ReWINS: A Distributed Multi-RF Sensor Control Network for Industrial Automation

Harish Ramamurthy, Dhananjay Lal, B.S. Prabhu and Rajit Gadh,
Wireless Internet for the Mobile Enterprise Consortium (WINMEC),
Henry Samueli School of Engg. and Applied Sciences,
University of California, Los Angeles,
44-116S, 420 Westwood Plaza, Los Angeles, California 90095

Abstract

Remote "Monitor and Control" systems are being increasingly used in various areas including security, transportation, manufacturing, supply chain, healthcare, biomedical, etc. A unit system with hardware and software components, sans network support, for providing limited monitoring and control for industrial automation was proposed in our earlier work. In this paper, we extend that architecture to address the issues faced by large-scale wireless industrial automation - networking, communication architecture, modularity, extendibility and fault tolerance. The RF link of the wireless interface is reconfigurable to accommodate different RF modules (Bluetooth, 802.11, Zigbee, RFID) providing Over-the-Air (OTA) plug-n-play capability. The application integration platform maintains a component level description of the system and is interfaced to a spatio-temporal visualization tool, imparting flexibility for implementing complex systems, where nodes can be addressed individually or on a group/cluster basis. Experimental results of the reconfigurable wireless interface and simulation results of network organization and healing are presented.

1. Introduction

Today, industrial applications demand real-time access to information for making intelligent decisions. Current generation control and monitoring systems are capable of sharing information over the network and are being increasingly employed for real-time decision support. These applications require stringent reliability and low latency, translating to specific design issues for the underlying system.

Currently, wired infrastructure is used for communications, which is plagued by problems of high deployment and maintenance costs sometimes requiring marshalling cabinets and mandated redundant runs for critical operations. Experts estimate that around $200 is spent for every foot of wire laid in an industry [11]. Thus, wired infrastructure constrains the viability of any smart real-time system.

Intelligent wireless sensor-based controls have thus drawn significant attention recently. These modules can be designed to combine sensing, in-situ computation, and contact-less communication into a compact device and placed in remote unattended locations. Further, advancement in technologies and standardization of communication, availability of hardware built on COTS components etc., have made the use of wireless sensors a viable option for monitor and control. Already large-scale wireless sensor networks having different capabilities are being used to monitor real-time application needs.

Industrial applications typically employ different types of sensors (thermal, photo, magneto, pressure, accelerometers, gyros, etc.), often deployed within the same network, having different interfaces and supporting different protocols for data and communications. Formation of monitoring and control systems from such diverse sensor elements thus entails deployment of controllers, that understand different sensor protocols. In addition, the problems are exacerbated when different RF communication links have to be used for satisfying the requirements of bandwidth, delay, jitter, range, noise immunity, costs. etc., for communication.

[13] had proposed a generic reconfigurable wireless interface for interfacing such sensors and the formation of systems from them. It was demonstrated how a basic system comprised of a few sensors within a small control area region can be implemented. This paper extends [13] to address design issues concerning wireless-enabling of large-

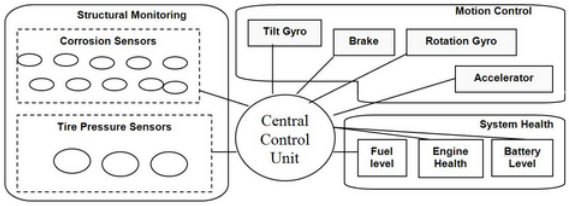

Figure 1. Application Scenario: Automotive Monitoring.

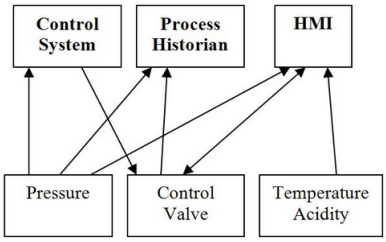

Figure 2. Application Scenario: Chemical Process Control.

scale industrial automation systems.

To illustrate these design issues more clearly, two sample application scenarios are presented below. One of them is an automotive sub-systems monitoring. As depicted in Figure 1, monitoring in an automobile involves the operational status and condition of number of sub-systems such as structural monitoring, system health, motion control, etc., involving data retrieval and processing from a high-density of sensors co-habiting in a small control area, to provide real-time decision support. Structural monitoring subsystem monitors the structural health; i.e. tire pressure, corrosion etc. The Motion Control sub-system is the most active/real-time sub-system that controls the key motion and navigation aspects. The System Health sub-system monitors the health of the peripherals, for e.g. battery charge level, fuel (gas and engine oil) level, etc.

The benefits of utilizing wireless communication such as space reduction, lower maintenance costs and flexibility of deployment can be suitably leveraged in this application. For e.g. the tire pressure, liquid level and corrosion can be sensed using low performance wireless technologies that are cheap and more power efficient like RFID or Zigbee, whereas highly active sensors like Encoder and Gyro need to be interfaced with high performance wireless technologies like Bluetooth or Wi-Fi.

A more demanding application is that of process control. To illustrate, a sketch of a small part of the process control system (liquid/gas flow control) is depicted in Figure 2. The sensors used are pressure, temperature and pH; the control valve is an actuator catering to three different applications. The control system maintains the pressure of the liquid, the process historian keeps track of the pressure, temperature and acidity and the Human Machine Interface provides a visual means to directly interact with the system. Unlike in the automotive system, here multiple sensors have multiple consumers of data putting an additional burden in terms of data reliability, consistency, synchronization, etc. Hence, a smart interface is required at the sensor end to manage the transmission of data to multiple sinks and avoid multiple queries being forwarded to the sensor (aggregates the queries). Interoperability among different applications can be achieved by using an open messaging architecture to communicate with the applications.

The organization of this paper is as follows. Section 2 covers related work on sensor networks, and specific initiatives for industrial automation. Application requirements of an industrial control network are presented in Section 3. Section 4 describes the intelligent wireless sensor architecture and Section 5, the developed networking architecture. Implementation details, snapshots, simulation and experimental results of the current implementation are presented in Sections 6 and 7. Finally Section 8 reports the conclusions on the research.

2. Related Work

2.1 University Research Initiatives

Early work in the field of wireless sensor networks was DARPAs military surveillance and distributed sensor network project, low-power wireless integrated micro-sensor (LWIM) and the SenseIT project. The Wireless Integrated Network Sensors (WINS) project and NIMS project [3, 8] at UCLA in association with the Rockwell Science Center, deals with ad-hoc wireless sensor network research, with a focus on building micro-electronic mechanical sensors (MEMS), efficient circuit design, and design of self-organizing wireless network architecture. These projects are oriented towards environmental and military applications, involving tens of thousands of nodes, and use non-standard RF communication technology.

The Motes and Smart Dust project [7] at UC, Berkeley focused on creating low-cost micro-sensors, with emphasis on the development of sensors and an embedded operating system, TinyOS. 'nesC' a product of this project have been utilized for sensor characterization. The project is node-centric, rather than system-centric.

The Pico-Radio [12] project at UC, Berkeley aims at developing a power-efficient wireless interface for the sensors, and a unified wireless application interface called Sensor Network Service Platform (SNSP), that abstracts the sensor

network and makes it transparent to the application layer, through a set of services.

Several initiatives like TinyDB [5], Cornells Cougar [14] etc. attempt to develop a declarative SQL-like language to query sensors and define certain standard query services. These solutions are applicable for environmental-monitoring where data is collected from thousands of sources. Similar research initiatives include MITs AMPS [4]. A detailed literature survey of various university research initiatives is available at [9].

Different protocol architectures for networking have also been proposed. The directed diffusion technique [6] deals with large-scale sensor networks where routing occurs based on the setup of gradients designed to draw event data based on interests. Geographic routing approaches use nodes locations as their addresses and forward data in a greedy manner towards them [2]. Other ways of achieving energy-aware routing have also been proposed in the literature [15].

Although there has been extensive research effort on ad-hoc wireless networks, the focus has been on developing low-cost wireless sensor interfaces and not much on actual application integration. The approach has been to develop wireless interfaces and protocols, which support the features/requirements for a particular class of applications (like military, environment sensing). There is almost no interoperability between the solutions.

2.2 Industrial Initiatives

The field of automation systems has continuously evolved- starting from early days of register level programming for data acquisition and point-to-point wired links for communication, to the current virtual instruments and DeviceNet, a communication paradigm for networking industrial systems through a common wired infrastructure. Developmental Efforts in this area can be broadly classified as:

- **Early Initiatives :** Included the design of industrial open protocols for communication also known as field buses like CAN, DeviceNet and ControlNet; proprietary system formation tools virtual instruments from National Instruments, Factory solutions from ABB etc. Further development involved open data exchange or messaging framework for e.g. OPC foundation which is trying to establish a standard data exchange standard so that interoperability among products (hardware and software) from different manufacturers is achieved [1].

- **Emerging Initiatives :** Industry believes that strong potential for wireless lies in enterprise-wide asset monitoring and maintenance on an open protocol for things to communicate with each other. ZigBee is touted as a promising technology for this to happen.

Industrial initiatives have put a lot of work on the system formation issues, but have been unable to exploit the advantages of wireless technology.

Deployment of wireless infrastructure in industries will occur incrementally and interoperability (between sensor-networks) and extendibility (different application needs) will form the requirements of prospective solutions. ReWINS research initiative is an attempt to develop such an end-to-end solution with support for incremental deployment through a transparent lower layer implementation and control architecture, and a user-friendly application interface.

3. Application Requirements

The sensor network that we envisage for an industrial automation scenario differs from the conventional definition of a sensor network, as considered in majority of the technical literature. These differences arise due to the nature/requirements of targeted application.

Reliability of the nodes and lower routing redundancy are key departures from the conventional view of sensor networks, when applicable to an industrial automation scenario. Another difference is that nodes, although power-efficient, often have access to power sources. Access to power sources facilitates in providing different levels of service for sensing - providing enhanced support for sensing and communication for time-critical data, or to be able to ignore this feature for non-time-critical data. Further, the physical locations of nodes can be manually logged even though the network may not have begun operation. For monitoring high-cost equipment, typically more advanced sensors with better communication capability (such as Wi-Fi) may be used.

Another difference in our sensor network paradigm is that routing is not data-centric, but is node-centric, as in a regular or ad hoc network, which lends individual addressability to different kinds of sensors, and allows specific predictable actuation. It also allows the nodes to have 4-byte IP addresses allowing remote monitoring and control from any location over the Internet. However, sensor network for industrial automation, differs from an ad hoc network in the following two ways: Firstly, it is application specific, i.e., tailored for performance with respect to fast and reliable delivery of control and actuation signals to a node or a group of nodes. Secondly, unlike an ad hoc network, the traffic is not accurately characterized as in a peer-to-peer manner. This is an important aspect of the network routing function.

The network structure is hierarchical with no need for peer-to-peer routing support at the lower levels, corollary

of the fact that an individual sensor node never needs to query another node or send an actuation command to it. Instead, data is collected from a certain region by an aggregator device that has custom-application dependent intelligence. Aggregators issue actuation commands, and may also provide a user-interface for taking geographically localized decisions by human intervention. It has been shown earlier, however, that there may be multiple data sinks for the same information. Hence, at the higher level of network hierarchy, there is in-built support for peer-to-peer routing. Thus aggregators, along with the central control unit, provide full peer-to-peer routing support amongst each other. Hence, the routing is a hybrid of a tree-based source routing (at nodes), and a full routing table maintenance at aggregators.

Focusing attention solely on an industrial automation application thus brings forth the following design issues:

- **Scalability :** Though the number of sensors/actuators etc. that need to be interfaced is sufficiently lower than the environment monitoring scenario (typically 20 high-end sensing nodes to 200 low-end sensor devices per aggregator, scalability still remains an important issue- especially since the network must be self organizing and self-healing. The idiosyncrasies of the different components of the system have to be carefully examined and considered. For e.g. consider Bluetooth can support a maximum of 7 connections per device. Thus, suitable and sophisticated networking techniques have to be applied for multiple sensors/actuators to co-exist.

- **Multiple Interface Requirement :** Typically the cost issues dominate here for e.g. corrosion sensors can use low performance wireless technologies like RFID. In addition, as performance and range present a tradeoff, it may be required that performance is sacrificed for range and vice-versa.

- **System formation :** A modular and hierarchical system formation technique enhances the system flexibility, robustness and reliability. Complex systems can be implemented and modified with little effort.

- **Fault Tolerance :** A certain level of service guarantee is required from the communication system as this directly affects the system reliability. This guarantee may be in form of a confidence level for a particular latency in a command/query message. Fault tolerance must be achieved in two differing domains- real-time and non-real time. Real-time fault tolerance accounts for route repairs when actual query or command messages are being sent over the air. Non-real time fault tolerance refers to in-built mechanisms in the network to detect node-failures and establish alternate routing topologies in the long-term, as well as notify the user of the node failure. The network protocol must provide recovery from both kinds of failure.

- **Interoperability :** As the deployment of sensor networks will happen in incremental stages, interoperability with existing legacy solutions is required. This can be achieved by using open and customizable message passing and network architecture.

- **Energy Efficiency :** In an industrial automation scenario, the primary design attribute is robustness of the network,, i.e., every node must be connected to the network, and be uniquely addressable, with a high-confidence level of low latency. Hence, energy-saving is not critical in network setup and organization. However, in the long-term operation of the network, periodic monitoring of the health of the network (i.e., good condition of the nodes) forms the largest proportion of the energy budget. Hence this function must be designed with a view to minimizing energy.

Further, the entire sensing system should have the functionality of upgrade or update with minimal effort and intrusion.

4. ReWINS Node Architecture

ReWINS system consists of a network of sensors, actuators and aggregators communicating with the central control unit using standard RF-links. The basic scenario is shown in Figure 3. 'D' represents the devices, which can be sensors or actuators and 'A' is the aggregator. For the sake of nomenclature, we term a sensor to be any kind of transducer which is capable of exchanging information (electrical signals) and similarly, an actuator is any kind of device which accepts data (electrical signals) and performs a measured action. Sensors and actuators are referred to as devices henceforth. The main function of the aggregator is to form an ad-hoc network of sensors under its purview and collect the data and signals from/to the devices and to transmit it back to the CCU using a backhaul link. Detailed description of the various modules follows.

4.1 Node Hardware

The hardware architecture of the wireless intelligent data collection device from [13] has been extended to support more number of sensors. Recognizing that most of industrial automation applications use serial data for communication, support for serial channels has been added. The modified hardware design is shown in figure 4. The current implementation can support several analog input/output channels (with sampling rates upto 200ksps), serial channels

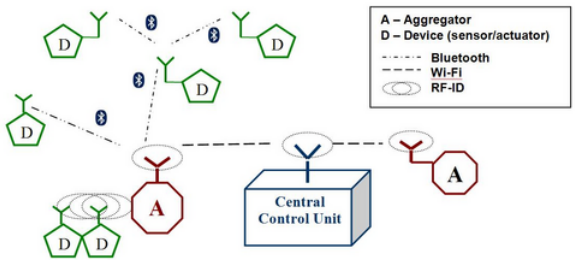

Figure 3. System Overview.

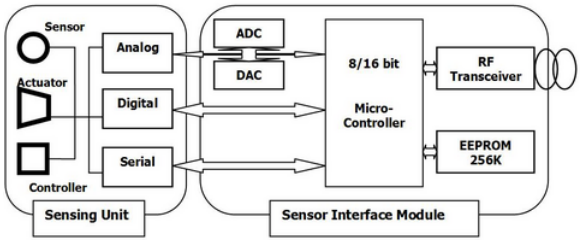

Figure 4. ReWINS Node Hardware Design.

(I2C, UART and Microwire) and digital/interrupt driven channels.

The device is composed of two components: a Sensor interface and an RF interface. A microcontroller acts as an intelligent interface between the sensor or the data collection unit and the RF module. It handles tasks related to data collection, data processing and wireless transmissions. The RF transceiver communicates with the aggregator or CCU over the RF-link.

The device is based on an interactive-flexible-modular-plug-n-play architecture. As ReWINS is modular in design, each module can be replaced without disturbing other modules significantly. Further using the capability of reconfigurability the system can be updated (sensor run-time parameters, microcontroller program variables, firmware/system software) over the air (OTA) without significant effort.

4.2 Node Software Architecture

In [13], the device remained in one of two modes command mode and data mode. The command mode was for configuring the device and the data mode for actual data transmission. The design was appropriate and efficient for raw data communication for small systems, but was restrictive in terms of flexibility and infeasibility of large scale deployments. Further the design in [13] treated the RF and sensor interfaces separately and allocated particular peripherals for each interface (at the firmware level) for e.g. UART to RF and analog/digital channels to sensor. As pointed out earlier, a lot of industrial sensors (specifically chemical sensors) and communication systems (fieldbus,

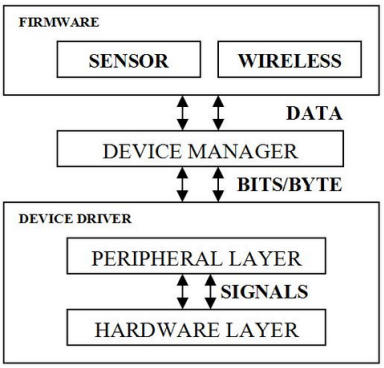

Figure 5. ReWINS Node Software Design.

IEEE 1451) use serial channels like I2C, microwire, field bus for communication and thus in our design the hardware peripherals have been separated from the firmware implementation.

We now briefly explain the different layers of the software stack (Figure 5):

- **Hardware layer :** Maintains hardware implementation details of the device for e.g. the mapping of I/O pins to peripherals, making the system hardware independent. The hardware design can hence evolve continuously with the software design.

- **Peripheral layer :** Manages different peripherals like ADC, DAC, and serial channels, etc. It implements peripheral specific code and facilitates access to the peripherals for higher layers through a set of pre-defined functions.

- **Device Manager :** Helps convert raw data into information and packages it or vice versa (for commands). It controls access to peripherals and provides a multi-threaded runtime environment through interrupts. It exports a component-based model for peripherals, which can be easily manipulated using scripts/visual tools.

- **Firmware :** Final applications are written on this layer this includes the sensor/actuator and RF implementations.

5. ReWINS Networking Architecture

Network establishment and communication between the nodes and devices at each of the two hierarchical levels is described in this section.

5.1 Inter-Aggregator Communication

An aggregator device has a preconfigured MAC address. The IP address is acquired dynamically from the central control unit. Addressing at the network layer occurs through IP addresses. The aggregator nodes maintain peer-to-peer routing capabilities through distance-vector routing tables.

5.2 Sensor Node Communication

The sensor nodes organize to form a tree topology rooted at an aggregator. The rationale being that all major decisions must necessarily be taken by aggregators or the CCU. However, low-level or routine decisions that involve single-parameter threshold matching can be taken at the node itself, and information about these actions must be propagated to the aggregator, albeit with low priority.

Tree formation occurs in the following manner for Zigbee enabled mica motes or Wi-Fi and Bluetooth.

- **Wi-Fi/Mica radio motes :** The aggregator (root of the tree) initiates topology formation. It sends a broadcast message, which is replied by its neighbors (with their MAC address) that accept it as their parent. The aggregator then assigns a network address and individually propagates this address to each of its children. The nodes acknowledge and accept their network address. Random backoff along with carrier-sensing (CSMA/CA) is employed at the nodes in replying to a prospective parent to avoid collisions. In assigning the network address to a child node, the parent uses a path-encoded scheme, which shall be explained shortly. The same technique is then used by the children to establish further communication with their children. Each child also propagates the MAC-to-network address mapping of its children to the aggregator. This process initializes the network. In the second phase, specific nodes are contacted by aggregator commands to connect them to the network.

- **Bluetooth :** Initially, the aggregator operates in master mode (i.e., inquiry), while all other nodes are in inquiry-scan, so they are prospective slaves. Once the master establishes a piconet (after inquiry and paging), the newly connected children of the master initiate the formation of a scatternet. Note that once again a path-encoded scheme is used by a master to assign network addresses to its children, and these addresses are propagated to the masters parent until it reaches the aggregator. In this case, although the network addresses are encoded, they are translated into 4-byte IP addresses for transmission. Thus, when a node gets connected to the tree topology as a child, it starts time-sharing between the current piconet (as a slave), and the formation of a new piconet (as a master) to further extend the tree topology.

5.3 Network Setup: Path-Encoded Addressing for Sensor Nodes

Network addressing within the sensor network is based on Class E network addresses, that are reserved for future use, and are not part of the Internet. Class E Internet addresses are characterized by 1111 as the most-significant bits of the 4-byte IP address. Aggregators have a unique 4-bit string associated with them. This allows individual addressing of at most 16 aggregators in the whole network, thereby permitting a reasonably large number of aggregators. The remaining 3-byte sequence of the address denotes a path from the aggregator to the node. This is assigned in a manner explained below. The method of addressing is similar in philosophy to source routing, however, unlike the source routing method a single network address of a node is sufficient to completely characterize the routing for communication between the node and the aggregator. All routing addresses, therefore, are assigned with respect the nodes aggregators. Nodes only maintain physical-network address information about their parent and children, minimizing routing information communication and maintenance.

Consider level 1 beyond the aggregator (level 0) in the topology tree. The aggregator may use n bits in addressing its children, thus providing a unique path to each of them. For example, it may assign its children unique path encodings of 001, 010, 011 etc., given that it uses 3-bit encoding per level of the tree. Note that address 000 is reserved. The encoding 000 in routing denotes that the destination has been already reached, i.e., a node has all 0s encoding beyond its unique path address string to indicate that no further parsing and message propagation needs to done. All further levels in the tree topology may also be uniquely addressed by bit-strings. The concatenation of all level strings lends a unique network address to a node in the topology. With a 21-bit string it is possible to address 7 different levels beyond the root, with 5,4,3,3,3,2,1 bits for each level respectively. This aspect of the routing design is flexible, and may be changed based on the node density and projected latency.

5.4 Network Recovery: Node Failure

When a node failure occurs in real-time query/actuation, then the parent sets the 3 least significant error bits (default 000) to the unreachable level, and sends a broadcast packet. All nodes that observe a packet to have a node-unreachable identifier, and lie in the tree topology at a level that is less than or equal to the level at which the node is unreachable,

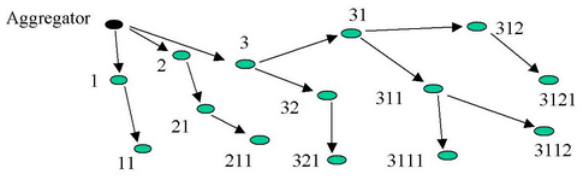

Figure 6. Path-Encoded Addressing at Sensor Nodes.

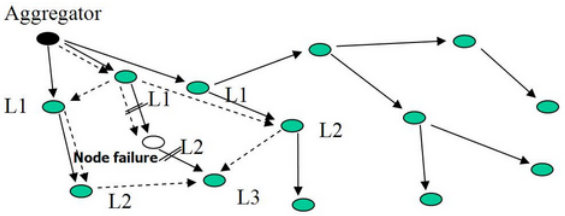

Figure 7. Network Recovery from Node Failure.

rebroadcast the packet. In this manner, a localized temporary route repair is attempted for packet delivery. This is illustrated in Figure 7.

The network performs periodic health monitoring by propagating messages from the root/aggregator to the nodes, and back-propagating messages from the nodes to the root. In summary, nodes have the capability of addressing sensed information to the aggregator, while aggregators have the capability to address any node for query or actuation. Node failures are repaired by efficient broadcast in real-time, and then by solicit messages in non-real-time.

6. Implementation

In the current version of the system different types of sensors, viz., rotary and linear have been interfaced. Bluetooth, Wi-Fi and Mote have been used for the RF communication link. Work on supporting other RF technologies like UWB and Zigbee is currently being done. In the current implementation, as the aggregator is just a logical component, the aggregator resides on the CCU itself. A snapshot of the partially assembled ReWINS node is shown in Figure 8.

The present implemented interface can support devices of different categories such as encoders (absolute/incremental), brushless motors, gyro sensors, linear position sensors, etc. Various industrial systems were implemented using the sensors and the devices, and the modular functionality was tested out. One application tested out was the gyro-motion control. The system was implemented using a gyro coupled with the motor-encoder system. The

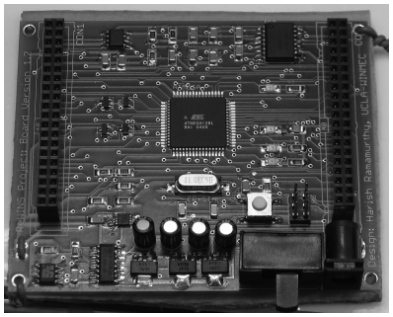

Figure 8. ReWINS Node Hardware, version 1.2.

Gyro sensed any tilt and passed the information to the CCU which instructed the motor-encoder system accordingly.

7. Results

7.1 Wireless Interface Performance Experiments

Delay and bandwidth have a significant effect on the fidelity and responsiveness of the system. To characterize the parameters, experiments were performed in an echo-scenario, i.e. whenever the CCU sends a packet to the device; the device simply echoes back the packet. Measuring the delay from start of transmission from CCU to end of reception at CCU gives the round-trip-delay of the link Bandwidth is measured by the data rate. Experiments for bandwidth testing have already been reported in [13]. In the current set of experiments we characterize the delay for both WiFi and Bluetooth interface specifically considering the effects of distance, packet burst size and traffic. We start by presenting the experimental results for each wireless interface individually, in an indoor laboratory environment.

Though the amount of data transmitted in one shot (i.e. packet burst) in industrial applications is of the order of few bytes but this data needs to be communicated in real-time. The delay performance of Bluetooth with varying packet burst size is plotted in Figures 9 .

From Figure 9, we see that though the delay increases by a small factor, the packet burst does not have a considerable effect on the performance of Bluetooth. Further, presence of traffic has no considerable effect on the performance of Bluetooth. In summary the delay of Bluetooth can be characterized by:

- Distance: The performance worsens with increasing distance and exhibits more jitter [13].
- Competing Traffic: It has no considerable effect on the performance (while working at a baud-rate of 115K).

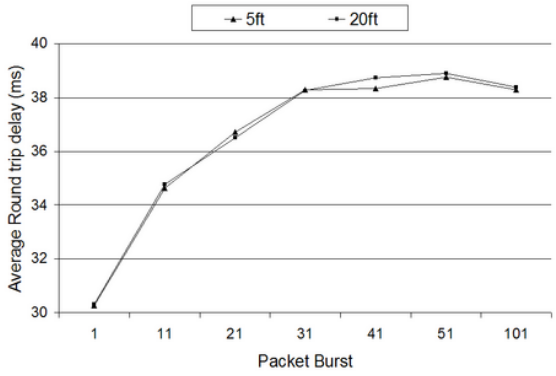

Figure 9. Variation of Delay with Packet Burstiness for Bluetooth.

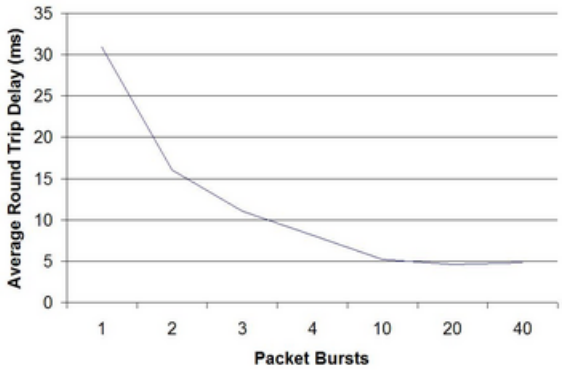

Figure 10. Mean Round Trip Delay with Packet Burstiness for Wi-Fi (5 ft).

- Packet Bursts: Mild effect with performance degrading with more packets per burst

Packet bursts were varied and per packet delays were calculated for varying distances for Wi-Fi and are presented in Figures 10 and 11. As can be seen, the per packet delay decreases as the packet burst is increased. However, the jitter (standard deviation) increases as can be seen in Figure 11. These peculiarities arise due to the nature of medium access control on Wi-Fi, which is not based on time-slotting, but is achieved through competition for the channel. In this mechanism, a node tends to maintain control over the channel once it gains access.

As can be seen from Figures 12 and 13, competing traffic is a key factor in determining performance, specifically when distance increases.

Summary for Wi-Fi

- Distance: Performance degrades with distance, delay increases and becomes jittery.

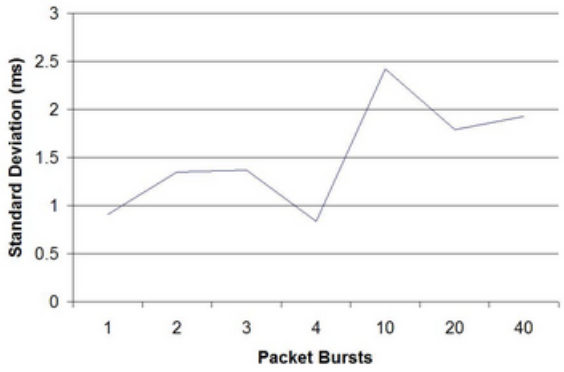

Figure 11. Standard Deviation of Round Trip Delay with Packet Burstiness for Wi-Fi (5ft).

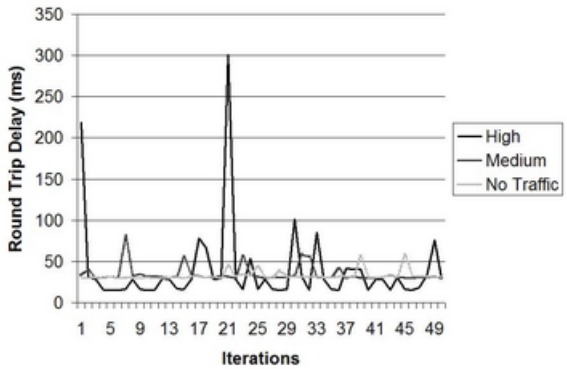

Figure 12. Variation of Delay in presence of Competing Traffic for Wi-Fi (20ft).

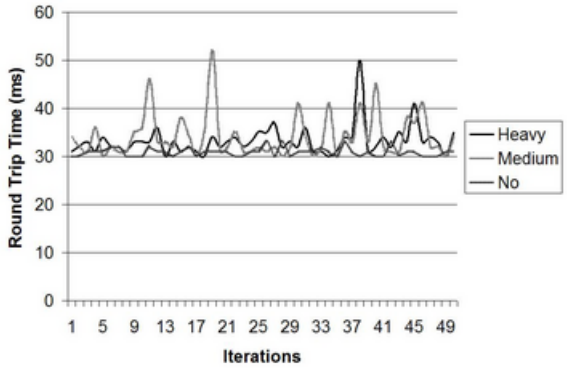

Figure 13. Variation of Delay in presence of Competing Traffic for Wi-Fi (5ft).

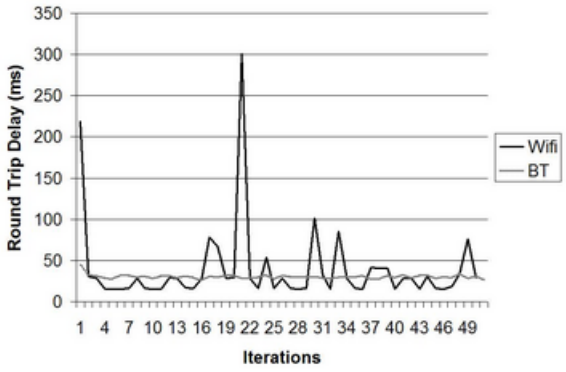

Figure 14. Round Trip Delay Comparison for Wi-Fi and Bluetooth (20ft).

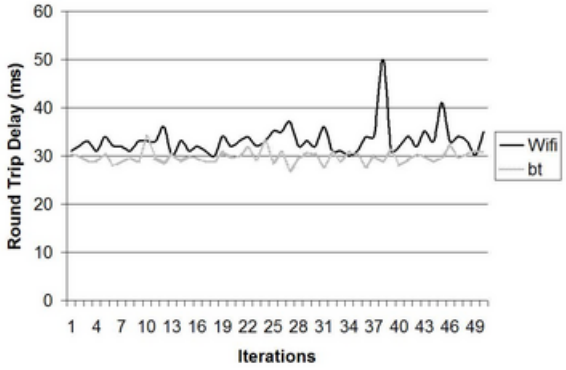

Figure 15. Round Trip Delay Comparison for Wi-Fi and Bluetooth (5ft).

- Traffic: Performance worsens with increasing traffic. The effect is more pronounced at larger distances.
- Packet Bursts: Bigger payloads experience less per-byte delays

Each factor enlisted above degrades the performance individually, i.e. heavy traffic with smaller packet bursts will perform worse than medium traffic with smaller packet bursts.

The variation in round trip delay for both Wi-Fi and Bluetooth have been plotted for various distances in figures 14 and 15.

Thus from the experimental results one can conclude that Bluetooth seems to perform better in industrial automation scenarios where limited bursts of data need to be delivered in real-time in a noisy environment and Wi-Fi seems to perform better in scenarios where a huge amount of data needs to be transmitted in a less noisy environment. Due to space constraints not all experimental results have been reported. Readers are encouraged to refer [13] for detailed results.

7.2 Simulation of Network Organization

In order to gauge the performance of network organization, we have simulated a network of nodes in Parsec, a C-based discrete-event simulator [10]. A rudimentary CSMA/CA based MAC protocol based on exponential backoff has been applied to each node to test the overlying network layer protocol suite. To separate the MAC layer functionality from the network layer, the mean of the exponential distribution for backoff has been set to a reasonably large value, and is related to the packet size as

$$BO_{mean} = 5 * n * PKT\_SIZE \quad (1)$$

Here n is the average number of nodes in a neighborhood, obtained roughly by taking the reciprocal of the probability that a particular node lies in an area of interest (based on uniform distribution of nodes over area). In this manner, we ensure low channel traffic and isolate the network functions to some extent. Figure 16 illustrates the results obtained on self-organization, with respect to scalability for three example scenarios. It shows the total number of nodes connected, as well as the percentage connected when nodes are uniformly distributed over a large area. The results shown are averaged over 10 random topologies. The nodes have a transmission range of 50m. In one scenario, 50 nodes have been distributed over a 200m x 200m area, and a single broadcast is used by a node for soliciting children. In the second case, three broadcast messages are employed by a node, for the same topologies. In the third case, 120 nodes are strewn randomly over a 230m x 230m area. Here, each node uses 7 broadcast attempts for soliciting children. Connectivity for a node is defined as the dual process of a node accepting a child solicitation, receiving a valid network address, and then back-propagating this network address to the control unit (aggregator) so that its network address is added in the already existing physical address and location log. Thus, the node is completely addressable by the aggregator.

We observe that for the 50 node case, close to 85% connectivity is obtained with 3 broadcast attempts, while for the 120 node case, 76% connectivity is achieved with 7 broadcasts per node. Thus, our protocol scales reasonably well when the number of nodes exceeds 100. However, the number of broadcasts needs to be increased for scaling of connectivity when the hidden terminals are more. This scaling is also non-linear. Also, for each level of the tree topology, there is an extra overhead both in forward broadcasting and reverse propagation, hence, the protocol does not scale linearly when the number of nodes are increased to cover a larger area while keeping the node density constant.

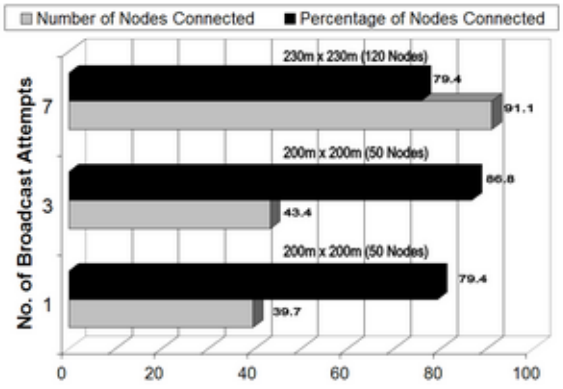

Figure 16. Scalability Performance of Network Self-Organization Protocol.

These results highlight the importance of a mechanism that is needed to bring connectivity to levels greater than 95%. An aggregator pointedly directs some nodes to attempt to connect to unconnected nodes (based on physical location mapping) in phase 2 of network organization. The network failure recovery has also been tested, and is found to yield satisfactory results in all cases except for nodes with a very sparse neighborhood density.

8. Conclusions

In this research, we have proposed an end-to-end solution for wireless monitor and control in industrial scenarios. We extended our earlier work to include the networking and system formation aspects in the design. We have demonstrated a proof-of-concept working model of our solution and have successfully integrated a variety of sensors/actuators and formed intelligent systems by using the developed application interface. For illustrative purposes, we also discussed couple of application scenarios for such remote data collection systems. Experiments were performed to characterize the different wireless links (Bluetooth and WiFi) in terms of distance, packet bursts and traffic. The test results have been very satisfactory. The network setup and real-time features of our protocol have been tested through simulation. We have observed that our initial network organization approach scales well for the targeted number of nodes that are of interest for deployment in industrial automation.

References

[1] *OPC HDA Specifications*, Version 1.20.1.00 edition, 2003. Available for download at: www.opcfoundation.org.

[2] M. Corr. "Geographic Based Ad-hoc Routing for Distributed Sensor Networks", 2001.

[3] M. J. Dong, G. Yung, and W. J. Kaiser. "Low Power Signal Processing Architectures for Network Microsensors". In *Proceedings of International Symposium on Low Power Electronics and Design*, August 1997. WINS project online at http://www.janet.ucla.edu/WINS.

[4] W. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. "An Application-Specific Protocol Architecture for Wireless Microsensor Networks". *IEEE Transactions on Wireless Communications*, 1:660–670, October 2002.

[5] J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek. "Beyond Average: Towards Sophisticated Sensing with Queries". In *International Workshop on Information Processing in Sensor Networks*, March 2003.

[6] C. Intanagonwiwat, R. Govindan, and D. Estrin. "Directed diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks". In *Mobile Computing and Networking*, pages 56–67, 2000.

[7] J. M. Kahn, R. H. Katz, and K. S. J. Pister. "Mobile Networking for Smart Dust". In *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, August 1999.

[8] W. J. Kaiser, G. J. Pottie, M. Srivastava, G. S. Sukhatme, J. Villasenor, and D. Estrin. "Networked Infomechanical Systems (NIMS) for Ambient Intelligence". Technical report, Center for Embedded Networked Sensing, 2003.

[9] B. Krishnamachari. *Sensor Networks Bibliography*. ANRG, Univ. of Southern California. http://ceng.usc.edu/~anrg/SensorNetBib.html.

[10] R. A. Meyer and R. Bagrodia. *PARSEC Simulation Language User Manual*. UCLA Parallel Computing Laboratory. http://pcl.cs.ucla.edu/.

[11] L. G. Paul. "Cutting the Cord". *Managing Automation Magazine*, September 2004.

[12] J. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets, and T. Tuan. PicoRadios for Wireless Sensor Networks: The Next Challenge in Ultra-Low-Power Design. In *Proceedings of the International Solid-State Circuits Conference*, February 2002.

[13] H. Ramamurthy, B. S. Prabhu, and R. Gadh. "Reconfigurable Wireless Interface for Networking Sensors (ReWINS)". In *Proceedings of IFIP TC6, 9th International Conference, PWC 2004*, September 2004.

[14] Y. Yao and J. E. Gehrke. "Query Processing in Sensor Networks". In *First Biennial Conference on Innovative Data Systems Research (CIDR)*, January 2003.

[15] M. Younis, M. Youssef, and K. Arisha. "Energy-Aware Routing in Cluster-Based Sensor Networks". In *IEEE/ACM MASCOTS 2002*, October 2002. Available at http://citeseer.ist.psu.edu/younis02energyaware.html.