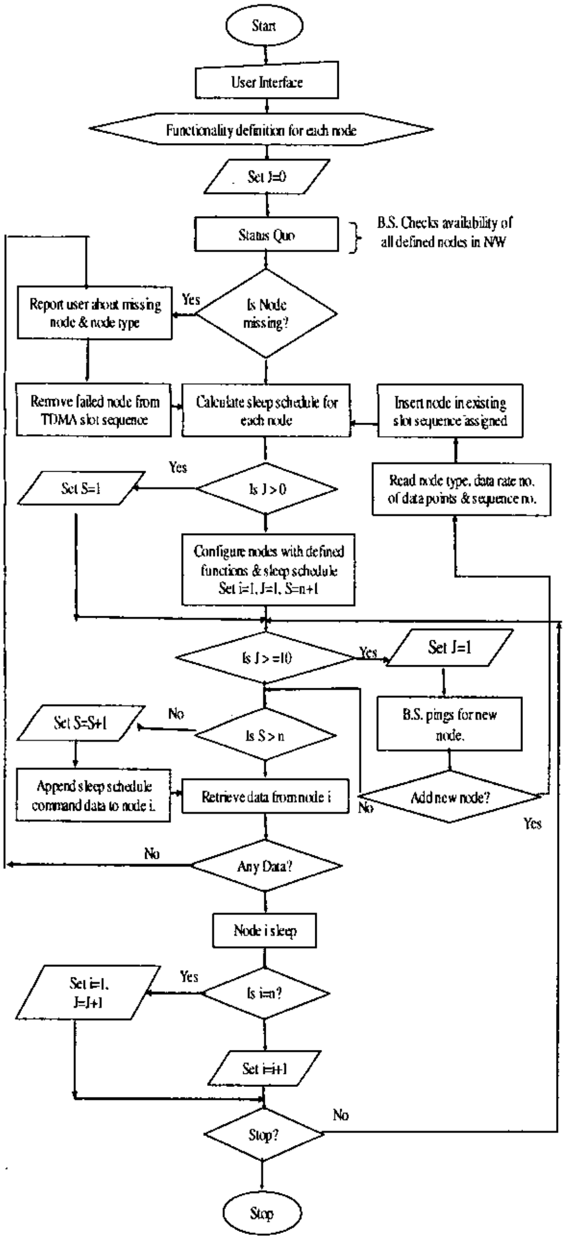

*Figure 2:* Flowchart for UC-TDMA protocol.

Collisions cause significant amount of energy wastage as messages are to be retransmitted. Also retransmission of messages some time causes a loop in schedule and may further spoil other future transmissions. It is thus wise to spend some energy in contention mechanism along with scheduling. Even with contention alone, collisions are not reduced to zero and latency is increased as sender waits for random duration of time before contending again for the channel. With this modified RTS-CTS, the BS generates the virtual RTS, there is no chance that any two nodes will simultaneously contend for the channel during normal network operation. Even if a node is scheduled to access the channel and BS is talking to some other node, the virtual RTS for the scheduled node will not be generated until the BS has finished communicating with the current node. This reduces the probability of collisions to zero. The virtual RTS scheme also reduces the control overhead for contention to zero, for nodes in the network. As these control overheads are short packets, they are highly energy exhausting, a large amount of energy is wasted in startup of transmitter electronics [2]. In original RTS-CTS mechanism [8], considerable amount of energy is wasted in switching of modes from sleep to receive then to transmit and then again to receive each time RTS signal is being sent. Also, processing required at the node to acquire the channel is reduced nearly to zero. The node simply sleeps and wakes up according to a timer. Modified RTS-CTS scheme thus saves fairly large amount of energy by tapping our overall network arrangement and data gathering requirements.

## 6.7. Adaptability and Reconfigurability

During normal operation of network in non-continuous mode, application data requirements might change for a set of measurements. Our network should then be able to adapt to such changes by adjusting certain node parameters (like sampling rate, number of data points and active channels). But then this adjustment should not disrupt the normal functioning of remaining network.

To render above mentioned adaptability, base station makes use of new node parameters to calculate new sleep schedule for the desired nodes. Here we have assumed that application do not compel network to change the sequence of nodes to adapt to the changes. To configure nodes with this new schedule and parameters, after completing the ongoing cycle, the BS appends these parameters and schedule to the CTS signal sent to the nodes whilst they acquire the channel.

To facilitate the re-configuration that also includes the change in the sequence in which nodes transmit, the BS calculates new sleep schedule for the entire network and uses new UC-TDMA frame for sequencing the nodes. It takes one complete TDMA frame to affect the re-configuration of the network.

## 6.8. Emergency Addressing and Alarm

To address any emergency situation, our nodes keep sensing the physical quantity even when radio is in sleep mode. The nodes compare the measured value with a set threshold value on a continuous basis. On determining that the measured value exceeds the set threshold, node declares an emergency situation to be addressed immediately. Node then wakes up its radio and transmits its node address to the BS until responded.

In continuous mode of network operation the BS remains busy, talking to some nodes, all the time. So when a node in emergency transmits its address on the channel already occupied by some other node, it results in a continuous checksum error due to collisions. As our MAC protocol assures that there are no collisions in any other situation, the BS interprets these continuous collisions as an indicator of emergency. To handle this, the BS hangs up the ongoing operation and receives the address of the node in emergency. After addressing the emergency the BS catches up with the node scheduled to access the channel at that particular instant.

## 6.9. Scalability

To address the requirements of scalability, the following two aspects are to be taken care of: failure of existing nodes and addition of new nodes. If a node is not able to transmit its data at the scheduled time, it is considered to be failed. It can be seen from the flowchart in figure 2 that if data from any node is not retrieved, it is declared to be failed. The BS then reports about the missing node (with its node type and node address) to the engineer. The UC-TDMA frame is then scaled by removing the failed node from the sequence. The sleep schedule for the remaining nodes is calculated again with new TDMA slot sequence. This new sleep schedule for each node is appended to CTS signal sent to the nodes according to the existing slot sequence. As it can be seen from the flowchart, new UC-TDMA frame is affected after sending the new schedule to all the nodes according to the existing schedule.

Addition of new nodes is not so frequent affair in the network. To scale the UC-TDMA frame with addition of nodes, after every ten repetitions of TDMA frame the BS pings for availability of newly installed nodes in the network. On detecting a new node, its node type, sequence number in TDMA slots, and other node parameters are read. These parameters are then inserted into respective arrays at the location specified by the sequence number. Newly calculated sleep schedule is appended and affected in a manner similar to that mentioned above. Refer to figure 2.

## 6.10. State Machine for Nodes

One of the important aspects of our network organization and protocol is to minimize the processing at the nodes required for enabling the communication of nodes with BS. The simple state machine running at each node is shown in figure 3. At power-on, nodes enter the receive state as it consumes lesser startup energy than that required by the transmit state. From the energy model given above and for typical radio specifications [11] we found that it takes 69.78 % more energy to startup the radio in transmit mode than that in receive mode.

In receive state, the node looks for commands from the BS. In set up state, the node sets up various parameters like active channels, sweep rate, number of data points, sequence number, node type etc. In sleep mode, the node turns its radio off but keeps sensing the physical quantity. It comes out of the sleep mode in case of emergency or time out. In transmit state, the node transmits the data or other parameters desired by the BS.

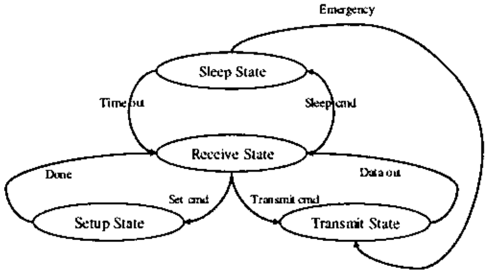

*Figure 3:* **State machine running on each node**

## 7. Implementation

We implemented most of the UC-TDMA protocol described above. We used 2 G-link sensor nodes, 1 SG-Link sensor node, 1 V-Link sensor node, 1 Base station, 1 Laptop and LabVIEW version 6.1 for the implementation.

All of the sensor nodes were physically installed at optimal locations in the Heating and Air-conditioning Plant at ARRI. It was found that sensors need to be tightly placed in order to get accurate measurements. Base station was connected to laptop using a 9-pin RS-232 serial connector. We created two separate GUIs viz. Network configuration wizard and an application GUI shown in figures 4 and 5 respectively.

The network configuration wizard is used to save all the node parameters in a separate configuration file which is used by the application GUI for setting up the network. More than one configuration file can be created and saved for later use. For each node following parameters are defined. 1. *Sweep rate,* it is the data sampling rate (one sweep represents one sample from all active channels). 2. *Number of Sweeps,* it is the number of data points transmitted each time node transmits (one data point represents one sample point from all active channels). 3. *Sequence Number,* it is the sequence at which a particular node will transmit. The node with sequence number two will transmit after the node with sequence number one and so on. 4. *Active Channels,* it is the number of channels that transmit each time a node

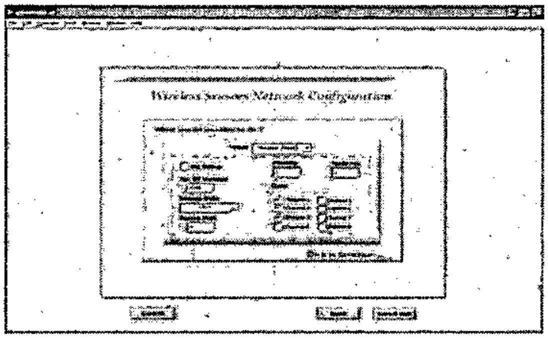

***Figure 4:* Screen shot of Network Configuration Wizard.**

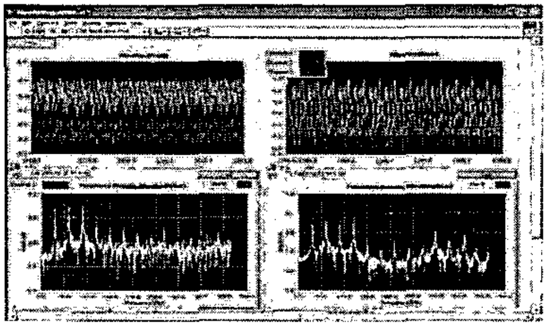

Figure 5: Screen shot of Application GUI

transmits. Each node can have maximum of eight channels. 5. *Node Number,* it is a 16-bit number which identifies the type of physical quantity being measured by any particular node.

The application program utilizes sweep rate, number of sweeps and sequence number arrays for creating the UC-TDMA frame. Sweep rate along with number of sweeps ascertains the time duration of TDMA slot for that particular node. Sequence number on other hand resolves the position of the slot in the frame. The application program communicates with nodes through the BS connected to the serial port. Figure 5 shows the time plots obtained from two G-link sensors in real time. The plots keep moving to left as new data keeps coming in real time. Below the time plots are their respective frequency plots obtained by taking FFT of data points transmitted by the senor node in one time slot. The FFT is thus updated after each time node transmits new data to the BS. These are thus time varying FFT plots of real time data acquired by the BS. These FFT plots thus represent instantaneous vibration frequency signatures.

The data acquired by each sensor over the time of network operation is saved in a data file selected by user. This data can be used by any other application program for further detailed data analysis using various tools like fuzzy logic, neural networks, etc. The network was operated in the continuous mode of the UC-TDMA protocol for gathering the data. The sensor nodes were configured for turn-wise transmission of 2000 number of sweeps at a sweep rate of 829 sweeps per second from three active channels on the node. Data transmission performance of the network was found to be satisfactory with 1.2 % data loss per time slot. This loss was determined by the checksum errors obtained at the base station. It was found that after a node finishes sending its data to BS, there is a little delay of the order of 100s of milliseconds before the next node in sequence starts transmitting. This delay is attributed to the time taken by the BS to generate CTS signal plus the time taken by the node in responding with data.

## References

[1] A. Mainwaring, et al, Wireless Sensor Networks for Habitat Monitoring, Intel Research, IRB-TR-02-006, 2002.

[2] E. Shih et al, Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks, in proc. ACM MobiCom'01, pp.272-286.

[3] E. Duarte-Melo, M. Liu, Data-Gathering Wireless Sensor Networks: Organization and Capacity, Preprint submitted to Elsevier Science, May 2003.

[4] F. Zhao, et al, Information-driven dynamic sensor collaboration for target tracking, IEEE Signal Processing Magazine, 2002, vol.19, pp. 61-72.

[5] G. Baweja, B. Ouyang, Data Acquisition Approach for Real-time Equipment Monitoring and Control, in Proc. IEEE/SEMI ASMC'02, pp 223-227.

[6] H. O. Marcy, et al, Wireless Sensor Networks for Area Monitoring and Integrated Vehicle Health Management Applications, Rockwell Science Centre, AIAA-99-4557, 1999.

[7] K. Shorabi, et al, Performance of a Novel Self-organization Protocol for Wireless Ad Hoc Sensor Networks, proc. IEEE VTC, 1999, pp. 1222-1226.

[8] LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE std 802.11-1997 edition, 1997.

[9] M. LaPedus, Intel to use wireless sensors for chip-equipment maintenance, EE TIMES, Oct. 2003.

[10] R. Kannan, et al, Energy and Rate based MAC Protocol for Wireless Sensor Networks, SIGMOD, vol.32, No.4, December 2003.

[11] RF Monolithic Inc., TR1000 Datasheet.

[12] S. Saha, et al, System Design Issue in Single-Hop Wireless Sensor Networks, proc. IASTED-CIIT'03.

[13] S. Singh, C. Raghavendra, PAMAS – Power Aware Multi-Access Protocol with Signaling for AD HOC Networks, in proc. ACM SIGCOMM '98, pp. 5-26.

[14] T. W. Carley, et al, Contention-Free Periodic Message Scheduler Medium Access Control in Wireless Sensor / Actuator Networks, in proc. IEEE RTSS'03, pp.298-307.

[15] W. Ye, et al, An Energy-Efficient MAC Protocol for Wireless Sensor Networks, in proc. IEEE INFOCOM'02.

[16] W. R. Heinzelmann, et al, Energy-efficient communication protocols for wireless microsensor networks, in proc., HICSS, 2002.

[17] W. B. Heinzelmann, et al, An Application Specific Protocol Architecture for Wireless Microsensor Networks, in IEEE Transactions on Wireless Communications, vol.1, No.4, 2002.