Politecnico di Torino

WIRELESS SENSORS NETWORKS

Final Project

Fco. Javier Osta Ardevines

08/03/2011

INDICE

Introduction

Today the sensors can be found in large number of systems and electronic devices. Most of these sensors lack the ability to process and analyze the data they detected, limiting itself to function like a transducer that performs the measurement of one or more environment variables and sends that information to a central processor. However, the researchers predict the arrival of a new generation of sensors, equipped with its own intelligence, able to organize themselves and to interface wirelessly with other fellows. Arises so-called Wireles Sensor Networks, WSN, consisting of ad-hoc macronetworks consist of many individual sensors that exchange information with each other wirelessly and through a protocol of pre-established communication.

In recent years, several research laboratories, and especially multinationals such as Intel, have bet heavily on this technology. It predicts that these networks will lead a technological revolution similar to the internet. There is talk of global surveillance network on the planet, capable of recording the habits of the people, make a monitoring about people and goods, monitor traffic, etc. Even if it will have to wait a few more years, it have arisen multiple initiatives and research projects interest and practical applicability. This paper will discuss some of these experiments that demonstrate the potential of this technology.

What is a wireless sensor network (WSN)?

Wireless sensor networks (WSN), are based on devices low cost and consumption (nodes) that are able to obtain information from their environment, process locally, and communicate via wireless links to a central coordinating node. The nodes act as elements of the communications infrastructure to forward messages received from nodes farther toward the focal point.

The wireless sensor network consists of numerous distributed devices spatially, using sensor to monitor various conditions at different points between including temperature, sound, vibration, pressure, motion or pollutants. Sensors can be fixed or mobile.

The devices are autonomous units that consist of a microcontroller, power energy (usually a battery), a radio transceiver (RF) and a sensor element.

Due to the limitations of battery life, the nodes are built keeping in mind the energy conservation, and generally spend much time at all 'sleeping' (sleep) of low power consumption.

The WSN are capable of self-restoration, ie failure of a node, the Web will new ways to route data packets. In this way, the network will survive as a whole even if individual nodes lose power or be destroyed. The capacity of auto diagnosis, self-configuration, self-organization, self-restoration and repair, are properties that have been developed for these networks to solve problems that were not possible with that they were not possible with the other devices.

Elements of a wireless sensor network (WSN)

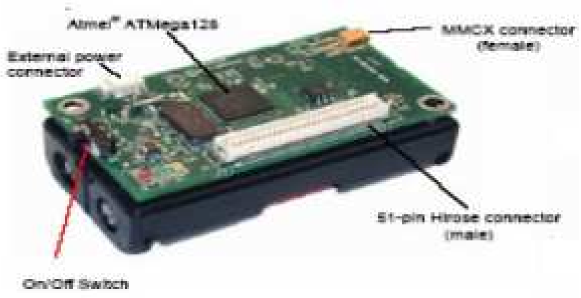
Two approaches have been taken. The first to integrate all components (sensor, radios and microcontrollers) on a single plate initiated by Moteiv Corporation. They have a lower production cost and are more robust in harsh environments or effects.

The second approach started by Crossbow Technology Inc. is to develop plate with transceivers that can be connected to the microcontroller board. This approach is more flexible. The nodes typically consist of a plate of sensor or data acquisition and a “mote or speck” (plate processor and transmit/ receive radio). These sensors can be communicated with a gateway, which has capacity to communicate with other computers and other networks (LAN, WLAN, WPAN…) and Internet.

In connection with the software they need, there are specific operating systems like TinyOS for embedded systems. Routing systems and security are fundamental structure of a wireless sensor network.

Research data system

The sensors are different in nature and technology. They take environmental information and converted into electrical signals. In the market-sensing plates measure many different, such as barometric pressure sensors, GPS, light, measure solar radiation, soil moisture, air humidity, temperature, sound, wind speed and many more.



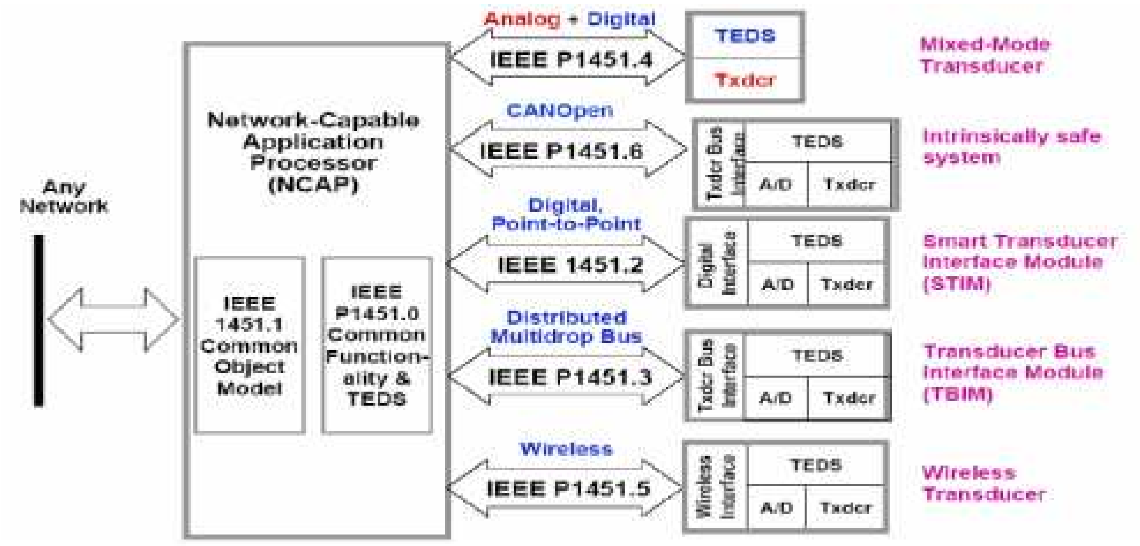
Examples: MTS300/310, sensor capable of detecting acceleration, light, microphone, sound, magnetometer, temperature, and the MTS420 sensor can detect temperature, humidity, luminosity, is sensitive to the light, contains a barometer.

Fig.1. MTS300/310

To control environments in real time systems, smart transceivers meet the IEEE 1451. Sensor standardization: A family of proposed standards. To define an interface for sensors and actuators, which is independent of protocols communication network used. The transceivers are equipped with sensors and micro actuators controller that provides them with “local intelligence” and communication skills. Designed as interface between 802.11 (WiFi), 802.15.4 (Bluetooth) and 802.15.5 (ZigBee).

The figure shows the architecture of the IEEE 1451 standars.

- The point-to-point interface that meets the IEEE Standard 1451.2
- The distributed multi-drop interface that meets the IEEE standard 1451.3
- The Analogic+digital interface that meets the IEEE standard 1451.4
- The wireless interface that meets the IEEE standard 1451.5 (WIFI, Bluetooth, ZigBee)
- The CanOpen interface that meets the IEEE standard 1451.6



Motes

Motes equipped with processing and communication to the sensor node. Processors radio take sensor data through its data ports, and send the information to the station base.

Typical components are:

Batteries
A CPU.
Flash Memory
Separate data memory programs
A plate of sensors: light, humidity, pressure, etc.
Radio to communicate with other motes.
ADC: analog-digital converter

They are resistant to weather and inhospitable terrain and able to run an application.

Gateway

They permit the interconnection between the sensor network and a TCP/IP. Example: MIB600Ethernet.

(TCP/IP) Network gateway which in turn serves as a programmer with Ethernet connection that we can connect from a PC.

Base Station

Data collector based on a common PC or embedded system.

Parameters of a WSN

The core values that characterize a wireless sensor network are:

- Lifetime
- Network Coverage
- Cost and ease of installation
- Response time
- Accuracy and frequency of measurements
- Security
- The core values that characterize the sensor node are:
- Flexibility
- Robustness
- Security
- Communication skills
- Computing Capacity
- Synchronization Facility
- Size and cost
- Energy cost

Architectures

The modular design is necessary in order to reuse the items. However, being involves modular design limitations and has to be careful to ensure that the interfaces between modules, hardware and software requirements are general enough to allow portability expected.

There are two architectures:

Centralized architecture in which nodes communicate only with the gateway and the others it is distributed architecture in which sensor nodes communicate only with other sensors in the same range.

Another aspect is the distributed computing where nodes cooperate and implement algorithms distributed to obtain a single global measure that is responsible for coordinating node communicate to the base station. The nodes not only capture the information but also use their computing capacity to developed measures.

Standard and proprietary wireless technologies for sensors Wireless

The most popular wireless standards are for the IEE 802.11b LAN (“WiFi”) for PAN, the IEEE 802.15.1 (Bluetooth IEEE 2002) and IEEE 802.15.4(ZigBee IEEE 2003).

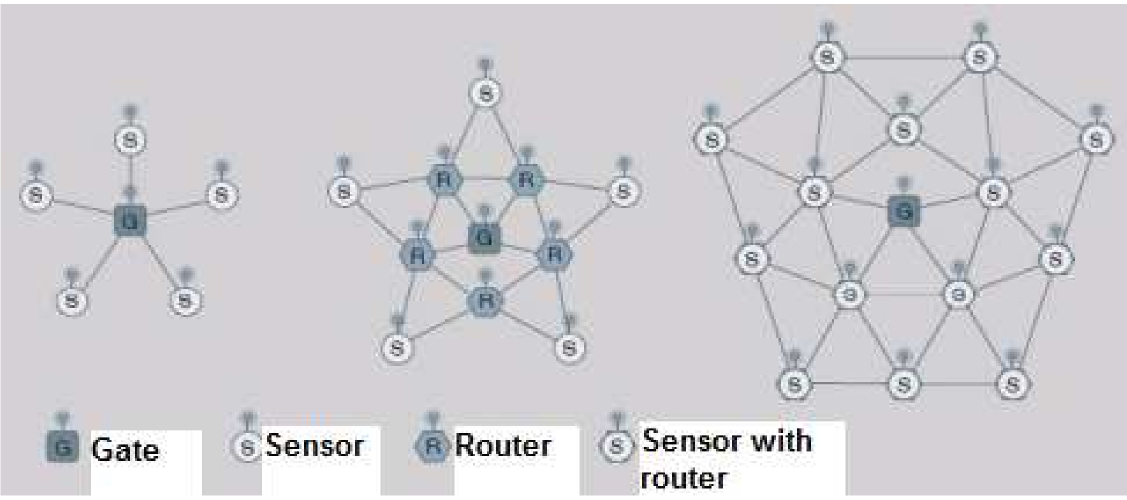
They use the ISM bands (Instrumentation, Scientific and Medical radiobands), 902-928 MHz (USA), 868 to 870 MHz (Europe), 433.05-434.79 MHz (USA and Europe) and from 314 to 316 MHz (Japan) and band of 2,400 GHz – 2.4835 GHz (universally accepted).

The current WSN are based on IEE 802.15.4 ZigBee 802.15.4, that which is more general WISA. Including multi-hop, which implies in what the messages uses different hops wave. Nodes are not assigned specific time intervals, but must compete to access the channel. This allows access more users to the wireless medium, but introduces uncertainty in the system, as the delay and consumption energy increase when a node is waiting for turn. Furthermore, intermediate nodes know the when they can be applied to route packets for others. It is therefore advisable to have intermediate nodes. ZigBee is ideal for active monitoring.

When the number of nodes to interconnect is very high, the networks solution is more than one level with different technologies (hybrid networks). Also it developed proprietary technologies (Crossbow Technology and Freescale Semiconductor). We understand that future developments should be based on standards.

Topologies

Besides the classic mesh network topology of WSN, there are others topologies. I.e. Topology star networks, wireless nodes communicate with a gateway device a bridge of communication with a wired network.



An emerging compromise is to have common WSN router devices that communicate with the gateway. The sensors only need to communicate point to point routers and can therefore be kept simple and power while improving the range and redundancy of the network itself.

Routing

The nodes have no knowledge of the topology of the network must discover it. The basic idea is that when a new node to appear on a network, announces its presence and listen broadcast of its neighbors. The node is informed about the new nodes to its scope and way of moving through them, in turn, can announce to the other nodes that can be accessed from a time, each node knows which nodes have around and one or more ways to achieve them.

Routing algorithms in wireless sensor networks must meet the following standards:

- Maintain a routing table reasonably small.
- Choose the best route to a given destination (either the faster, more reliable better capacity or the route of least cost).
- Keep the table regularly to update the fall of nodes, their change of position or appearance.
- Require a small number of messages and time to converge

Routings Models

There are several types of routing protocols.

Direct Broadcast Protocol (one-hop):

This is the most simple and direct communication is. All nodes in the network transmitted to the base station. It is an expensive model in terms of energy consumption and infeasible because nodes have limited transmission range. Its transmissions can’t always catch the base station, have a maximum distance of radio, so the communication directs is not good solution for wireless networks.

Multi-hop model (multi-hops):

In this model, a node transmits to the base station to resend the data to one of its neighbors Wireless Sensor Network which is closer to the base station, while he sent to another node closer to you get to the pot base. Then the data travels from source to destination hop by hop from one node to another until it reaches the destination. Given the limitations of the sensors is a viable approach. A large number of protocols use this model, including all Multihop Tmote Sky and Telos: multihop LQI, Miniroute.

Schematic model based on clusters:

Some protocols use optimization techniques to improve the effectiveness of the previous model. One of them is the aggregation of data used in all routing protocols based on clusters. A schematic approximation breaks the layered network clusters. Nodes grouped into clusters with a head, the head of router from that cluster heads other clusters or base station. The data travels from one cluster to lower layer coating higher. While jumping from one to another, he is doing from one layer to another, so it covers more distances. This means that, in addition, the data is transferred faster to the base station.

Theoretically, the latency in this model is much lower than that of multi-hop. Creating clusters provides an inherent capability of optimization of cluster heads. Therefore, this model is better than previous networks with large number of nodes in a large space (about thousands of sensors and hundreds of meters away) distances. This means that, in addition, the data is transferred faster to the base station.

Data centric protocols:

If we have a huge number of sensors, it is difficult to identify which sensor we get a data, in a certain area. One approach is that all sensors send data that they have. This causes a great waste of energy. In this type of protocol, data is request a waiting area and referred to it. The nodes of the trading area including more valid information. Only this is sent, thus saving energy.

Location-based protocol

It exploits the sensor location to route data over the network.

Operating Systems

There is a wide range of operating systems for microcontrollers. The main examples are listed below.

Bertha (pushpin computing platform)

A software platform designed and implemented to model, test and deploy a distributed sensors many identical nodes. Its main functions are divided into following subsystems:

- Process Management
- Management data structures
- Organization of neighbors
- Network Interface

Nut / OS

It is a small operating system for real-time, working with 8-bit CPUs.
Has the following function:

- Multithreading
- Synchronization Mechanisms
- Dynamic Memory Management
- Asynchronous Timers
- Serial Ports I / O

Designed for processors with the following resources:

-0.5 kBytes RAM
- 8 kBytes ROM
- 1 MIPS CPU speed

Contiki

It is a freely distributed operating system for use in a limited type of computers, from the 8-bit microcontroller embedded systems including wireless motes.

eCos (embedded Configurable operating system)

Is a free operating system, real time, designed for applications and embedded systems they only need a process. You can set many options and can be customized to meet requirement, offering the best real-time performance and minimizing hardware requirements.

EYESOS

It is defined as an environment for Web-based desktop, allow you to monitor and access a remote system using a simple search engine.

MagnetOS

It is a distributed operating system for sensor networks or ad hoc, aimed at implementing network applications requiring low power consumption, adaptive and easy to implement.

MANTIS (Multimodal Networks In-situ Sensors**)**

CORMOS (Communication Oriented Runtime System for Sensor Networks**)**

Specifically for wireless sensor networks as the name suggests.

TinyOS

The TinyOS operating system multitasking is a small group, useful for small devices such as motes. It is an operating system “event-driven” means it works produced from events that call functions. It has been developed for sensor networks with limited resources. The TinyOs development environment directly supports the programming of different microprocessors and allows you to program each with a unique identifier to differentiate, or what is the same thing can be compiled on different platforms by changing the attribute.

TinyOs system libraries and applications written in nesC, a version of C that was designed for programming embedded systems. In nesC, programs are composed by components that are linked to form a complete program.

The components are linked through their interfaces. These interfaces are bidirectional and specify a set of functions that are implemented either by vendors or by those who use it. NesC expect the code to be generated to create a program with a executable containing all the elements thereof, as well managers of disruption of higher-level programs.

TinyOS has the following characteristics:

- Small kernel footprint (executable footprint OS) between code and data 400 bytes.
- Component-based Architecture.
- Abstraction Layers well established clearly limited level of interfaces, while those components can be represented by diagrams automatically.
- Extensive resources to develop applications.
- Tailored to the limited resources of the motes energy, processing, storage and bandwidth.
- Operations divided into phases (Split-phase).
- Directed by events (Event Driven) reacts to sensors and messages.
- Concurrent event-based tasks and implementation in nesC.
- Perform services interfaces.
- Bidirectional interfaces commands and events.
- The provider implements commands.
- The events are implemented by the user.
- A module implements an interface.
- The components provide and use interfaces.
- A configuration refers to the internal and external interfaces.
- An application consists of a high-level configuration and all associated modules.

T-Kernel

Is an operating system which accepts the applications and instructions executable image basic. Therefore, no matter if is written in C++ or assembly language.

LiteOS

Operating system originally developed for calculators, but has also been used
for sensor networks.

FreeRTOS

Typically used for embedded applications, has the following characteristics:

- It uses lots of memory
- Any event in the hardware can run a task
- Multiarch (source ports for other UCP)
- Many have predictable response times for electronic events

Programming Languages

The sensor is difficult to programmed, among other difficulties is the limited capacity calculation and the amount of resources. And just as in traditional computer systems programming environments are practical and efficient to debug code, simulate, in these microcontrollers is still no comparable tools.

We can find language like:

- **nesC**: language we use for our spots, and it is directly related to TinyOS.

- **Protothreads**: specifically designed for concurrent programming, provides wireless two bytes as a base of operation.

- **SNACK**: facilitates the design of components for wireless sensor networks, especially when the information or calculation is very cumbersome to manage, complicated byNESC, the programming language makes it easier and more efficient. Then it is a good substitute for NESC create high-level libraries to be combined with more efficient applications.

-**c @ t**: initial computer kneel at a point in space in time (Computation at a point in space (@) Time).

- **DCL**: Language-ended distributed (Distributed Compositional Language)

- **galsC**: Designed for use in TinyGALS is programmed using a language task-oriented model, easy to debug,it allows competition and is compatible TinyOS modules NESCO.

Security

Security in these networks is not resolved. The term wireless is associated with " unreliable and unsafe". Because the technology in inherently insecure due tip its close relationship with physical environment (a sensor node can be easily accessible, and data are sent through a wireless medium), it is necessary to create security services to ensure the robustness and reliability of these systems, such as:

- Primitives and Infrastructure security key
- Self-monitoring systems (eg. systems auditing and IDS)
- Protocols for safe operation, etc.

Energy efficiency

The objective of energy efficiency is to maximize lifetime of the network while that the application meets its QoS requirements. The technological improvements that allow increasing battery capacity progress slowly. This means that efficiency energy will remain a challenge for such networks in the near future.

Node design for low power consumption means to choose low-power components. The first parameter to consider is the consumption of CPU power, sensor, radio transceiver and possibly other elements such as external memory peripherals during normal operation.

The choice of low power components typically involves accept compromises on the average yield. As a rule generally, allow power CPU operates in a tight loop clock, with less features on the chip to other units counterparts who consume more energy.

The optimization of energy consumption in the nodes for maximize the lifetime of the network, is an objective basics. The elements to consider are: Communication is the biggest consumer of energy. A distributed system means that some sensors need communicate over long distances, the resulting in increased consumption. So it is a good idea to process locally as much energy to minimize the number of bits transmitted.

CPU is able to be in "sleep" while "has nothing to do." Sending data from nodes can be of three forms: continuous mode at set intervals, led by events (sent when a certain condition is met,) and directed by consultation (only when request). There are also hybrid systems that use a combination of the above.

Integration of wireless sensor networks in environmental monitoring cyber infrastructure.

In current systems monitoring spatial coverage is limited by wires that connect them. In contrast, the spatial coverage and resolution of WSN can be conveniently configured to be meaningful to domain scientists.

Despite the limitations of traditional system, which has made a big investment, it is difficult to apply the wireless network system drastically, but it is important to carry out the change.

Many of the monitoring systems should be placed in areas which are difficult to reach. This supposed to set up a central sensor and a data-logger to store the data in its internal memory and always depends on human action for the data collection. With the WSN technology this is no longer necessary and you can handle real-time data to support time-sensitive environmental studies.

On the other hand the publicity available web services for sensor data visualization and dissemination is important also to make publicly available to benefit a broad range of entities, to enable data exchangeability and interoperability.

Overall system architectures

The new environmental monitoring system can be divided into four major layers as shown in Fig. 1:

Physical data layer

In this layer there are a variety of sensors that are used to measure environmental parameters. The sensor data are transmitted from monitoring site to the Central Data Collection (CDC) Server. It incorporates a wireless modem for telemetry, a single-board computer (SBC) Serves RFG, and a WSN for distributed soil moisture monitoring, remote field gateway (RFG) server provides effective control, management and coordination of two relatively independent sensor datalogger-based wired sensor system and the WSN-based wireless sensor system.

Logical data layer

Postgre SQL Database (DB) Server collects the data from the distributed monitoring stations. The central data collection (CDC) Server acts as an intermediate component of different physical layer devices and support data validation required by the DB Server. A Sensor Web Enablement (SWE) compliant data repository is installed to enable data exchange, accepting data from both internal DB Server and external sources trough the Open Geographic Consortium (OGC) web services.

Web presentation layer and user layer

The web presentation layer consists of a web portal where you can find: K-12 Education Outreach, Real time environmental monitoring, historical data download and modeling analysis synthesis. This can be implemented with a Sensor Observation Services (SOS) at this layer to facilitate data exchange. The SOS web service will be published to a catalog service in the OGC SWE framework to make it publicly accessible on Ineternet.

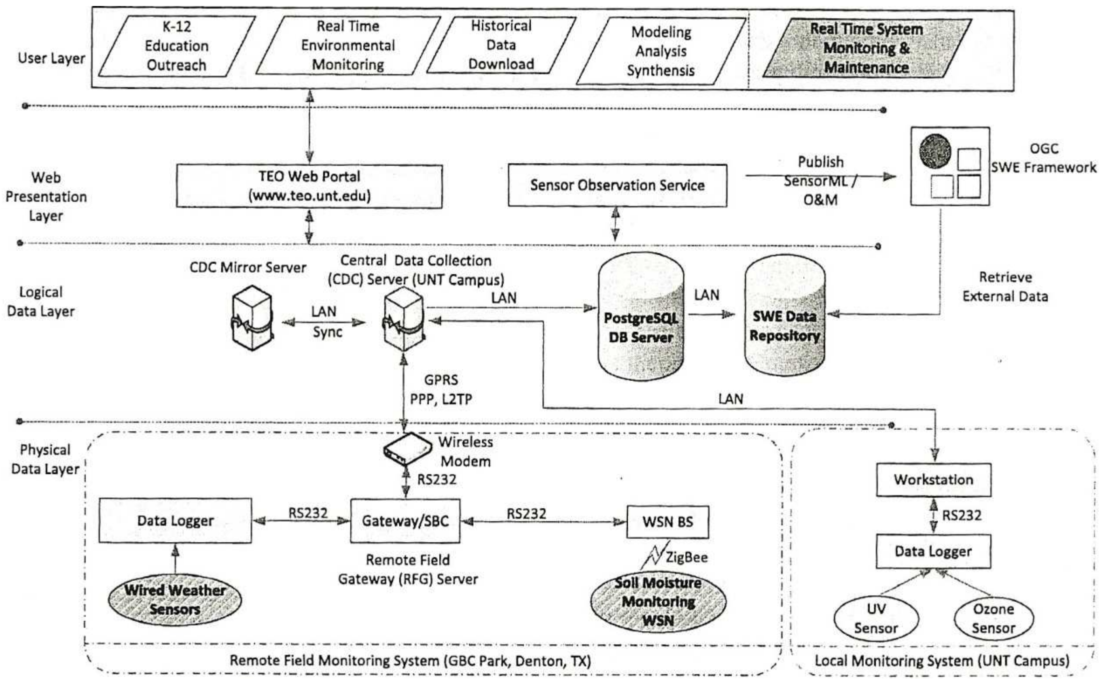


Fig. 1. System architecture of integrated environmental monitoring system.

System architecture of sensor nodes

The WSN hardware platform used in this current design is the IRIS mote from Crossbow Technology. The Base Station (BS) node transmits aggregated data to the remote field gateway (RFG) Server through serial port (RS232). To accomplish this without human intervention, these devices are equipped with a solar cells and rechargeable batteries.

In figure 2 we can see the functional block diagram of sensor node. In general, in environmental monitoring applications, every sensor node periodically carries out three main tasks, including data generation through sensing, data processing, and data reporting through multihop wireless communications. To perform the data generation tasks, sensor reading are collected periodically, which necessitates global time synchronization in the network. After in the data processing task, sensor nodes calibrate, aggregate, summarize, and compress the data. Finally, in the data reporting task data are transmitted to the BS node trough multihop wireless communications.

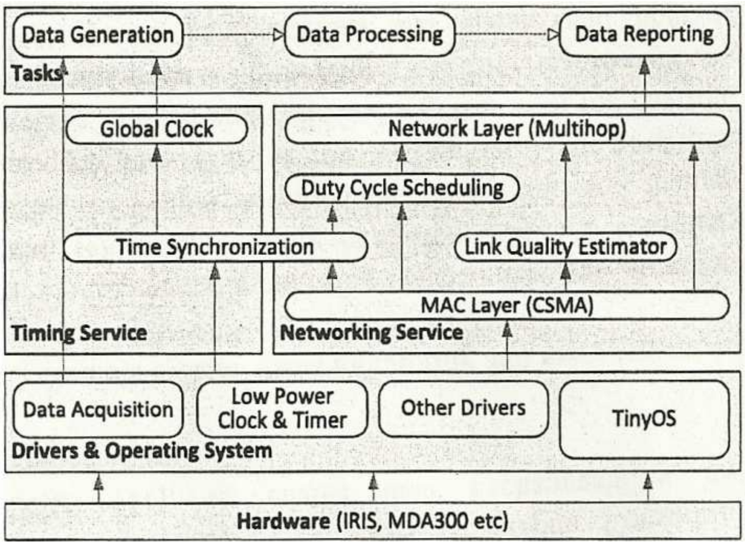


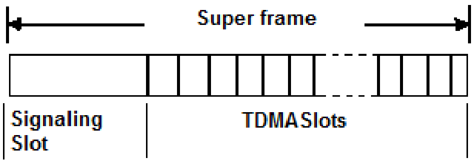
Fig.2. Functional block diagram of sensor node.

Networking protocols for multihop data collection

Energy efficiency is one of the major design considerations in environmental monitoring sensor networks. Therefore is important to choose a good design of networking protocols. In wireless sensor networks, Media Access Control (MAC) layer are broadly categorized in 2 groups, Schedule-based and contention-based.

A contention-based protocol is highly autonomous but relatively energy-inefficient while schedule-based protocol may eliminate overhearing and collision among neighboring nodes to store high energy efficiency, but it may suffer co-channel interference from other types of devices operating in the same frequency band.

In this system they developed a hybrid MAC layer protocol that integrates Carrier Sense Multiple Access (CSMA) and duty-cycle scheduling to achieve high energy efficiency to support long-term, low-rate and large-scale sensor networks applications. It uses a distributed duty-cycle scheduling algorithm to coordinate sensor nodes' sleeping.



Our protocol is divided in superframes and these in time slots.

Fig.3. Time slot structure of a super frame in the hybrid MAC protocol.

The super frame starts with a signaling slot where the nodes actively broadcast and received packets. Sensor nodes compete for Time Division Multiplexing Access (TDMA) slots and exchange control information with neighbors during this period. In many-to-one sensor data collection networks, all of the data packets are routed from child nodes to their parents.

With the TDMA slots we can guarantee that, there will be only one parent-child pair active in any two-hop neighborhood. This system is mainly designed to provide a collision-free channel forwarding such data packet. However other neighboring nodes within the vicinity of the parent-child pair can save energy by avoiding overhearing unnecessary packets. This is the biggest difference between our duty cycle scheduling protocol and most of the existing protocols.

Other protocol like B-MAC can save energy because it does not need time synchronization, however, this time synchronization can't be deleted in the system without requiring other components as well.

The flooding time synchronization protocol (FTSP) proposed in time-stamps synchronization messages at the MAC layer, with removes the non-deterministic delay at both sender and receiver caused by uncertain processing time in the operating system for context switches. This protocol can synchronize multiple receivers with a single broadcast message. In a tree-structucted network, if every node synchronizes to its parents, ultimately all of the nodes in the network could synchronize to the root to achieve global time synchronization.

In this experimentally study, it is observed that FTSP can achieve less than 1 ms timing errors in a three-hop network when the power management functionality is turned off. To save energy in low-power-modes, existing timer drivers pull down clock frequency at which repeat timers request when there are no active one-shot timers. Unfortunately, switching between high and low frequencies results in inconsistent time stamps. Therefore, had been developed a new two-layer time driver to replace the original drivers which employed two individual hardware clock to tackle the two types of timer separately. A high speed clock is used to drive one-shot timers. On the contrary, it allows speed and thus low power clocks runs continuously to support the repeat timers.

Although the data flow is unidirectional in our tree structure, the nature of the wireless structure makes difficult to maintain a reliable multihop routing hierarchy. Here we implemented a link quality estimator based on exponentially weighted moving average (EWMA) estimation method. The multihop routing protocol makes use of the link quality estimator to maintain a reliable routing topology.

Wireless Telemetry system

Wireless telemetry system hardware design.

To integrate a variety of devices in field is implement a RFG Server using a compact, rugged, ultra-low-power single-board computer SBC. The SBC provides a standard set of on-board peripherals and a software power consumption control for on-board peripherals. This saves energy by controlling between sleep and actives modes. The devices deployed in the field are commonly equipped with a RS232 serial port, including data loggers, wireless modem, and the WSN BS node. Thus, with five serial ports onboard, SBC is well suited to served as a gateway server

Wireless communication from field to CDC Server is implemented by using a General packet radio service (GPRS) modem. It is a packet-oriented mobile data service. These GPRS use Point-to-point (PPP) protocol. To be energy-efficient, the wireless modem is powered off during the system's sleep period.

The target supporting the system with a fully charged battery for at least a week without recharging, for this; has been implemented a power budget analysis system. Thus it can use solar energy with a large solar panel and a lead-acid rechargeable battery.

Remote data collection services

Our SBC deployed the full-feature Debian GNU/Linux. It is convenient to develop remote data collection services by taking advantage of the software packages that Debian provides.

The RFG Server wakes up periodically to carry out data collection services. The wireless modem is powered on at the same time as RFG server. Then, several independent data collection processes are started to poll data from the WSN BS node and dataloger trough RS232 ports. The data collected by the RFG Server is inserted into a local file system or directly CDC server through the wireless modem. The CDC Server then synchronizes its database to the RFG database.

The time of synchronization depends on the amount of data and traffic load condition in the network. We implement a simple duty cycle negotiation protocol between the RFG Server and the CDC Server to enhance energy efficiency of solar-powered remote monitoring system. As fig 4. shows, the RFG server and the CDC Server are protected from potential networks failures by timers T1 and T2, respectively; that is data collection process is terminated when the timers expired.

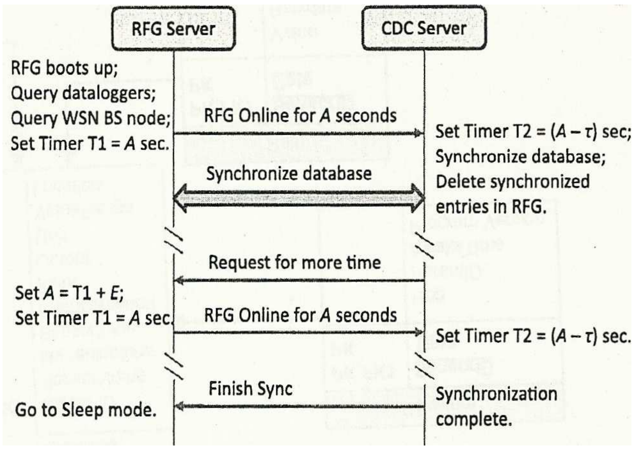


Fig.4. Duty cycle negotiation protocol between the RFG Server and the CDC Server.

Various system status data are also collected in the same way as sensor data to enable remote monitoring and management of the monitoring system deployed in the field.

Sensor data management

There are two ways to bring data to CDC server, or the CDC; it periodically connects to the data source and pulls the data, or opens the port and wait for the rod to be pushed from the data source. Or received data are stored in local file system. Later are tested according to a set of predefined validation rules, and sent to the PostgreSQL DB Server. Data handler may also require the RFG server to recollected and retransmit missing data packet.

Sensor data base design is driven by the emphasis on system extensibility because of the need to handle a large volume of data collected from heterogeneous sources in long-term operations. All sensor information is contained in one relation or table, whereas each observation is stored in a separate relation. An example of such database schema is shown in Fig 5

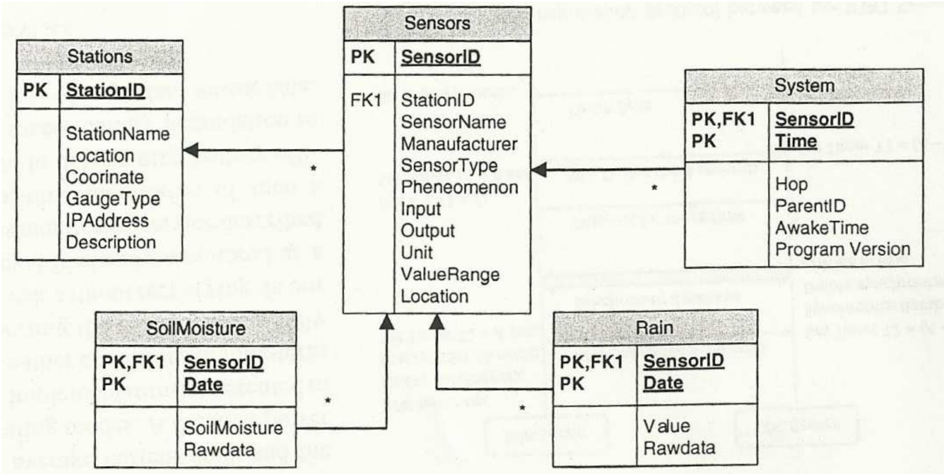


Fig.5. Illustration of the database schema.

Sensor data visualization and dissemination

The research was developed a dedicated web portal, Texas Environmental Observatory (TEO) Online, for the sensor data visualization and dissemination. To take the most of the flexible Google Maps APIs to associate sensor with their geographical locations intuitively as shown in Fig 6. Such an interfaces provides direct visualization of special distribution of sensors data.

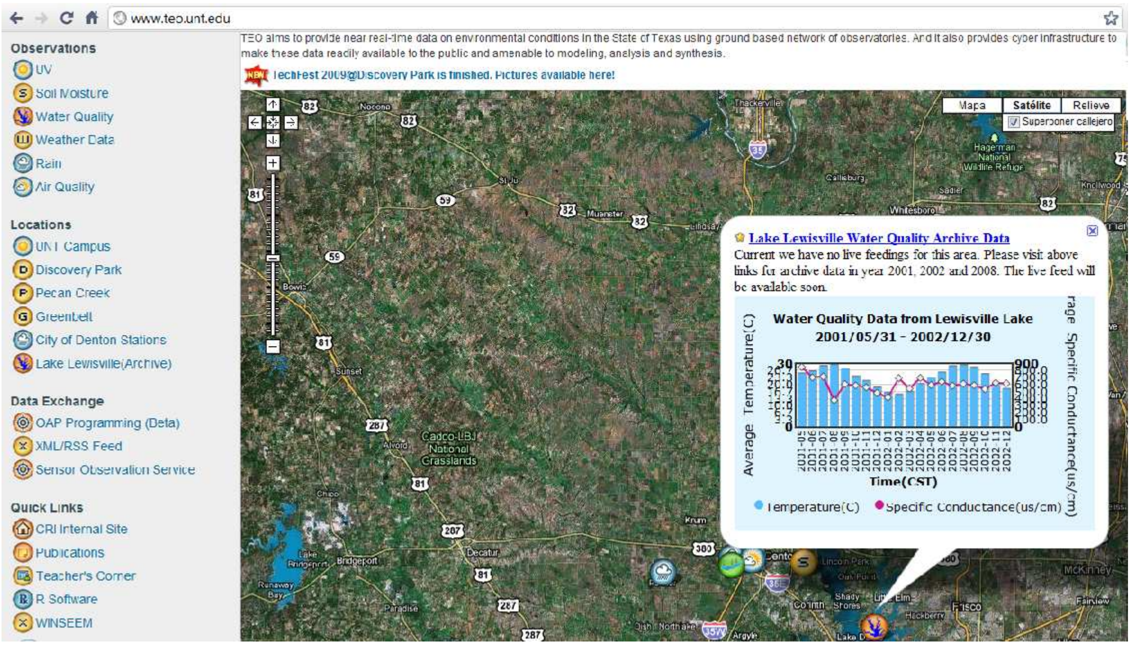


Fig.6. A snapshot of the TEO web interface for data visualization

TEO Online portal provides a variety of ways to explore sensor observation data. For example, data can be browsed under different overlapping categories, such as type of observations (UV, soil moisture, etc).Several predefined functions allow the analysis of data statistics such as average and maximum values.

Data interoperability rises as an important issue with the wide application of sensor networks and different field. Data exchange can be performed through Extensible Markup Language (XML) data exchange, Really Simple Syndication (RSS) feed, and Sensor Observation Services (SOS), as show in fig 7

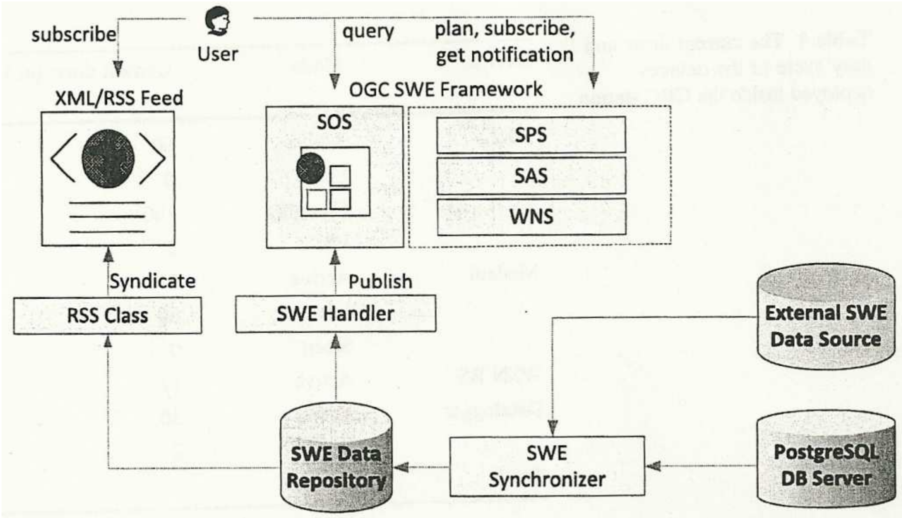


Fig 7. Sensor data dissemination and exchange frame work.

The sensor data interoperability is achieved by a dedicated backend SWE data repository, as well as front-end RSS feed and web services building upon the repository. RSS feed items and links are stored in a RSS table in the repository, while the live data is encapsulated in the RSS page by the web layer RSS class functions. Thus the general users can subscribe and watch the data through his browser or can request to receive the most recent sensor observation data automatically.

The SWE is a new standard that specifies interoperable interfaces and metadata encodings to enable real-time integration of heterogeneous sensor data. Major encoding standards include SensorML that describes sensor system information and O&M (Observation and Measurement) that encodes actual live data. The interoperability interface standards include SOS (sensor Observation Services), SAS (Sensor Alert Services), SPS (Sensor Planning Services), and WNS (Web Notification Services).

These modules collaborate together and can be used to customize various web-based or desktop-based applications for environmental scientists.

The database is syndicated with a sensor table, which allows the formation of the SWE standard. Thus SWE is synchronized with the DB server. During this synchronization, SWE synchronizer can retrieve external data. The data in the SWE repository is converted to the SensorML and O&M

format by an SWE handler, which then feed the information to the upper-layer web services.

Deployment and field testing results

Deployment of sensor networks in the field

The weather station in Denton, Texas has nine years to control the weather, with a wired system, but in Match 2008 was extended with a modem wireless system of 8 motes as shown in Fig.1.

It way to 16 motes along a cross-sectional transect as shown in fig.8. Support a long-term, each sensor node collects data every 10 minutes from soil moisture sensor and onboard temperature and relative humidity sensor. This cross network topology is perfect to check the moisture and a vegetation changes as a function of the height above the river and soil type.

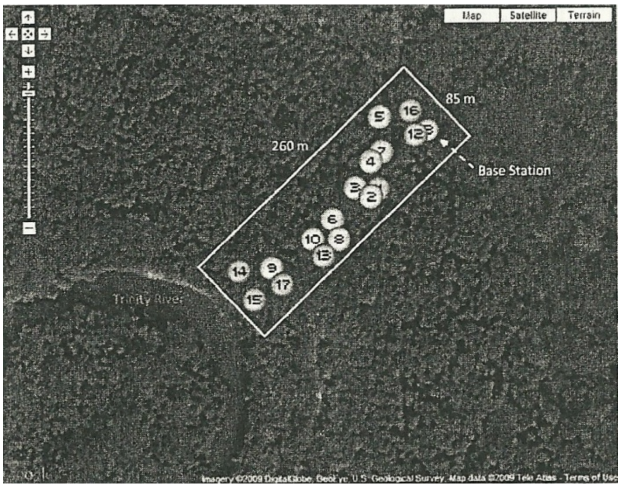


Fig. 8. Sensor deployment topology in the field as shown on the Google Map-based Teo Online web portal.

The motes are installed in weatherproof boxes and on top of metal poles to avoid flooding water. Before we put the motes we'll have to check to measure the one-hop radio communication range. We observed that with a maximal transmission power of 3dBm, IRIS motes are able to transmit on overage 30 m with 95% packet reception rate (PRR) and 50 m with 80% PRR. Thus, we deployed motes with a maximum one-hop distance of about 30 m.

The TEO Online web portal has been operational since March 2008 with most of the basic web services implemented.

System performance characterization and field testing results

Duty cycling provides an effective way to achieve energy efficiency. The RFG wakes up for 90 s every 10 min. The Table 1 shows the current draw and duty cycle devices deployed inside the GBC station, powered by solar panel with a peak current of 960 mA and a 12 Ah lead-acid rechargeable battery. The battery supports 7 days without recharding. In general, the capacity of a solar panel should be at least 10 times the average power consumption.

Device	Mode	Current draw (mA)	Duty cycle (%)	Avg. current draw (mA)
SBC	Active	60	15	9.0
	Sleep	0	85	
TS-BAT3	Charging	100	10	11.8
	Idle	2	90	
Modem	Active	250	5	13.5
	Idle	60	10	
	Sleep	0	85	
WSN BS	Active	11	100	11
Datalogger	Active	30	1	2.28
	Idle	2	99	
Total				47.58

Tab.1 The current draw and duty cycle of the devices deployed inside the GBC station

With the duty cycle scheduling algorithm, motes are only active during a few TDMA slots to report and relay sensor data and a signalind slot to synchronize time, manage neighbor list, and update parent information. Motes remain in the sleep mode and consume much less power than in the active mode. Fig 9 shows a measurement result of current draw and duty cycle, capture with an oscilloscope.

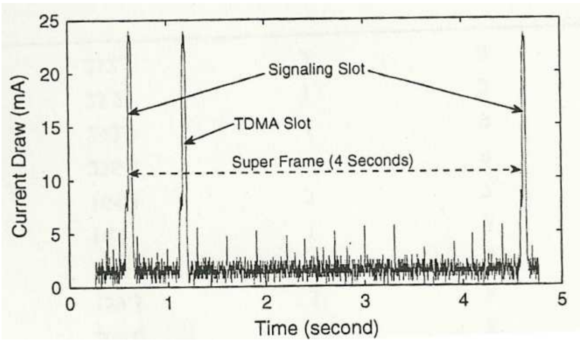


Fig 9 Measurement of the current draw and duty cycle of a mote.

The spender an average current of 24 mA for 16 us in active mode and 2.73 mA for 90 us in sleep mode. Thus, two fully charged 2500 mAh NiMh batteries with a self-discharging rate of30% can sustain a mote for about 4 weeks without recharging. The solar cell used at the power motes is able to provide 100 mA peak current at 3 V, about 36 times the average current draw of the load.

The variation of weather condition was also captured by sensor on motes as shown in Fig 10. The difference in data among the three soil moisture sensors reveals spatial variation characteristics of the soil moisture condition n that area. Mote 3 is installed in a transparent box to put a solar cell inside the box, Mote 4 is installed in a non-transparent box with the solar panel.

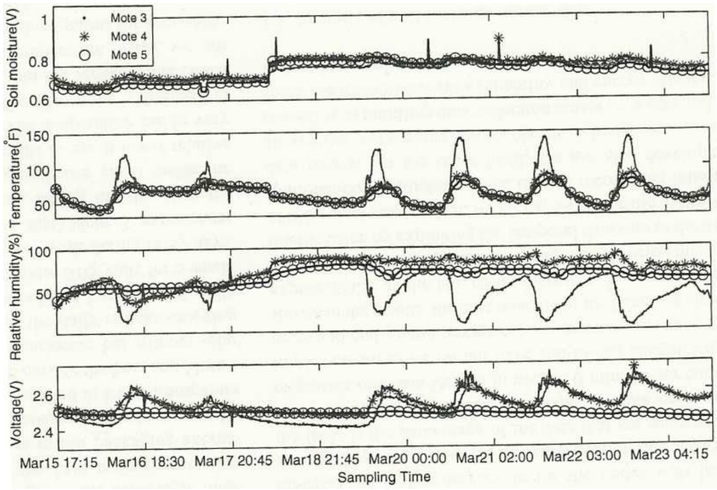


Fig 10. Sample sensor data collected in field testing.

Table 2 shows a few statistics of the transect sensor network status data that has been collected from field tests over a one-month period. From the hop count measurements (between each node and BS) we can clearly observe to the tree structure of the multihop sensor network experiences dynamic variations and the sensor network is able to reorganize autonomously in the face of environmental and network changes.

Node ID	Distance to BS (m)	Min. hop count	Max. hop count	Avg. hop count	Avg. duty cycle (%)	Data reception rate (%)
1 (BS)	0	0	0	0	6.2	1
2	88.1	1	6	2.3	5.0	95.0
3	92.8	1	5	1.6	4.1	96.0
4	65.6	1	3	2.0	4.5	97.8
5	50.2	1	2	1.0	4.0	97.3
6	130.7	2	6	2.9	4.7	91.0
7	51.9	1	3	1.6	4.4	97.2
8	141.2	2	6	3.5	3.8	92.6
9	209.0	3	8	4.3	6.4	76.0
10	159.3	3	8	4.6	4.3	76.9
11	77.3	1	5	1.5	4.2	88.7
12	14.1	1	3	1.1	2.5	98.0
13	164.3	3	7	3.7	4.4	76.1
14	236.6	4	9	5.7	4.5	68.5
15	242.7	4	9	5.9	4.1	71.5
16	25.2	1	2	1.1	2.0	97.3
17	212.3	3	8	5.1	4.8	76.1

Tab 2. Statistics of sensor network status data collected from field tests.

Mobile sensor network programming system (MSNPS)

In traditional sensor networks, the sensor are deployed manually the manual installation and upgrading of physical infrastructures will not scale if we want to build the sensor networks in a larger scale.

Many new WSN applications use sensor carried by moving objects. I.e. in a people centric sensor network environment, people wear sensor that are embedded in cells phones or PDAs. When people move around with sensor, the sensor can be use for various applications. In this way, a mobile sensor network can be formed without any manual facility installations and can scale to a large area.

MSNPS contains a gateway protocol that bridges the Internet and sensor network so that control and data messages can be exchanged across networks. MSNPS can disseminate new configurations or new binaries to the targeted sensor nodes instead of the entire network. MSNPS is optimized for mobile sensor networks.

MSNPS overview

In figure 1 is shown the mobile sensor network in MSNPS works. The MSNPS client agent, which works in the Internet, is used to provide the user interface to users from the Internet to disseminate the code images. The roadside micro servers collected de sensed data and send the sensed data to the sensing application server. A gateway is running in the micro servers so that sensor network can exchange control and data messages with the client agent. Micro servers have at least two interfaces, one to connect to the sensor network and the other to the IP network through devices such as wireless mesh routers. These mesh routers form a IP-based wireless mesh network and connect to Internet trough gateway routers.

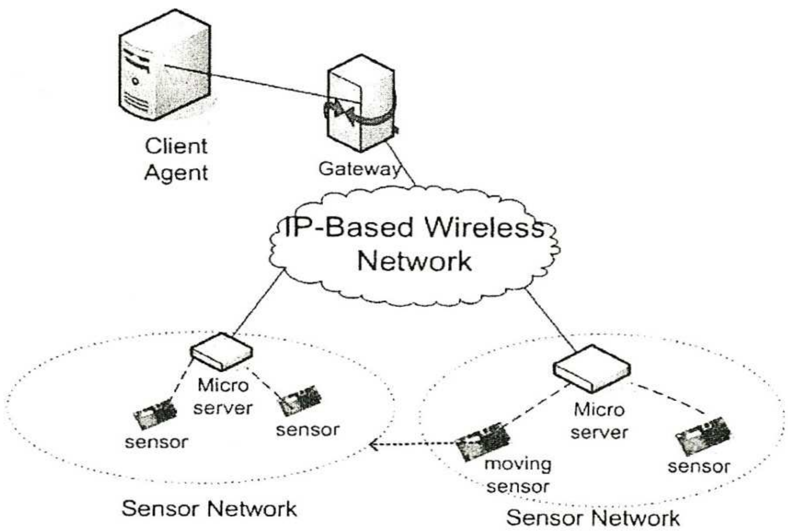


Fig 1 MSNPS architecture

There are four types of messages in MSNPS. Code version messages contain information of the code to be disseminated. Code request messages are used for sink nodes to request the code image. Code data messages are used to send the content of code image. Route update messages are used to maintain the MSNPS routing paths.

The code version messages is first broadcasted from the client agent to micro servers, and then injected into the sensor networks. The code version message broadcasting will form a spanning tree. If a sensor node fits the code image type and it does not have the code image with the same or larger version number, it will become a sink node.

Design of MSNPS

MSNPS gateway protocol

MSNPS needs gateway protocol

MSNPS needs gateway functionality to bridge the client agent and the sensor network so that the micro servers can communicate with the client agent through the Internet and communicate with the sensor nodes in the mobile sensor network.

MSNPS gateway protocol is designed to support TinyOS based WSNs.

The gateway support two mode:

Passive mode

In a passive mode, the gateway converts datagrams coming from the Internet to active messages in the sensor network and vice versa.

When the micro servers are in passive mode, the client agent becomes the root of the code dissemination tree. Before an application level message such as a code data message is sent down to the UDP layer at the client agent, an extra header called gateway header will be added to support the gateway protocol at the micro servers.

Active mode

The gateway can act in active mode too. In active mode, the micro servers not only act as the gateways, they can also aggregate, fragment and reasamble the messages. In this way, a sensor in WSN and a host in Internet can exchange large amounts of application data.

A reliable transport protocol such as TCP should be used. On each micro server, a gateway process is running. The TCP connection between the gateway process in the micro server and the application process on the host can be initialized by either side.

Micro server can send long messages to the sensor nodes.

MSNPS network layer design

In MSNPS, acode dissemination tree is maintained for packets routing. MSNPS constructs a bidirectional tree using code version messages and code request messages.

MSNPS routing information is maintained at each sensor node by two routing tables: a node table, which contains the routing information to the root, and a request table, which contains the routing information to the sink nodes.

In Fig. 2 every node in the mobile sensor network that receives the code version messages sets up a gradient pointing to its parent node.

A sink node joins the distribution tree of the code and sends the code request to the root along the gradients. The intermediate nodes receiving the registration message code of the route is joined by updating the request table for source data messages can be directed from the root to the sink nodes.

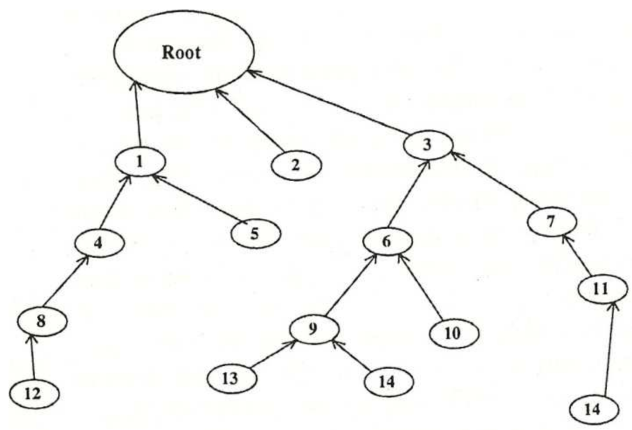


Fig 2. Gradient set from node table

In the mobile sensor network, the probability of link failure is much larger than that in the static sensor network. The root needs to broadcast messages to repair the tree.

Wireless sensor networks for detection and monitoring of nuclear sources.

The developed system aims to provide functionality to control applications of radioprotection. It is designed to be installed at a nuclear facility and this fixed it at all times to the levels of radioactivity.

To achieve this objective has performance characteristics that differentiate it from common applications. The ability to store data in database, monitors, alarms, warning of abnormal events and to periodic statistical or real-time.

In this application there are two types of motes the fixed and volatile, the latter can enter and exit the system.

All information is stored, processed and managed on a server that meets the following two basic functions: gateway between the motes and the end user as the data server to client applications. This project works with the system TinyOS.

Wireless Sensor Networks for Fire Rescue Applications

1 Fire Rescue: State-of-the-Art

Fire rescue is one of the most important public safety activities. A typical scenario of the fire rescue process is depicted as follows. After the fire department gets a fire alarm call, it will send a fire fight team to the fire field. Normally, a fire fight team consists of one incident commander vehicle, two engine vehicles, one ladder vehicle, and the most important role, a set of firefighters, who are grouped as squads associating with one of the above vehicles. During the process of fire rescue, the incident commander is in charge of the whole fire rescue situation, including monitoring the fire field and making real-time schedule on firefighter assignment. The two engine vehicles carry water, which will be used in the case when water is lack near the fire field. And the ladder vehicle holds the utilities like ladders that are needed by the firefighters. The firefighters are organized into different squads based on their specialty and fight cooperatively to eliminate fire in the fire field.

This fire rescue operation mode has several shortages. First of all, the incident commander could not have a clear view of neither the status of firefighters after the rescue work starts nor the accurate situation of the fire field, so that it is difficult for him to make an optimized schedule. Second, the firefighters in the fire field do not know the dangerous situation around him in time, which increases the danger to the firefighter as well. Finally, it is inflexible for the fire depart-ment headquarter located far away from the fire field to get fresh and timely fire rescue information, which is particularly important for big cities which might have multiple fires at the same time.

Current fire fight may be supported by some systems such as Geographic Information System (GIS) and FINDER. FINDER uses some technology to guide the fightfighter to find the injured crew member, as a result, the incident commander and fire department have no clear view of the whole fire field so that it does not help to make good schedule of fire fighting operation. GIS is another popular system used in the fire rescue applications. It allows a user to view and analyze information graphically and spatially, and maintain a database of that information. Thus, it is good to be used in fire protection and management but not very suitable to be applied in applications that have high real-time requirement. GEOMAC is a successful system built based on GIS to help fire fight, which links GIS technology to the Internet, using remote-sensing data from automated remote weather stations and some image-interpreting skills to provide managers scattered over three states with almost realtime information to base their decisions on, but this system also not monitoring the firefighter conditions which is essential for a good fire rescue operation. So this system is suitable to be used in wildland fire management.

In summary, current fire rescue is not efficient and current fire rescue systems either focus on fire protection and management or focus on only guide firefighter in inefficient way.

Requirements

Wireless sensor networks are application-specific systems. Different applications may have specific requirements of the design of the underlying WSN, including protocols as well as the hardware and software support. Fire rescue is a special application which is different from the previous WSN supported applications, such as environment or habitat monitoring, and object tracking, thus it has some specific requirements for the design of WSN system. In the following paragraphs, is described them one by one in detail.

Accountability of firefighters. Accountability of fire-fighters is the first and most fundamental requirement of any fire-related public safety applications. Continuing with the basic scene of a fire rescue application described above, when firefighters start fighting in the fire field, the incident commander and fire department need the information of the firefighters, one of which is for the accountability of firefighters. At the current stage, each firefighter is associated with a squad and one vehicle. When the firefighters enter in the rescue operation, they leave their badges in the vehicle, which has their name and ID only. After firefighters come back from the rescue work, they get their badge back. If the badge is left in the engine, the incident commander assumes that the owner of the badge is missing in fire rescue. In this way, the firefighter will only have two status, alive or missing. As a matter of fact, this approach for accountability is too obsolete to collect useful information. We argue that for the complete accountability of firefighters, we need the following information of each firefighter, including both static and dynamic information. A list of static information could be name, ID, age, specialty, primary squad assignment and second squad assignment. Primary squad assignment denotes the squad the firefighter belongs to in the normal condition. And second squad assignment denotes the temporarily assignment of the firefighter to another squad to substitute some absent firefighter with the same specialty. Besides these static information, we would also like to know the dynamic information of firefighters in the field, such as the total time the firefighter participated in the fire rescue, the workload of the firefighter, continuous locations of the firefighter, and the physical condition of the firefighter. These information can be collected by WSN and reported to the incident commander, who will evaluate the efficiency of the fire rescue operation based on these accurate accounting information.

Real-time monitoring. Besides the accountability information of firefighters, more real-time information about the firefighter and fire field is wanted by the incident commander and the fire department. First, the incident commander needs the real-time location information of firefighters, because thefirefighters keep moving according to the fire situation during the process of fire rescue. Having the location information, the incident commander can make better schedule, e.g., he can find firefighters with some specialty and send them to where they are needed. In addition to the information about firefighters, the fire field information is also very useful for the incident commander to judge the real situation of the fire rescue and make real time decision and schedule. Thus, the WSN could be used to collect the environment information of the fire field, including the humidity and temperature of the fire field, the wind speed, the density of the smoke, and so on.

Furthermore, the information about some vital events in the fire field need to be monitored as well, e.g., the death of the firefighters and the dramatic changing of the environment parameters, such as temperature, chemical and biological leak.

Based on these information, the incident commander and fire department will have a clear view of the fire situation and make effective schedule to fight the fire. Intelligent scheduling and resource allocation. The quick changing of environment and the happening of emergent events in the fire field require that the fire rescue schedule is made in a timely fashion, in fact, which is very difficult, if not impossible, for the incident commander in current fire rescue operation. Thus, to improve the speed of response to the changing environment and emergency, we argue that an intelligent software, which capable of automatically mining the collected data, is essential to aid the incident commander and fire department to make quick but effective decision on how to fight the fire. The requirements of this software are listed as below. First, this software needs to analyze the information collected from the fire field, calculate the accountability, and detect the events. Second, it should present these data to the incident commander and fire department in an easily understandable way. Third, the software can automatically generate the schedule of fire rescue process based on the collected data. Finally, the software should notice the incident commander when some important events are detected.

In WSN, this software will run either on the powerful laptop acting as sink deployed on the incident commander's car or on the machines located in the fire department. Web-enabled service and integration. Not only does the incident commander sitting near the fire field need the information collected by the WSN, but also the officers sitting in the fire department, which is located far away from the fire field. In a big city like New York, there maybe several fires happened at the same time, so the officers in the fire department need to make schedule on how to control these fires effectively and concurrently. Thus the real-time information from different fire fields is needed by the fire department, and the optimized schedule will be made based on this global information. Web-based service is one of the most convenient ways to provide these information to these officers. By doing so, the real-time information from each fire field is wrapped as a web-enabled service, accessible through regular Web browsers. Because the fire department is located far away to the fire field, the traditional Internet will act as the bridge to connect the fire field and fire department. First, the webenabled service should provide the information that the fire department interested via the network, e.g., it continuously reports the live situation of each fire field. Second, it will automatically generate some events to the fire department to ask aid when more firefighters or vehicles are needed. Moreover, the collected data can be stored and analyzed later to find some good rescue models to support the future fire fight.

2 FireNet Architecture

Having known the requirements of the fire rescue application, we are now in a position to propose FireNet, a wireless sensor network architecture for the fire rescue application.

Compared with traditional wired sensor systems, the novelty of WSN lies in that the sensors deployed in the sensor field can self-organize into a connected ad-hoc network via wireless communication. We call the specific wireless sensor networks deployed for the fire rescue application FireNet.

The architecture of the FireNet is shown in Figure 1. In the FireNet, the vehicles and firefighters are equipped with sensors which form a self-organized heterogenous wireless sensor network. In the incident commander's vehicle, a powerful laptop connected with a powerful sensor acts as the gateway of WSN. The ladder vehicle and the two engine vehicles are loaded with sensors having GPS equipped. These vehicles can act as the landmark for the whole WSN because they will have relatively stable location, i.e., other sensors can calculate their location based on the location of these vehicles. Each firefighter in the sensor field carries a sensor, such as MICA2 or MICAz from Crossbow attached with available sensor board which can sense interested parameters, acting as the active badge for each firefighter. The active badge records all the information expected by the incident commander and fire department (for later analysis), such as the firefighter information, the fire field environment information and emergent events, as listed in the Figure 1.

The role of each active badge (i.e., sensor) has two-fold: sensing the data and forwarding the packets. We can also install different program on the sensors for the firefighters with different specialty, thus these sensors can have different functionality. Furthermore, the sink of FireNet, located in the incident commander's vehicle, is connected to the fire department headquarter via traditional Internet so that the WSN and the fire department can communicate each other and keep contact.

After the fire fight team starts their work, the sensors attached to vehicles and firefighters are self-organized into a WSN via wireless communication. Then, the sensors start to operate according to their pre-installed program. For example, the sensors attached to firefighters will collect the information of firefighters, sample the environment parameters, and generate the vital events happened in the fire field, as shown in the right part of Figure 1. These data will be reported to the sink by the multi-hop routing protocol and further delivered to the fire department via Internet. Then, both the incident commander and the officers in the fire department have the accountability and real-time information from the fire field, which is abstracted and presented by the powerful pre-installed software in the sink or fire department. By doing so, the whole fire field is monitored and the status of each firefighter is clear to the incident commander and fire department. Based on this, the incident commander and fire department will make optimized fire schedule according to the suggestion of the intelligent software. For instance, in the case of fire rescue in September 11, 2001, some firefighters were covered by the dust or buried in some part of fire field of ground zero, which is a very dangerous situation. If WSN is used, it will be much easy to detect such a situation and the location of firefighters in anger, a rescue can be arranged immediately to help them out of the disaster. Moreover, the schedule commands from the incident commander can be sent to firefighters via FireNet to instruct the firefighters to move or take some other ctions as well.

In addition to the communication ability, each sensor has its CPU and local memory so that it can do some calculation such as data aggregation and store the data for a period of time. With the development of the semiconductor technology, the computational capability and the memory size have been extended a lot in the last several years. We believe that wireless sensors will be more powerful in the near future so that more information can be stored and more efficient realtime decision making algorithms can be employed. In summary, as described above, wireless sensor network is a very promising technology in the application of fire rescue. It is very useful in terms of not only monitoring the whole fire field including the firefighters and environment information, but also sending schedule information and commands to the firefighters in a more intelligent way.

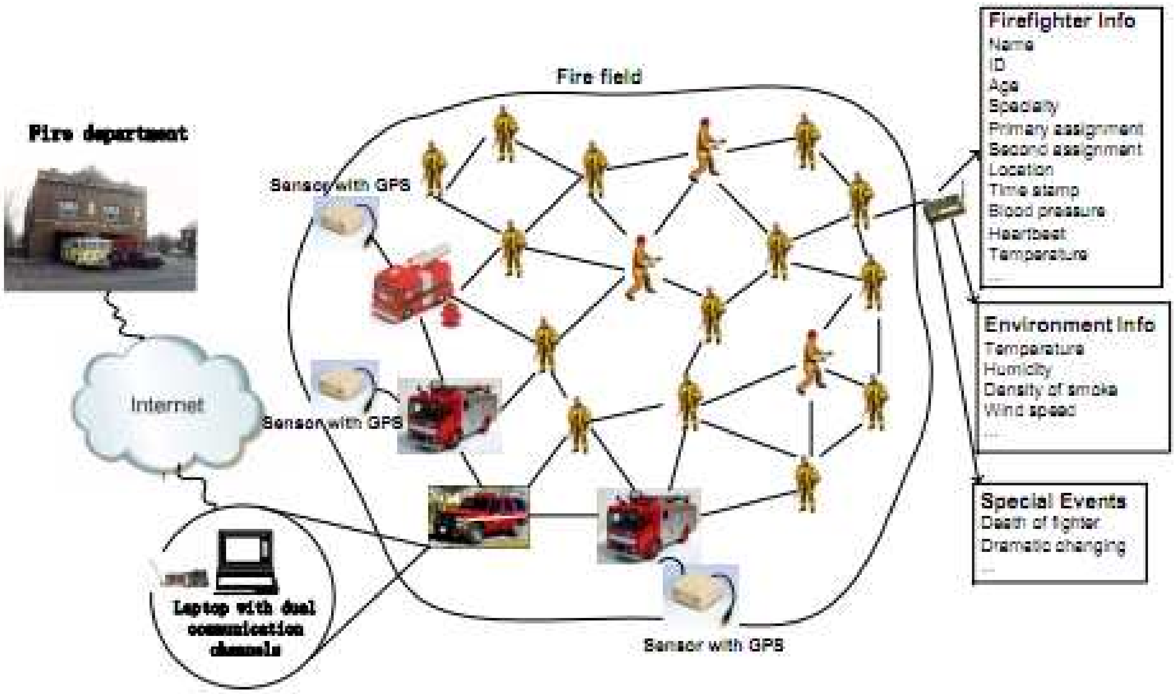


Fig 1. The architecture of FireNET

3 Research Challenges

As described above, the fire rescue application has its special characteristics, such as high mobility and real-time, and specific requirements as described in Section 1.2. Previous protocols are not satisfactory and need to be revisited for this application. New protocols and support from hardware as well as software are needed to build the sensor system. Next, we narrate them in detail.

Protocol Challenges

First, four protocol challenges, namely real-time selforganization, fault-tolerant routing, service differentiation, and real-time and mobile localization, need to be addressed for this application.

Real-time self-organization. FireNet is a pure ad-hoc wireless network, thus the self-organization of FireNet is very crucial. Unlike other WSN applications like environment monitoring, the fire rescue application of WSN has two specific characteristics, the high dynamics of sensors due to the high mobility of firefighters in the fire field, and the realtime requirements of data collection. These two inherent features make the self-organization of FireNet become more challenge. Moreover, we expect that the wireless communication between sensors in a fire field has extremely high loss rate due to the harsh environment. Thus, it is not easy to achieve the goal of real-time self-organization. A good selforganization protocol should be proposed to support the automatic re-configuration of the sensor network, which should have the capability to make all the sensors attached to fire-fighters and other utilities in the sensor field connect or reconnect to the sensor network in a limited period of time after they newly join the network, lose the connection and reconnect, or change their location. In such a protocol, several functions are highly wanted. First, the relationship between each firefighter and his/her associated vehicle should be captured automatically; Second, in a short period of time after firefighters start the fire fight process, these firefighters and the utilities are all self-organized into a connected adhoc WSN; Third, during the procedure of firefighter moving, the sensor network should keep connection in spite of the absence of several firefighters from their original area, and the moving sensors should re-connect to WSN in a limited time at the new location. The challenge of this protocol lies in how to satisfy the real-time requirement, and always keep the connectivity and coverage of the whole WSN.

Fault tolerant routing. As argued above, one significant characteristic of the fire rescue application is the high dynamics of network resulting from the mobility of firefighters or failure of sensors. Although the real-time self-organization protocol takes care of the re-configuration of WSN, it cannot assure the successful delivery of messages to the sink.

However, the fire rescue application has high requirement on accuracy and real-time to the collected data. A fault tolerant 4routing protocol is essential to guarantee the successful delivery of the data and events in the case that the mobility and absence of some sensors as well as the unreliable wireless communication will cause a high package lost rate. The fault tolerant routing protocol should deal with both the mobility of sensors and the failure of sensors happened in the sensor field in a timely fashion. Previous research on the routing protocol in WNS usually assume that sensors are located in fixed position, neglecting the high dynamics of the network. Thus, novel routing protocols are expected to handle the problem of sensor mobility and real-time delivery. One possible solution is to build multi-path routing protocols in the real time, which can dramatically improve the probability of successful delivery in a high dynamic environment.

Service differentiation. The main function of the WSN in this application is to collect different information from the fired field, e.g., the environment parameters such as temperature, humidity, wind speed, and chemical and biological leak, the emergent events such as the dramatic changing of the environment parameters and the death of firefighters, and the status of the firefighters such as location, and the workload of firefighter. Although all these data need to be monitored and collected, their importance is different and then these parameters should be treated in a different way in the system. Thus, we argue that an efficient service differentiation scheme, including timeliness, storage requirement, and processing priority, should be provided to implement multiple quality-ofservice (QoS) in such an environment. As a matter of fact, how to arrange the collection and delivery of these data in an effective and efficient way is a challenge because sensors have limited low bandwidth and share the wireless communication medium. A good service differentiation scheme is needed to schedule the data collection and transmission through the FireNet architecture. One possible direction is exploiting the inherent consistency requirements of those parameters. For example, an event of firefighter death is much more important than any other parameters in the network, so it will be set the highest priority and reported to the sink as soon as possible. In addition, the tradeoff between the energy efficient and data consistency including data accuracy and timeliness should also be examined in the protocol.

Real-time and mobile localization. As we described in Section 1.2, the location, especially the real-time location, of firefighters in a fire scene is a very important and valuable piece of information. Given this piece of information, the incident commander could have a clear view of the distribution of deployed firefighters, and make real-time decisions.

Moreover, location information is very useful for other protocols in WSN as well. Most research topics in WSN, e.g., fault tolerant routing, aggregation, event detection and tracking, and so on, directly or indirectly lend on accurate location information provided by the underlying localization service.

Admittedly, location in the fire rescue application is not a trivial task given the fact that firefighters are moving very fast and randomly in a real rescue operation as well as the inherent ad-hoc feature of FireNet. Localization in WSN has been extensively studied in the literature [11]; however, as a reality check, few of practical localization algorithms are deployed in the real applications, and practical localization, especially mobile localization, is still a challenge from the perspective of real deployment. Intuitively, Global Positioning System (GPS) is a pretty good positioning system at outdoors, however, it is not accurate enough for indoor tracking. Moreover, most of existing localization solutions did not take the mobility into consideration, i.e., they always assume the location of sensors is static, which is obviously not the case in FireNet.

The few mobile localization algorithms such as do not consider the moving speed and the dynamics of the system. Therefore, we argue that the localization protocol for a WSN in fire rescue application needs to address the following issues: mobility, heterogeneity, locality, robustness, feasibility, and accuracy, each of which is described as follows.

• Mobility The fast movement of firefighters makes the localization a big challenge in a timely fashion.

• Heterogeneity Due to the heterogeneity of the sensors used in FireNet, the localization algorithm should take these diverse platforms into consideration, e.g., the computing devices on some vehicles/equipments, such as laptops or tablets, could be integrated with GPS support which provides some reference points for further location resolving, while on the other side, the sensors carried by firefighters would be very simple and possess only limited computing resource and energy support.

• Locality Each sensor has limited computing power, memory, and communication range, thus only a completely localized algorithm is applicable in FireNet, where each sensor interacts with its neighbors only.

• Robustness Sensors in FireNet are working in a very harsh and highly failure-prone environment. Robustness is a key requirement of the localization algorithm, i.e., the failure of some sensors should not affect the calculation of other sensors location.

• Feasibility The localization algorithm has to be practical enough so that its computing cost could be affordable by the limited hardware/software supporting of those tiny sensors.

• Accuracy The real scheduling by the incident commander and fire department is based on the accurate location of firefighters. Accurate positioning in a static environment is already nontrivial, it becomes more challenge in such a highly dynamic environment.

Software Challenges

As analyzed in Section 1.2, software components are required in the sink and fire department. These software components are used to analyze the collected data and make good schedule suggestion. Because the sink and fire department have different requirements to the software, two types of software should be designed for them respectively.

The software for the sink should has the following functions. First, it needs to analyze the collected data and abstract the useful information from the data such as the accountability information of each type of firefighters. Second, it should make some schedule suggestions based on the collected data and some preset rules. Third, it needs to present the abstracted information and the schedule suggestion to the user of this software. The design of this software includes two parts, the graphic user interface (GUI) part and the intelligent decision making and scheduling part. All the information needed by the incident commander and fire department will be presented in the GUI interface. For instance, the accurate real-time location of firefighters will be shown in a map of the fire field. And the accountability information as well as the status of firefighters will be provided by the GUI. The scheduling part has the function of analyzing data and generating schedule suggestion. A well designed rule set is essential to provide good scheduling schemes. Some technologies from artificial intelligence community may be useful to design the rule set through self-learning. Besides, to support the remote access of these real-time information, we need to provide a web-enabled interface, which needs to take the security into consideration because of the confidential information of firefighters.

Another software component is needed at the fire department side. We can use the similar software as in the sink. However, the officer in the fire department may not care such detailed information as a incident commander. For them probably the general information is enough. Meanwhile, the decision rules at the fire department are different from those of a incident commander. Comparing with the incident commanders, the officers in the fire department headquarter need to re-schedule the firefighter squads as a whole across multiple fire fields.

Hardware Challenges

The fire rescue application also posts some specific requirements on the hardware support for sensors. Normal commodity-of-the-shelf wireless sensors, e.g., motes from Crossbow, are applicable in this application in terms of functionality. However, to our knowledge, they haven’t considered the extremely harsh environment like fire field, which is a very important issue to the success of the WSN deployment. The FireNet architecture proposed for the fire rescue application is running in a dangerous environment which normally has fire, water, dust, and extremely high temperature. Moreover, the sensors are shaking and moving dramatically with the moving of firefighters. If the sensors are exposed to such an environment without well protection, it will stop working immediately. So we argue that two issues should be considered to address the hardware challenge. One is packaging, and we need to come out an ideal packaging scheme for sensors to make sure they are water-proof, fireproof, and vibration-proof. The other is the internal design of wireless sensors. Some fault tolerance features should be taken into consideration during the design procedure.

Monitoring structures

An application of WSN is monitoring of structures has been mainly conducted in the United States and Canada. Here are estimated to have about 25 trillion of dollars invested in civil structures and investment so they want to take control of the structures built. The technology used is called SHM Structural Health Monitoring of English and works with the identification and monitoring of bizarre behavior. These may be a flaw in a structure such as bridges, buildings or other structures.

Vibration control of bridges

Thanks to new microelectromechanical systems (MEMS) and we can have acceleration sensors that can measure wireless media; we can control the vibrations in buildings. The University of Berkeley, California, conducted a study on a pedestrian bridge over the highway I-80 in Berkeley.

This application gives information on the state of life of the structure as well as events that occur during monitoring it, such as an earthquake.

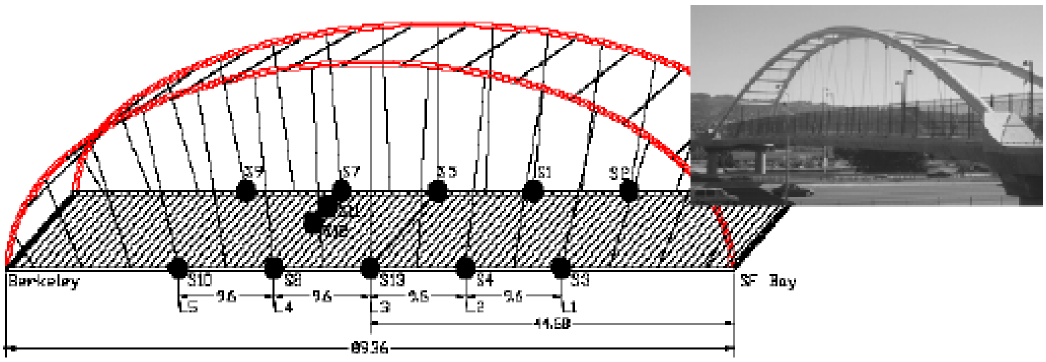


Fig 1. Location of sensor on the bridge.

Health monitoring of civil infrastructure (Golden Gate Bridge)

Another example of simple harmonic motion (SHM) application is held in the Golden Gate Bridge in Sa Francisco, Where the nodes are designed to monitor in various parts of the structure vibrations that have occurred, whether by the passage of vehicles or or by atmospheric conditions.

Sixty-four nodes were implemented in a system of 46 jumps measured the vibration environments with an accuracy of 30 g. Vibration environments were sampled at 1kHz with an exposure time less than 10 s.

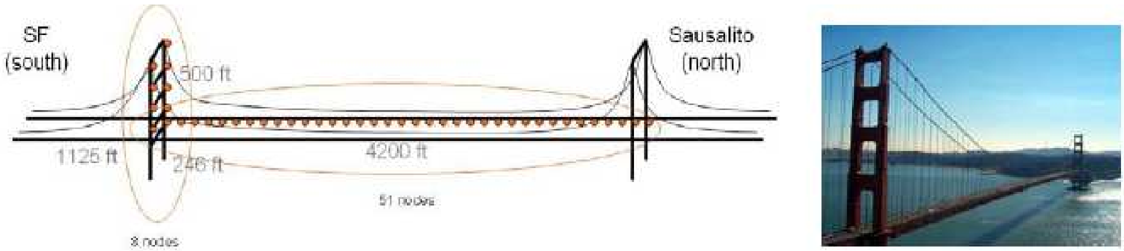


Fig.2 Location of sensor on the bridge Golden Gate

Automotive

With the characteristics of WSN, cars may soon be available to talk to each other and with infrastructure in roads and highways. The sensors can be applied to the wheels of the vehicle to assist the driver and warn of possible warning messages.

For example, during an emergency braking, emergency message from the car that stops may be sent to all cars nearby so that they take action with regard to this event. Another interesting application is the collection of traffic data in real time. The information that a car may have come in the opposite direction can be valuable. A vehicle can receive information from other fixed sensor information.

All this information can be passed from vehicle to avoid congestion and plan alternate routes.

Ford Capstone Project.

With 70% of Ford motor company vehicles being remodeled is important to keep current and learn the new features and releases. Ford is looking at new ways to innovate when it comes to collect measurements from their vehicles. Sensor networks are also increasing their intelligence and data can be obtained not only limited to light, temperature, humidity and movement.

A team from the University of Michigan, along with Ford, is designed and developed a way to identify how many times a vehicle is inspected by a potential buyer. This could include how many times a door is opened, the hood was raised or luggage was inspected. The events studied could be extended to where a person took a seat and how much time remains in it.

The environment, by definition, cam be very dynamic, making the ability to connect in a mesh network critical to long term.

Agriculture and ranching.

Farming and ranching are two areas where this technology can be important since a situation where monitoring outdoor conditions that help to improve production and quality in agricultural production or control of cattle on the move can be very difficult with traditional technology.

Camalie vineyards

Camalie vineyards, in the United States have one of the most advanced systems for measuring soil moisture. They use wireless technology developed by Berkeley University in collaboration with Intel and marketed by Crossbow.

The application is optimized irrigation, reducing water consumption, energy used in pumping and improving the quality of the grapes. It provides monitoring of the irrigation system, showing faults that may have a substantial impact in the long-term.

Fig 1Devices implanted in vineyards.

Once implemented the system found a significant increase in production and a decrease in energy consumption when using the facilities.

Routine monitoring in pigs

Dr. Philippe Bonnet University of Copenhagen developed an application designed to facilitate the work of veterinarians, controlling several variables in the daily routine of pigs on a farm.

Bibliography

Trade magazine:

Measurement and Wireless Networks

Other works:

Wireless Sensor Networks by Chris Townsend and Steven Armas

Conclusion

After studying in depth look at wireless sensor networks are quite clear improvement compared to traditional wireless networks due to several factors as are the durability of the lifetime of the batteries, allowing greater portability of sensor nodes and that can record more events to power stay longer in some places, the routing protocols networks sensors allow gain than in durability also gain in efficiency the avoidance of collisions between packets, which also ensures a lower number of unnecessary network traffic.

Because of the great features of such networks are currently using sensor networks in many projects related to different fields such as: environment, health, military, construction and structures, automotive, home automation, agriculture, etc.

This type of network currently is leading a technological revolution similar to that had appearance of internet, because the applications appear to be infinite, also speaks global surveillance network on the planet capable of recording and tracking people specific goods and research projects have generated great interest for application in practice.