Wireless Netw (2010) 16:1091–1108
DOI 10.1007/s11276-009-0190-1

Integration of wireless sensor networks in environmental monitoring cyber infrastructure

**Jue Yang · Chengyang Zhang · Xinrong Li ·
Yan Huang · Shengli Fu · Miguel F. Acevedo**

Abstract Wireless sensor networks (WSNs) have great potential to revolutionize many science and engineering domains. We present a novel environmental monitoring system with a focus on overall system architecture for seamless integration of wired and wireless sensors for long-term, remote, and near-real-time monitoring. We also present a unified framework for sensor data collection, management, visualization, dissemination, and exchange, conforming to the new Sensor Web Enablement standard. Some initial field testing results are also presented. The monitoring system is being integrated into the Texas Environmental Observatory infrastructure for long-term operation. As part of the integrated system, a new WSN-based soil moisture monitoring system is developed and deployed to support hydrologic monitoring and modeling research. This work represents a significant contribution to the empirical study of the emerging WSN technology. We address many practical issues in real-world application scenarios that are often neglected in the existing WSN research.

Keywords Duty cycle scheduling · Remote monitoring · Sensor data visualization · Sensor data dissemination · Sensor networks · Telemetry

1 Introduction

Environmental monitoring applications have become major driving forces for research and development of wireless sensor networks (WSNs) [1, 2]. Ecological and environmental scientists have been developing a cyber infrastructure in the form of environmental observatories, consisting of a variety of sensor systems, sophisticated computational resources and informatics, to observe, model, predict, and ultimately help preserve the health of the natural environment. Such an infrastructure becomes more important as we recognize that the natural world is inextricably linked to the human society to form an extremely complex ecosystem. WSN-based environmental monitoring systems promise to enable domain scientists to work with data sets of unprecedented fine spatiotemporal resolution.

Despite significant advances in recent years, there are still many challenging issues to be addressed to fulfill the full potential of the emerging WSN technology. The importance of the empirical study of WSN has been recognized by the research community, and considerable efforts have been put into the development and deployment of WSN testbeds for various practical applications, including environmental monitoring [1–8]. However, there are many limitations in the existing WSN testbeds. For example, many deployments are in controlled environments instead of real-life application environments. Most

J. Yang · C. Zhang · X. Li (✉) · Y. Huang · S. Fu ·
M. F. Acevedo
College of Engineering, University of North Texas,
1155 Union Circle #310470, Denton, TX 76203, USA
e-mail: xinrong@unt.edu

J. Yang
e-mail: jy0074@unt.edu

C. Zhang
e-mail: cz0022@unt.edu

Y. Huang
e-mail: huangyan@unt.edu

S. Fu
e-mail: fu@unt.edu

M. F. Acevedo
e-mail: acevedo@unt.edu

of the deployments are designed for short-term experiments or proof-of-concept demonstrations, instead of long-term autonomous operation to support ongoing work by domain scientists and practitioners. Most of the deployments are stand-alone WSN-only systems, monitoring very few environmental parameters, instead of being a part of the ever-growing environmental monitoring cyber infrastructure. As a result, it is difficult to consolidate a broad range of sensor data systematically to study the cross-correlation among various environmental parameters.

In this research, we strive to fill the aforementioned gaps and make significant practical contributions in WSN research by developing a soil moisture monitoring WSN and integrating it into a large-scale environmental monitoring infrastructure for long-term operations [9]. Specifically, we present a novel environmental monitoring cyber infrastructure that features (1) soil moisture monitoring with flexible spatial coverage and resolution, (2) seamlessly integrated wired and wireless sensors, (3) long-term, autonomous, remote, and near-real-time monitoring, (4) publicly available web services for sensor data visualization and dissemination, and (5) remote system monitoring and maintenance. Although the focus of the paper is on overall system architecture for seamless integration of wired and wireless sensors for long-term, remote, and near-real-time monitoring, we also present a suite of sensor networking protocols and various related device drivers that are developed and optimized for environmental monitoring applications, and address many practical issues in real-world application scenarios that are often neglected in the existing literature.

The new WSN-based soil moisture monitoring system is developed to support long-term hydrologic monitoring and modeling research. Increasing urbanization brings changes to the land cover of a given drainage area, which in turn increases the quantity of water flowing overland and decreases the amount of time to reach peak flow [10], increasing in some cases the risk of flash floods. Hydrologic models are helpful in predicting how changes in land cover in rapidly urbanizing areas translate into changes in the stream flow regime. These models require inputs that are difficult to measure over large areas, especially variables related to storm events, such as soil moisture antecedent conditions and rainfall amount and intensity. In addition, the ability to monitor in real time rapidly changing variables before, during, and after storm events will contribute to the improvement of rainfall estimations from meteorological radar data and enhance hydrological model forecasts. The new monitoring system that we are developing is ideally suited for such applications.

The rest of the paper is organized as follows. We first identify key design requirements for the new environmental monitoring system in Sect. 2 and describe the overall system architecture in Sect. 3. Then, in the next three sections, we describe three major components of the new system, including a WSN for soil moisture monitoring, a wireless telemetry system for remote near-real-time monitoring, and web services for data visualization and dissemination. Some initial field testing results are presented in Sect. 7, followed by a summary and future works in Sect. 8.

2 Design requirements

In this research, we develop a new environmental monitoring system to significantly improve the capability and usability of the system that is currently deployed at the Greenbelt Corridor (GBC) Park, Denton, Texas, operated by the Texas Parks and Wildlife Department. Some key design requirements are identified in this section.

2.1 Soil moisture monitoring with flexible spatial coverage and resolution

In the existing system, all sensors are deployed inside a small fence-enclosed area, a situation typical of many environmental monitoring systems. There is a need to provide flexibility to extend the spatial coverage and adjust the spatial resolution of soil moisture sensors. The spatial coverage of the system is limited by the physical limitation of the length of the cable connecting the sensors to the datalogger. In contrast, the spatial coverage and resolution of WSN can be conveniently configured to be meaningful to domain scientists.

2.2 Integration of WSN with existing environmental observatories

Despite their limitations, traditional environmental monitoring systems with various wired sensors are capable of accomplishing many monitoring tasks, and substantial investments are in place to monitor temperature, wind speed and direction, rainfall, and solar radiation. Drastically replacing the existing systems with an immature technology such as the WSN is considered unacceptable to many domain scientists and practitioners. Therefore, it is important to introduce the new WSN technology without disrupting the ongoing operation of environmental observatories through seamless integration of wired and wireless sensors.

2.3 Long-term, autonomous, remote, near-real-time environmental monitoring

Many environmental monitoring systems are deployed in remote areas that are inconvenient to access for data retrieval and system deployment and management. Traditionally, a stand-alone field station consists of a datalogger and a variety of sensors. Datalogger is programmed to sample at a fixed rate and data are stored in its internal memory. The data are retrievable via the RS232 port using a computer. Thus, accessing the data requires a visit to the field station, which is inconvenient and is extremely difficult, if not impossible, during harsh conditions, for example flood events. In addition, it has been recently recognized that many ecological and environmental studies need long-term data collection and management. Thus, environmental monitoring systems need to be survivable in extreme environmental and weather conditions for long-term operation with limited human intervention, making energy harvesting and energy efficiency major design considerations. Near-real-time data collection is another important feature to support time-sensitive environmental studies, which necessitates a convenient yet reliable long-haul wireless communication link.

2.4 Publicly available web services for sensor data visualization and dissemination

It is important to make data publicly available to benefit a broad range of entities such as environmental researchers, local citizens and government policy makers, and K-12 teachers and students. In addition, the explosive growth of environmental data collected by a variety of sensors in long-term operation necessitates a unified framework for data collection, management, integration, visualization, and dissemination. Such a framework should conform to standards, such as the Sensor Web Enablement (SWE) standard proposed by the Open Geographic Consortium (OGC) [11], to enable data exchangeability and interoperability.

2.5 Remote system status monitoring and management

For environmental monitoring systems deployed in remote areas, remote monitoring of system status is extremely useful for system development, debugging, and maintenance purposes. Thus, various system status data need to be carefully defined and collected together with the environmental sensor data. Furthermore, it is important to remotely adjust system configurations and update and upgrade software programs.

3 Overall system architecture

The new environmental monitoring system can be divided into four major layers as shown in Fig. 1, including physical data layer, logical data layer, web presentation layer, and user layer. Such a layered approach makes it possible to implement the system in a flexible, extensible, and efficient way. At the physical data layer, a variety of sensors are used to monitor environmental parameters. Sensor data are transmitted from a monitoring site to a Central Data Collection (CDC) Server. To address the design requirements, we incorporate a GPRS modem for

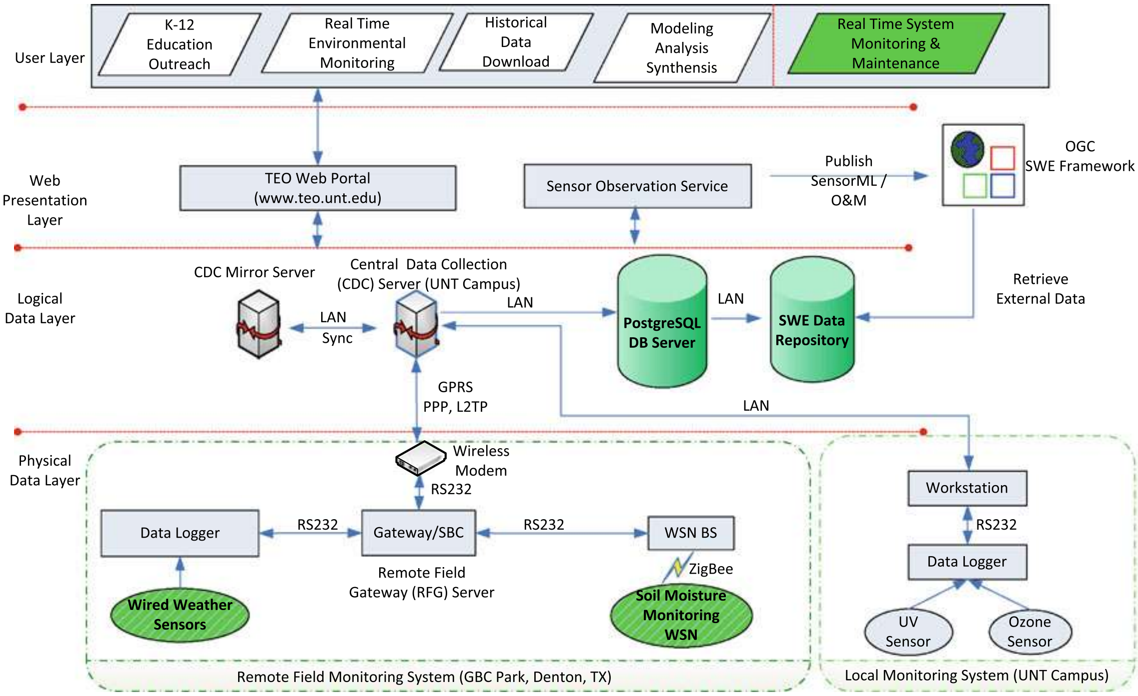


Fig. 1 System architecture of the integrated environmental monitoring system

wireless telemetry, a single-board computer (SBC) as Remote Field Gateway (RFG) Server, and a WSN for distributed soil moisture monitoring. The RFG Server provides effective control, management, and coordination of two relatively independent sensor systems, i.e., a traditional datalogger-based wired sensor system and the WSN-based wireless sensor system. The Linux-based RFG Server supports remote login to allow maximum remote manipulation of the devices in the field such as the SBC, datalogger, and WSN.

At the logical data layer, sensor data collected from the distributed monitoring stations are stored in a PostgreSQL Database (DB) Server. The CDC Server acts as an intermediate component to hide the heterogeneity of different physical layer devices and support data validation required by the DB Server. The CDC Server and its mirror server also archive raw data on local file systems. Daemon programs running on the CDC Server pre-process the data before it is inserted into the database, and periodically perform synchronization tasks. An SWE-compliant data repository is installed to enable data exchange, accepting data from both internal DB Server and external sources through the OGC web services.

The web presentation layer consists of a web portal, i.e., TEO Online [9], and a sensor web implementation. The web portal serves as a user-friendly interface for data visualization, analysis, synthesis, modeling, and K-12 educational outreach activities. It also provides useful capabilities for system developers and operators to remotely monitor system status and remotely update software and system configuration, which greatly simplify system debugging and maintenance tasks. We also implement Sensor Observation Services (SOS) at this layer, conforming to the SWE standard to facilitate data exchange. The standard SensorML/O&M data representation makes it easy to integrate our sensor data into the existing Geographic Information Systems (GIS) web services and exchange the data with other organizations. The SOS web service will be published to a catalog service in the OGC SWE framework to make it publicly accessible on the Internet.

Finally, the user layer abstracts a variety of needs for education, outreach, research, and system development and management purposes.

4 Wireless sensor networks

4.1 System architecture of sensor nodes

The WSN hardware platform used in our current design is the IRIS mote from Crossbow Technology [12]. The IRIS mote provides a highly integrated, cost-effective hardware solution for low-power WSN applications. At each sensor node, a soil moisture sensor probe is connected to an optional MDA300 data acquisition board. The base station (BS) node is installed on an extension board MIB510, which interfaces with the RFG Server through the RS232 serial port. The data collected by motes are periodically transmitted to the BS node through multihop communications. Then, the BS node transmits aggregated data to the RFG Server through serial port. To accomplish long-term operation with minimum human intervention, motes are powered by solar cells and rechargeable batteries. The capacities of the rechargeable battery and the solar cell are determined through power budget analysis. A detailed power budget analysis of our current implementation is presented in Sect. 7.

Figure 2 shows the functional block diagram of sensor node as we have implemented in this research. 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 accomplish the data generation task, sensor readings are collected periodically at certain frequency and sensor data are time-stamped upon sampling, which necessitates global time synchronization in the network. Then, in the data processing task, sensor nodes calibrate, aggregate, summarize, and compress the data. For example, data aggregation and summarization techniques such as the E2K method [13] may be used to reduce communication load by exploiting the spatiotemporal correlation properties that are inherent in many sensor data. Lastly, during the data reporting task, data are transmitted to the BS node through multihop wireless communications. The data reporting task is enabled by a variety of software services as shown in Fig. 2, which implements essential timing, communication, and networking protocols for energy-efficient multihop data collection in distributed networks.

The manufacturer of the motes provides a series of tools for compiling, building, and testing motes programs. The

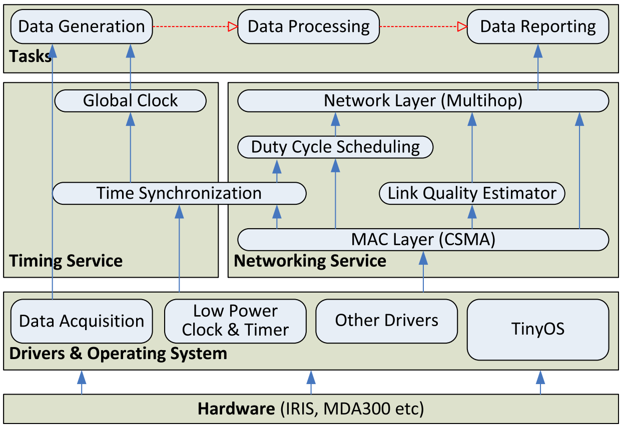


Fig. 2 Functional block diagram of sensor node

embedded TinyOS 1.1 source tree includes most of the essential drivers for motes and sensor boards, as well as a proprietary full featured multi-hop, ad-hoc, mesh networking protocol stack, XMesh. Unfortunately, the XMesh stack does not offer duty-cycling operation in IRIS platform and its non-open-source nature does not allow us to customize it for application-specific requirements. Therefore, we decided not to adopt the XMesh solution but design and develop our own networking protocol stack. The component-based NesC language, with which TinyOS is written, promises that designers can easily compose new applications and services by wiring together existing components. However, implementing a complete working system is not as easy as bringing building blocks together. For example, we tried to replace the default Media Access Control (MAC) protocol in TinyOS (i.e., CSMA) with the more power-efficient S-MAC [14] and combine it with the default multihop routing protocol. The result turned out to be disappointing: they failed to form a stable routing tree. The default multihop routing protocol uses active probing as well as eavesdropping for link quality estimation, which works well with CSMA. However, S-MAC implements overhearing avoidance mechanism and thus yields inaccurate and erroneous link quality estimation, which is detrimental to proper routing path selection. Moreover, it is necessary to restructure or even redesign most of the building components, especially the networking protocols, to construct a reliably functioning system. It is well understood that in a resource-constrained platform such as wireless sensor node, the stack of protocols should be jointly optimized in order to maximize the overall network performance and to minimize energy consumption.

A number of lower-power protocols are available in the literature, but most of the existing ones are not optimized for environmental monitoring applications. For example, in environmental monitoring, distributed sensor nodes typically form a spanning-tree structure, rooted in a single data-collection BS (or sink) node, so that majority of the data traffics are from sensor node to the BS node. However, most of the existing protocols are intended for peer-to-peer applications where data are routed between any pair of nodes in the network. Such a system architecture usually introduces a large amount of communication overhead and cannot exploit the unique tree-structure to minimize energy consumption. In the next section, we briefly describe the networking protocols that we have developed for environmental monitoring applications.

4.2 Networking protocols for multihop data collection

As discussed in Sect. 2, energy efficiency is one of the major design considerations in environmental monitoring sensor networks. Wireless communication and networking are among the most energy-consuming operations that a node performs so that the issue of energy efficiency must be addressed in the design of networking protocols. In wireless networks, MAC layer protocols are broadly categorized into two groups, schedule-based and contention-based methods [14, 15]. In schedule-based protocols such as time, frequency, and code division multiple access methods (i.e., TDMA, FDMA, and CDMA), wireless devices are scheduled to occupy different channels that are physically or logically independent. In contrast, in contention-based protocols such as the carrier-sense multiple access (CSMA) method, wireless devices compete for a single shared channel.

A contention-based protocol is highly autonomous but relatively energy-inefficient due to high collision rate in the shared channel and idle listening. A schedule-based protocol may completely eliminate overhearing and collision among neighboring nodes to achieve high energy efficiency, but it may suffer co-channel interference from other types of devices operating in the same frequency band, especially in the unlicensed ISM band that sensor networks typically employ. Thus, schedule-based protocols are widely used in single-hop, many-to-one systems with licensed band such as cellular networks. However, in multihop WSN, though most of the traffics are many-to-one, some of the control and signaling packets must be broadcasted among neighboring nodes in order to establish multihop routes. Consequently, most of the MAC protocols designed for sensor networks adopt CSMA as the baseline mechanism, and implement time-slot scheduling algorithms to coordinate duty-cycling [14, 16–18]. For example, a simple duty-cycling scheme is implemented in [19], where a mote keeps radio on for 1 s in a 2 min period. In [20], the authors employ the S-MAC [14], which is based on CSMA and synchronous sleep scheduling. Both of the soil moisture sensor networks presented in [19, 20] are limited to one-hop range, making the systems not scalable to capture spatial variation characteristics in a large area.

In this research, we develop a hybrid MAC layer protocol that integrates CSMA and duty-cycle scheduling to achieve high energy efficiency to support long-term, low-rate, and large-scale sensor network applications. The hybrid MAC protocol uses a distributed duty-cycle scheduling algorithm to coordinate sensor nodes' sleeping. Similar to most of the low-power MAC protocols available in the literature, our protocol divides the entire time axis into super frames, each of which is then divided into time slots, as illustrated in Fig. 3. Each super frame begins with a signaling slot where all nodes actively broadcast and receive packets. Sensor nodes compete for TDMA slots and exchange control information with neighbors during this period. The duty cycle of each node is scheduled by its parent; that is, a parent node assigns a TDMA slot to its

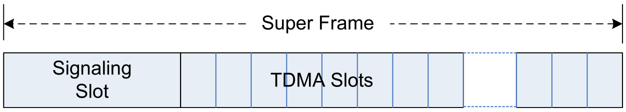


Fig. 3 Time slot structure of a super frame in the hybrid MAC protocol

child upon request. Then, sensor nodes turn off their transceivers and remain asleep except in their own active TDMA slots to conserve energy. During each TDMA slot, CSMA is still used to avoid any unexpected collision due to inaccurate time synchronization, co-channel interference from other types of devices, etc. Thus, the hybrid protocol strives to retain the flexibility of contention-based protocols while improving energy efficiency in multihop networks.

To guarantee the uniqueness of the TDMA slot assignation, we adopt a slot reservation protocol similar to the RTS-CTS collision avoidance mechanism in the 802.11 standard. Each node maintains a network allocation table (NAT), similar to the NAV in 802.11. But instead of keeping only the duration field, NAT keeps track of the TDMA slot allocation of neighboring nodes. In a many-to-one sensor data collection network, all of the data packets are routed from child nodes to their parents. Hence the slot assignment involves only the parent–child pair. Specifically, a child node randomly picks a slot that is not in the NAT, i.e., not occupied by any neighbors and requests for the slot by sending a request (REQ) packet to its parent. The parent node responds with a reply packet (RPL) if the requested slot is not in its NAT, i.e., not in use. Then the child node acknowledges the slot allocation and concludes this request. Other neighboring nodes eavesdropping the conversation updates their NAT, which in turn is used to ensure that when a node negotiates slots with its parent, the slots occupied by their neighbors will not be reused.

The TDMA slots are exclusive for communications between parent–child pairs. With the help of the slot assignation algorithm, we can guarantee that, for any TDMA slot, there will be only one parent–child pair active in any two-hop neighborhood. In a multihop monitoring system such as ours, most of the data traffic comes from relaying measurements along the routing tree. The TDMA slots are mainly designed to provide a collision-free channel for forwarding such data packets. Additionally, other neighboring nodes within the vicinity of the parent–child pair can save energy by avoiding overhearing unnecessary packets. This is the major difference between our duty cycle scheduling protocol and most of the existing protocols that are designed for peer-to-peer applications, where data are routed between any pair of nodes in the network. For example, in S-MAC, neighboring nodes form virtual clusters and the nodes in the same cluster share a common slot. The slot is not uniquely assigned but randomly picked. Thus, to transmit a packet, a node competes for the slot not only with the nodes in the same cluster, but also with nodes from other clusters. To avoid possible collision, the RTS-CTS procedure is used in every transmission, which results in a large amount of overhead. For an irrelevant node, which is neither the sender nor the receiver in the cluster, it still needs to wake up in order to receive potential incoming packets. Though S-MAC implements an overhearing avoidance mechanism where a node turns off transceiver when receiving RTS that is not addressed to it, it is inevitable for the node to overhear the RTS packet and waste energy in turning transceiver on and off.

T-MAC [16] follows the design of S-MAC but outperforms S-MAC with variable load by introducing adaptive duty cycling scheme. However, it suffers from the same problem as S-MAC. B-MAC [17] utilizes an unsynchronized approach, which is different from S-MAC and our hybrid protocol. In B-MAC, sensor nodes independently follow a sleeping schedule based on target duty cycle and periodically sense channel activity. Before message transmission, the sender is required to transmit a very long preamble to wake up every node in the neighborhood, including the receiver. Since sensor nodes are not synchronized, the preamble must be longer than the sleep period so that the receiver is able to detect it. Although B-MAC eliminates the overhead of time synchronization, it spends considerable amount of energy in sending the wake-up preamble. However, time synchronization cannot be removed from our system since it is required by other components as well. Z-MAC [18] is yet another TDMA-CSMA hybrid protocol. Similar to our protocol, Z-MAC employs a distributed slot assignment protocol called DRAND, to ensure unique slot assignment. Prior to normal operation and when topology changes, DRAND protocol needs to be executed to assign slots. The large amount of overhead associated with DRAND makes it unsuitable for networks with frequently changing topology. In addition, Z-MAC is built on top of B-MAC so that it inherits the shortcomings of B-MAC that we have just discussed.

The performance of schedule-based MAC protocol depends on the accuracy of time synchronization. The flooding time synchronization protocol (FTSP) proposed in [21] time-stamps synchronization messages at the MAC layer, which removes the non-deterministic delay at both sender and receiver caused by uncertain processing time in the operating system for context switches, system call overhead, interrupt handling, and other essential operations. The FTSP is able to synchronize multiple receivers with a single broadcast message. Such a flooding-based method is also insensitive to topological changes. In a tree-structured network, if every node synchronizes to its parent, ultimately all of the nodes in the network could synchronize to the root to achieve global time synchronization.

In this research, a modified FTSP is developed to exploit the unique tree-structure of environmental monitoring WSN. Unlike FTSP where the synchronization root is randomly picked up, every node in our network synchronizes to BS, which has access to the real-time clock. In the new design, similar to FTSP, each node maintains a buffer containing the latest time stamps for estimating clock skew and offset. The buffer window is also used for outlier detection to filter out corrupted time measurements. In our experimental 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. However, in low-power modes, FTSP results in errors of several hundred milliseconds. Through carefully examination of the timer driver shipped with FTSP, it is found that the driver is not able to return consistent time stamps in low-power modes. In TinyOS, timer driver is required to implement two types of timer interfaces: one-shot timer and repeat timer. A one-shot timer fires only once whereas a repeat timer fires periodically until being called off. All existing timer drivers rely on a single hardware clock to handle both types of timers. To support the one-shot timers, which usually fire in a few milliseconds, the clock has to run at high frequency, which leads to high CPU usage and high power consumption. 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, we developed a new two-layer timer driver to replace the original drivers, which employs two individual hardware clocks to tackle the two types of timer separately. A high speed clock is used to drive one-shot timers, which remains active for a short period of time in normal mode. On the contrary, a low speed and thus low power clock runs continuously to support the repeat timers. Two clocks are synchronized from time to time to ensure consistency in time stamps.

The multihop routing protocol at the network layer is responsible for establishing and keeping up the routing hierarchy in the distributed WSN. Although the routing structure in environmental monitoring applications is a simple tree and the data flow is almost one-directional, the dynamic and unreliable nature of wireless communications poses great difficulty in organizing and maintaining a reliable multihop routing hierarchy. In this research, we implement a link quality estimator based on the exponentially weighted moving average (EWMA) estimation method [22]. The weight used here is the normalized received signal strength of the synchronization packet, which will then be halved in each cycle. Such an estimator reacts quickly to potentially large changes in link quality while being stable enough when changes are small. The multihop routing protocol makes use of the link quality estimator to maintain a reliable routing topology.

5 Wireless telemetry system

5.1 Wireless telemetry system hardware design

To seamlessly integrate a variety of devices in the field, as shown in Fig. 1, we implement an RFG Server using a compact, rugged, ultra-low-power SBC TS-7260 from Technology Systems, Inc. [23]. The SBC provides a standard set of on-board peripherals and includes software power consumption control for on-board peripherals, making it ideal for power sensitive designs, such as solar or battery-powered embedded systems. To minimize energy consumption, the environmental monitoring system is automatically duty-cycled between the active and sleep modes. The sleep mode of SBC is enabled by the optional battery backup board TS-BAT3, which also serves as an embedded uninterruptible power supply (UPS) for contingency power support. The devices deployed in the field are commonly equipped with an RS232 serial port, including data loggers, wireless modem, and the WSN BS node. Thus, with five serial ports onboard, SBC is well suited to serve as a gateway server. Other alternative products in the market typically provide fewer serial ports and have much higher power consumption as compared with TS-7260.

The long-haul wireless communication from the field to the CDC Server is implemented by using a GPRS modem. GPRS, standing for General Packet Radio Service, is a packet-oriented mobile data service, available to the subscribers of the GSM cellular networks. The GPRS link is maintained by SBC using the Point-to-Point (PPP) protocol, a data link protocol commonly used to establish a direct connection between two nodes over serial cable, telephone line, cellular phone, or dial-up networks. Upon boot up, SBC automatically dials to the GPRS network and keeps the link alive during the entire active period. To enable secure system access, the Layer 2 Tunneling Protocol (L2TP) is used to support virtual private network that establishes a secure point-to-point connection between the RFG Server and the CDC Server through the public Internet. To be energy-efficient, the wireless modem is powered off during the system's sleep period.

Wireless telemetry system in the field is powered by solar energy with a large solar panel and a lead-acid rechargeable battery. The required capacities of the rechargeable battery and the solar panel are determined through power budget analysis. In power budget analysis, average power consumption of each power load device is determined by measuring or estimating the average current draw and the time spent in each of its operating modes. A detailed power budget analysis of our current implementation is presented in Sect. 7. To survive extreme weather conditions in long-term operations, we target at supporting the system with a fully charged battery for at least a week without recharging. In our system, the battery voltage level is closely monitored as a part of the remote system status monitoring service described in the next section. Near-real-time monitoring of such a system status data is important in determining battery efficiency and early detection of severe battery degradation to prevent system failure and the loss of important sensor data.

5.2 Remote data collection services

The SBC deployed in our design supports the full-featured Debian GNU/Linux, which may be customized to meet various low-power embedded computing needs. Thus, it is convenient to develop remote data collection services by taking advantage of the software packages that Debian provides, including a complete GNU C/C++ development environment, many Linux services such as PPP, FTP, Telnet, and MySQL database server, and various GNU/Linux libraries and utilities.

The RFG Server wakes up periodically to carry out data collection services. Upon boot up, the RFG Server executes a series of scripts to initiate various services, including an event logging daemon, a MySQL database server, an FTP server, an SSH terminal, and a Telnet terminal. A PPP daemon is also initiated to establish and maintain a PPP link to the GPRS network. The wireless modem is powered on at the same time as the RFG Server. Then, several independent data collection processes are started to poll data from the WSN BS node and dataloggers through RS232 ports. The data collected by the RFG Server are inserted into a local MySQL database, instead of being saved in the local file system or directly sent to the CDC Server through the wireless modem. The database server provides proficient data management that facilitates efficient data search, enables concurrent data access, minimizes data redundancy, enforces data integrity, and improves data consistency. With the in situ database, sensor data can be readily retrieved through a uniform interface and securely warehoused in the field, even in the event of network failures between the RFG Server and the CDC Server. After acquiring all data, the RFG Server notifies the CDC Server that new data is ready for retrieval. The CDC Server then synchronizes its database to the RFG database.

The duration of the database synchronization process is random in nature due to the inherent uncertainties in the amount of data to be synchronized and the traffic load condition in the network. Thus, instead of adopting a fixed-length active period, we implement a simple duty cycle negotiation protocol between the RFG Server and the CDC Server to enhance energy efficiency of the solar-powered remote monitoring system. As shown in Fig. 4, the CDC Server may request more time when it is needed. If synchronization is finished before the timer T2 expires, the CDC Server sends a Finish Sync command to the RFG Server to put it into sleep immediately. The RFG Server and the CDC Server are protected from potential network failures by the timers T1 and T2, respectively; that is, data collection process is terminated when the timers expire.

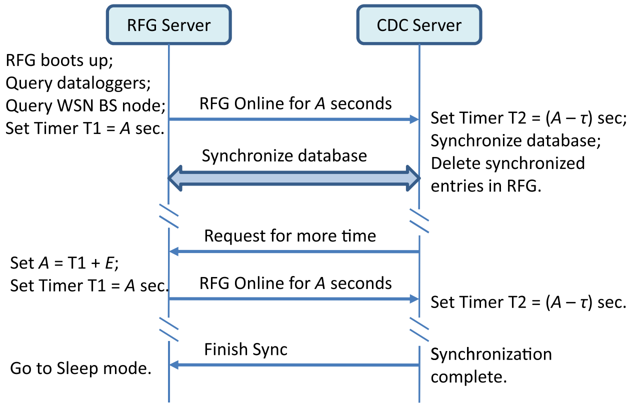


Fig. 4 Duty cycle negotiation protocol between the RFG Server and the CDC Server. The constant $\tau = \mathrm{RTT} + \varepsilon$, where RTT is the estimated round-trip time and ε is an appropriate guard-band time

Various system status data are also collected in the same way as sensor data to enable remote monitoring and management of the monitoring systems deployed in the field. Sensor nodes of WSN report system status along with sensor measurement data, such as battery voltage level, network topology data, and network performance statistics. The voltage level of the rechargeable battery, which powers the SBC, wireless modem, and dataloggers inside the station, is monitored by a datalogger and the battery voltage data is reported along with the wired sensor data. The RFG Server logs abnormal events in its local file system and reports to the CDC Server immediately as long as it is online. Authorized users can adjust system configuration such as duty cycle and sampling rate in near-real-time from the web portal by sending commands to the RFG Server through the CDC Server. Remote reprogramming of dataloggers and WSN follows the same steps, but because a large amount of data that needs to be downloaded from the CDC Server to the RFG Server, duty cycling of the RFG Server is temporarily disabled. Once the RFG Server receives a new program image, sensor nodes are reprogrammed through an over-the-air programming protocol.

5.3 Sensor data management

The CDC Server receives data through either pull or push operations. In a pull operation, the CDC Server periodically connects to the data source and pulls the data. In a push operation, the CDC Server opens a port, such as an FTP port, through which the data can be pushed by the data source. Such a mechanism allows the CDC Server to flexibly adapt to different types of data sources. The sensor data collected by the CDC Server is first archived in the local file system. For each data source, a back-end data handler (daemon program) is used to check the integrity of the data. Based on a set of predefined validation rules, the data are cleaned up before sending to the PostgreSQL DB Server. Data handler may also require the RFG Server to recollect and retransmit missing data packets. Data handlers run on separate user spaces to avoid conflict among different data sources. A new data handler is added for each new data source with minimal change in the database and web visualization layers. Therefore, the system's scalability and extensibility are greatly enhanced.

Sensor database 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. When more sensors are deployed as the system expands, they are registered as new records in the sensor relation by data handlers on the CDC Server. Observations from new sensors are added as new tables. The observations and sensors are linked through unique sensor identification codes. Thus, they will be automatically recognized by the upper-layer web applications once being added to the database. Such a design allows flexible system development and web interface design. It also facilitates conversion of the data to the SensorML format [24] to enable sensor data exchangeability and interoperability conveniently through web services.

6 Sensor data visualization and dissemination

As an integral part of the new environmental monitoring system, we have developed a dedicated web portal, TEO Online [9], with a set of publicly available web services for sensor data visualization and dissemination. In designing the data visualization framework, we take full advantage of the flexible Google Maps APIs [25] to associate sensors with their geographical locations intuitively on the satellite-view map interface as shown in Fig. 6. Each sensor is represented by a KML (Keyhole Markup Language [26]) placemark and displayed as an interactive marker that links to a drilldown information page for detailed observation charts. Such an interface provides direct visualization of spatial distribution of sensors and sensors data, which is an important aspect of environmental studies.

Designed with flexibility in mind to address various user interests, 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.), sensor location, or type of sensors (wired, wireless, etc.). In the drilldown information page, real-time and historical sensor data as well as their temporal variation statistics may be viewed in several different formats. Depending on the type of natural phenomenon, various types of single- and multi-series flash charts are provided. For a single-series chart, users can adjust the data reporting interval and the zooming level of time range to view data trends at different temporal granularities. Several predefined functions allow the analysis of data statistics such as average and maximum values. With a multi-series chart, users can compare readings from sensors deployed at different locations to analyze spatial distribution characteristics of natural phenomena.

Data interoperability rises as an important issue with the wide application of sensor networks at different fields.

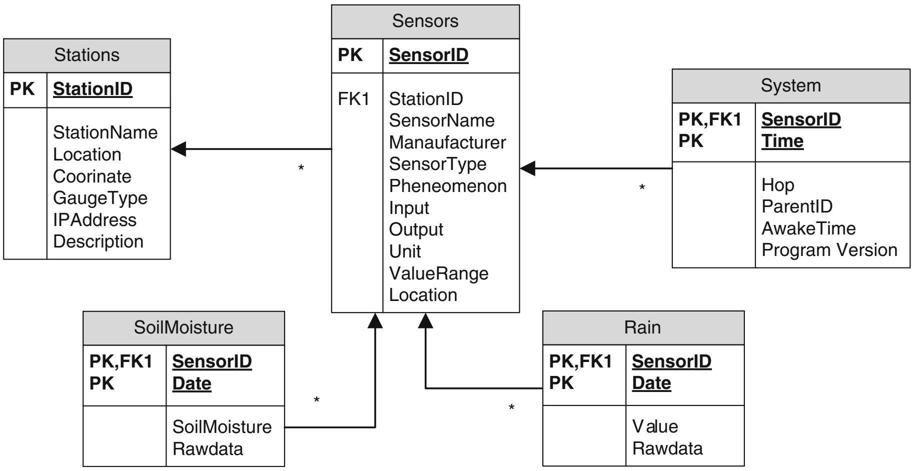
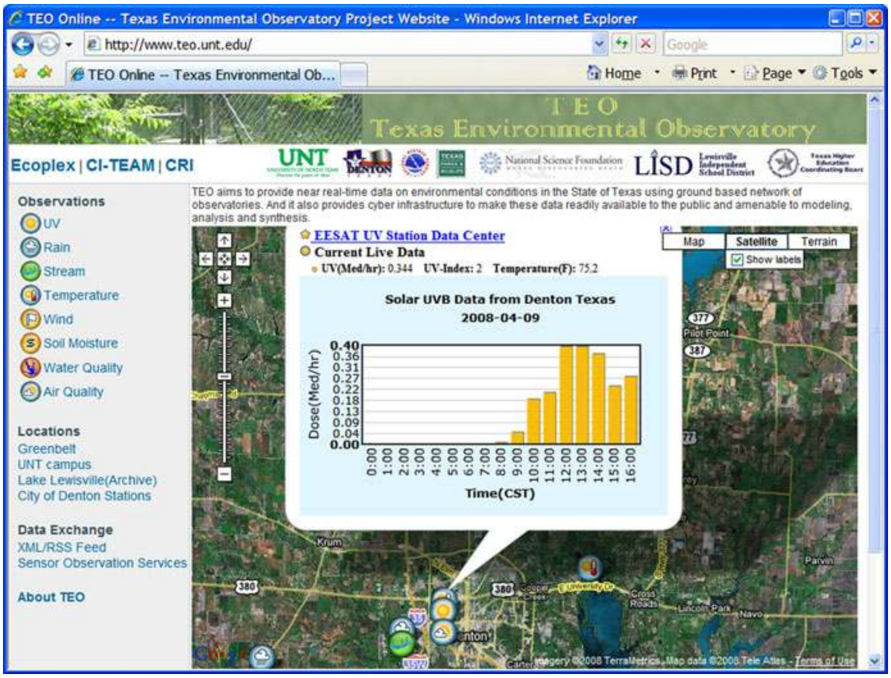


Fig. 5 Illustration of the database schema

Research in wireless sensor networks has seen increasing interests in data exchange and interoperability [27–31]. In our systems, we develop a flexible framework aiming at tailoring data exchange requirements of both general users and domain experts. Data exchange can be performed through Extensible Markup Language (XML) data exchange, Really Simple Syndication (RSS) feed, and Sensor Observation Services (SOS), as shown in Fig. 7.

The sensor data interoperability is achieved by a dedicated backend SWE data repository, as well as front-end RSS feed and web services built upon the repository [32]. RSS feed has been widely used to publish frequently changing data and allow users to subscribe to the data. RSS feed items and links are stored in an RSS table in the repository, while the live data is encapsulated in the RSS page by the web layer RSS class functions. General users can subscribe to RSS feed and retrieve live data automatically using client side RSS syndication tools that are widely available nowadays. For example, when a user first visits our RSS feed page using an up-to-date web browser (such as the latest Internet Explorer, Firefox, etc.), a "Subscribe Now" button (or a "Subscribe to this feed" link) will be shown, through which the feed can be subscribed to. Alternatively, a user can also add the feed address into the subscription library of dedicated RSS tools (such as Google Reader). After the subscription, user will



TEO aims to provide near real-time data on environmental conditions in the State of Texas using ground based network of observatories. And it also provides cyber infrastructure to make these data readily available to the public and amenable to modeling, analysis and synthesis.

Current Live Data

UV(Med/hr): 0.344 UV-Index: 2 Temperature(F): 75.2

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

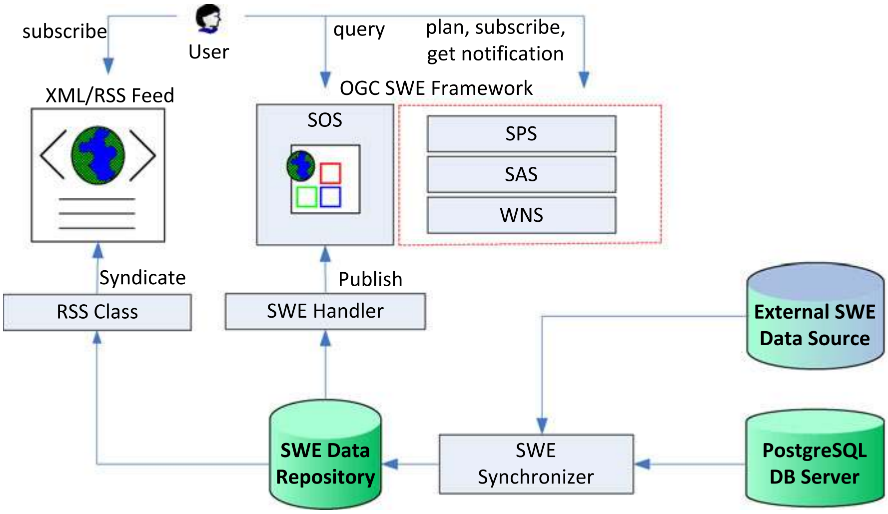


Fig. 7 Sensor data dissemination and exchange framework

be able to receive the most recent sensor observation data automatically though the RSS tool without visiting the TEO website.

For domain experts, we provide an even more powerful and flexible data exchange interface through the OGC's SWE framework. The SWE is a new standard that specifies interoperable interfaces and metadata encodings to enable real-time integration of heterogeneous sensor data [11]. Major encoding standards include SensorML that describes sensor system information and O&M (Observations and Measurements) that encodes actual live data [24]. The interoperability interface standards include SOS (Sensor Observation Services), SAS (Sensor Alert Services), SPS (Sensor Planning Services), and WNS (Web Notification Services). Specifically, SOS provides an API for near-real-time retrieval of sensor observation data, SAS provides standard web service interface to allow subscription to alerts from sensors, SPS serves as an intermediate layer between the user interface for requesting user-driven acquisitions and the sensor system, and WNS allows asynchronous transmission of sensor alert messages.

These modules collaborate together and can be used to customize various web-based or desktop-based applications for environmental scientists. For example, when designing web-based applications, the standard HTTP POST method can be used in any web forms to retrieve data from our SOS server (http://sensorweb.cse.unt.edu:8080/teo/sos). The HTTP requests are expressed using a few simple messages in XML format. A "DescribeSensor" message with the parameter of a sensor name (defined within the "procedure" tags) can be used to get the sensor schema information as a SensorML document. A "GetObservation" message with the parameters of data type, time, phenomena, and response format may be used to retrieve the sensor observation data as an O&M document. The details of each request message and response format can be found in the SWE standards [11]. TEO website also provides a sample web-based SOS application at http://www.teo.unt.edu/xml/sos.php to demonstrate the usage of web service requests. The returned results from our SOS server are also in XML format that can be easily parsed and converted into various formats needed for domain specific purposes. Similarly, when a desktop-based client application is preferred, different desktop interface development tools such as Java Development Kit (JDK) and Adobe Flex may be used as long as such tools support the above standard web service message formats. In JDK, for instance, web services can be supported using the Metro service stack [33]. Most other development kits also have similar built-in support for such standards.

Our database is syndicated with a sensor table, which automatically enables conformation to the SWE standard. Thus, the SWE data repository can be easily synchronized with the DB Server through an SWE synchronizer. Meanwhile, the SWE synchronizer can also retrieve external data from any SWE-compatible data sources and confederate our repository. 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.

7 Deployment and field testing results

7.1 Deployment of sensor networks in the field

The GBC weather station in Denton, Texas, has been operational for nine years with temperature, solar radiation, rain gauge, wind speed and direction, and soil moisture sensors, all of which are connected by wire to dataloggers and are deployed inside a small fence-enclosed area. In March 2008, we expanded the GBC station by deploying a wireless modem, an SBC, and a small pilot WSN consisting of 8 motes, to implement the integrated system shown in Fig. 1. A snapshot of the station and mote deployment setup (in the lower left-hand side corner) is shown in Fig. 8.

One year later, in March 2009, we expanded the WSN to a deployment consisting of 16 motes (in two sets of eight motes each) along a cross-sectional transect as shown in Fig. 9. To support long-term hydrologic monitoring and modeling in the floodplain area of the Elm Fork of the Trinity River, each sensor node collects data every 10 minutes from soil moisture sensors (connected by wire to the motes) and onboard temperature and relative humidity sensors. This network topology provides an opportunity to collect a duplicated set of soil moisture variation along a cross sectional transect from the river bank (higher elevation and sandy soil) to the weather station (lower elevation

Fig. 8 The GBC weather station and the weatherproof installation box for motes

Fig. 9 Sensor deployment topology in the field as shown on the Google Map-based TEO Online web portal

and clay soil). Characterizing soil moisture variation with respect to elevation and soil type is vital to understanding vegetation distribution along the floodplain as well as responses to flooding.

To survive extreme weather conditions, we installed the motes in weatherproof boxes, and the boxes are installed 4 feet above the ground on top of metal poles to avoid flooding water and prevent fallen leaves from covering solar cells. Prior to the deployment, we conducted a site survey to measure the one-hop radio communication range between motes in the deployment environment that features densely populated trees and grasses. Radio propagation characteristics of the environment vary significantly over time due to seasonal variation of the vegetation in the area. From the measurement results collected in the summer, we observed that with a maximal transmission power of 3 dBm, IRIS motes are able to transmit on average 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. Sensors are not deployed at regular grid points, mainly because of the irregular layout of trails, trenches, trees, dense bushes, etc. in the field.

The TEO Online web portal [9] has been operational since March 2008 with most of the basic web services implemented. Various environmental sensor data and system status data from several monitoring sites can be viewed and downloaded from the web portal using the methods described in Sect. 6. Currently, the sensor data are being shared with the City of Denton and the National Weather Service in Fort Worth for various monitoring, modeling, and prediction purposes, which is made extremely convenient by the new data exchange framework that we implemented in this project.

7.2 System performance characterization and field testing results

As discussed in Sect. 2, energy efficiency is one of the major design considerations of remote environmental monitoring systems. Duty cycling provides an effective way to achieve energy efficiency. In the current setup, the RFG Server wakes up for 90 s every 10 min for data collection with a duty cycle of about 15%. The wireless modem is powered off during inactive periods. Table 1 shows the current draw and duty cycle of the devices deployed inside the GBC station, powered by a solar panel with a peak current of 960 mA and a 12 Ah lead-acid rechargeable battery. In practice, a lead-acid battery cannot be 100% discharged repeatedly. Therefore, it is necessary to de-rate the battery by some amount, generally 25% [34]. Thus, the battery deployed in the field may support the system for about 7 days without recharging. In general, the capacity of a solar panel should be at least 10 times the average power consumption of the load [34]; the solar panel deployed in the field meets such a requirement.

In our current implementation, each super frame of 4 s consists of a signaling slot of 40 ms and 99 TDMA slots of

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

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

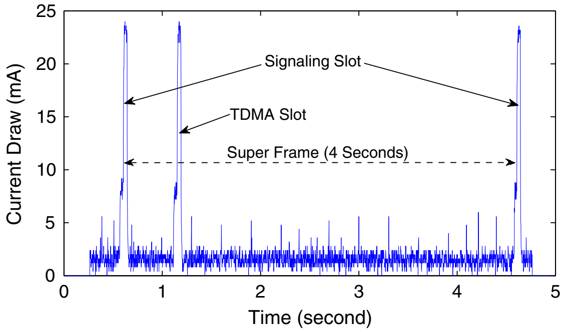


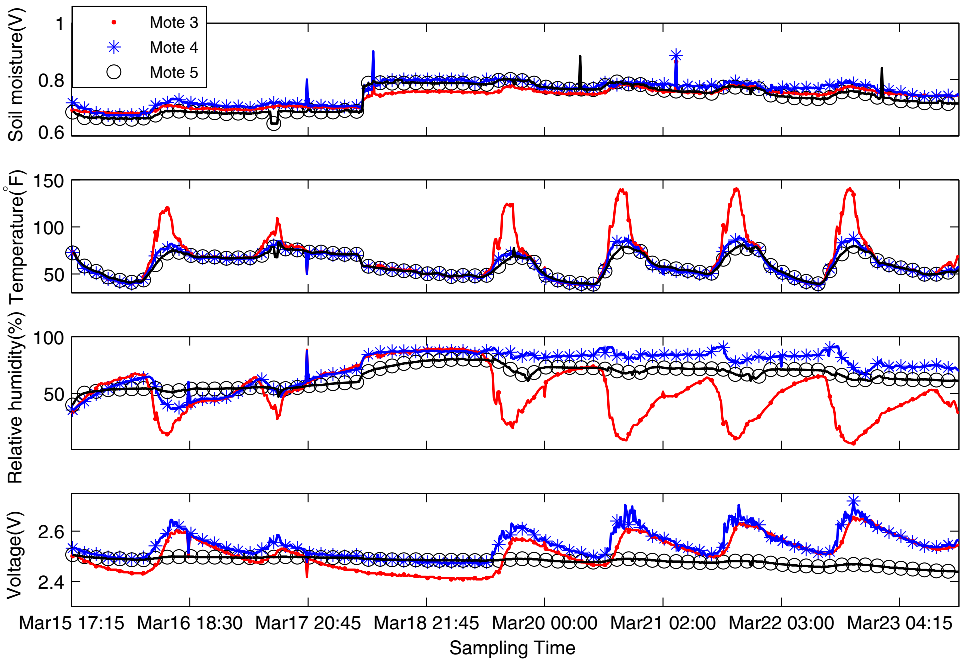
Fig. 10 Measurement of the current draw and duty cycle of a mote

40 ms each. With the duty cycle scheduling algorithm, motes are only active during a few TDMA slots to report and relay sensor data and a signaling slot to synchronize time, manage neighbor list, and update parent information. Other than these active periods, motes remain in the sleep mode and consume much less power than in the active mode. Figure 10 shows a measurement result of current draw and duty cycle, captured with an oscilloscope, of a non-parent node (i.e., leaf node in the tree structure) from an experimental setup in the laboratory. A parent-node will have at most 9 active TDMA slots within a super frame with a maximum of 8 children, which leads to a duty cycle of no more than 10% for a mote.

An average current draw of 24 mA in the active mode shown in the figure matches well with the manufacturer's data. In the sleep mode, the radio transceiver is disabled and the CPU wakes up occasionally to handle hardware interrupt routines and software events, such as timer services and counter updates, in order to maintain network stack and remain synchronized. According to the manufacturer's data, the average current draw is about 16 µA when both radio transceiver and CPU are in the deep sleep mode. With all peripheral devices turned off we can achieve a minimum of 90 µA in sleep mode; however, the use of the optional MDA300 board and the LED indicators results in an average current draw of 1.53 mA. In our implementation, based on the measurement results shown in Fig. 10, the average current draw of a mote is about 2.73 mA with 5% duty cycle. Thus, two fully charged 2500 mAh NiMh batteries with a self-discharging rate of 30% can sustain a mote for about 4 weeks without recharging. The solar cell used to power motes is able to provide 100 mA peak current at 3 V, about 36 times the average current draw of the load; a much larger design margin is selected for the solar cell (powering motes) than the solar panel (powering the devices inside the weather station) because solar cells, being installed closer to the ground than the solar panel, are more likely to be shadowed by trees in the forest.

Figure 11 shows a subset of data collected by three motes of the 2008 pilot WSN during field testing (one year prior to the deployment of the transect sensor network discussed above). On March 18, 2008, the GBC area experienced heavy rainfall and a significant drop of temperature. The rain event was monitored by a rain gauge connected to a datalogger. The variation of weather condition was also captured by the sensors on motes as shown in Fig. 11. The soil moisture exhibits a jump on that day while the temperature was falling. The difference in data among the three soil moisture sensors reveals spatial variation characteristics of the soil moisture condition in that

Fig. 11 Sample sensor data collected in field testing



area, which is an invaluable input to the hydrologic modeling research. As a part of the field testing, motes are installed in two different boxes to test packaging alternatives. 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 cell installed outside the box, and Mote 5 uses the same non-transparent enclosure but without solar panel. The results clearly show the daily voltage charging and discharging phenomena from Mote 3 and Mote 4 while the voltage of Mote 5 dropped about 0.05 volts for a week. Also notice that solar cell did not work during rainy days. Additionally, we can observe that Mote 3 experiences significantly higher temperature during the day than the other two as a result of the greenhouse effect inside the transparent box, which also results in much lower relative humidity readings for Mote 3. The temperature can be very high in Texas in the summer. The IRIS mote is designed to operate in harsh environments, but the rechargeable NiMh batteries cannot tolerate high temperature. Thus, we are currently evaluating more packaging options for survivable long-term deployment in the field.

Table 2 shows a few statistics of the transect sensor network status data that we have collected from field tests over a one-month period. The distance between each node and BS is determined using GPS coordinate measurements. From the hop count measurements (between each node and BS) we can clearly observe that 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. The average duty cycle of each node is roughly around 5% and we do observe the tendency of higher duty cycle for the nodes with larger average hop count values. The data reception rate shown in the table is the percentage of the data that are successfully received by BS from each sensor node while each node originates one data sample in every 10 min. In our current implementation, we do not have end-to-end acknowledgement and end-to-end retransmission mechanisms so that as shown in the results, the data reception rate decreases almost exponentially as the hop count increases. For analysis and modeling, the missing data are being reconstructed through interpolation by exploiting the temporal structure of the data of individual sensors. But, we are exploring the use of the full spatiotemporal structure of the data to reconstruct missing data points. On the other hand, we are also developing alternative networking protocols to achieve end-to-end reliability in multihop data collection sensor networks and to study the tradeoff between reliability and energy efficiency in practical application scenarios.

7.3 Analysis of soil moisture sensor data

Soil moisture data from the sensors installed in the duplicated transect according to Fig. 9 were analyzed for a 12-day time period from March 25, 2009–April 02, 2009. We will refer as south (S) transect to the sensors numbered {15, 17, 13, 8, 2, 11, 12, 16} and as the north (N) transect to the sensors numbered {14, 9, 10, 6, 3, 4, 7, 5}. Each set of sensors is given in the order in which they appear starting from the river bank.

Because the time-stamps of sensors vary in the order of several minutes within the 10 min sampling interval, all

Table 2 Statistics of sensor network status data collected from field tests

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

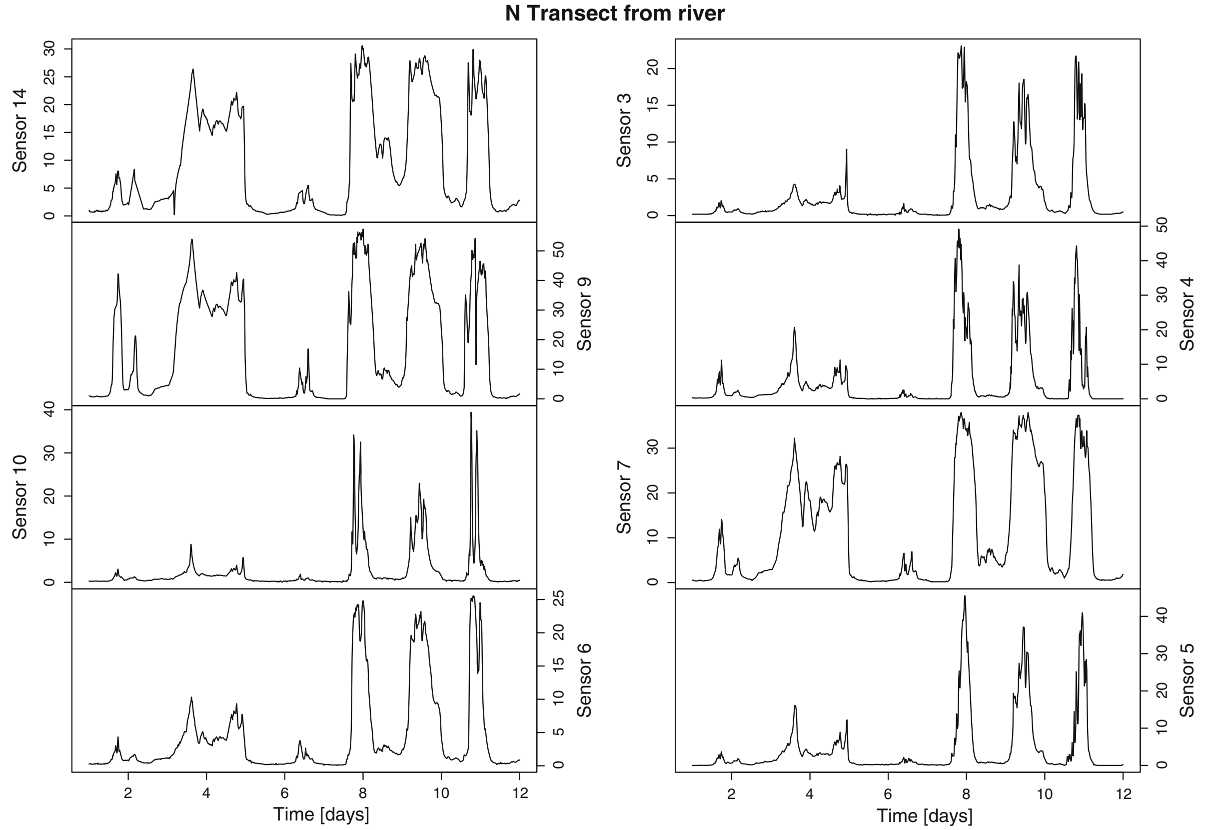


Fig. 12 Soil moisture (in %) sensor data along the N-transect

data were interpolated to the 05 min instant of the interval, which is to say minutes 05, 15, 25, etc. Figure 12 shows the time series obtained from all sensors along the N-transect. We can clearly observe how all sensors show similar time responses, but differ in magnitude. The time series for the S-transect displays similar results and thus it is not shown in this paper for the sake of space.

To analyze the relationships among the sensors, pair-wise cross-correlation functions were calculated along each transect starting from the sensor at the river bank as shown in Fig. 13. From the results, we can observe that correlation decreases as moving away from the river bank, while the maximum correlation shifts to positive time lags of about 0.1 days. An exception is Sensor 7, which shows high correlation at almost zero lag. This is the type of results that can then be explored in the field to understand changes along the elevation profile. Cross-correlation analysis can lead to using the data to build forecasting models of soil-moisture at various points of the floodplain, a valuable tool for flooding prediction.

Spatial patterns were analyzed by looking at variations along the two transects at the same time or as time-average for periods of interest. In particular, Fig. 14 shows the time average for the entire 12-day period for each transect, including an indication of time variability (vertical bar with extremes at mean ± 1 standard deviation). It can be appreciated how variability is lower for the S-transect, and how some locations (such as the one given by Sensor 9) are highly variable. Although both transects start at similar values of soil moisture for the river bank, the locations further from the river bank display contrasting magnitudes of soil moisture. Such spatial pattern can be correlated with vegetation distribution to understand plant response to soil moisture.

8 Summary and future works

A remote near-real-time environmental monitoring system developed to support long-term environmental studies is presented in this paper with a focus on the overall system architecture for seamless integration of the emerging WSN-based system and the legacy wired sensor system. A unified framework for sensor data collection, management, dissemination, and exchange is also presented. Because of space limitation, many technical details of the system components are not covered in this paper; more design and implementation details of the soil moisture monitoring WSN and the web services for sensor data visualization, management, dissemination, and exchange will appear in separate publications.

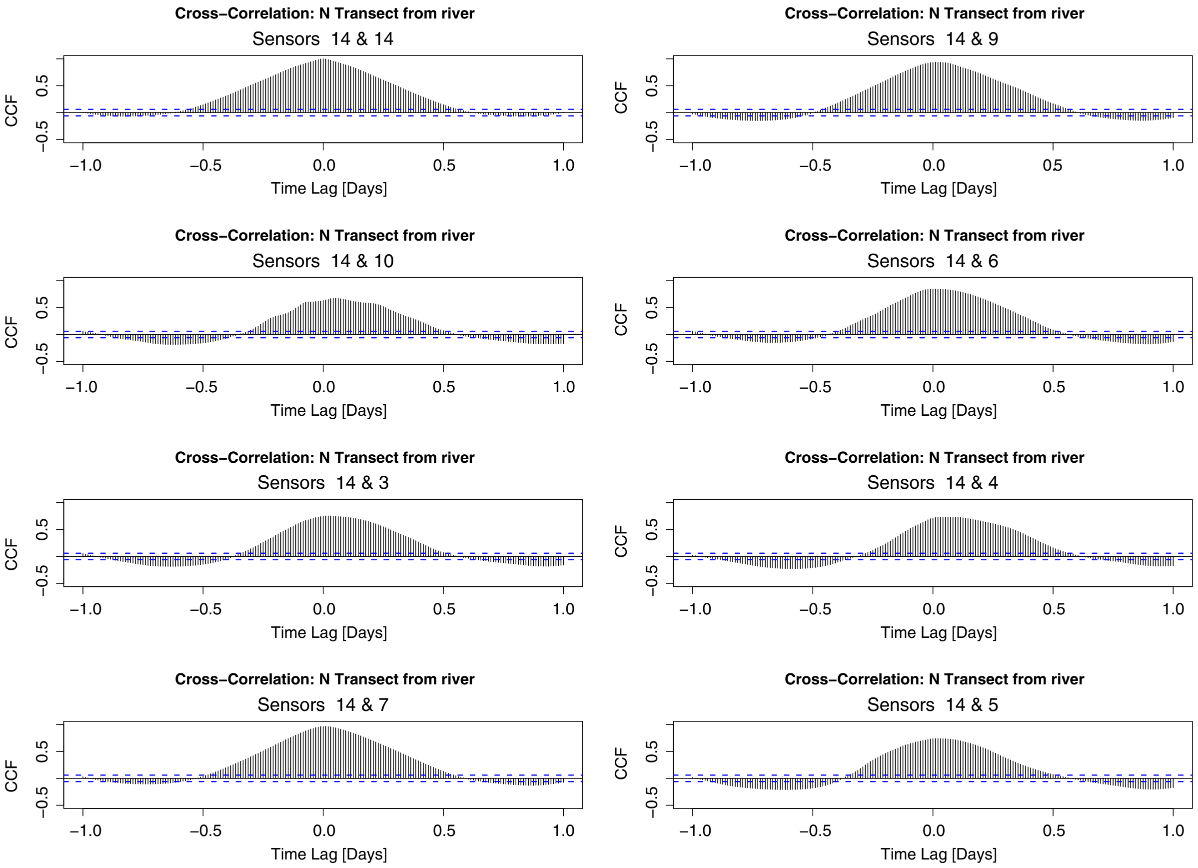


Fig. 13 Pair-wise cross-correlation functions along the N-transect

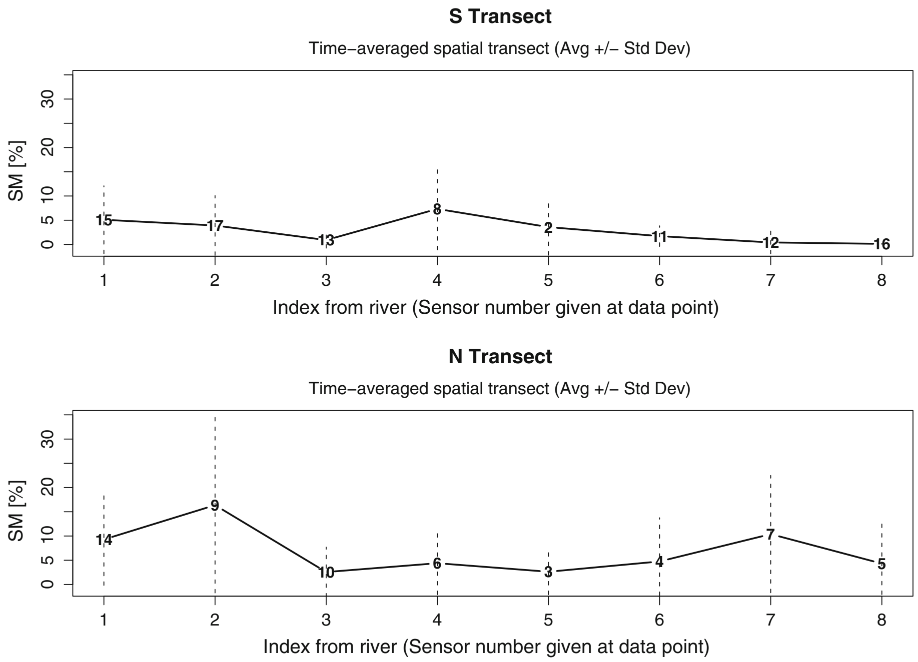


Fig. 14 Time-average data for the entire 12-day period for spatial pattern analysis

Currently, with support from the National Science Foundation, we are in the process of scaling up the soil moisture monitoring WSN at the GBC site to around 100 motes to have a much larger geographic coverage than the current deployment [9]. In addition, in cooperation with the City of Denton, the current remote monitoring system with

WSN and wireless telemetry are being replicated at five other weather stations across the North Texas area. We are also deploying a large-scale WSN of around 100 motes in the GBC site to serve as an open research infrastructure for the WSN research community. To fulfill this goal, secure web services are being developed for remote over-the-air programming and configuration of WSN. Access to such web services will be provided to interested researchers who may carry out experimental study of WSN from thousands of miles away. The environmental monitoring cyber infrastructure presented in this paper has been very useful for education and outreach purposes as well. More publicly available web services will be developed for visualization of raw sensor data and modeling results to better inform the public and government decision makers about the environmental conditions and to help them make environmentally responsible decisions.

Acknowledgements This work is supported in part by the National Science Foundation of the United States under Grants OCI-0636421, CNS-0709285, and EEC-0431818 and by the Texas Advanced Research Program under Grant 003594-0010-2006. The authors would like to thank David Hunter (City of Denton), Sonny Solis (Texas Parks and Wildlife), Bob Carle (National Weather Service), Wilfredo Franco (ULA), and members of the TEO Research Group at UNT, especially Rudi Thompson (Biology), Duane Gustavus (CITC), Adam Skelton (EE MS student), Sanjaya Gurung (EE MS student), and Liping Chen (Applied Geography MS student) for their contributions to the development of the TEO system.

References

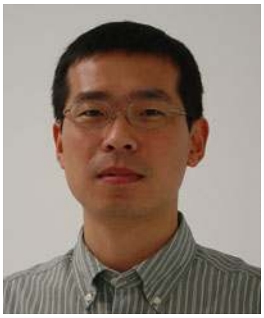
1. Culler, D., Estrin, D., & Srivastava, M. (2004). Overview of sensor networks. *IEEE Computer, 37*, 41–49.
2. Martinez, K., Hart, J. K., & Ong, R. (2004). Environmental sensor networks. *IEEE Computer, 37*, 50–56.
3. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., & Anderson, J. (2002). Wireless sensor networks for habitat monitoring. In *Proceedings of the ACM international workshop on Wireless Sensor Networks and Applications (WSNA)*.
4. MoteLab. Harvard sensor network testbed. Available at http://motelab.eecs.harvard.edu
5. ExScal. Extreme scale wireless sensor networking. Available at http://cast.cse.ohio-state.edu/exscal
6. Kansei. Sensor testbed for At-scale experiments. Available at http://ceti.cse.ohio-state.edu/kansei
7. CitySense. An open, urban-scale sensor network testbed. Available at http://www.citysense.net
8. SCADDS. Scalable coordination architectures for deeply distributed systems. Available at http://www.isi.edu/scadds/testbeds
9. Texas Environmental Observatory. Available at http://www.teo.unt.edu
10. Cheng, S., & Wang, R. (2002). An approach for evaluating the hydrological effects of urbanization and its application. *Hydrological Processes, 16*, 1403–1418.
11. Sensor Web Enablement WG., Open Geographic Consortium Inc. Available at http://www.opengeospatial.org.
12. Crossbow Inc. Available at http://www.xbow.com
13. Harrington, B., Huang, Y., Yang, J., & Li, X. (2009). Energy-efficient map interpolation for sensor fields using Kriging. *IEEE Transactions on Mobile Computing, 8*(5), 622–635.
14. Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 21st international annual joint conference of the IEEE Computer and Communications Societies (INFOCOM)* (Vol. 3, pp. 1567–1576).
15. Stallings, W. (2004). *Wireless communications & networks*, 2nd edn. Prentice Hall.
16. van Dam T., & Langendoen, K. (2003). An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the international conference on Embedded Networked Sensor Systems (SenSys)*.
17. Polastre, J., Hill, J., & Culler, D. (2004). Versatile low power media access for wireless sensor networks. In *Proceedings of the international conference on Embedded Networked Sensor Systems (SenSys)*.
18. Rhee, I., Warrier, A., Aia, M., & Min, J. (2005). Z-MAC: A hybrid MAC for wireless sensor networks. In *Proceedings of the international conference on Embedded Networked Sensor Systems (SenSys)*.
19. Musaloiu-E, R., Terzis, A., Szlavecz, K., Szalay, A., Cogan, J., & Gray, J. (2006). Life under your feet: A wireless soil ecology sensor network. In *Proceedings of the third workshop on Embedded Networked Sensors (EmNets)*.
20. Cardell-Oliver, R., Smettem, K., Kranz, M., & Mayer, K. (2005). A reactive soil moisture sensor network: Design and field evaluation. *International Journal of Distributed Sensor Networks, 1*(2), 149–162.
21. Maroti, M., Kusy, B., Simon, G., & Ledeczi, A. (2004). The flooding time synchronization protocol. In *Proceedings of the ACM conference on Embedded Networked Sensor Systems (SenSys)*.
22. Woo, A., Tong, T., & Culler, D. (2003). Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the international conference on Embedded Networked Sensor Systems (SenSys)*.
23. Technologic Systems Inc. Available at http://www.embeddedarm.com
24. SensorML, UAH VAST. Available at http://vast.uah.edu/SensorML
25. Google Maps API, Google. Available at http://code.google.com/apis/maps/
26. Keyhole Markup Language, Google. Available at http://code.google.com/apis/kml
27. Delicato, F. C., Pires, P. F., Pinnez, L., Fernando, L., & da Costa, L. F. R. (2003). A flexible web service based architecture for wireless sensor networks. In *Proceedings of the 23rd international conference on distributed computing systems workshops*.
28. Zhang, J., Hart, Q., Gertz, M., Rueda, C., & Bergamini, J. (2009). Sensor data dissemination systems using Web-based standards: A case study of publishing data in support of evapotranspiration models in California. *Civil Engineering and Environmental Systems, 26*(1), 35–52.
29. Garcia, R. M., Carvalhal, P., Ferreira, M. J., Silva, L. F., Almeida, H., Santos, C., et al. (2009). A flexible framework for data exchange and presentation between wireless sensor networks and personal devices. In *The international conference on "Computer as a Tool"*.
30. Sleman A., & Moeller, R. (2008). Integration of wireless sensor network services into other home and industrial networks using device profile for web services (DPWS). In *The 3rd international conference on Information and Communication Technologies: From Theory to Applications (ICTTA)*.
31. Kawahara, Y., Kawanishi, N., Ozawa, M., Morikawa, H., & Asami, T. (2007). Designing a framework for scalable coordination of wireless sensor networks, context information and web services. In *The 27th international conference on Distributed Computing Systems Workshops (ICDCSW)*.

32. 52North Sensor Web, 52north.org. Available at http://52north.org/
33. Metro Web Services, Sun Microsystems, Inc. Available at http://java.sun.com/webservices/
34. NovaLynx Corporation. *Power budget calculations*. Available at http://www.novalynx.com

Author Biographies

Jue Yang received the B.S. degree in Computer Science at Fudan University, Shanghai, China, in 2005. He is currently a Ph.D. student in the Department of Computer Science and Engineering at the University of North Texas, Denton, TX. Since 2005, he has been with the Department of Electrical Engineering at the University of North Texas as a research assistant. His current research interests include statistical signal processing and wireless sensor network.

Chengyang Zhang received his B.S. degree in industrial automation from University of Science and Technology, Beijing, in 2000. He is currently a Ph.D. student in the Department of Computer Science and Engineering at the University of North Texas, Denton, TX. His research interests include location-based services, spatio-temporal databases, and sensor streaming data management.



Xinrong Li received B.E. degree from the University of Science and Technology of China, Hefei, China, in 1995, M.E. degree from the National University of Singapore in 1999, and Ph.D. degree from Worcester Polytechnic Institute (WPI), Worcester, MA, in 2003, all in Electrical Engineering. From 2003 to 2004, he was a post-doc research fellow at the Center for Wireless Information Network Studies, WPI. Since 2004, he has been with the Department of Electrical Engineering at the University of North Texas, Denton, TX, as an assistant professor. His recent research has been focused on statistical signal processing for geolocation, wireless communications, and wireless sensor networks.

Yan Huang received her B.S. degree in Computer Science from Beijing University, Beijing, China, in 1997 and Ph.D. degree in Computer Science from the University of Minnesota, Twin Cities, MN, in 2003. She is currently an assistant professor at the Computer Science and Engineering Department of the University of North Texas, Denton, TX. Her research interests include geo-sensor networks, geo-stream processing, spatial databases, and data mining. She is a member of the IEEE Computer Society, the ACM, and the ACM SIGMOD. Her research is supported by Texas Advanced Research Program (ARP), Oak Ridge National Lab, and NSF.

Shengli Fu received his B.S. and M.S. degrees in Telecommunication Engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1994 and 1997, respectively; the M.S. degree in Computer Engineering from Wright State University, Dayton, OH, in 2002; and the Ph.D. degree in Electrical Engineering from the University of Delaware, Newark, DE, in 2005. He is currently an assistant professor in the Department of Electrical Engineering at the University of North Texas, Denton, TX. His research interests include coding and information theory, wireless sensor network, and joint speech and visual signal processing.

Miguel F. Acevedo is a Regents Professor in the Department of Electrical Engineering and the Department of Geography at the University of North Texas, Denton, TX. During 1992–2007 he was in the Department of Geography and the Graduate Program in Environmental Sciences of the same university. He obtained his Ph.D. degree in Biophysics from the University of California, Berkeley (1980) and master degrees in Electrical Engineering and Computer Science from Berkeley (M.E., 1978) and the University of Texas at Austin (M.S., 1972). Before joining UNT, he was at the Universidad de Los Andes, Merida, Venezuela, where he taught since 1973 in the School of Systems Engineering. His expertise and interests are in ecological and environmental modeling; landscape and forest ecology; human–natural interactions, watershed and reservoir management; biosensors and environmental observatories; global climate change and variability. He has served on the Science Advisory Board of the U.S. Environmental Protection Agency (EPA).