ROPE: A Reactive, Opportunistic Protocol for Environment Monitoring Sensor Networks

Rachel Cardell-Oliver

School of Computer Science & Software Engineering, The University of Western Australia, rachel@csse.uwa.edu.au

Abstract

Sensor networks for outdoor environmental monitoring are required to provide a robust service in a hostile environment. Their purpose is to detect and report the temporal and spatial dynamics of monitored variables across a landscape. Protocols for these networks must adapt to changing properties of the environment: spatial and temporal variability in monitored environmental variables and changes in wireless connectivity. We present a novel adaptive protocol for outdoor environmental data gathering and report the results of tests with a prototype implementation using CSIRO Fleck hardware.

Keywords: Sensor Network, Adaptive Protocol, Robustness Testing, Radio Noise, Self-stabilising protocols

1. INTRODUCTION

A. Problem

Sensor networks for outdoor environmental monitoring are required to operate in a hostile environment in which the environmental variables being measured, communication connectivity, and physical presence and absence of nodes can all vary significantly, and unpredictably over time. The purpose of these networks is to detect and report the temporal and spatial dynamics of monitored variables across a landscape. Developing protocol software that provides a robust, reactive data delivery service in this environment has proved to be a challenging task [1], [14].

We have observed the behaviour of outdoor sensor networks and their nodes, and identified several significant properties about the variability of that performance which have not been considered in previous protocol designs. Given these observations, we examined the requirements for environmental monitoring applications, prioritising their different, sometimes conflicting, goals. We then asked, what are the consequences, if any, of these observed properties of outdoor sensor networks, and how should protocols be built to address them? In this paper, we propose a novel data gathering protocol, ROPE, to address constraints of environmental monitoring applications, and their implementation environment. We report the results of implementation trials using CSIRO Fleck hardware [13] and Mica2 motes [8] with TinyOS programs [15] to test software components of the ROPE protocol.

Outdoor, environmental applications require robust delivery of data describing the dynamics of monitored variables in a landscape. The data gathering network must be able to run unattended for several months at a time. Additional

requirements are imposed by the constraints of the network nodes and the environment in which they are deployed. First, environmental variables typically have both short periods of highly dynamic behaviour and also long, stable periods in which the variable changes very little [12]. For example, soil moisture values changes significantly over a short time scale during rainfall events, but not during long, dry periods. In this context, we need only report “interesting” data; that is, only parts of the temporal sequence for a given sensor’s data in which the value of soil moisture changes significantly. The conditions for communication of results by low power radio are also changeable [1], [14]. Finally, nodes have a limited energy budget, and so the amount of data transmitted over the sensor network must be minimised, because data transmission is both expensive of limited node energy, and subject to data loss through noise, over both short and long timescales.

B. Related Work

Many protocols have been proposed for data gathering in sensor networks. These protocols typically provide routing of information from data gathering producers to consumer nodes such as base stations, and their focus is on scalability and energy efficiency. In protocols such as Directed Diffusion [5], scalability is achieved by making local routing decisions based on information from a node’s neighbours, instead of constructing and maintaining end-to-end routes.

Tree based data gathering protocols, such as Diffusion, have several disadvantages for environmental monitoring applications. Local decision making can lead to undesirable global effects such as oversize routing trees, causing energy to be wasted in unnecessary packet transmissions [9]. Also, when network connectivity changes significantly, as observed in several empirical studies [14], [3], [1], then the cost of building and maintaining a permanently available routing tree becomes a high overhead, conflicting with the goal of energy efficiency. The overhead of tree maintenance is particularly pronounced when the volume of data to be delivered over the network is very low, say as a result of effective in-network processing.

Data gathering protocols for sensor networks need to be robust to communication failures including variable reliability of communication links [14], [1], asymmetric links between nodes [17], [3], and long-range transmission interference between nodes [3]. Packet losses in wireless networks are addressed in the SMAC protocol implementation used in trials at Pinjar [16], [1] by request-clear handshaking and an acknowledgement for each data packet. SMAC uses repeated

transmissions (by default, up to 7) when an acknowledgement is not received. In the Duck Island sensor network [14] acknowledgements are not used, but every packet is transmitted several times (by default, 3). However, neither of these schemes addresses the problem of long periods of poor communication or asymmetric links, and so they can waste significant energy by failing to react to these properties of their environment.

The SECOAS firefly MAC protocol [4] does adapt to communication losses. This protocol, based on the biological model of firefly synchronisation, provides a robust, data gathering tree using time division scheduling. Each node has a single transmission slot in every round. It uses the redundancy of multiple routes to overcome random packet loss, and overhearing neighbours' activity to cease transmission when communication conditions are poor, and resume when they improve.

Scheduling the transmission of packets, by time division of slots between nodes, reduces data loss caused by collisions between the transmissions of different nodes. Scheduling transmissions is beneficial for sensor networks because one node's transmission can interfere with reception for nodes well beyond the carrier sense range of the transmission [3]. Chain protocols [10] and SECOAS firefly [4] both use scheduling to maximise the use of nodes' waking, transmission time. Chain protocols [10] minimise transmission energy and scheduling costs by electing a leader node to transmit to the base station, with all other nodes forming a chain to the leader, each saving energy by transmitting their data in turn over a short distance. The chain can be repaired by skipping a node if links break.

Both the chain and firefly protocols are more adaptive and so more energy efficient than previous protocols. However, a disadvantage of both is that the routing tree or chain structure is maintained all the time, to support a regular, periodic reporting schedule. This is too high an energy overhead for applications such as ours where nodes have infrequent, and variable requirements for reporting their data.

One of the most energy-expensive operations in a sensor network is transmitting a packet. Thus, the lifetime of a network can be improved significantly by using in-network data processing to reduce the number of packets that need to be transmitted. In-network processing, such as local compression of gathered data [11], [12] and aggregation of data as it is transmitted through the network [6] can both be used to reduce network load.

Biological protocols compress their data in order to maximise individual nodes' fitness. For example, if a node has low batteries, a nearly full log, or hears its near neighbours transmitting similar data, then it will reduce its frequency of data gathering, and thus the amount of data it must later transmit. Conversely, if interesting changes are occurring in its local environment, then data gathering frequency is increased. In trials with a sea-bed sensor network, such a protocol achieved 50% compression of gathered data [11]. Lightweight temporal compression [12], is a simpler scheme in which a node processes a temporal stream of periodic sensor readings

and saves linear fragments, to approximate the full sequence, up to sensor error bounds. This scheme has the advantage of being simple to implement and run on the small processors used in our applications [8], [13], and it has been shown to deliver significant 20-to-1 compression. For this reason, we had no need to implement more complex compression or aggregation schemes, and we use a variant of lightweight compression in the ROPE protocol.

C. Contribution

This paper proposes a novel protocol for data gathering in an environmental monitoring network, and evaluates its performance by testing software components of the protocol using CSIRO Fleck hardware [13] and Mica2 motes [8]. We show that the ROPE protocol meets the requirement of gathering interesting spatial and temporal dynamics of environmental variables, whilst respecting the constraints of the environment in which data gathering occurs. The protocol is opportunistic in making the best use of both each node's resources and the resources of its environment. The novel combination of features for this protocol are:

- **Opportunistic transmission times** are chosen to transmit bursts of data when conditions are good, and otherwise wait;
- **Scheduled communication on a path** from gatherer to base is used to minimise shared wireless channel interference and contention;
- **Minimal state information** is stored by each node so that the network can be self-organising and fault-tolerant. For example, path information is only generated when needed, and data packets are not acknowledged individually;
- **Lightweight in-network data processing** is used to minimise the amount of data transmitted both saving network energy, and aiding scientific analysis of the delivered data.

The paper is organised as follows. In Section 2 we identify requirements for a class of environmental monitoring applications, and also the constraints imposed by outdoor sensor network environments. Section 3 presents the ROPE protocol, showing how it addresses these requirements and constraints. Section 4 reports the results of implementation tests and our conclusions are presented in Section 5.

2. REQUIREMENTS AND CONSTRAINTS

A. Environmental Monitoring Requirements

The design of the ROPE protocol is motivated by an application for soil moisture monitoring in Banksia woodland sandy soils over an area of approximately 500 square metres, using a cluster of up to 25 data gathering nodes connected to a base station and GSM gateway [7]. Each node gathers data from up to 4 Echo20 dielectric soil moisture sensors [2] placed at 2m to 5m spatial distance from the node at depths of between 10cm and 1m below the surface of the soil. Larger scale monitoring projects can be constructed using several such independent cluster networks.

Sensors are placed in the field site, by a human expert, to detect significant temporal and spatial changes in soil moisture, and also with sufficient redundancy to allow for both node and sensor failures. Sensor placement is neither uniform, nor random, but clustered around areas of interest in the monitored landscape. The main functional requirement for the network is to collect a temporal series of soil moisture readings from each sensor across a landscape, and report this data to the base station.

B. Non-Functional Requirements

The highest priority non-functional requirement for environmental monitoring sensor networks is to maximise the field life of the network: the time it is effectively gathering and reporting soil moisture data. In order to meet this goal trade-offs can be made against reliability and latency. Delivery of most but not every data point is a sufficient requirement for most environmental monitoring applications, including ours. Spatial redundancy and local back-ups are used to compensate for potential network data loss. Latency of up to 24 hours in reporting readings is also acceptable, since the main purpose of the gateway link is for monitoring node health, and identifying any management changes that may be required, rather than for immediate action in response to the current state of the environment. In our field trials, nodes log their data to non-volatile memory as well as delivering it to a base station over the wireless network.

When designing networks for outdoor monitoring, both normal and exceptional conditions should be considered. A *robust* protocol is required to meet one specification under normal conditions and a possibly weaker specification under exceptional circumstances. For example, under normal operation (with packet reception above 99%), the network is required to deliver at least 98% of significant data to the base station. When a node becomes isolated by communication or energy failure, then the exceptional specification requires that as much as possible of the node's data should be stored locally, and delivered to the base when communication is restored.

Environmental sensor networks are required to be *self-stabilising* in that starting from any global state, the system should converge to a state of equilibrium. Self-stabilising protocols allow for continued operation of the network despite node failures and the addition and removal of nodes.

The data gathering network is also required to be *fair*, in that resources and opportunities are shared fairly between the nodes and the network gathers data from all reachable parts of its monitored landscape. In normal operation, every node's data can be reported in full. In extreme conditions, nodes reduce the amount of data they transmit, taking into account other nodes in the area, and the dynamics of their own data, so that the most significant data that has been gathered is given priority for delivery.

C. Communication Constraints

The low power radios used in sensor network nodes are noisy. Not only is there a probability that each message transmitted

may be lost, but there are also long range effects in which node to node connectivity may be lost for periods of several hours or even days [1], [14], [3]. Network connectivity also changes as nodes are added and removed from the network, or fail and recover. Although an informed choice of antennae, radio hardware and the positioning of nodes, can minimise radio errors, protocols for environmental monitoring still need to adapt their data gathering strategies to changes in radio connectivity caused by changes in the environment that the network can not control. Significantly, poor connectivity is likely co-inside with environmental events of particular interest such as rain storms.

Another constraint for sensor network communication is that one node transmitting can cause interference for receiving nodes up to 3 times more distant than the transmission range of the node [3]. Furthermore, the transmission range itself varies significantly for receivers in different directions from the transmitting node [3], [17].

D. Energy Constraints

Finally, protocol design for long-lived data gathering is constrained by the energy budget of individual nodes. Transmitting a packet requires approximately 5 times more energy than taking a sensor reading and storing it locally. Thus, minimising the number of transmissions required is a critical step for extending the lifetime of battery powered nodes. Our Fleck nodes use solar panels to recharge two 2300mAh NiMH batteries. In Australian conditions this significantly reduces energy concerns, at least during daylight. However, since to minimise the cost per node of the network, we plan to use in the field the small solar panels provided with domestic solar garden lights, the amount of recharge energy available is limited and it is still necessary to conserve energy by minimising packet transmissions, and the time nodes spend awake listening for radio messages, or taking sensor readings.

3. ROPE PROTOCOL DESIGN

In this section we develop a protocol that addresses the application requirements and environmental constraints identified in Section 2.

A. System Model

The ROPE protocol is designed for small cluster networks of up to 25 monitoring nodes with low power resources (batteries and limited solar recharge) and short range radios. These nodes process data locally, and report readings of interest to a base station. The base station has higher power resources (larger batteries) than the nodes and a long range GSM mobile phone link to a persistent database store. Network nodes perform co-ordinated sleep-wake cycles, synchronised by a high power broadcast by the base node. Nodes wake and listen for a synchronisation signal from the base. If a node misses a synchronisation signal, it estimates the time and increases its listening interval for the next round. When new nodes are added to the network, they listen for up to two cycles to achieve synchronisation with the network's sleep cycle. If no

Data	Bits
Node ID (usually location)	16
Number of Packets Remaining $\in [0,256]$	8
Day $\in [1,200]$	8
Minute $\in [1,1440]$	16
Sensor ID $\in [1,4]$	3
SM milliVolt $\in [300,1000]$	12

TABLE 1: ROPE DATA PACKETS

signal is heard, a new node continues gathering data, but waits for several hours before trying to resynchronise.

At the beginning of each waking cycle, any node which has sufficient data to report, or any node whose reporting period has expired, contends for a transmission path to the base station. After a few packet times, any nodes not required for data delivery return to sleep until the next global reporting round.

B. Bandwidth Requirements

Our soil moisture monitoring application requires up to 4 sensors per node each to take one 2 byte reading per 3 minutes, indexed by a 2 byte time stamp (minute within day). With no data compression, $(4 \times 2 + 2) \times 20 \times 24 = 4800$ bytes are gathered by each node in one day. Typical TinyOS MAC packets have a 6 byte header with 29 byte payload. Allowing one byte per payload for the day, 24 hours data can be reported in $4800 \div 28 = 172$ packets. The ROPE protocol adds a very low control overhead of 22 packets to deliver these 172 data packets. Transmitting and processing one packet takes approximately 40ms, and so one day's data can be reported in 8 seconds, including overheads. Thus the required duty cycle of the whole network is 200 seconds per day, that is a 0.23% network duty cycle. In a multihop network of average depth 3, the required duty cycle is 0.69 %. We have chosen to use a 15 minute sleep cycle for nodes since it makes the correction of any clock drift manageable, it allows timely feedback of information from the base to the nodes, and it also gives significant energy savings by maximising sleeping time for the network. In the next section, we show that there are further significant energy savings to be made on the worst case 172 data packets per day per node, by in-network processing of the data before transmission.

Table 1 summarises the information contained in ROPE data packets. The node identifier, number of packets remaining to send and the day of the first reading are included once per packet. In addition each packet contains one or more sets of data, each consisting of the time in minutes that the reading(s) were taken and from 1 to 4 soil moisture readings each tagged with a sensor identifier.

C. In-network Data Processing

The environment monitored by an outdoor sensor network is a 2- or 3-dimensional, often discontinuous, space of values, such as volumetric water content of the soil. The monitored value at any point in this space itself changes over time. Thus

we are monitoring a function of one or more environmental values over a 3-dimensional real number ($\mathcal{R}$) landscape,

$$Environment : \mathcal{R}^3 \rightarrow Time \rightarrow Values$$

The goal of local processing of gathered data at each node is to minimise the amount of data to be transmitted, whilst preserving as many interesting features of the underlying spatial and temporal dynamics of the measured landscape as possible. It is assumed that the sensors have been placed so that significant spatial variations in monitored landscape variables can be detected, and that there is sufficient redundancy in the placement to allow for loss of some data, and to allow the detection of sensor or node failures. Temporal dynamics are captured by a time sequence of readings for each probe. The sensing rate is one soil moisture reading per 3 minutes. This rate was chosen after it was shown to be both necessary and sufficient to capture significant dynamics in a 46 day field trial which included two extreme flooding events and subsequent drainage of the site.

The temporal data series gathered by each node is represented by a sequence of soil moisture values $v_{t1}^{ns}, v_{t2}^{ns}, v_{t3}^{ns}$, for sensor s on node n . The time ti is given by a pair, $day \in \{1, 200\}$ and $minute \in \{1, 1440\}$.

ROPE uses a lightweight data compression scheme based on [12] that approximates the underlying data series by linear fragments. The accuracy of sensor readings from the Echo20 soil moisture probes is typically $\pm 3\%$, but with soil specific calibration, the error can be reduced to $\pm 1\%$. Output values from the probes range from 375 to 1000 milliVolts. Thus a $\pm 1\%$ volumetric soil moisture error is equivalent to 30 milliVolts, since we have calibrated the probes for our Pinjar field-site soil.

We record only sensor readings which differ by more than a threshold, T , equal to the sensor's specified margin of error (30 mV). For each of its probes, s , node n remembers its ultimate and penultimate values, v_t^{ns} and v_{t-}^{ns} (respectively). It also remembers the last logged value, $v_{t'}^{ns}$, for that probe. The difference between pairs of readings is used to determine whether to log or discard the readings. To account for a cumulation of small errors causing a change exceeding the threshold, we also compare the latest reading with the last stored reading. By default, all probe readings are logged at least twice a day (once per 720 minutes) so that faulty probes or nodes can be detected.

- 1) if $|v_t^{ns} - v_{t-}^{ns}| > T$ then log v_t^{ns} and log v_{t-}^{ns}
- 2) if $|v_t^{ns} - v_{t'}^{ns}| > T$ then log v_t^{ns}
- 3) if $t - t' \geq 720$ then log v_t^{ns}

In Section 4 we show that this simple compression scheme is highly effective, compressing real data by a factor of up to 20-to-1. In future work we plan to allow nodes to vary their default reading interval of 3 minutes, up to say 1 reading per hour, depending on the current conditions in the environment. A node uses local information such as the volatility of its recent soil moisture readings, soil temperature or rain gauge readings, to decide whether a faster or slower

data rate is applicable. Since nodes overhear messages from their neighbours and the base station, they could also make use of significant network environment information and user requests that are gossiped through the network, at no extra energy cost [4].

D. Finding a Time Slice for Reporting

There are two triggers for a node to report its gathered data: at least 500 bytes of data have been logged (that is 50 to 125 sets of readings), or 12 hours or more has elapsed since the node last reported. If either of these events occurs, then a node attempts to transmit during the next reporting slot. In the first case, 500 bytes can be sent in 18 standard TinyOS packets requiring approximately 0.7 seconds to transmit and process. In the second case, the minimum data stored is, by default, 10 bytes (one reading from each sensor) and this can be transmitted in a single packet taking 40 ms with processing. In the worst case, a node collects data at the maximum rate from all 4 sensors and no compression is possible. In this case it gathers 20 bytes per hour, and so generates 500 bytes every 150 minutes. In dry conditions, saving say 1 set of readings per hour, it would take 50 hours to generate 500 bytes of data. In normal conditions, 48 slots are available each 12 hours for 25 nodes to report. If more than one node tries to transmit in the same slot, the first requestor is chosen and the rest defer to one of the next slots.

E. Fairness and Priority of Data

A standard random backoff algorithm for shared wireless channel access is not sufficient to achieve a fair covering of all parts of the monitored landscape when the network is under stress. For example, it does not take into account the importance of the data being delivered by a node, and may allow some groups of nodes, such as those nodes closest to the base station, to dominate others. To overcome this problem, nodes need to know the importance of the data they are reporting in terms of the network's global data gathering purpose. For example, data from nodes in a region that has not recently reported should have priority over data from regions in which one or more nodes have recently reported their data. Establishing priority is particularly important in extreme conditions when there are more nodes waiting to transmit than slots immediately available. In the ROPE protocol nodes wait for a random backoff time chosen in inverse proportion to the significance of their data before requesting to transmit.

How can a node assess the significance of its data? One method for implementing data priority is for the base station to arbitrate between requests. However, it is more robust, and improves self-organisation of the network, to use a distributed algorithm in which each node can determine, or at least approximate, the relative importance of its own data. The priority of a node's stored data depends on [11]:

- how much data the node has stored;
- how long the node has been waiting to report it;
- whether any close neighbours have recently transmitted similar data;

- the resources currently available to the node to store or to transmit the data.

This information can be determined by each node examining its local state, and by listening to the transmissions of its neighbours.

If a node waiting to request a transmission slot, hears one or more, higher priority nodes' requests to send then it defers to the next or later transmission period. In exceptional conditions, that is when nodes have been waiting too long or have too much data, then more than one node can share a single transmission period by a node waiting for higher priority neighbours to complete their transmissions, and then transmitting its own data. In such poor communication conditions, nodes use all available information from themselves and neighbouring nodes, to determine how important their data is, and hence how persistent they should be in trying to send it, and also how much of the data to send.

When isolated from the rest of the network, a node must make local decisions about the significance of its data. However, when in communication with other nodes, it can use information gathered by those nodes, such as rainfall detected by a rain gauge sensor or a user request relayed by the base, in order to adjust its gathering regime. Such information is gossiped through the network as nodes overhear each other's requests.

F. Fault-Tolerant Reporting

Communication links are noisy, and so a percentage of packets transmitted, even in good conditions, will be lost. When communication is very poor because of environmental conditions such as bad weather or low batteries waiting for solar recharge, it is better to delay and try again later.

A node that is ready to deliver its locally processed data needs to make three judgements:

- 1) Does it currently have sufficient energy to transmit its logged data? If not, then wait for solar recharge.
- 2) Does it have a reliable connection to the base station to start transmission? If not, then wait for another slot.
- 3) Whilst transmission is in progress, is communication reliability sufficiently high to be worth continuing? If not, stop transmitting packets and wait for a more reliable slot.

In order to make these judgements, a node needs feedback from the base station or its neighbours, on the success of its transmissions. There are trade-offs to be made here. Acknowledging every packet transmitted by a sensor node doubles the energy cost of delivering that data. However, if no acknowledgements are used, or nodes simply always transmit multiple times, then significant energy is wasted in useless transmissions. Our compromise is to use one feedback packet per 10 data packets transmitted. Each feedback packet contains the number of packets received in the last burst. If the loss rate is too high, say, more than 2 out of 10 packets were lost, then the sensor node abandons its connection and tries in another slot. Recall that sensors are placed redundantly in order to compensate for up to 20% packet losses. A similar

feedback scheme is used when a node requests to send its data, but this time only 3 copies of the request are transmitted for feedback from the base node. Preliminary field trials, described in section 4, have shown this scheme to be well suited to data reporting in environmental sensor networks.

When a node is ready to report its data, it:

- 1) Listens for a time interval inversely proportional to the priority of its data;
- 2) If no higher priority request is heard whilst waiting, then send a request (3x) to transmit data; otherwise sleep until the next wake cycle;
- 3) Await a request feedback packet from the base station;
- 4) If feedback OK then start transmitting data in bursts of 10 packets; otherwise sleep until the next wake cycle;
- 5) Every 10 packets await a data feedback packet from the base station;
- 6) If feedback shows the delivery rate is OK, and the node has data remaining, then repeat steps 4 to 6; otherwise sleep until the next wake cycle.

G. Multi-hop ROPE

We are currently implementing ROPE for a single hop network in which the transmissions of each sensor node are able to reach the base node. These conditions are satisfied by the high quality radios in Fleck hardware, and our Pinjar field site. In future work we plan to extend the ROPE implementation for a multi-hop network since, even in a single hop network, the transmission range of nodes may be reduced to 18% of normal range in poor weather conditions [3], so creating a multi-hop network for a time.

If the base node hears a request packet, then it replies immediately to the requesting node. Neighbouring nodes over-hearing the request also listen for the reply, and so if a node is out of range of the base, then one of its neighbours can reply, and then relay the request message, and so on until the base station is reached. Each node knows its location (either GPS at installation, or by estimating its hop count to the base), and so can estimate its “goodness” as a router to the base for the requesting node. The base station, if it can hear a request strongly, is the best neighbour, and so it has highest priority to respond. Routing nodes prioritise the time of their responses so that better neighbours respond earlier and are chosen as the next link in the path. This gives a local, distributed scheme for creating best paths that is scalable as the network size grows.

If any node hears further requests whilst a path is being set up, then it sends a wait message to the requestor who normally delays within the same slot, using that path once higher priority nodes have finished transmission. Alternatively the requestor may choose not to wait, but to compete again in the next transmission slot.

4. IMPLEMENTATION TESTS

We have implemented and tested components of the ROPE protocol using CSIRO Flecks, Mica2 Motes and TinyOS programs. The Fleck uses the same Atmega128 processor as the Mica motes, but has a different radio: a 433Mhz, Nordic

Fig. 1: Raw and Compressed Soil Moisture Data

nRF903 radio transceiver with range up to 500m [13]. The Fleck is powered by re-chargeable batteries and a solar cell that trickle recharges the batteries; it operates on supply voltage from 1.3 to 5V. Like the motes, Flecks can be programmed with TinyOS. To date, we are using a prototype sensor board for one Echo20 soil moisture probe, built by Fleck researchers; a full sensor board for 4 probes is planned.

A. Measuring the Effectiveness of In-Network Processing

The ROPE protocol uses data compression for energy efficient operation. In order to determine the potential benefits of this approach, we analysed several soil moisture data sets ranging from a trial of 132 minutes in garden soil to a 46 day field trial at our test site in Pinjar. We applied the data compression algorithm outlined in Section 3C to these data sets. These data sets were collected using Mica2 motes with MDA300 sensor boards and two Echo 20 soil moisture probes per node.

As shown in Figure 2 the algorithm is extremely efficient, reducing the data sets from 5% to 24% of their original size. For a sample data set of 132 readings at 1 per minute, including a watering event, compression reduced the original 132 readings to 32, a saving of 76%. For Pinjar field trial data sets, where readings were recorded at 30 minute intervals over 8 to 46 days, the data sets were compressed to 5% to 13% of their original size. The trial period included two major flooding events, and subsequent drainage of the site, but conditions were otherwise dry and stable. An example of the compression for a sample of over 500 data points over 11 days from the 46 day trial is shown in Figure 1.

Analysis of these data sets also identified some interesting points for future work. Soil moisture values change very fast, within minutes, during exceptional events, such as flooding over the monitored area. However, outside these events, values are extremely stable. Whilst it is important to have high frequency monitoring so as not to miss significant dynamics, there are energy savings to be made if we can find ways of predicting when the measuring rate should be increased or decreased, for example using soil temperature, or rainfall to predict imminent changes in soil moisture.

A few of the data readings were outside the expected range or dynamics of the sensors. Such readings may indicate a

original size	compressed size	final as percent
132	32	24%
311	15	4.82%
311	18	5.79%
409	20	4.89%
409	21	5.13%
910	48	5.27%
910	54	5.93%
2208	215	9.74%
2208	287	13.00%

Fig. 2: Compression of Soil Moisture Field Data

failing sensor, hardware or software errors or low batteries. Battery power is a likely contributor, since the quality of data gathered tends to be worse towards the end of trial data sets.

B. Data Transmission Performance

In this section we consider how well nodes are able to identify the quality of their communication links, and how much that quality changes over time. In order to measure link quality we programmed 3 Flecks, each following a 15 minute sleep-wake cycle, and transmitting 10 standard TinyOS packets each time they awoke. These trials measured,

- 1) the temporal pattern of single hop packet transmission losses over an extended period;
- 2) the effectiveness of solar recharging in an outdoor environment, using three different solar panels;
- 3) clock drift between nodes, and hence potential clock synchronisation problems;

In normal conditions, packet delivery is extremely reliable. Of 820 packets sent by each node over a 20 hour period, 815, 816 and 812 were received, and in the following 13 hours 520, 520, 512 out of 520; that is over 98.5% in all cases. Of the following 1350 packets sent per node 100%, 99.78% and 99.26% were correctly received. Thus, in normal conditions, the design decision not to use packet by packet acknowledgements is justified. Packet reception in extreme conditions was tested using a fleck with very low battery, on the ground, with many obstacles between transmitter and receiver. Figure 3 shows the resulting reception rate per 10 packets transmitted at 15 minute intervals. Further trials in field conditions are planned, in order to investigate the extent and time range of periods of poor communication in more detail.

The above trials were run in perfect conditions of sunny, clear days without rain. The main tests were run with three Flecks, each within 20m of the base station. We also tested longer transmission ranges and found that, for a base station placed 50cm above the ground transmission range was up to 60m with 95 to 100% reception rates. When the base station was placed 1.2m above the ground then 95 to 100% reception rates were measured up to 200m from the transmitter.

The ROPE protocol is designed to minimise the energy used by each node and so to maximise the overall lifetime

Fig. 3: Reception rate over time for a Fleck with very low battery

of a network. Figure 4 shows energy usage and recharge for the three transmitting Flecks over a typical 24 hours during the trial period. Three different solar panels were tested for recharging the batteries of each Fleck: 2 from cheap domestic solar garden lights (flecks 10 and 11), and a higher quality scientific solar panel (fleck 12). Fleck 12 was in shade for all but one hour of morning sun, and has an 11 x 8 cm solar panel. Fleck 11, with a 5.5 x 5.5 cm panel, was in full sun for 6 hours from 10am to 4pm and Fleck 10, with a 10 x 10 cm panel, was in light shade but with around 7 hours of sun per day. Figure 4 shows the varying recharge current received by each Fleck when in full sun over a typical day during the trial. The on-board temperature for each Fleck over the same period is shown in the same figure for cross reference. The scientific panel generates much higher current than the garden light panels, and so could be used for applications with high energy requirements. The garden panels, however, were shown to be sufficient for the maximum load generated by the ROPE protocol.

We measured the drift between Fleck clocks, and timing slippage introduced in software, as 20 seconds gain per 12 hours on average (min 19s, max 21s), or 0.4375 seconds for each 15 minute cycle. This is well within the synchronisation listening window of ± 15 seconds per wake up cycle. The main part of this drift is likely to be bugs in the timing control of our protocol software, and this will be addressed in future implementations.

5. CONCLUSIONS

This paper introduces a novel reactive protocol, ROPE, for outdoor, data-gathering sensor networks. ROPE closely couples the application of soil moisture monitoring, and the properties of its operational environment. The protocol is opportunistic in choosing good transmission times, and making effective use of in-network processing to reduce the network's overall transmission load. ROPE adapts to the variability of monitored variables, communication conditions and network topology that are experienced by outdoor environmental monitoring applications. Thus, in changeable environments, ROPE is more energy efficient than existing protocols which maintain permanent routing trees, and also more robust.

Fig. 4: Battery Voltage, Solar Charge and On-Board Temperature

ACKNOWLEDGEMENTS

The author is grateful for support and advice from members of the CSIRO Fleck group in Brisbane, Peter Corke, Pavan Sikka, Phil Valencia and Leslie Overs, who provided Fleck hardware and software, a prototype soil moisture sensor board and driver for the Echo20 sensor for this work; Keith Smettem and the WA Water Corporation for running the large data set field trials at the Water Corporation's Pinjar site; and Ian Marshall, Antonio Gonzales and Chris Roadknight, of the SECOAS project, for insightful discussions on adaptive algorithms for sensor networks.

REFERENCES

[1] Rachel Cardell-Oliver, Keith Smettem, Mark Kranz, and Kevin Mayer. Field testing a wireless sensor network for reactive environmental

monitoring. In *International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne*, December 2004.

[2] Decagon Echo-20 Dielectric Aquameter. [Online] Available at <http://www.decagon.com/echo/>, April 2004.

[3] Alessio Falchi. Sensor networks: Performance measurements with motes technology. Master's thesis, Dept. of Information Engineering, University of Pisa, 2004.

[4] Antonio Gonzalez, Ian W Marshall, and Lionel Sacks. A self-synchronised scheme for automated communication in wireless sensor networks. In *International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne*, December 2004.

[5] Chalermek Intanagonwivat, Ramesh Govindan, and Deborah Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks (MobiCOM 2000) August 2000, Boston, Massachusetts*, 2000.

[6] Samuel Madden, Michael Franklin, Joseph Hellerstein, and Wei Hong. TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks. In *5th Symposium on Operating Systems Design and Implementation*, pages 131–146, December 2002.

[7] K. Mayer and K. Taylor. An Embedded Device Utilising GPRS for Communications. In *International Conference On Information Technology and Applications, Bathurst, Australia*, 2002.

[8] Mica Motes Specifications. [Online] www.xbow.com/Products/Wireless_Sensor_Networks.htm Accessed May 2003.

[9] Sule Nair and Rachel Cardell-Oliver. Formal specification and analysis of performance variation in sensor network diffusion protocols. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 170–173. ACM Press, 2004.

[10] Minh-Long Pham, Daeyoung Kim, Yoonmee Doh, and Seong eun Yoo. Power aware chain routing protocol for data gathering in sensor networks. In *International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne*, December 2004.

[11] Chris Roadknight, Laura Parrott, Nathan Boyd, and Ian W Marshall. A layered approach to *in situ* data management on a wireless sensor network. In *International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne*, December 2004.

[12] T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow, and D. Estrin. Lightweight temporal compression of microclimate datasets [wireless sensor networks]. In *Proceedings of 29th Annual IEEE International Conference on Local Computer Networks*, pages 516 – 524. IEEE, 2004.

[13] Pavan Sikka, Peter Corke, and Leslie Overs. Wireless sensor devices for animal tracking and control. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*, pages 446–454. IEEE Computer Society, 2004.

[14] Robert Szweczyk, Alan Mainwaring, Joseph Polastre, John Anderson, and David Culler. An analysis of a large scale habitat monitoring application. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 214–226. ACM Press, 2004.

[15] TinyOS: a Component-Based OS for the Networked Sensor Regime. [Online] <http://webs.cs.berkeley.edu/tos/>, Sep 2003.

[16] Wei Ye, John Heidemann, and Deborah Estrin. Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 12:493–506, 2004.

[17] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 125–138. ACM Press, 2004.