INVITED PAPER

Environmental Wireless Sensor Networks

This paper reviews recent experiments with networks for environmental and agricultural applications; it also provides a critical review of recent research and considers future challenges and opportunities.

By Peter Corke, *Fellow IEEE*, Tim Wark, *Member IEEE*, Raja Jurdak, *Member IEEE*, Wen Hu, *Member IEEE*, Philip Valencia, *Member IEEE*, and Darren Moore, *Member IEEE*

ABSTRACT | This paper is concerned with the application of wireless sensor network (WSN) technology to long-duration and large-scale environmental monitoring. The holy grail is a system that can be deployed and operated by domain specialists not engineers, but this remains some distance into the future. We present our views as to why this field has progressed less quickly than many envisaged it would over a decade ago. We use real examples taken from our own work in this field to illustrate the technological difficulties and challenges that are entailed in meeting end-user requirements for information gathering systems. Reliability and productivity are key concerns and influence the design choices for system hardware and software. We conclude with a discussion of long-term challenges for WSN technology in environmental monitoring and outline our vision of the future.

KEYWORDS | Environmental monitoring; wireless sensor network (WSN)

I. INTRODUCTION

Wireless sensor networks (WSNs) are an important technology for large-scale monitoring, providing sensor measurements at high temporal and spatial resolution. The simplest application is *sample and send* where measurements are relayed to a base station, but WSNs can also perform in-network processing operations such as aggregation, event detection, or actuation.

The first WSN papers a decade ago [1] clearly articulated the promise of the technology for a diverse range of monitoring applications including forests, waterways, buildings, security, and the battlefield, and how it would transform the way we do science and business. One decade on, it is clear that progress has not been as fast as was predicted. Instead of smart dust sprinkled from aircraft we have large nodes[1] connected by myriad wires to transducers. The research community is still concerned with networking and maximizing the lifetime of networks powered from finite electrochemical primary cells. While many have reported sample-and-send systems with tens of nodes and operational durations from days to years, other features envisaged at the outset such as event detection, sensing *and* actuation, or the integration of robots and sensor networks have not become commonplace. It seems that the technology is still emerging.

WSN technology has followed a hype cycle [2] triggered by the availability of low-cost low-power feature-rich microcontrollers and single-chip radio transceivers. This led to excitement, expectation, the founding of startups, and the establishment of new conferences and journals. Early adopters embraced the technology as end users, willing to suffer the inconveniences of bleeding-edge technology in order to gain an advantage. Activity peaked and the trough of disillusionment followed. The early adopters became frustrated, the researchers felt that the early adopters were too needy, and the expected markets did not materialize. Then follows the slope of enlightenment when expectations are moderated all round, the bugs are ironed out, the real rather than supposed applications become

Manuscript received November 16, 2009; revised July 8, 2010; accepted July 8, 2010. Date of publication October 7, 2010; date of current version October 20, 2010.

P. Corke was with the Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Qld. 4064, Australia. He is now with the School of Engineering Systems, Queensland University of Technology, Brisbane, Qld. 4001, Australia (e-mail: peter.corke@qut.edu.au).
T. Wark, **R. Jurdak**, **W. Hu**, **P. Valencia**, and **D. Moore** are with the Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Qld. 4069, Australia (e-mail: tim.wark@csiro.au; raja.jurdak@csiro.au; wen.hu@csiro.au; philip.valencia@csiro.au; darren.moore@csiro.au).

Digital Object Identifier: 10.1109/JPROC.2010.2068530

[1]Typically 1 L in volume.

clear, and the field grows steadily. This paper reflects our experience on this journey for WSN technology.

An early claim was that WSNs are a new *instrument* for gathering data about the natural world [3] and our collaborators who are primarily scientists have high, and not unreasonable, expectations of something that purports to be an instrument. In particular, they expect a high level of system integration, performance, and productivity.

System integration means creating an end-to-end system that delivers data to an interested user. This meant that our mind set had to change from a narrow focus on just the WSN component to creating an information system. The WSN is just one small part of a complex system that includes internet links from the WSN to a server, databases, and web presentation tools. Each of these components was critical for success and we underestimated each of them. For example, the internet link from a sensor network to the server presupposes that an internet endpoint exists and this was not always the case—3G modems are ideal in principle but problematic in practice. We naively assumed that servers within our organization were always up, but did not know that network infrastructure upgrades happen late at night and on weekends, making those servers inaccessible from the outside world.

Performance has many aspects. One is reliability of the node itself: its power source, radio links and overlying protocols, and reliability of the application and operating system software. Lack of performance leads to gaps in the data record, negating the claim about high temporal precision, and incorrect data (due to all manner of root causes) lead to a lack of trust and confidence in the system. Accuracy and calibration are critically important—it is not enough that the network returns numbers to a central database which is the engineers concern; the numbers must accurately reflect the state of the environment [4]. Sensor calibration and detecting stuck values from broken transducers are very important.

Productivity has two aspects. Most important is how well it assists the end user to do their science or business, and is largely related to human interface design and the performance of the underlying database and presentation software. Databases and web presentation tools are simple student projects when they have to deal with trivial amounts of data, but as the data volume grows performance falls dramatically and will frustrate the end user and lower their productivity. For developers and maintainers of WSNs, productivity is about reducing the total cost involved in a sensor network over its lifetime—analogous to total cost of ownership (TCO) for a computer system. Costs that must be accounted for include planning, node hardware, deployment, troubleshooting, and maintenance. Only semiconductors follow Moore's law; the other costs follow a learning curve and are already mature.

Applications of sensor networks are very diverse but we use as case studies some of the many applications that we have tackled which are outdoor, and concerned in some way with the natural environment and/or agriculture. Our choice of applications was dictated by interest from our collaborators in those fields and the national challenges facing our country such as land degradation, water shortage, and climate change. Our collaborators have high expectations and they have been an important part of our learning.

These applications share common challenges such as robustness to unpredictable events and harsh outdoor conditions, and goals such as network longevity. In most of our deployments, delays in data delivery were not a major issue, neither was providing absolute guarantees on data delivery. This has guided our design choices and distinguishes our focus on outdoor network deployments from other sensor network application classes, such as industrial automation or indoor tracking.

In this paper, we attempt to explain the slow progress of the field by exploring our own progress, the pitfalls we experienced, and the lessons we learned. The next section of the paper discusses in chronological order a number applications, the technological challenges that they presented, and a summary of the technologies we developed in response to these challenges. Section III describes our current state-of-the-art end-to-end system. In Section IV, we discuss challenges for the field such as value proposition and alternative technologies, and in Section V, we take stock and discuss what can be done with the technology now and into the future.

II. APPLICATIONS

In this section, we discuss the major sensor network applications that we created over the past six years (Table 1), the unique technical challenges they posed, and the lessons we learned from them. The applications all share a common theme: understanding the natural and agricultural environments in response to major challenges faced by Australia. The applications include microclimate monitoring for farms and rain forests, water-quality monitoring, and cattle monitoring and control. Two applications involve actuation in addition to sensing: cattle are actuated by applied stimuli, and a robotic boat is actuated to perform sampling.

Each application presented different challenges which included mobility, actuation, energy, and intermittent connectivity. Our design choices and technologies were responses to these challenges and have evolved over time but the rate of change is slowing and, after six years, we are now at a stage where the core technology is reliable and configurable enough that it is has been deployed, unaided, by domain scientists.

A. Cattle Monitoring

We developed a network at a research farm over 500 km from our laboratory. The project has had several phases and technology generations, and it has been the primary driver of our technology development. The first phase involved recording the positions of cattle over time,

Table 1 Major Sensor Network Deployments. The Deployments Marked With * Are Discussed in Section II in Detail. More Details Can Be Found at www.sensornets.csiro.au

ID	Application	Unique Challenges	Lessons	Deployment	Year	Hardware	Software	Size	Network Protocols
A	Micro-climate	Proof-of-concept		Lab outdoors (Stage 1)	2005–2007	Fleck1C	TOS 1.x	25	ZTDMA
B*	Cattle monitoring	Static/mobile nodes	Flooding not scalable, network management difficult	Farm (Stage 1)	2005–2006	Fleck1C	TOS 1.x	28	ZTDMA
C*	Ground water quality	Long range low-power wireless communications (>1km)	Link layer retransmissions beneficial	Burdekin	2006–2007	Fleck3	TOS 1.x	9	MintRoute
D	Testbed	Self-maintained operating system	Link quality based communication benefits	Lab outdoors (Stage 2)	2006–2009	Fleck3, Fleck3B	FOS	45	Diffusion, LQ
E*	Virtual fencing	Localisation and actuation, static and mobile sensor network integration	Threaded OS more productive, trade-off between hardware features and range	Farm (Stage 2)	2007–2009	Fleck 3, Fleck Nano	FOS	80	Diffusion
F*	Rain forest micro-climate	Non-line-of-sight low-power communication, low energy harvest	Solar energy limitations, radio transmission in dense foliage	Springbrook (Stage 1)	2008–2009	Fleck3	FOS	8	LQ, LPL
G*	Water quality at different depths	WSN - robot integration, low-power wireless communication over water	Challenges of wireless communications on floating nodes	Wivenhoe (floating)	2008–2009	Fleck3	FOS	50	LQ, LPL
H	Environmental impacts monitoring	Solar power base-station		Wivehoe (catchment)	2009	Fleck3B	FOS	55	LQ, LPL
I	Biodiversity monitoring	System energy		Springbrook (stage 2)	2010	Fleck3B, multimedia nodes	FOS, Visual DSP++	55	LQ, LPL

and also soil moisture at various points in a paddock. Soil moisture is an important indicator of how quickly pasture will grow and therefore important for planning stocking rates (number of animals per unit area).

1) *Challenges:* Information from static and mobile nodes was to be relayed to a base station and then over the internet to a remote server. We later added nodes with cameras that periodically transmitted images of key locations such as water troughs. This was our first deployment that included both mobile and static nodes and this raised new challenges for routing and network topology maintenance.

2) *Technology:* From our early experiments with Mote hardware we saw the need to develop a device with improved radio range, solar power capability (for a country where solar insolation is of the order of 20 MJ/m^2/day), mechanical and electrical robustness, and ease of interfacing to transducers. This became the Fleck series of nodes shown in Fig. 2.

The Fleck-1 used the Atmega 128 processor, with 128 KB of program flash memory, 4 KB of RAM, and a stream-based Nordic nRF903 radio transceiver at 433 MHz which provided a 72-kb/s channel and a range of 500 m using a quarter-wave antenna. The board was 60×60 mm^2 in size and includes power supply, solar charging circuit, and sensing for on-board temperature, battery voltage, and charging current.

In retrospect the most inspired part of the system design was the simplest. A screw terminal block on the Fleck made it very easy to install. Battery, solar cell, serial, and analog and digital transducers could be connected using just a screwdriver—no expansion board was required. However, for more complex interfacing, a simple microprocessor pinout expansion bus was provided and an expansion board for the soil moisture transducers was developed.

The nodes carried by the animals were the Fleck-2 which was specialized for animal tracking applications. It has the functionality of the Fleck-1 with additional on-board global positioning system (GPS), three-axis magnetometer (electronic compass), three-axis accelerometer, and a multimedia card (MMC) socket for local bulk data storage as a safeguard against intermittent network connection or the case where data rate exceeds network capacity. The Fleck-2 nodes were built into collars that were worn by the cattle [Fig. 1 (left)].

For image processing the Atmega processor lacks memory and computational power. Unlike the Cyclops [5], we

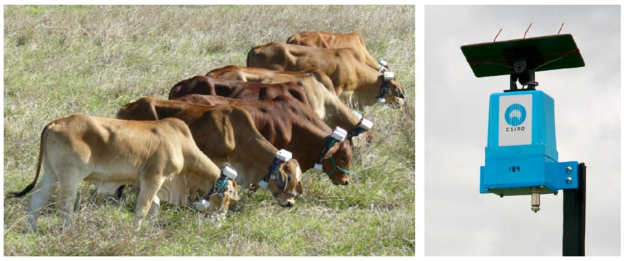


Fig. 1. ***Sensor nodes in the farm deployment: (left) cow collars, (right) second generation environmental housing with solar cell on top.***

chose to implement the camera as an expansion board with a Texas Instruments 32-b 150-MHz digital signal processor (DSP) with 1 MB of SRAM connected to a 640 × 480 color image sensor. The DSP performs image capture, buffering, and processing and sends the image in small blocks over the serial peripheral interface (SPI) bus to the Fleck for transmission over the radio network [6].

The nodes were programmed under TinyOSand and used an inhouse developed self-organizing time-division multiple-access scheme (ZTDMA) which cooperated with the Deluge [7] protocol for over-the-air reprogramming. The BMAC [8] protocol became part of TinyOS after we started development and we did not adopt it for two reasons. First, we were convinced there was enough power from the sun to not warrant duty-cycled radio communications. Second, we had forked the TinyOS code tree to support our Fleck platform and BMAC was integrated into a different version of TinyOS.

3) Lessons Learned: ZTDMA had very poor end-to-end delivery rates since it used network-wide flooding to achieve multihop communication, and therefore scaled poorly with network size. This became increasingly problematic as nodes were added, particularly the camera nodes which periodically transmitted large bursts of data. The rapid development of TinyOS at that time made it difficult to add desirable new features into our Fleck-1 port of TinyOS, partially negating the benefits of a common software platform.

Software became increasingly complex and took longer to debug which, combined with the remote location, made our productivity low and led to tension with our collaborator. Managing a remote network was hard, and we had no easy way of accessing the state of any of the nodes. Abundant solar power and a simplistic battery charging circuit led to overcharging of the batteries which greatly reduced their lifetime.

We had significant problems with the environmental housings, a prosaic but critical part of a sensor network system. Our first generation housings were standard gray plastic electrical boxes and the quarter-wave 433-MHz antennas were externally mounted. Box penetrations for transducers, solar power, and radio frequency (RF) were potential sources of water and insect ingress, but the bigger problem was the time taken to assemble each unit—nearly half a day. The bigger form factor of the Fleck-2, 60 × 120 mm^2 was problematic for mounting in the animal collars and there was, in retrospect, no clear advantage in cost or reliability of the single board solution. In fact the small number that was manufactured led to higher unit costs.

B. Ground Water Quality Monitoring

Deployment C was a relatively small network, nine nodes, located 2000 km from our lab. Its purpose was to monitor the salinity, water table level, and water extraction rate at a number of bores within the Burdekin irrigated sugar cane growing district. This is a coastal region and over extraction of water leads to saltwater intrusion into the aquifer. The area we monitored was approximately 2 × 3 km^2.

1) Challenges: The WSN had to operate unattended, and compared to previous sensornet deployments the network was very sparse with very long wireless transmission ranges (with average link length over 800 m). One simplification was that many nodes could be mains powered (since they were colocated with pumps).

2) Technology: The Fleck-3 series used a Nordic nRF905 radio which had a more sensitive receiver giving longer radio range. It has an inbuilt proprietary media access control (MAC) that supports packets up to 32 B and required only an SPI-bus connection to the processor. Like its predecessor, this transceiver did not provide received signal strength indication (RSSI) or link quality (LQ) information. It also used a different modulation scheme making it incompatible with the Fleck-1 and -2 radio, and we also made the decision to move from 433 to 915 MHz primarily for the smaller antenna size (which could then be placed *inside* the enclosure). This was the period when the rest of the WSN community was moving *en masse* to 2.4 GHz. The node also included an improved solar charger that allowed the solar cells to be disconnected from the battery, under software control, to prevent over charging. Developments in microelectronics also allowed the board to be slightly smaller, 50 × 60 mm^2.

We used the TinyOS operating system and implemented the MintRoute[2] network protocol which uses a shortest-path-first algorithm to route packets to the base station on the basis of a definable routing metric. We chose bidirectional Expected number of Transmissions (ETX) as our routing metric which works well with the radio chip we selected. We also use hop-to-hop retransmission to increase end-to-end delivery rates.

3) Lessons Learned: The technology was transitional: our last system to use TinyOS[3] and the Surge protocol and our first to use the new Fleck-3 node. We deployed no other network using this technology combination. The importance of protocols became very clear, and this system was able to achieve more than 95% end-to-end delivery rates in deployment when the individual LQ was higher than 15%. We conducted a radio survey to determine achievable communications distances in the environment, but we did this when the fields were bare. We neglected to account for the sugar cane which is up to 4.5 m tall when fully grown and interferes with line-of-sight wireless communication [9].

[2]TinyOS multihop routing, http://www.tinyos.net/tinyos-1.x/doc/multihop/multihop_routing.html.

[3]Although FOS was prototyped the project time lines were such that we considered it prudent to use TinyOS.

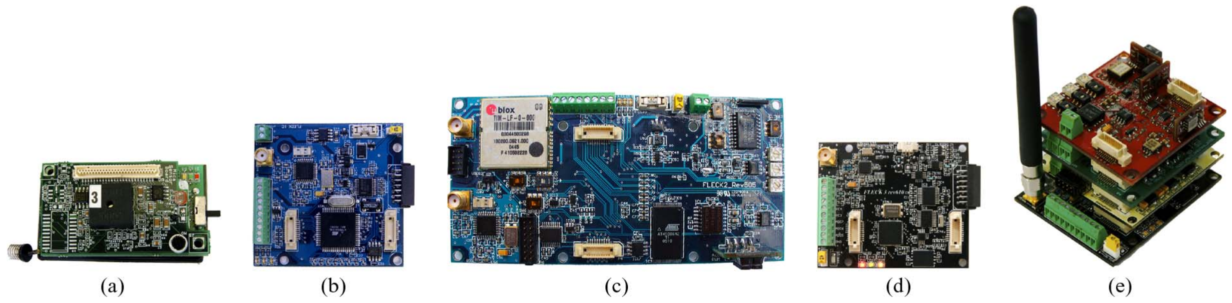


Fig. 2. ***Evolution of CSIRO WSN mote platforms. (a) Mica Mote. (b) Fleck 1c. (c) Fleck 2. (d) Fleck 3. (e) Fleck 3b stack.***

The network was deployed in 2006 and operated for 1.5 years, delivered more than 1 million water quality readings, and required only two maintenance visits. One visit was to repair a number of nodes damaged in a violent electrical storm.

C. Virtual Fencing

A new group of large-scale and remotely operated deployments was looming that included cattle control. The lessons learned so far stressed the need for better foundational technology. This led us to develop our own operating system and a new node with a better radio but which was incompatible with the Fleck-1 and Fleck-2. Deployment D, at our laboratory, was a testbed to shakedown this "new technology."

We also developed a moulded plastic case that was quick to assemble and had the antenna inside the enclosure [Figs. 1(right) and 5]. The new unit could be assembled in just 15 min.

1) Challenges: Key requirements included the ability for access to status and control of remote nodes, and this was particularly important to meet the ethics requirements for the virtual fencing [10] experiments.

2) Technology: The new Fleck-3 was incompatible with the Fleck-2 nodes used in the cow collars. Rather than redesign a Fleck-3 variant for animal tracking, we chose this time to exploit the expansion bus on the Fleck. Expansion boards are stackable [Fig. 2(e)] and the whole assembly could be bolted together PC-104 style leading to systems that were extremely robust mechanically and electrically. The Fleck-2 functionality now fitted on two 50×50 mm^2 expansion board: inertial sensing (accelerometers, gyroscopes, and compass), and GPS/MMC combination.

The growing complexity of our deployments led us to develop our own operating system. A sensor network operating system must abstract underlying hardware, facilitate resources sharing, and be power conscious and sleep whenever possible. TinyOS [11] was the first open-source operating system for sensor nodes and is event based. However, we and others have found that event-driven code is difficult to write, understand, maintain, and debug [12]. It does not scale well with program size, leading to difficulty in developing and maintaining large applications [13]—it requires that logically blocking sequences be written in a state-machine style [14]. The result is that the control flow for a single conceptual task and its state are split across several language procedures, effectively discarding language scoping features [15].

Our Fleck operating system (FOS) sits at the "sweet spot" identified by Adya [15]. FOS provides a priority-based nonpreemptive (cooperative) threading environment. This simple concurrency model means that semaphores are not required. The scheduler is responsible for CPU power management and enters the lowest mode consistent with thread resource requirements. Other approaches to threads on sensor nodes include preemptive threading [16] and protothreads [17]. Interestingly TinyOS itself eventually supported the threading model [18] and justified it in terms of ease of use and significantly greater expressivity compared to the event-based model.

FOS provides uniform access to underlying resources via a POSIX-like application programmer interface (API) with blocking read and write primitives. FOS supports the many transducer interface boards we have developed. Time-critical operations such as analog data sampling or high-speed timers are handled by interrupt-level callbacks. A virtualized timer based on an event-time queue is provided for non time-critical delays. The kernel is a relatively small piece of software, around 12 000 lines of C code.

The most well-known and cited disadvantages of threads are that each thread must reserve its own stack space that cannot be shared and, because it is difficult to know in advance how much stack space a thread needs, the stack is typically overprovisioned. We addressed these concerns in several ways. We developed a static analyzer for estimating required stack sizes for each thread which eliminates wastage. We use an interrupt stack so that we do not have to allocate space for interrupt handlers on *every* thread stack. Finally, the kernel checks sentinel bytes in the stack on each system call and invokes a panic if a stack overflow has occurred. In practice, the memory required for thread stacks has not proved to be a limitation, and applications with ten or more threads are routinely created.

Other support tools include a wireless boot loader that uses our remote procedure call (RPC) protocol [19] for command and status, and a negative-acknowledgment-based protocol for efficient reloading of multiple nodes simultaneously over the radio. Software failures return control to the boot loader allowing remote intervention or restart. A postmortem memory dump analyzer provides complete state and symbolic debugging information from memory images uploaded from nodes that have failed in the field.

FOS needed a routing protocol and we chose diffusion routing to meet the mobility requirements [20]. In diffusion a node (source) periodically broadcasts/advertises *interests*. Other nodes (destination) can subscribe to the *interests* and the sensed data will be routed from source to the destination via the reverse path.

As an alternative to programming the node we can consider that the node exports a predefined set of services which a client program can access. The services are accessed using RPCs. Other RPC systems have been proposed for sensor networks [21]–[23] but our system is now the mainstay of many long-lived deployments and has proven invaluable for diagnostics and network retasking.

The exported services are code segments, called *actions*, which closely resemble C-language functions. An RPC generator takes a set of actions and creates two sets of output files: the server-side C-code to be included on the node, and the client-side Python class. Our tiny RPC system uses a stateless protocol, and the RPC server runs as a separate thread on the node. Every node includes a set of standard services including reading or writing RAM or electrically eraseable programmable read only memory (EEPROM), reading transducers or battery condition, and rebooting.

An application, which is a WSN client written in Python, invokes a method call on an RPC object which causes the arguments to be marshaled, and sent via a gateway node to the WSN. There the action is executed and the return values marshaled and returned to the application. In the case that the function call is broadcast rather than unicast the method call returns a list of return values. Thus, a client program running on a host computer can seamlessly access services on one or more sensor nodes.

Using this facility, we have developed the network equivalent of a "symbolic debugger," a keyboard-interactive tool which allows node state to be examined or altered [19]. If the node application's symbol table is available we can address remote node memory symbolically. Our experience validates the experience of others, for example, [24], regarding the importance and utility of remote monitoring and control mechanisms.

As the number and complexity of deployments grew we had a problem with the number of different message formats. This deployment (E), for example, was highly heterogeneous with many different kinds of transducers such as soil moisture, battery voltage, and GPS position, as well as virtual sensors such as node and network performance metrics. We refactored several applications and created a general message format called tagged data format (TDF).

TDF is a self-describing schema for any data transmitted within the network, including RPC call and return frames. When a node sends data, it is packed into the payload of a message and tagged, prepended by a unique byte value, to identify the type of data and implicitly its length in bytes. This enables the creation of generic "back-end" tools that can parse the message payloads without knowledge of the application that created them.

3) Actuation: "Virtual fencing" (VF) [10] not only requires sensing of position and velocity information, but also actuation which in this case is the application of audio and mild-electrical stimuli to the animals. While applying stimuli is technically trivial, an independent ethics oversight committee required that it be applied ethically. With a nonzero probability that software bugs could breach the ethics commitment, we designed our stimulation hardware with a dedicated low-level controller that enforced our ethics constraints. Additionally, we designed the application to deactivate the VF logic should a communications link back to the base be lost. This meant that we could monitor the animals in real-time, deactivate collars if necessary via RPC, and know that when we could not contact the collars, that no stimuli was being applied. As the confidence in the logic matured, the need for a continuous communications link was relaxed, however the embedded ethics-enforcing logic on the hardware has remained. We have successfully demonstrated significant reduction of grazing on exclusion zones designated using our "VF" algorithm on hundreds of cattle not previously exposed to the technology [25].

4) In-Network Processing: The cattle monitoring work also gave us the opportunity to investigate some challenges around in-network processing of high sample-rate GPS data. The cattle nodes have a fixed-sized memory buffer to which position data are added at a constant rate, and from which data are downloaded at a nonconstant rate when they come into contact with static sensor nodes. We have developed a novel algorithm that performs online summarization of position data within the buffer, where the algorithm naturally accommodates data input and output rate mismatch, and also provides a delay-tolerant approach to data transport [26]. The algorithm has been extensively tested in a large-scale long-duration cattle monitoring and control application.

Data summarization and aggregation has been a growing area of focus within the sensor networks community [27], [28]. For many applications, the "sample-and-send" paradigm is not the right one, especially in applications which require periods of streaming high-fidelity data and periods where no data are required. Growth in solid-state memory capacity and shrinking costs mean that availability of local storage space is increasingly not a challenge [29]. The key research opportunities here revolve around the most effective ways to compress data which allow for efficient search and communication of the most valuable information contained within.

5) Lessons Learned: The decision to change both hardware and software was correct in retrospect, even if it was painful at the time. From a hardware perspective, the pain was in obsoleting all the nodes we had previously built at a significant cost, and this was difficult to "sell" to our partners. Writing code on a developing operating system was difficult for all concerned, but we did achieve the goal of a programming environment that was more *convenient* for applications developers. FOS was rapidly adopted and led to a huge jump in productivity—we found that students and visitors were up and running in hours. We routinely created complex applications with ten or more threads. This is counter to the choices made by other groups and advocated in the literature. In our experience, memory limits turned out not to be an issue, and in retrospect the tradeoff made in early WSN research where memory was saved at the expense of increased program complexity was perhaps not the right one.

Increased productivity and several parallel projects meant that message formats got out of control, they were not defined by FOS, and each project defined its own message format for data and for commands. This reached crisis point and was refactored into the self-describing data format TDF and a tiny remote procedure call system. These tools were quickly adopted by the team and led to another lift in productivity and in our ambition—we could contemplate building much more complex systems because the foundations and the debugging tools were strong.

The downside of adopting custom hardware, a radio with no LQI or RSSI and a small packet size, and custom software was that we cut ourselves off from the mainstream of sensor network development. We made do with "good enough" networking algorithms and devoted our effort to system integration. One interpretation of this is that the gains from protocol development are diminishing, and that systems issues now dominate. In fact our code base has a kernel with 12 000 lines of code (LOC) and the Python support tools and utilities have 23 000 (LOC).[4]

D. Rainforest Monitoring

Deployment F was part of a major initiative to provide reliable, long-term monitoring of rainforest ecosystems. Our target was a rainforest area in South-East Queensland (Springbrook, Australia) which had a high priority for monitoring the restoration of biodiversity. The first phase of the project was to develop a better understanding of the challenges in deploying long-term, low-power WSNs in rainforest environments [30]—the engineering testbed. These environments are typically characterized by areas with very limited solar energy with adverse and dynamic radio environments. In order to develop the network and energy management protocols required for robust and reliable performance of long-term, rainforest networks, we had to first quantify the performance of current WSN technology under these conditions.

1) Challenges: A well-recognized constraint on sensor networks, identified in the earliest research, has been energy resources. Sensor nodes are expected to be deployed in environments away from the energy grid and must either hold sufficient energy resources to last the required lifetime, or have their energy store replaced manually or replenished continuously with energy harvested from the surrounding environment. Sensor networks nodes comprise a number of core components such as a microcontroller, radio, flash memory, transducers, and other peripherals. Each of these components can be in one of a number of states each of which has different power consumption. Fig. 3 illustrates the energy consumption rates of a node as a function of the states of its core components. Based on these data and given a typical radio duty cycle of 5%, and transducer sampling/sending rate of around 5 min, we can estimate that node's average current consumption is around 2 mA.

2) Technology: Our current approach to managing battery storage is to combine both rechargeable and nonrechargeable batteries in each device. In the default mode of operation, all energy for the device will come from three rechargeable 1.2-V 2700-mAh NiMH batteries working in combination with monocrystalline solar panels capable of supplying up to 300 mA of current. In the event no further energy is harvested for long periods, the system will switch to the nonrechargeable (Alkaline) energy supply when the rechargeable battery voltage falls below a threshold, and switches back whenever this voltage rises again.

Fig. 4 shows the results of an experiment over two weeks with nodes deployed in open and covered areas. All nodes are deployed in the open with the exception of nodes "20," "Log-runner," and "19," which are placed in forested areas. Whereas nodes in the open typically harvest over 10 kJ/day, the covered rainforest nodes harvest as little as 100 J on average—only 1% of the energy of nodes in the open. A node drawing an average of 2 mA requires 600 J/day so sustainable operation would require an average current consumption of less than 0.33 mA.

The engineering testbed deployment showed that the links were highly dynamic and asymmetric. To meet high end-to-end delivery requirements we implemented a LQ-based routing protocol for FOS which became the

[4]Measured usinghttp://www.dwheeler.com/sloccount by D. Wheeler.

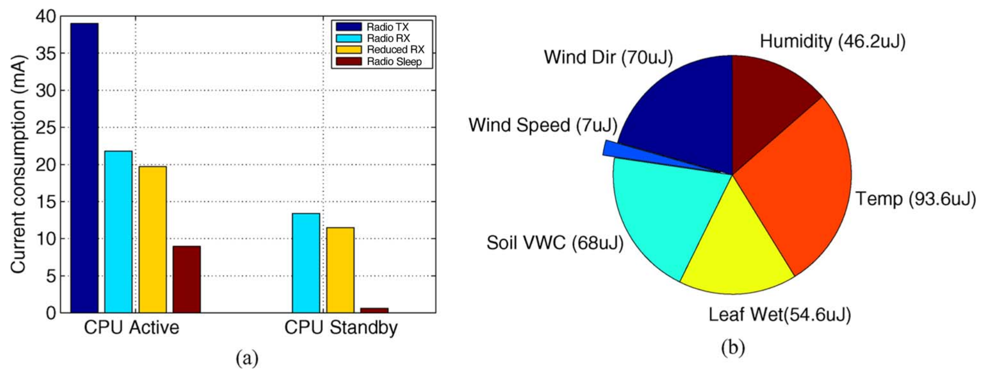


Fig. 3. ***(a) Breakdown of total current consumption by CPU and radio (100% duty cycle) for different states. (b) Breakdown of transducer energy consumption per sample. (a) CPU and radio. (b) Sensors.***

protocol used across all future deployments, and was retrofitted to deployment E. LQ is similar to the collection tree protocol (CTP) in TinyOS [28] and uses ETX as routing metrics and takes bidirectional LQ into account. However, hardware LQ indicators, such as RSSI and LQI, are not available from the Fleck-3's transceiver. Instead packet reception rates (PRRs) are estimated by snooping neighbor traffic. Snooping a packet from a neighbor has the same energy cost as receiving a packet but is better than the alternative of using node goodput to estimate PRRs. The estimation of LQ to sibling and child nodes based only on replies to infrequent beacon messages will not be as fresh as the estimate obtained by snooping neighbor traffic. For a difficult rainforest communication environment, we trade off energy for the robustness of communications. As a result, LQ achieved more than 99% end-to-end delivery rates when the network was connected.

To help meet the power budget, we implemented a low-power MAC protocol based on low-power listening (LPL) [8]. In LPL, nodes wake up periodically (every 57 ms) for a short period (3 ms) to check for communication activities and attempt to receive messages. Consequently, a source node needs to transmit a long "preamble" to wake up destination node before transmitting a message and this incurs a slightly higher cost for the sending node. Because traffic is typically low in a habitat monitoring sensor network, messages are sent every 5 min; nodes using LPL can sleep most of the time and conserve energy.

The nodes used the now standard Fleck-3 boards and environmental housings as shown in Figs. 5 and 6. A custom expansion board was built to interface to the many transducers: wind speed and direction, soil moisture, leaf wetness, temperature, and relative humidity.

We also started the next generation of video and audio interface. The system developed for deployment B used a DSP that had a poor software development environment. We redesigned this to use an Analog Devices BlackFin processor and we experimented with both uCLinux and the Visual DSP++ environment [6] before settling on the latter as our development environment of choice. The new interface has a mega-pixel color image sensor, two audio

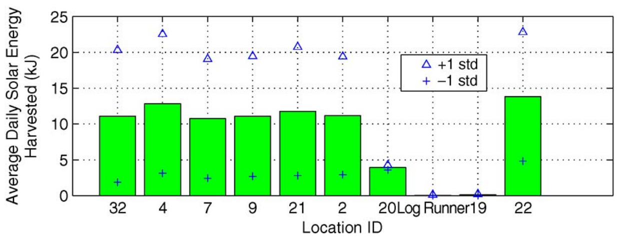


Fig. 4. ***Average daily energy harvested by each node.***

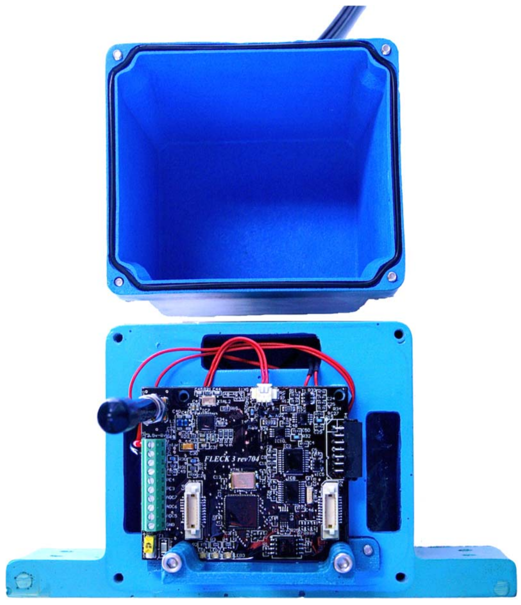


Fig. 5. ***Inside the second generation environmental housing showing the seals.***

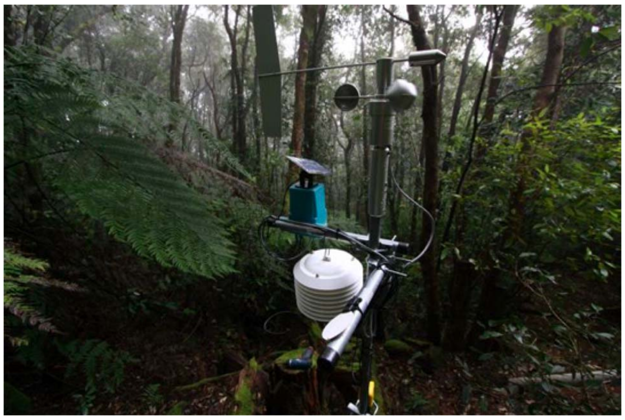


Fig. 6. ***A microclimate node deployed in the Springbrook rainforest.***

channels and interfaces to passive infra red (PIR) transducers for triggering. The applications include event-based and periodic image capture or sound recording. Some initial work into classification of vocalizing species has been undertaken.

3) Lessons Learned: We learned that radio propagation through dense and wet forest is poor, and that the decision to move from 433 to 900 MHz may not have been the right one. The LQ protocol did work very well despite the limitations of the radio transceiver. Results from experiments [30] showed that throughput for nodes in open area ranged from $> 99\%$ for one-hop nodes down to 80% for nodes up to four hops from the sink. In the case of forest nodes, throughput ranged from 95% to less than 20% in the worst case periods. In particular, we found that many links of forest nodes would completely breakdown during and after heavy rain events.

This was also the deployment where we learned that available solar power is not always enough to power a node—especially when a large suite of transducers must be powered. This does however raise an important question about an unwritten assumption for WSNs—that nodes are deployed and never revisited. In practice, our rainforest nodes are visited every few months to remove leaves, sticks, and insect nests from the transducers. It would therefore be quite feasible to replace the battery on each visit, eliminating the complexity associated with energy management discussed above.

E. Lake Water Quality Monitoring

The purpose of this deployment was to measure vertical temperature profile at multiple points on a large water storage that provides most of the drinking water for the city of Brisbane, Australia. The data, from a string of temperature transducers at depths from 1 to 6 m at 1-m intervals, provide information about water mixing within the lake which can be used to predict the development of algal blooms.

1) Challenges: Low-power wireless communications over water proved to be a challenge due to multipathing (radio waves reflected from the water surface destructively interfere with waves traveling directly) and the nonvertical orientation of the antennas in windy or wavy conditions (effective channel gain is reduced if antennas are not parallel). Interfacing a robotic boat to the static sensor nodes was another challenge.

2) Technology: The network comprises floating sensor nodes which contain the now standard platform of a Fleck-3, FOS, and a custom expansion board for the one-wire temperature transducer string. The node is mounted on an anchored float [see Fig. 7(left)], along with a solar cell and a high-gain (6 dB) whip antenna mounted atop a 1.5-m mast. Bright, but low-power, strobes are activated at night to prevent collision with other water craft.

The most novel element in this network [Fig. 7(right)] is a 14-ft twin-hull solar-powered robotic boat. Navigation is by GPS and depth sounder, and a scanning laser range-finder mounted high and looking forward detects obstacles. The onboard navigation computer communicates via a serial port with a Fleck which serves as a gateway to the floating sensor network. Specific RPC calls received by this Fleck are forwarded to the navigation computer for execution, and the return status is returned to the RPC caller [31].

We use the robot to crosscheck the calibration of deployed nodes using its own higher quality temperature transducer which can be set to different depths, and also to measure temperature along a transect between nodes. This architecture allows us to achieve high-rate measurements in parallel with automated measurements across space, and to access both high- and low-precision transducers. In the future, we plan for anomalous events detected by the network to be automatically investigated by the robotic boat.

Robots have been used previously to deploy and repair sensor networks [32], localize nodes post deployment [33],

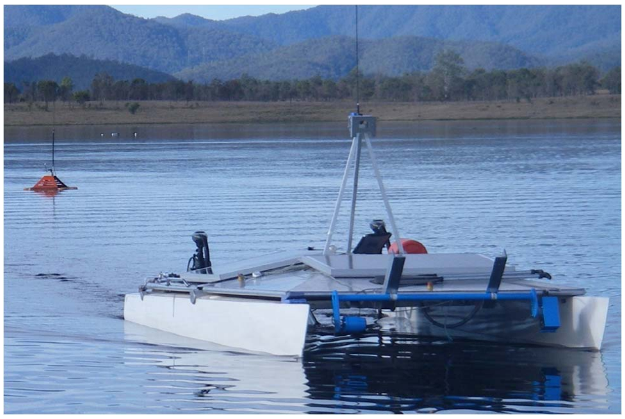


Fig. 7. ***Lake deployment. Robotic boat in the foreground and a floating node in the background.***

[34], and to collect data from nodes and physically transort it back to base using underwater robots [35], or fixed-wing unmanned aerial vehicle (UAV) [36].

3) Lessons Learned: The RPC framework proved to be a very effective way to remotely command the robotic boat, as well as to determine the status of the floating nodes. It is interesting to note the increasing level of abstraction with which we consider the network over the six-year period. We started with TinyOS where continuations and context must be managed by the user and then to FOS where they are managed by the operating system. We then moved from programming in NesC/C to application-specific virtual machines [37], special languages [38], and eventually to a version of Java [39] that ran on a node and supported classes, threads, and exceptions. We then moved away from node-level programming entirely by using remote-procedure calls. Primitives within the nodes could be composed using programs written in Python running on a host computer anywhere on the internet.

We also learned that communication across water is difficult with links typically less than 500 m, compared to the reliable 1000 m we experience at the farm. Preliminary trials with nodes running at 433 MHz show much better performance across water, and combined with the experience from deployment F argue that sensor nodes for environmental sensor networks should perhaps consider radios operating in the very high frequency (VHF; 30–300 MHz) band.

F. Summary

In retrospect the factors most critical to our success across these applications have been:

1) choosing a radio transceiver that gave low-power long-range links;
2) a robust MAC protocol based on bidirectional LQ estimation;
3) easy network reconfiguration based on RPC;
4) simple uniform data representation (TDF);
5) early adoption of solar power for sensor networks [40].

In a narrow sense, none of these are contributions to the field of sensor networks but individually and in combination they are critical to the technology tackling long-term real-world problems. Our developments were based on results already in the literature but with significant effort in implementation to ensure their reliability and usability.

III. END-TO-END SOFTWARE SOLUTION

Our end-to-end WSN software solution has evolved to conveniently present data to the end user (the ultimate purpose of a WSN) and to automate recurring tasks that are common to all deployments. It is the culmination of our numerous real-world deployment experiences, mainly outdoor solar-powered deployments, and substantial software development effort. It comprises a comprehensive suite of back-end tools and a generic sensor-node application *ToSense*.

The architecture of our end-to-end system is illustrated in Fig. 8 for the case of deployment E. Our back-end toolset consists of software that manages sensor data at the network gateway level, a central relational database, a web portal (Fig. 9) for data visualization, and a collection of Python utilities to remotely control and monitor deployed nodes and networks. Automated monitoring utilities detect hardware failures and alert relevant personnel. Data retrieval and visualization tools allow slow degradations in transducer performance or battery capacity to be detected and rectified. Data can be queried directly from nodes [41] but our focus was on providing the greatest flexibility across a wide range of application domains, and this was the basis for our decision to use a central database for data management.

ToSense runs under FOS, uses the low-power MAC and LQ routing, and provides functionality for sensor management (add, remove, retask, etc.), as well as interrogation of node health using remote procedure calls. This is another example of increasing functional abstraction for WSNs and is enabled by the underlying software tools. The ability to query the health of individual nodes is crucial to discover

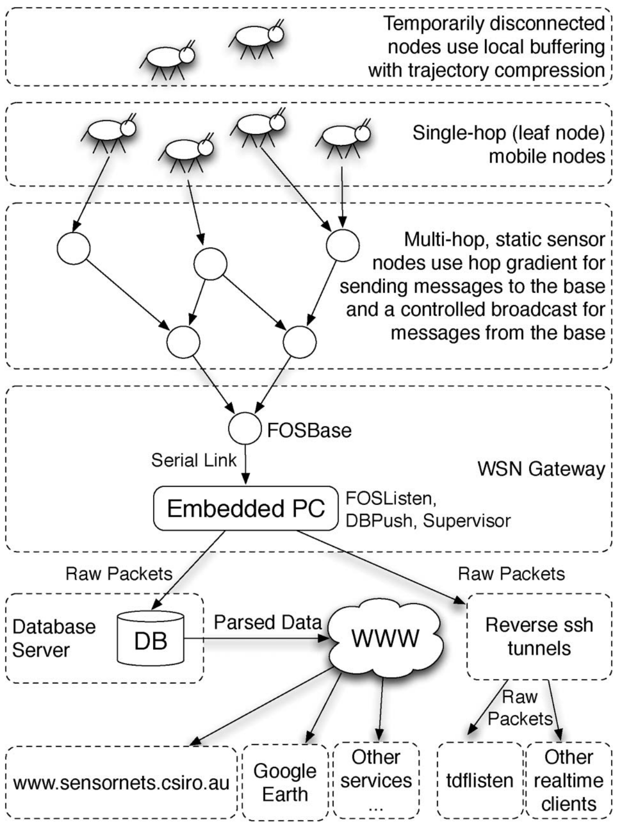
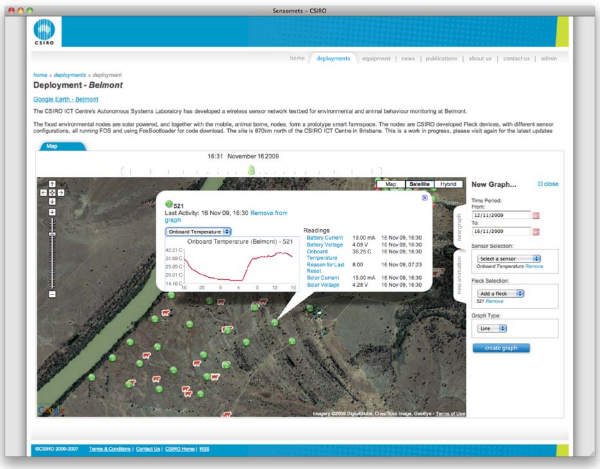

Temporarily disconnected nodes use local buffering with trajectory compression
Single-hop (leaf node) mobile nodes
Multi-hop, static sensor nodes use hop gradient for sending messages to the base and a controlled broadcast for messages from the base
FOSBase
Serial Link
WSN Gateway
Embedded PC
FOSListen, DBPush, Supervisor
Raw Packets
Raw Packets
Database Server
DB
Parsed Data
WWW
Reverse ssh tunnels
Raw Packets
www.sensornets.csiro.au
Google Earth
Other services ...
tdflisten
Other realtime clients

Fig. 8. ***Data flow in and out of the WSN and interaction with the back-end Python tools.***


Sensornets – CSIRO
CSIRO
home | deployments | equipment | news | publications | about us | contact us | admin
home » deployments » deployment
Map
Map | Satellite | Hybrid
New Graph... close
new graph
new animation
Time Period:
From:
To:
Sensor Selection:
Select a sensor
Fleck Selection:
Add a fleck
Graph Type:
Line
create graph
Onboard Temperature (Belmont) - 521
42.21 C
31.69 C
25.85 C
20.01 C
14.16 C
16
20
0
4
8
12
16
Imagery ©2009 DigitalGlobe, Cnes/Spot Image, GeoEye - Terms of Use
©CSIRO 2006-2007 | Terms & Conditions | Contact Us | CSIRO Home | RSS

Deployment - *Belmont*

Google Earth - Belmont

The CSIRO ICT Centre's Autonomous Systems Laboratory has developed a wireless sensor network testbed for environmental and animal behaviour monitoring at Belmont.

The fixed environmental nodes are solar powered, and together with the mobile, animal borne, nodes, form a prototype smart farmspace. The nodes are CSIRO developed Fleck devices, with different sensor configurations, all running FOS and using FosBootloader for code download. The site is 670km north of the CSIRO ICT Centre in Brisbane. This is a work in progress, please visit again for the latest updates

16:31 November 16 2009

521
Last Activity: 16 Nov 09, 16:30 Remove from graph

Onboard Temperature

Readings

Battery Current	19.00 mA	16 Nov 09, 16:30
Battery Voltage	4.09 V	16 Nov 09, 16:30
Onboard Temperature	36.25 C	16 Nov 09, 16:30
Reason for Last Reset	8.00	16 Nov 09, 07:23
Solar Current	15.00 mA	16 Nov 09, 16:30
Solar Voltage	4.28 V	16 Nov 09, 16:30

Time Period: From: 12/11/2009 To: 16/11/2009

Sensor Selection: Onboard Temperature Remove

Fleck Selection: 521 Remove

Fig. 9. ***Web view of a deployment.***

the cause of data irregularity. Sensor configuration data are stored in the Fleck's external flash memory, which is organized with the Coffee file system [42]. *ToSense* supports over-the-air multihop code replication for remote software upgrades. A dedicated thread manages the sampling of transducers at the correct times, a solar charge management thread prevents overcharging, and a watchdog thread restarts the node in the event of software failure.

IV. DISCUSSION

From the outset, the promise of WSNs has been to develop the next generation of distributed sensing technology—free of the need for external infrastructure such as cell towers or satellites. This in turn increased expectation around a new frontier—large-scale, pervasive, environmental sensing which would transform the way we observe and sustainably exploit the natural world. More than a decade on from this original vision however, we are far from seeing widespread use of large-scale sensor networks becoming a reality. Networks are typically relatively small in size (< 30 nodes) and/or only deployed over short periods (days to months) of time. Network nodes are almost exclusively programmed by experienced software engineers and maintenance costs required to sustain continuous operation of networks are usually significant and usually borne by the sensor network researchers.

In the remainder of this section, we give some insight from our own firsthand experiences, described in Section II, as to why the barrier for widespread adoption of this technology is still so high. Based on learnings from our own design, development, and deployment experiences, we identify some of the key technical challenges of this field which remain to be solved. We also consider some of the future challenges around value-proposition and alternative technologies.

A. Ongoing Technical Challenges

The field of sensor networks has become very popular in many ways due to the breadth and depth of its technical challenges. In moving to an environment free of fixed communications infrastructure, and introducing significant constraints around energy and computational resources, much of the standard thinking around communications, networking, operating systems, hardware platforms, and sensing has had the opportunity to be rethought from first principles. While this has undoubtedly raised a compelling new set of computer science and engineering research questions, many of the technical advances in the field are arguably reaching the point of diminishing returns.

In the majority of sensor-network applications, radio clearly dominates the energy consumption. As such, much of the community has focussed on ways to reduce the radio duty cycle to save energy. As it stands, duty cycling at the communications link layer has plateaued at around 1%–5% [8] for most practical deployment scenarios and is unlikely to improve significantly with current radio technology. Reliability of data communications is also an important problem for WSN. As a number of our deployments showed, the variability in conditions in many environmental areas (e.g., foliage, rain, humidity) means that communication LQ between nodes is highly dynamic and unpredictable. Given the constraints around radio output power and fixed antennas used with most nodes, guaranteeing the delivery of data over multiple node hops is extremely difficult, with low data throughputs expected for nodes with long hop counts to a sink.

Energy has been and remains a challenge for sensor network deployments. The energy state of a node places a constraint on the performance that a node can deliver. A node's energy state reflects its stored battery energy, actual and predicted harvested energy (through solar and other sources), and its energy load. Progress in battery technology has been much slower than increases in processing and communication rates, which emphasizes the importance of energy-efficient operation. We have observed through deployments that the amount of harvestable energy from solar current is highly dependent on geographic location, season, and deployment environment. For instance, the canopy cover in the rainforest dramatically reduced the harvestable solar energy compared to our lake deployment, where summer solar energy far exceeds the nodes' daily energy usage. Solar prediction models that account for these variations are key to forecasting the node's energy state.

The lack of resources for building and deploying networks in the order of thousands of nodes means that practical issues around scale are yet to be fully explored. In the case of typical collection tree protocols back to a single gateway, it is clear that these protocols cannot scale by orders of magnitude given the current capacity of network links. Whereas a "sample-and-send" paradigm may be suitable in some applications, there are increasingly compelling opportunities around nodes taking on more adaptive "event-driven" roles such as responding to queries from users or significant changes in environmental phenomena [43]. This is evident from our deployments such as cattle monitoring where a sample and send of raw position or movement data is not feasible—instead the network must be able to send back summary information, either when requested or when the opportunity arises.

This also opens the question of where "intelligence" should reside within these networks as the research community seeks to expand the capabilities from simple, distributed sensing devices to distributed, intelligent networks. Increasing the computational load at the node to extract information from data is appealing from an energy perspective in that it greatly reduces the communication cost which is dominated by radio budgets. It does however place a higher priority on the need for high-quality links to ensure that information can be returned with minimum latency. Almost certainly the appropriate decision on these kinds of questions will be determined by the specifics of each application.

B. Cost Benefit

For any emerging technology, economic drivers and cost benefit are pivotal issues which could have a dramatic effect on its market growth. Sensor networks face a number of challenges in this regard. The field arguably emerged due to the commoditization of cheap, low-power, single-chip microcontrollers and radios. These components emerged due to the rapid growth of global industries such as cell phones, wireless remotes, and car locks. Likewise, battery technology, while not following Moore's law, has still seen significant increases in energy density and reduction in price due to the increased demand for portable electronic devices.

While these components form the core of the typical platform for environmental sensing, the value proposition is greatly reduced by the remaining cost components: transducers, housing, and deployment. Compared with cell phones or television remotes, environmental sensing is a miniscule market. As a result, the current cost of transducers and the housing usually dwarfs the cost of the computational and communications elements of a WSN node. Until some step change in tranducer technology occurs, widespread environmental monitoring with hundreds or thousands of nodes will not be economically feasible, apart from a few areas of extreme scientific interest. Areas of increasing global interest such as improved understanding around greenhouse gas emissions and carbon sequestration will be likely areas that could see increased investment in large-scale long-term monitoring initiatives. Likewise the increasing scarcity of water resources in many regions could well see increased investment in innovative water monitoring and management practices which utilize sensor networks. This could in turn bring about a new generation of transducer technology, reducing current costs by orders of magnitude.

C. Alternative Technologies

While large-scale WSNs offer some clear potential for improved environmental sensing into the future, there are competitive technologies which will also improve over the next 5–10 years. Satellite remote sensing is already used extensively to infer a lot of information about the planet. Using systems such as SPOT-5, the technology already exists to extract multispectral features at resolutions better than 10 m^2—albeit at a significant cost.

Increased demand for connectivity in rural and remote regions should also see an increased spread of 3G, 802.11, and 802.16 coverage into regions where this has never been previously available. At the moment, the power consumption of 3G and 802.11 class devices is too high to be practical for most long-term environmental monitoring applications, however new generations of low-power 802.11 and 3G radios are expected to grow in the market, making these potentially viable for applications where multihop wireless nodes are currently the only option.

In many ways, only time will tell which technologies will become the preferred options for specific environmental sensing applications. It is likely that a combination of the technologies available today will be utilized for future deployment scenarios. Growing focus around the use of cognitive radios opens up new opportunities for current "mote-class" networks to merge with other wireless services in order for systems to overcome some of the current limitations around unreliable links, or the need for short periods of high network capacity.

V. THE ROAD AHEAD

Over the years we have deployed sensor networks for a wide range of applications, mostly to monitor outdoor environments that require long-range operation for prolonged periods without maintenance. We are currently at a stage of high confidence and stability in deploying these types of networks and tailoring the technology for each application. This stability suggests a fork in the road for environmental sensor networks: 1) either the technology is mature enough for larger scale adoption by scientists, farmers, and the wider industry; or 2) the technology is ready for a next phase of research and development of more advanced functionality to meet the demands of the end users. It is most likely that a combination of commercialization and development activities will characterize the future direction for this technology.

Further R&D in sensor networks could represent a shift in data flow, storage, and communication. Sensor networks have been so far treated as data gathering tools. While the initial wave of sensor network deployments has focused on periodic sample-and-send applications, the next wave will rely increasingly on performing in-network processing for a higher degree of adaptability to dynamic physical environments. Technological improvements in processing capacity and form factor will enable nodes to decide on-the-fly whether to process, compress, store, act on, or send sensor data back to users. Such decisions will depend on the fusion of multiple streams of information, including data from different transducers, channel state, local energy state, and application-specific user policy.

To drive such decisions, sensor nodes will process data locally, fusing readings from multiple transducers with current radio and energy states to transform transducer data into more useful information. There is also a shift from the use of simple scalar transducers, such as temperature, humidity, and light, to more complex multimedia transducers, such as audio, image, and video sensors. The increased sensing modality of sensor nodes also poses new challenges, such as the design of appropriate triggering strategies among various transducers and the distributed coordination among multiple nodes in a heterogeneous sensor networks.

Another timely issue is that of standardization. As with other maturing technologies, sensor networks have reached a stage where standard protocols are consolidating diverse research proposals that exist. IEEE 802.15.4 and 6loWPAN lay the infrastructure for future networks with IPv6 support. Building on these standards, and to support the wide range of emerging applications, we are in the process of building the next generation of sensor nodes by moving towards software-defined radio, with support for frequency-, antenna-, modulation-, and data-rate diversity. We expect these new nodes to provide WSNs with a more versatile and robust communication architecture on which to build services for the diverse application space. The shift towards IP support enables simple web servers to run on sensor nodes, so that users can access individual transducer data through standard web browsers. This may drive a shift in data storage and reporting models for WSNs, where portions of data may reside on the node, in raw or compressed form, for provision to the user only upon request.

The increasing integration of sensor networks with the internet, through IP support, raises new security challenges. While physical security challenges, such as vandalism to sensor nodes in our lake network, will always exist, mechanisms for securing communication, processing, and storage of data at sensor nodes will need to be addressed at every layer in the communication stack. Efforts in this direction, such as the TPM [44] and secure wireless key exchange, have already started, but more work is needed for an integrated cross-layer approach to the security issue.

The issue of network scale will be one of the next big hurdles for our development and deployment work. While newer radios, processors, and communication protocol pave the way for larger networks, scaling up to deployments with hundreds of nodes that run for months or years will require some form of hierarchy in the network. Another major challenge for scaling up is detecting node malfunctions, which become more probable, and recovering from these faults seamlessly. We expect that development of effective network programming models will also increase in importance for larger deployments, as current model for per-node programming will not scale well.

While our node software and hardware development processes have reached a mature stage that allows nonspecialists to deploy nodes, there remains a need for defining more streamlined deployment models to ensure wider uptake of the technology. Easily configurable software, such as our ToSense application, and hardware, such as the stackable daughterboard are key components of this model. The next step is to ensure that the process of composing working systems from these components is easy for nonspecialists.

In summary, the next wave in environmental and agricultural sensor networks will combine commercialization of current technology and development of more advanced functionality. This will include nodes with multiple sensing modalities and diverse radio configurations, as well as the continuous redefinition of the design space to identify and address challenges that emerge in this application space, and indeed in transferring lessons learned to new applications as well. Adaptive power management strategies that efficiently manage these activities without compromising performance quality also remain an open direction for continued investigation. ■

Acknowledgment

The authors would like to thank their colleagues: P. Sikka, G. Winstanley, S. Brosnan, L. Overs, J. Whitham, G. Foley, C. Crossman, B. Mackey, D. Swain, G. Bishop-Hurley, M. Bruenig, S. Sen, P. Flick, B. Wood, D. Palmer, G. Salagnac, L. Klingbeil, C. Richter, D. O'Rourke, M. D'Souza, M. Dunbabin, R. Dungavell, D. Hugo, P. McCarthy, C. Knight, K. Tane, and Powercom Ptd Ltd; and their students: S. Rothery, P. Schmid, V. Ho, M. Ung, T. Wen-Chan, and N. Nourani-Vatani.

REFERENCES

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proc. 5th Annu. ACM/IEEE Int. Conf. Mobile Comput. Netw.*, 1999, pp. 263–270.

[2] Gartner Group, *Understanding Hype Cycles*. [Online]. Available: http://www.gartner.com/pages/story.php.id.8795.s.8.jsp

[3] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Instrumenting the physical world with pervasive networks," *IEEE Pervasive Comput.*, vol. 1, no. 1, pp. 59–69, Jan.–Mar. 2002.

[4] J. Beutel, S. Gruber, A. Hasler, R. Lim, A. Meier, C. Plessl, I. Talzi, L. Thiele, C. Tschudin, M. Woehrle, and M. Yuecel, "Permadaq: A scientific instrument for precision sensing and data recovery in environmental extremes," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, 2009, pp. 265–276.

[5] M. Rahimi, D. Estrin, R. Baer, H. Uyeno, and J. Warrior, "Cyclops, image sensing and interpretation in wireless networks," in *Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst.*, 2004, pp. 311–311.

[6] T. Wark, P. Corke, J. Liu, and D. Moore, "Design and evaluation of an image analysis platform for low-power, low-bandwidth camera networks," in *Proc. ImageSense08 Workshop/ACM Sensys08*, 2008, pp. 7–11. [Online]. Available: http://research.microsoft.com/en-us/um/redmond/events/imagesense08/CameraReady/imagesense2008-wark.pdf

[7] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst.*, 2004, pp. 81–94.

[8] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst.*, 2004, pp. 95–107.

[9] T. L. Dinh, W. Hu, P. Sikka, P. I. Corke, L. Overs, and S. Brosnan, "Design and deployment of a remote robust sensor network: Experiences from an outdoor water quality monitoring network," in *Proc. 2nd IEEE Int. Workshop Practical Issues in Building Sensor Netw. Appl.*, Dublin, Ireland, 2007, pp. 799–806.

[10] T. Wark, D. Swain, C. Crossman, P. Valencia, G. Bishop-Hurley, and R. Handcock, "Sensor and actuator networks for protection of environmentally sensitive areas," *IEEE Pervasive Comput.*, vol. 8, no. 1, pp. 30–36, Mar. 2009.

[11] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "Tinyos: An operating system for sensor networks," *Ambient Intell.*, pp. 115–148, 2005.

[12] A. Dunkels and O. Schmidt, "Protothreads lightweight, stackless threads in C," Swedish Inst. Comput. Sci., SICS Tech. Rep. T2005:05, Mar. 2005.

[13] O. Kasten and K. Römer, "Beyond event handlers: Programming wireless sensors with attributed state machines," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, 2005, pp. 45–52.

[14] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer, and D. Culler, "The emergence of networking abstractions and techniques in TinyOS," in *Proc. 1st Conf. Symp. Netw. Syst. Design Implement.*, Berkeley, CA, 2004, pp. 1–1.

[15] A. Adya, J. Howell, M. Theimer, W. Bolosky, and J. Douceur, "Cooperative task management without manual stack management," in *Proc. Usenix ATC*, Jun. 2002, pp. 289–302.

[16] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han, "MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms," *Mobile Netw. Appl.*, vol. 10, no. 4, pp. 563–579, 2005.

[17] A. Dunkels, O. Schmidt, T. Voigt, and M. Ali, "Protothreads: Simplifying event-driven programming of memory-constrained embedded systems," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst.*, 2006, pp. 29–42.

[18] K. Klues, C.-J. M. Liang, J. Paek, R. Musăloiu-E, P. Levis, A. Terzis, and R. Govindan, "Tosthreads: Thread-safe and non-invasive preemption in tinyos," in *Proc. 7th ACM Conf. Embedded Netw. Sensor Syst.*, 2009, pp. 127–140.

[19] P. I. Corke, "A remote procedure call system for FOS," CSIRO ICT Centre, Brisbane, Qld., Australia, Tech. Rep. 09/041, 2009.

[20] S. Rothery, W. Hu, and P. Corke, "An empirical study of data collection protocols for wireless sensor networks," in *Proc. Workshop Real-World Wireless Sensor Netw.*, 2008, pp. 16–20.

[21] K. Whitehouse, G. Tolle, J. Taneja, C. Sharp, S. Kim, J. Jeong, J. Hui, P. Dutta, and D. Culler, "Marionette: Using RPC for interactive development and debugging of wireless embedded networks," in *Proc. 5th Int. Conf. Inf. Process. Sensor Netw.*, 2006, pp. 416–423.

[22] T. May, S. Dunning, and J. Hallstrom, "An RPC design for wireless sensor networks," in *Proc. Mobile Ad hoc Sensor Syst. Conf.*, Nov. 7–10, 2005, DOI: 10.1109/MAHSS.2005.1542785.

[23] A. Dunkels, R. Gold, S. A. Marti, A. Pears, and M. Uddenfeldt, "Janus: An architecture for flexible access to sensor networks," in *Proc. 1st ACM Workshop Dyn. Interconnection Netw.*, 2005, pp. 48–52.

[24] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "The hitchhiker's guide to successful wireless sensor network deployments," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, 2008, pp. 43–56.

[25] G. Bishop-Hurley, D. Swain, D. Anderson, P. Sikka, C. Crossman, and P. Corke, "Virtual fencing applications: Implementing and testing an automated cattle control system," *Comput. Electron. Agriculture*, vol. 56, no. 1, pp. 14–22, Mar. 2007.

[26] T. Wark, C. Crossman, P. Valencia, P. Corke, D. Swain, and G. Bishop-Hurley, "Poster abstract: A sensor network for compression and streaming of GPS trajectory data," in *Proc. ACM Sensys*, 2008, pp. 439–440.

[27] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proc. 22nd Int. Conf. Distrib. Comput. Syst.*, 2002, pp. 575–578.

[28] T. He, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Aida: Adaptive application-independent data aggregation in wireless sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 3, no. 2, pp. 426–457, 2004.

[29] G. Mathur, P. Desnoyers, P. Chukiu, D. Ganesan, and P. Shenoy, "Ultra-low power data storage for sensor networks," *ACM Trans. Sensor Netw.*, vol. 5, no. 4, pp. 1–34, 2009.

[30] T. Wark, W. Hu, P. Corke, J. Hodge, A. Keto, B. Mackey, G. Foley, P. Sikka, and M. Bruenig, "Springbrook: Challenges in developing a long-term, rainforest wireless sensor network," in *Proc. Int. Conf. Intell. Sensors Sensor Netw. Inf. Process.*, 2008, pp. 599–604.

[31] M. Dunbabin and P. Corke, "A framework for marine sensor network & autonomous vehicle interaction," in *Proc. IEEE OCEANS Conf.*, Sydney, Australia, May 24–27, 2010, pp. 1–7.

[32] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Deployment and connectivity repair of a sensor net with a flying robot," in *Springer Tracts in Advanced Robotics*. Berlin, Germany: Springer-Verlag, 2006.

[33] P. Corke, R. Peterson, and D. Rus, "Networked robots: Flying robot navigation using a sensor net," *Robot. Res. 11*, vol. 15, P. Dario and R. Chatila, Eds. Siena, Italy: Springer-Verlag, Oct. 2003, pp. 234–243.

[34] P. Corke, R. Peterson, and D. Rus, "Localization and navigation assisted by cooperating networked sensors and robots," *Int. J. Robot. Res.*, vol. 24, no. 9, pp. 771–786, Oct. 2005.

[35] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage and retrieval with an underwater sensor network," in *Proc. ACM SenSys*, 2005, pp. 154–165.

[36] S. Teh, L. Mejias, P. Corke, and W. Hu, "Experiments in integrating autonomous uninhabited aerial vehicles (UAVs) and wireless sensor networks," in *Proc. Australasian Conf. Robot. Autom.*, 2008.

[37] D. Palmer, P. Sikka, P. Valencia, and P. Corke, "An optimising compiler for generated tiny virtual machines," in *Proc. IEEE Workshop Embedded Netw. Sens.*, Sydney, Australia, May 2005, pp. 161–162.

[38] S. Sen and R. Cardell-Oliver, "A rule-based language for programming wireless sensor actuator networks using frequency and communication," in *Proc. 3rd Workshop Embedded Netw. Sens.*, May 2006.

[39] N. Brouwers, K. Langendoen, and P. Corke, "Darjeeling, a feature-rich VM for the resource poor," in *Proc. ACM Sensys*, 2009, pp. 169–182.

[40] P. Corke, P. Valencia, P. Sikka, T. Wark, and L. Overs, "Long-duration solar-powered wireless sensor networks," in *Proc. Emnets*, Cork, Ireland, Jun. 2007, pp. 33–37.

[41] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: An acquisitional query processing system for sensor networks," *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 122–173, 2005.

[42] N. Tsiftes, A. Dunkels, Z. He, and T. Voigt, "Enabling large-scale storage in sensor networks with the coffee file system," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, 2009, pp. 349–360.

[43] M. Welsh, G. Werner-Allen, K. Lorincz, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, "Sensor networks for high-resolution monitoring of volcanic activity," in *Proc. 20th ACM Symp. Oper. Syst. Principles*, 2005, p. 13.

[44] W. Hu, H. Tan, P. Corke, W. C. Shih, and S. Jha, "Towards trusted wireless sensor networks," *ACM Trans. Sensor Netw.*

ABOUT THE AUTHORS

Peter Corke (Fellow, IEEE) received the Ph.D. degree in robotics from the University of Melbourne, Melbourne, Vic., Australia, in 1995.

Currently, he is a Professor at the School of Engineering Systems, Queensland University of Technology, Brisbane, Qld., Australia. Previously, he was the founding Research Director of the Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Qld., Australia. His research activities span machine vision, vision-based robot control, field robotics, and sensor networks.

Dr. Corke is the Editor-in-Chief of the IEEE Robotics and Automation Magazine, a member of the editorial board of the *International Journal of Robotics Research* and a founding editor of the *Journal of Field Robotics*.

Tim Wark (Member, IEEE) received the Ph.D. degree in multimodal signal processing and pattern recognition from the Queensland University of Technology, Brisbane, Qld., Australia, in 2000.

Currently, he is a Principal Research Scientist at the Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Qld., Australia. His current research interests are in in-network processing in wireless sensor networks, with a particular focus on the applications in the environmental domain.

Raja Jurdak (Member, IEEE) received the Ph.D. degree in *ad hoc* and sensor networks from the University of California, Irvine, in 2005.

He has been a Principal Research Scientist at the Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Qld., Australia, since October 2008, where he currently leads the Sensor Networks Research Team. He has over 40 peer-reviewed publications, as well as a book *Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective* (New York: Springer-Verlag, 2007). His current research interests are modeling, optimization, and real-world deployments of energy-efficient and highly responsive sensor networks.

Wen Hu (Member, IEEE) received the Ph.D. degree in sensor networks from the University of New South Wales, Sydney, N.S.W., Australia, in 2006.

Currently, he is a Researcher at the Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Qld., Australia. His current research interests are low-power communications, compressive sensing, and security issues in sensor networks.

Philip Valencia (Member, IEEE) received a double bachelors degree in engineering and computer science from the Queensland University of Technology, Brisbane, Qld., Australia, in 2001, where he is currently working towards the Ph.D. degree with a research focus of distributed online learning for wireless sensor and actuation networks.

Currently, he is a Research Engineer at the Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Qld., Australia, with eight years experience of programming motes and wireless sensor network deployments.

Darren Moore (Member, IEEE) received the M.Eng. degree in microphone array processing from the Queensland University of Technology, Brisbane, Qld., Australia.

Currently, he is a Research Engineer at the Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Qld., Australia. His current research interests are in acoustic applications of wireless sensor networks for environmental monitoring.