Autonomous monitoring framework for resource-constrained environments

Nazir, Sajid; Hamdoun, Hassan; Verdicchio, Fabio; Fairhurst, Gorry

Taylor & Francis
Taylor & Francis Group

ARTICLE

🔓 OPEN ACCESS ✅ Check for updates

Autonomous monitoring framework for resource-constrained environments

Sajid Nazir[a]*, Hassan Hamdoun [b]**, Fabio Verdicchio[a]* and Gorry Fairhurst[a]*

[a]School of Engineering, University of Aberdeen, King's College, Aberdeen, UK; [b]School of Natural and Computing Sciences, University of Aberdeen, King's College, Aberdeen, UK

ABSTRACT

The availability of low-cost imaging devices for embedded applications has enabled development of wireless monitoring systems capable of acquiring and transmitting both image and video data. Remote deployment of such systems is often constrained by limited power resources, thus a system must operate autonomously, balancing operational needs against available resources. This paper describes a framework for the design and implementation of an autonomous embedded remote monitoring system employing information-driven sensing to conserve energy and extend the system deployment lifetime.

The results from two case studies show improvements over a conventional system and other similar systems through the use of intelligent algorithms for reliable event detection and enhanced system operational lifetime by efficient utilisation of limited resources. The results are applicable to low-power battery-operated field devices offering better resource utilisation in disaster management systems, intelligent transportation and remote monitoring.

Introduction

Remote surveillance of assets or infrastructures [1,2] requires reliable and autonomous monitoring systems. There are major challenges when deploying a monitoring system in a remote location that lacks mains supply and communication infrastructure to backhaul the acquired information to the user. Often harsh conditions and/or distance prevent visiting remote sites, requiring autonomous monitoring systems to operate efficiently conserving their finite energy sources (e.g., battery). Communications can rely on radio (e.g. cellular networks [3] or satellite communication [4,5], which consume significant energy to transmit data to the user; energy harvesting techniques may be

CONTACT Hassan Hamdoun ✉ hassan.hamdoun@abdn.ac.uk 🖃 School of Natural and Computing Sciences, University of Aberdeen, King's College, Aberdeen, UK

*Now with School of Engineering and Built Environment, Glasgow Caledonian University, Glasgow, G4 0BA, Glasgow, UK.

**Now a Senior Researcher at BT Telecommunications PLC, Technology, Services and Operations, Adastral Park, Ipswich, IP5 3RE, Suffolk, UK. Email: hassan.hamdoun@bt.com

employed, but energy management [6] is still required to sustain long-term system operation. Conventional monitoring systems transmit all the data collected by sensors to the user, who extracts the information that is critical to identify 'events of interest'. In most cases, the data need further processing to extract information and resources (i.e. energy) are consumed to transmit data that are in most part irrelevant.

Research on autonomous monitoring systems recently focused on information-driven schemes that are energy saving because this reduces the cost of transmissions, providing better energy management. In comparison, an event or time-driven scheme that involves periodic data communications will result in relatively higher energy consumption. Proposed methods have ranged from the use of complex sensors [7], through efficient relay node selection [8], to cooperative data gathering [9] and game-theoretical energy management designs [10]. However, each approach addresses a specific application and none introduce a general framework to guide balancing the energy cost of on-board image processing with that of communication.

This paper presents a framework for the design and implementation of autonomous monitoring systems that combines building blocks such as: *sensing, data interpretation* and *transmission*. The energy requirements of these blocks must meet the available energy supply for sustained system operation. The energy saving can be realised by, for example, reducing the sampling frequency or by interpreting data to select and transmit only the information contributing to understanding of the process or event. As an example from environmental studies, consider a monitoring system where a motion-activated camera performs image analysis on a snapshot and determines whether a target (e.g., animal) of interest is in the frame prior to transmitting the picture to the user. Employing an energy management strategy, this system can balance the usage of power-intensive resources (camera, image processing and communications units) against battery provision. Our framework guides the selection of energy trade-offs, ensuring communication of relevant information (rather than all available data). Examples of resource-constrained monitoring applications include structural health monitoring, critical infrastructures, disaster management system and medical monitoring.

Related work, paper contribution and organisation

Resource-constrained monitoring systems described in the literature have applied algorithms to the sensory data collected by acoustic sensors [8], cameras [11,12] and a combination of sensors [7,9,13]. These approaches process sensor data to identify events of interest to be reported to a user. The research seeks to limit system energy requirements while retaining accurate detection. However, each work describes a specific system or algorithm,

rather than on an overarching framework for energy efficient monitoring that can be applied to different scenarios.

Each sensor is limited in its capacity for sensing, computation and transmission, with transmission energy being the dominant cost [14]. Various schemes have been reported in literature for example, to reduce transmitted data and decrease power usage. A detailed taxonomy is provided in [15]. An adaptive sampling algorithm [16] has been proposed for scenarios where sensing consumed more power than transmission, reducing the samples by 79% compared to a fixed rate scheme.

A data-driven adaptive sampling scheme [17] was found to save battery energy by 5.31% in sensing applications for slow varying sensed data. Another approach for energy saving is proposed in [18–20] using spatial and temporal aggregation of sensor values over a region to reduce the amount of transmitted data.

Information-driven sensing architectures have been reported to benefit from on-board processing to reduce the need for communication and save energy [21]. A system exploiting temporal correlation to reduce the transmissions by not reporting the values in acceptable error range is described in [21,22]. Some schemes use adaptive sampling, but need to maintain a buffer of raw values at the sensor node to determine the sensor values to be transmitted. A data reduction scheme based on Differential Pulse Code Modulation (DPCM) is reported in [23]. These studies trade-off the high cost of sending raw values by applying on-board algorithms such as Fast Fourier Transform (FFT) [24] is close to our case study 2, and describes a system measuring temperature and exploits spatial and temporal correlation between sensed values. They define a data quality metric by dividing measurements into different regions and treating values on their closeness to a region's boundary. Values around the boundaries are tracked for small changes and values far from the boundaries are not reported. The authors do not report significant energy savings for one node based on their algorithm although clustering and other cooperative transmission strategies are discussed. In [25], time series forecasting predicts future values to reduce sensor sampling and the number of messages transmitted. The sampling frequency is reduced once a sensor reports a constant, but increased if it changes. An energy saving of 87% is reported, but requires more sensing energy (due to complex sensors) compared to transmission energy and the gain is due to reduction in sensing rather than transmission energy.

In contrast, this work differs from above approaches as we keep the sensing frequency constant, but vary the transmission frequency based on the difference between the sensed value and the threshold limit, and send a periodic keep-alive message where the sensed value does not meet transmission condition. Thus the system transmits more frequently (with more accuracy) closer to a threshold and less frequently (with less accuracy) away from the threshold.

Although temperature data has a temporal correlation between successive readings, our system has the advantage of keeping the operator informed of significant events and providing reassurance that the system is alive by sending a reading every hour if no information is to be transmitted.

We provide a framework that does not rely on complex computations and a buffer at the sensor node to detect an event and autonomously deduce the threshold for event detection from the sensed values. Unlike schemes that aggregate spatial or temporal data to reduce traffic, our framework relies on deterministic thresholds to report actual sensed values. It assumes a system of units performing one of the following tasks: sensing, data processing and transmission. Data processing enables *information-driven sensing*, that is extraction of useful information from sensor data. This systematically combines the feature with *energy management* to balance on-board processing and transmission against available energy resources.

Two case studies evaluate practical system implementations of our framework. The first case study is a wildlife monitoring system. Traditional systems based on camera traps [26] are affected by 'false-positives', i.e. captured images containing no subject of interest, caused by spurious triggering of motion sensors. Such images needlessly consume transmission resources when the data is collected remotely. The second case study presents a system designed to monitor isolated roads and report the onset of road freezing conditions, thus initiating gritting. Timely notification of events of interest is of essence, whereas continuous reporting of non-critical conditions wastes resources.

This work makes the following main contributions:

- A general framework for embedded monitoring systems with on-board intelligent processing to extract information for onward transmission that can be applied across any power-constrained monitoring application.
- A general power management strategy to extend system operational life by providing a trade-off between processing and communications cost based on the distance of the sensed value to the chosen limits.
- Design and implementation of autonomous sensor nodes that can report events of interest under various field conditions by adjusting its parameters.
- Substantiates this framework through two case studies in environmental and road condition monitoring, each involving monitoring for occurrence of a random event based communication.

The remainder of the paper is structured as follows; Section 2 describes data interpretation and power management processes. Sections 3 and 4, describe two systems implementing the proposed framework. Conclusions are drawn in Section 5.
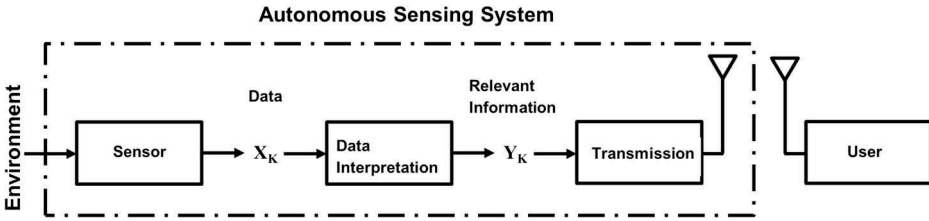
Autonomous Sensing System

Figure 1. Block diagram of a monitoring system with data interpretation.

Information-driven sensing for energy-constrained applications

A block diagram with the proposed architecture for autonomous monitoring is shown in Figure 1. The system senses the environment and reports to a remote user via a wireless link. The sensor data consists of a discrete set of samples, denoted as X_k. A sample can represent a scalar (e.g., a temperature measurement) or vector value (e.g., an image) depending on the application scenario. The data are coded after sensing (e.g., using JPEG images) and the size (in bytes) of a coded sensor measurement is denoted as x_k.[Replace this figure with attached]

A conventional system transmits X_k; the required transmission energy $E_T(x_k)$ is assumed proportional to the amount of transmitted bytes: $E_T(x_k) = c_T x_k$, where c_T (J/B) is the energy required to send a byte. In contrast, our system analyses the sensor data and only sends *relevant information* to the user. The system in Figure 1 features a processing unit that interprets the data sample X_k to extract the relevant information Y_k. Processing data requires energy $E_P(X_k)$, which depends on the data sample X_k. The y_k bytes of the relevant information Y_k are then communicated to the end user, requiring energy $E_T(y_k) = c_T y_k$. Note that $0 \leq y_k \leq x_k$: a data sample deemed irrelevant after processing results in zero information bytes. As an example, consider an outdoor motion-activated surveillance camera (sensor) taking a short video (X_k) that is processed to ignore any wind-blown bush and only retain frames where a trespasser is present. These frames, if any, are the relevant information transmitted to the user (Y_k).

For the k-th sample, the energy gain our system attains over a conventional one is:

$$g_k = E_T(x_k) - [E_P(X_k) + E_T(y_k)] = c_T[x_k - y_k] - E_P(X_k) \qquad (1)$$

Energy saving results when $g_k > 0$ for each sample (or over a given period, $\sum_{k=k_1}^{k_2} g_k > 0$). Transmitting only relevant information saves communication energy, but this saving must not be outweighed by the energy required for processing. Therefore we introduce the Energy Management stage shown in Figure 2. This block estimates whether data interpretation will yield an energy advantage for the current sample: if this is the case, data interpretation is
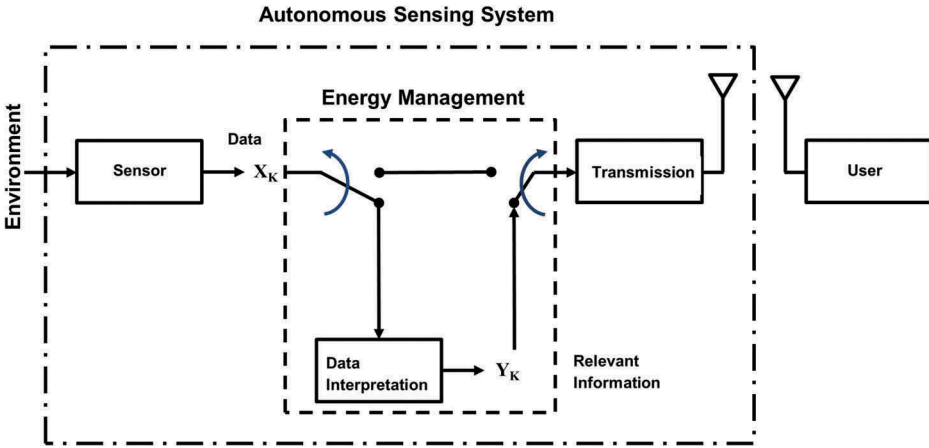
Figure 2. Monitoring system with energy management stage. Data interpretation is performed when a net energy gain is expected (switches open, as shown); otherwise sensor data proceed to transmission (switches closed).

performed prior to transmission (the switches in Figure 2 are open, as shown); otherwise the sensor data are transmitted without processing (the switches in Figure 2 are closed). Without knowledge of $E_P(X_k)$ and y_k, the system cannot compute whether $g_k > 0$; our system builds an estimate of: (*i*) the energy required to process the current sample and (*ii*) the size of the useful information derived from the data. These estimates, denoted as $\hat{E}_P(X_k)$ and $\hat{y}_k$ respectively, are used to forecast the energy gain *before* attempting to process the current sample; this *expected* energy gain $\hat{g}_k$ is given by:

$$\hat{g}_k = E_T(x_k) - \left[\hat{E}_P(X_k) + E_T(\hat{y}_k) + \varepsilon_k\right] = c_T[x_k - \hat{y}_k] - \left[\hat{E}_P(X_k) + \varepsilon_k\right] \tag{2}$$

[Replace this figure with attached]

where ε_k accounts for the energy required by the system to compute the estimates $\hat{E}_P(X_k)$ and $\hat{y}_k$ on the basis of statistics derived from past samples. The decision process is shown in Figure 3: only if the expected gain is positive, the data interpretation stage takes place, and the statistics used to compute $\hat{E}_P(X_k)$ and $\hat{y}_k$ are updated.

The term ε_k in is the computational cost of executing the decision block in the flowchart of Figure 3, which essentially requires creating (and updating) the statistics used to obtain the estimates $\hat{E}_P(X_k)$ and $\hat{y}_k$. In many practical cases, as in the examples discussed in this paper, these statistics are obtained by a moving average of past observation of $E_P(X_k)$ (where processing energy is approximated using processing time) and y_k; this is an inexpensive process for the control unit executing the algorithm and, in such cases, $\varepsilon_k \approx 0$.

Information-driven sensing and scheduling reduces the network traffic. The transmission system in our framework, discussed in this section, achieves the reduction to transmissions by processing sensor samples to filter irrelevant
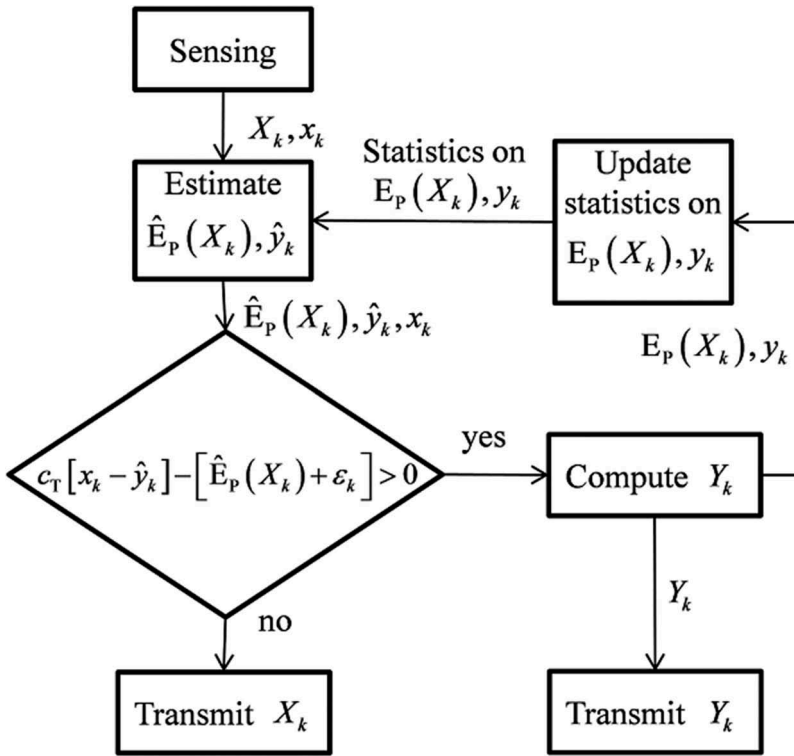
Figure 3. Flowchart of the decision process implemented by the energy management block.

data, and transmit only relevant information to save battery power. An effective system must not be too 'aggressive' in filtering data, which may make it prone to 'miss' important events. The following sections detail two specific implementations of the autonomous monitoring system of Figure 1. For each, we discuss a specific application scenario, the algorithms guiding the extraction of relevant information and the energy management process; we show that these systems achieved power savings without sacrificing event detection accuracy.

Application I: wildlife monitoring

This system was developed to support ecological research by the University of Aberdeen [26] and deployed in the Cairngorms National Park, Scotland [27]. The application requires detection of an animal in the field of interest and capture of image/video of the event. The target location has a low-density population of animals and only a few detections are expected, at random points in time over several months of deployment. The location is isolated, with no communication or power infrastructure, thus requiring satellite communication and solar panels to support a battery pack. The

deployment period spans the winter, when maintenance or emergency-repair visits to the site are impossible due to snow. Energy management is crucial, as solar power input is minimal due to limited exposure to the sun and snow cover of the panels.

In this scenario, it is normal to employ passive infrared (PIR) sensors that are triggered by motion occurring within the range of the sensor. PIR sensors require almost no power to operate continuously and respond to sudden changes in the thermal radiation they receive from the area facing the sensor. These variations are associated with a target (an animal) moving in front of the sensor. However, PIR sensors are prone to 'false-positives' [28]. These spurious detections are triggered by thermal fluctuations in the area facing the sensor, for instance due to moving cloud cover or to patches of terrain that gain/lose heat differently from their surroundings; moving vegetation (reflecting heat from the sun) also triggers the PIR. It is critical to identify the relevant information (events with an animal in view of a camera) rather than waste resources transmitting images containing moving foliage. The proposed system can identify the 'events of interest' in the monitored area and eliminate 'false positives'.

System architecture

The system architecture, shown in Figure 4, comprises:

- A **camera node** that incorporates a PIR sensor, a 5 Mega-pixel camera (capable of operating in darkness using an embedded IR LED lamp) and a Raspberry Pi, acting as embedded node controller and processing unit. This node sits close to the monitored area and is wired to a Gateway, which supplies power and collects data to be transmitted to a remote user via satellite link. Following a PIR trigger, the node controller activates the on-board camera to capture an image. Image processing can be applied by the node controller to determine the significance of the motion event: only relevant pictures are passed to the Gateway for reporting to the end user. The details of this process are discussed in the following section.
- A **wireless sensor node**, with PIR and other sensors onboard. It detects movements occurring in the middle of the monitored area, where this node is placed. A Gateway, placed in a peripheral location with the bulk of the equipment, is alerted of a detection event via wireless messages; in response, the Gateway activates a high-resolution video camera mounted at the edge of the monitored area to capture high-quality video footage of the scene. The wireless sensor node uses an AVR RFA1 micro-controller platform [29] to manage a PIR sensor, an X-band radar motion detector [30] and a wireless communication unit (using IPv6 Low Power Wireless
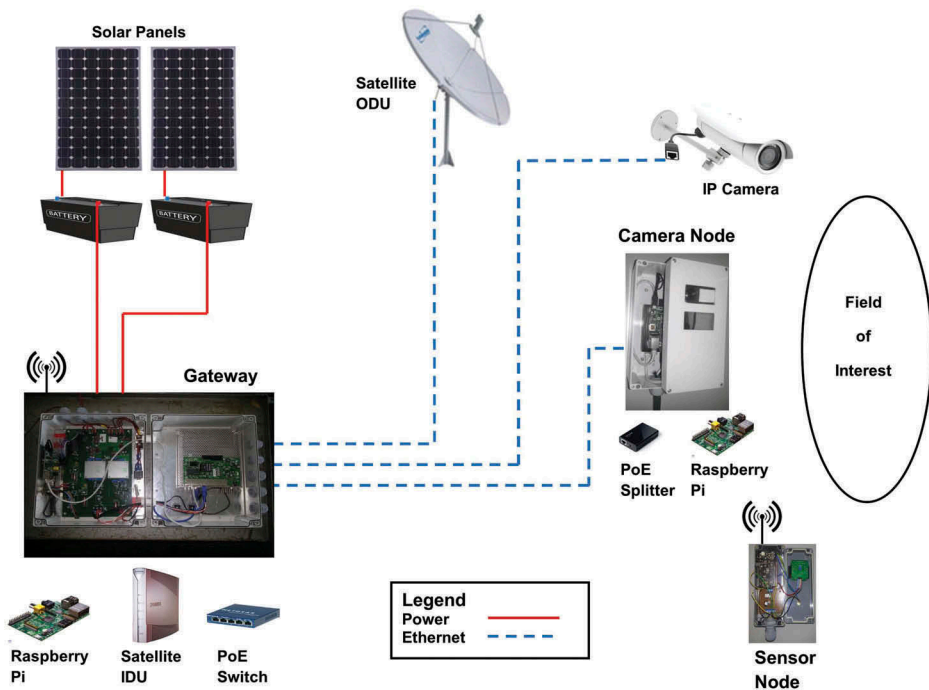
Figure 4. Autonomous wildlife monitoring system.

Personal Area [31,32]). The real-time OS Contiki[33] runs on the AVR platform to coordinate the actions required to report sensor trigger signals to the Gateway. The sensor node offers a first line of defense against spurious PIR triggers, e.g., those generated by terrain heat exchange due to moving cloud shadows. Following PIR activation, the node controller powers the X-band motion detector (an active sensor requiring non-negligible power that is not continuously operated). This uses the Doppler Effect to detect a moving object within its surroundings; animals and plants have sufficient water content to reflect the X-band microwave pulse and be detected. Using this sensor, the node controller is able to dismiss PIR triggers generated by terrain heat exchanges, but it may confirm those caused by wind-blown vegetation. If the X-band sensor confirms motion, the Gateway's camera is triggered and takes an image of the scene.

- A **Gateway** hosting a high-resolution video camera, a Gateway controller (Raspberry Pi B [34] coupled with an ADC board v2 [35]) a satellite Indoor Unit (IDU) and Outdoor Unit. (ODU). The Gateway power supply comprises two battery banks and solar panels. A Controller monitors internal and external temperature and battery voltages for the two battery banks; it powers on/off the satellite IDU and ODU. On-site LAN equipment provides wired connection for the high-resolution video camera and the camera

sensor node; Power over Ethernet (PoE) is used to power the camera node. The Controller is responsible for: (1) Activation of high-resolution camera based on sensor input and processing of the images; (2) Storage of sensor data, images and labelling of 'events of interests'; (3) Satellite activation for data backhaul based on available power and data importance. The Controller software is implemented using Python (and its widely available libraries) obtaining an open-source platform that can be made available to the research community.

[Replace this figure with attached]

Image analysis and energy management

The *wireless sensor node* and the *Gateway* form a sub-system implementing the framework of Figure 1. The discrete data stream X_k consists of images acquired by the high-resolution camera; these can be time-lapse or triggered by the wireless sensor node (combining PIR and X-band sensor information). Data interpretation is performed by the Gateway, which uses image analysis to confirm the presence of a target (of interest) in the captured image. The system basic settings imply that relevant images are scheduled for satellite transmission, $Y_k = X_k$, whereas irrelevant ones are discarded, $Y_k = \varnothing$. Advanced settings allow extracting low-resolution version of relevant images (e.g. using scalable image coding) that retain the important information and require less power to be transmitted: $Y_k \subset X_k$ and $Y_k > X_k$.

The *camera node* and the *Gateway* form a second sub-system that operates in a similar fashion: the node hosts the motion sensor & camera (generating X_k) and the processing unit that discriminates between relevant images ($Y_k = X_k$) and irrelevant ones ($Y_k = \varnothing$); relevant images are collected by the Gateway for transmission to the user.

In both cases, image processing algorithms identify relevant information in the data. By default, these algorithms rely on a simple image differencing technique. A reference image is captured and updated multiple times over a day (time-lapse); a motion-activated image is compared to the most recent reference and a *difference image* is calculated. The number and size of connected areas ('blobs') in the difference image are used to determine whether the motion-activated image contains an actual subject, rather than (for example) moving grass. The algorithm controlling the system operation is provided in Algorithm 1 below.

Algorithm 1: Motion detection and communication
update background image
if no activity detected through PIR for 5 min

>   *put processing, and communications sub-systems to sleep*
> *if motion is detected by PIR*
>   *wake-up system*
>   *if motion confirmed through microwave radar*
>   *take picture*
>   *perform image differencing*
>   *if motion confirmed through image differencing*
>   *switch on the satellite IDU*
>   *transmit image*

The energy management in Figure 2 and Figure 3 is implemented as follows. The computational cost of processing image X_k is approximated with a worst-case estimate, independent of the content of X_k. An image is transmitted as such when a target is confirmed ($Y_k = X_k$ and $Y_k = x_k$) and discarded otherwise ($Y_k = \varnothing$ and $y_k = 0$). Prior to image processing, the expected information size is $\hat{y}_k = \Pr(X_k) \cdot x_k$, where $\Pr(X_k)$ is the probability that detection is confirmed for X_k; this is approximated with the current confirmation rate.

Experimental results

We report results from a test deployment to assess the ability of the framework to capture and identify all events of interest, that is imaging animals in the camera field of view. Results collected over a two-day period are reported in Table 1. The first column refers to time-lapse images collected by the system every two minutes; these images provide *ground truth* data. The second column refers to pictures taken in response to a motion-sensor trigger; a conventional system would communicate to the user all such images.

The motion detector is highly sensitive to ensure nearly all targets passing in the area of interest are detected. We found that in only two cases the motion sensor was not triggered, despite a target being visible in the background of a time-lapse image; this was due to the animal being too far to trigger the PIR sensor. The downside of a highly sensitive detector is that it could be triggered by moving vegetation, as many sensor-triggered images did not contain an actual target. However, a difference image had white blobs corresponding to animal location. With no animal or other moving

Table 1. Event generation for a PIR and image-based system.

	Image Category		
Events	Time Lapse	Motion Triggered	Image Based
Communicated	1392	66	19
Confirmed	2	66	19
Missed	17	2	0

object in the scene, this difference would be a totally black image. In all such cases, image processing on the captured images eliminates all false positives (47 images) without discarding any relevant information (19 images).

Image processing by the simplified scheme consumed 0.15W for 0.6 secs on Raspberry Pi (700 MHz) whereas the transmission of a compressed JPEG image of size 640 × 480 required 1.5W over 20 secs.

Application II: road condition monitoring

A second system was developed for transport infrastructure monitoring at the University of Aberdeen [36]. The system monitors environmental conditions (temperature, humidity) along roads to identify the onset of freezing and alert council authorities that can efficiently coordinate road gritting operations when icing is expected. The target locations were isolated hillside roads where ice-forming conditions occur with different frequency and timing than for city centre roads. To ease deployment, the system did not rely on dedicated communication or power infrastructures in the area; it also needed to operate without maintenance over the deployment period. A sensor node is powered by batteries sized to operate for months and uses cellular network communication. Similar to the previous scenario, energy management is crucial. In this context, this means recognising conditions that indicate ice formation is likely; when conditions are too warm or dry for ice to form ('quiet' periods), the system avoids unnecessary reports to the user.

System architecture

The system comprises:

- Independent wireless sensor nodes, each equipped with temperature & humidity sensors, a camera and a GSM modem (Figure 5). Nodes are deployed along the monitored road network. The sensor node combines Raspberry Pi and Lilypad micro-controller [37] with associated peripherals. The Lilypad node controller requires only a small power to operate continuously and manages the temperature and humidity sensors.
- Images are processed locally by activating a Raspberry Pi to confirm the presence of frost/snow on road surface, triggered by the Lilypad. The Raspberry Pi is hence only activated when conditions are close to the threshold for ice-formation (see next section). This substantially reduces average power.
- The Raspberry Pi also controls a GSM communications unit, and could optionally activate an on-board camera to take an image of the road. Each node connects to the cellular network to upload data and any images to an Internet server.
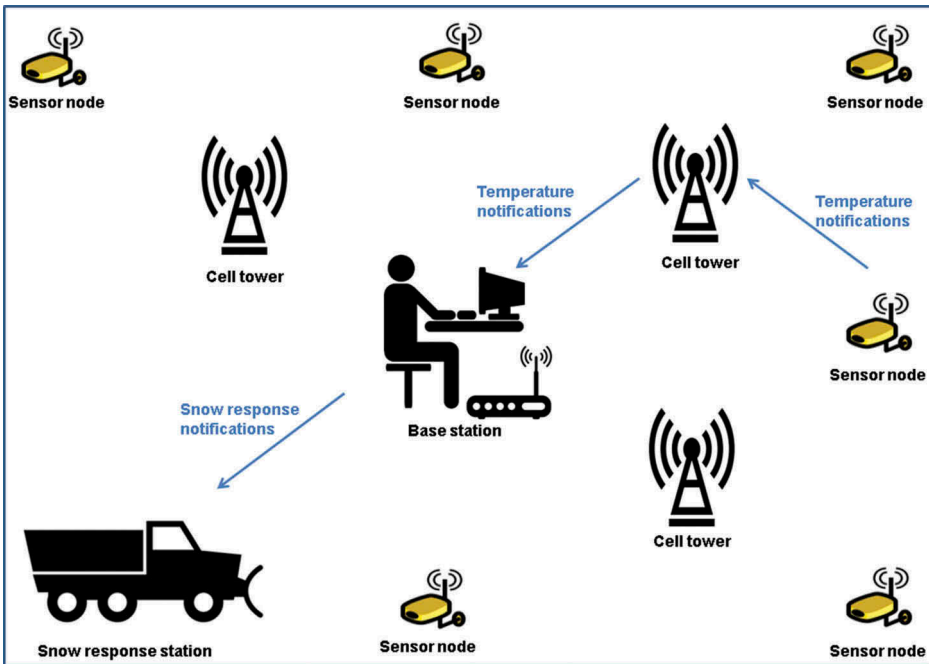
Figure 5. Architecture for road condition monitoring.

- The Internet server acts as content portal; allowing users to view an updated history of the measurements for all the nodes in the system. This generates visual alarms on a graphical interface and can send alerts (email and/or SMS) to selected staff to initiate gritting when set conditions are met at one or more locations along a road.

Environmental data analysis and energy management

With reference to the generalised system of Figure 1, the data stream for the road conditions monitoring system (under basic settings) comprises temperature & humidity samples (X_k), acquired every minute. The 'data interpretation' block uses sensor data to determine when ice formation is likely; the relevant information (Y_k) comprises those reading that convey new information to the user and are indicative of ice build-up on the road; these readings are communicated to the server via GSM link, whereas other are discarded.

We denote the k-th sensor reading as $X_k = \{T_k, H_k\}$, where T_k is the air temperature (°C) and H_k is the relative humidity (percent). The node processes the data sample and determines its relevance: T_k needs to be sufficiently different from the last temperature reading communicated to the server, denoted as T_l, $l < k$. Formally, the temperature sample T_k must satisfy the following condition:

$$|T_k - T_l| > \alpha\delta_k \tag{3}$$

where $\alpha \in [0, 1]$ is a scaling factor and $\delta_k \geq 0$ is a threshold discussed below. If condition is not met, no update is sent to the server, which continues to approximate the current temperature as T_l. When $\alpha = 0.1$, condition means that the user is updated only if the temperature variation exceeds 10% of δ_k. This is a *data-dependent* threshold defined as:

$$\delta_k = \left|T_k - T_k^{crit}\right| \tag{4}$$

with

$$T_k^{crit} = \min\left\{T_k^{dew}, T_{min}\right\} \tag{5}$$

where T_{min} is a system parameter close to 0°C and T_k^{dew} is the *dew-point temperature* associated to T_k, i.e. the temperature at which water vapour will condense out of the air, which is approximated as [38]

$$T_k^{dew} = T_k - (100 - H_k)/5 \tag{6}$$

In essence, T_k^{crit} is the *critical temperature* at which water vapour will condense and freeze on the road surface (including T_k^{dew} in accounts for cases when the air is too dry for sufficient water vapour to condense even at temperatures close to 0°C).

In broad terms, for the server to be updated with current data, a node must observe either small temperature variations close to the critical temperature or large variations around values far from the critical temperature. Consider an example case where $T_{min} = 2$°C and $T_k^{dew} > T_{min}$, then a 1°C temperature variation is required to trigger an update message for a current temperature reading of 12°C; conversely, a 0.1°C variation is sufficient for a temperature reading of 3°C. When energy constraints are particularly strict, the rate of messages sent to the user can be reduced, even when the readings are close to T_k^{crit}. In this case, our system can process a set of consecutive measurements and only report when a predetermined trend is found in those readings. Specifically, we considered sending a report only when T_k satisfies and three consecutive readings (T_{k-2}, T_{k-1}, T_k) exhibit a trend approaching T_k^{crit}. In a practical implementation, a minimum reporting rate is maintained by enforcing that at least one measurement is transmitted over a set period (every hour in our system), to act as an 'I-am-alive' message from the sensor node.

For this application, the energy management in Figure 2 and Figure 3 is simplified as follows. A temperature and humidity reading is transmitted when is verified ($Y_k = X_k$ and $y_k = x_k$) and discarded otherwise ($Y_k = \varnothing$ and $y_k = 0$). The computational cost of evaluating - with the low-power Lilypad microcontroller is so small that it is not worth estimating it in advance, as in section 3.2. However, the Raspberry Pi used an additional 0.3W for 6 secs, to transmit a

temperature value over the GSM modem thus emphasising the need to carefully choose when and which values to transmit.

A sensor node acquires data every minute, because this leads to prompt detection of the onset of freezing. In a conventional system, this increases the communication rate, with corresponding costs in node energy, but yields no benefit when freezing conditions are not approaching. The strategy discussed above allows detecting the onset of condensation with high temporal resolution without incurring a high communication cost, because only relevant temperature changes are communicated. The server collects the data sent by the nodes; when data show that conditions for the onset of condensation are present, the server can send an explicit alert to gritting vehicles.

Experimental results

A performance comparison for two identical sensor nodes, one implementing the proposed scheme and the other performing conventional monitoring, is reported in Table 2. The table shows the number of reports sent by the nodes and the energy consumption for a deployment period of approximately 18 days. With the conventional approach, a node acquires and communicates a temperature reading every minute via SMS. Two versions of the proposed approach are considered. Version 1 communicates every reading that satisfies ; Version 2 imposes the additional requirement on consecutive samples discussed in the previous section. For a timely response, it is useful to report events at a safe level above freezing, hence $T_{min} = 2$ ºC as suggested in the previous section.

The communication energy consumption was proportional to the number of messages transmitted. Therefore compared to a conventional system, the proposed system will save 91% of the communication budget. The savings are further widened with Version 2 of our system, where only 885 messages needed to be transmitted, saving nearly 97% of a conventional system communication budget. For both versions, Table 2 includes the transmission of 429 hourly updates.

Air and dew-point temperature variations over four days are shown in Figure 6. The red crosses correspond to events reported by Version 2 of our system. Green circles identify the hourly temperature notifications (sent regardless of event detection). Over a typical four day period, the number of events for a conventional system was 5760, whereas only 162 temperature notifications were sent using our system (Version 2) (with 66 notifications of relevant information and 96 hourly notifications). The estimation accuracy of

Table 2. Event generation for a conventional and proposed system.

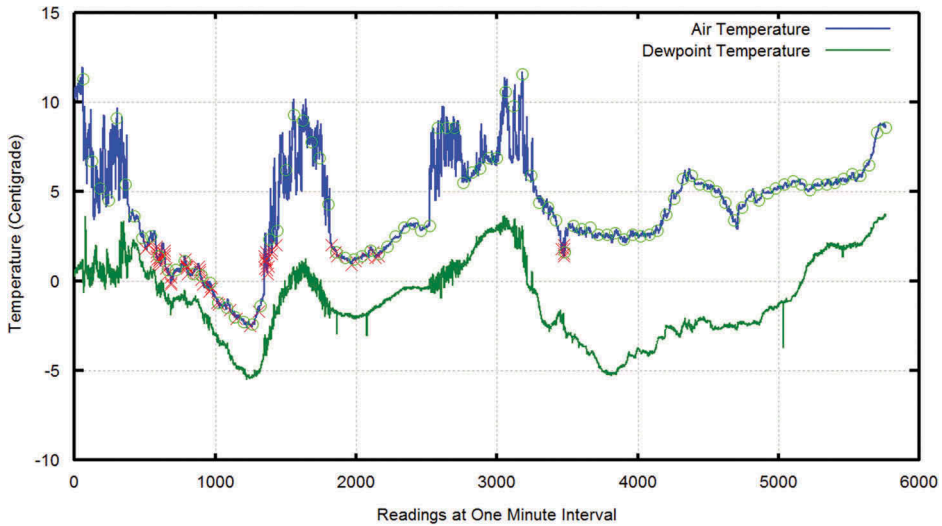|	Conventional System	Proposed System	
		Version 1	Version 2
Events Communicated	25,801	2262	885

Figure 6. Air and dew point temperature data for four consecutive days. Circles denote samples communicated by (both versions of) our system as hourly updates. Crosses denote the samples identified for transmission by Version 2 of our system.
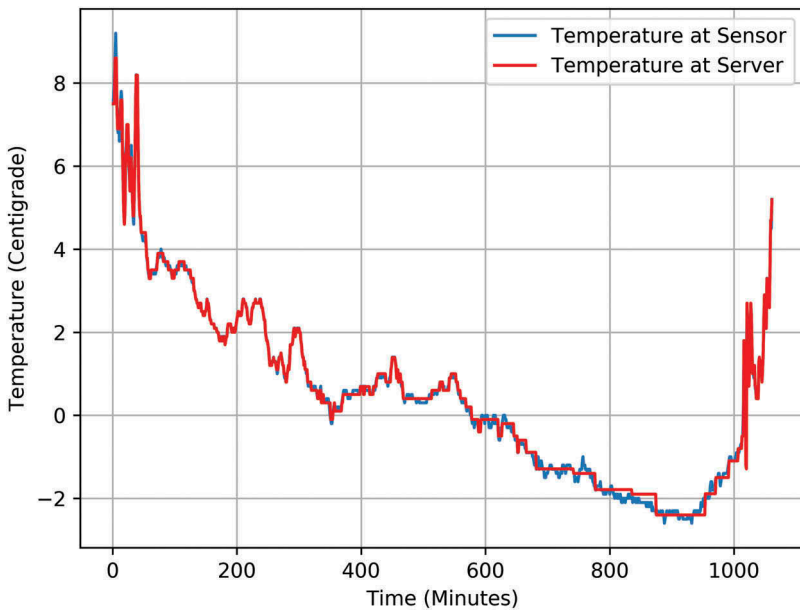

Figure 7. Estimation differences between the temperature readings at the sensor node and those transmitted to the server.

the system is shown in Figure 7. It can be seen that the system provides more accurate values around the selected temperature threshold of 2 ℃.

Ongoing developments of our system will introduce the ability to capture an image of the road when icy conditions are detected and may be able to send this to the user on demand. Frost can build quickly, leading to rapid

deteriorating and difficult driving conditions: an image of such conditions complements simple temperature trends and we expect to aid better assessment and response coordination by the road network managers and users.

Conclusions

Remote monitoring of sites without power and communication infrastructure remains challenging and requires development of new technology and algorithms to facilitate continued access to near-real time data from remote locations. The proposed framework for autonomous sensing and the implemented algorithms allow an embedded monitoring system to make autonomous decisions for data collection and processing, information extraction and communication frequency. This enables the deployed system to meet its mission with optimum utilisation of available resources over an extended period.

The developed framework and the two case study implementations form a basis for further research into the challenges of remote monitoring on a larger scale, as required by the transport sector, where monitoring of larger areas is required (long bridges, viaduct, river banks, etc.). New approaches may focus on distributed systems, where cooperation between nodes (units that capture, process and transmit information) goes beyond simple communication and involves distributed sensing and cooperative processing of the acquired data.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Hassan Hamdoun http://orcid.org/0000-0002-1444-859X

References

[1] Lynch JP, Loh KJ. A summary review of wireless sensors and sensor networks for structural health monitoring. The Shock and Vibration Digest. 2006 March;38(2):1-38.

[2] Hodge VJ, Keefe SO, Weeks M, et al. Wireless sensor networks for condition monitoring in the railway industry. IEEE Trans. Intell Transportation Systems. 2015;16(3):1088–1106.

[3] Peijiang C, Xuehua J. Design and implementation of remote monitoring system based on GSM, PACIIA '08. Pacific-Asia Workshop on Computational Intelligence and Industrial Application. 2008 Dec 19-20;1:678–681.

[4] Celandroni N, Ferro E, Gotta A, et al. A survey of architectures and scenarios in satellite-based wireless. Int J Satell Commun Network. 2013;31:1–38.

[5] Nazir S, Fairhurst G, Verdicchio F. WiSE- A satellite-based system for remote monitoring. Int J Satell Comm Network. 2016;35(3):201-214.

[6] Kroener M. Energy harvesting technologies: energy sources, generators and management for wireless autonomous applications. International Multi-Conference on Systems, Signals and Devices. 2012;1–4.

[7] Jokela M, Kutla M, Le L Road condition monitoring system based on a stereo camera, IEEE 5th International Conference on Intelligent Computer Communication and Processing, ICCP; 2009 Aug, p. 423–428.

[8] Su R, Venkatesan R, Li C. An energy efficient relay node selection scheme for underwater acoustic sensor networks. Cyber-Physical Syst. 2015;1:160–179.

[9] Mednis A, Elsts A, Selavo L Embedded solution for road condition monitoring using vehicular sensor networks, 6th International Conference on Application of Information and Communication Technologies (AICT); 2012 Oct, pp. 1–5.

[10] Zhou Z, Bai J, Dong M, et al. Game-theoretical energy management design for smart cyber-physical power systems. Cyber-Physical Syst. 2015;1(1):24–45.

[11] Allen MP, Graham E, Ahmadian S, et al. Interactive environmental sensing: signal and image processing challenges, IEEE International Conference on Acoustics, Speech and Signal Processing; 2008 Apr, pp. 5161–5164

[12] Wawerla J, Marshall S, Mori G, et al. BearCam: automated wildlife monitoring at the arctic circle. Springer J Machine Vis Appl. 2009;20(5):303–317.

[13] Dickinson P, Freeman R, Patrick S, et al. Autonomous monitoring of cliff nesting seabirds using computer vision, International Workshop on Distributed Sensing and Collective Intelligence in Biodiversity Monitoring; 2008 Dec; Amsterdam.

[14] Yunji L, Li P. Event-triggered sensor data transmission policy for receding horizon recursive state estimation. J Algorithm Comput Technol. 2017;11(2):178–185.

[15] Anastasi G, Conti M, Francesco MD, et al. Energy conservation in wireless sensor networks: a survey. Ad Hoc Networks. 2009 May;7(3):537–568.

[16] Cesare A, Giuseppe A, Francesco MD, et al. An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors. IEEE Trans. On Instrumentation and Measurement; 2010 Mar.

[17] Tongxin S, Min X, Jiahong C, et al. An energy efficient adaptive sampling algorithm in a sensor network for automated water quality monitoring. MDPI Sensors. 2017;17(11):2551.

[18] Anthony DW, Leo S, John AS. SenQ: an embedded query system for streaming data in heterogeneous interactive wireless sensor networks. In: Nikoletseas SE, Chlebus BS, Johnson DB, et al., eds. Distributed computing in sensor systems. DCOSS 2008. Lecture notes in computer science, Vol. 5067. 2008. Berlin, Heidelberg: Springer.

[19] Faster In-Network DG Evaluation of spatial aggregation in sensor networks. In Proceedings of IEEE International Conference on Data Engineering; USA; 2006 Apr 3-8, p. 148.

[20] Samuel M, Michael F, Joseph H, et al. TAG: a Tiny aggregation service for ad-hoc sensor networks. SIGOPS Operat Syst Rev. 2002;36:31–46.

[21] Wang Y, Lynch JP, Law KH Information driven wireless sensing and control for civil structures. [cited 2018 Jun 01]. Available from: https://pdfs.semanticscholar.org/a105/ae752e61591135ca3f83696cd114a649c083.pdf

[22] Raja KA, Karlmarx LR. Information driven data gathering for energy efficient wireless sensor network. Circuits Syst. 2016;7:3886–3895.

[23] Yui-Su Y, Sung-Ho K, Young-Hoon J. Intelligent data reduction algorithm for sensor network based fault diagnostic system. Int J Fuzzy Logic Intell Syst. 2009;9(4):301–308.

[24] Masoum A, Meratnia N, Havinga PJM An energy-efficient adaptive sampling scheme for wireless sensor networks. 2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing; Melbourne, VIC, 2013, p. 231–236.

[25] Chatterjea S, Havinga P. An adaptive and autonomous sensor sampling frequency control scheme for energy-efficient data acquisition in wireless sensor networks. International Conference on Distributed Computing in Sensor Systems, DCOSS 2008: Distributed Computing in Sensor Systems; pp 60–78

[26] Nazir S, Fairhurst G, Verdicchio F, et al. Resource Monitoring with wireless sensor networks and satellite, Digital Economy Conference, DE-2013; Manchester, UK.

[27] Report, Eastern Cairngorms site of special scientific interest operations requiring consent from Scottish natural heritage. 2012. [cited 2018 Jun 01]. Available from: http://gateway.snh.gov.uk/sitelink/.

[28] Newey S, Davidson P, Nazir S, et al. How suitable are camera traps for wildlife research? AMBIO Digital Conservation, Special Issue. 2015;44(4):624–635.

[29] Atmel http://www.atmel.com/devices/atmega128rfa1.aspx. Accessed 2016 Aug 2

[30] X-Band Motion Detector. [cited 2018 Jun 01]. Available from: https://www.parallax.com/sites/default/files/downloads/32213-X-BandMotionDetector-v1.1_0.pdf.

[31] Shelby Z, Bormann C The wireless embedded Internet - Part 1: why 6LoWPAN? EE Times Design, 2011. [cited 2017 Nov 15]. Available from: http://www.eetimes.com/document.asp?doc_id=1278794.

[32] Transmission of IPv6 Packets over IEEE 802.15.4 Networks, RFC 4944; 2007 Sep.

[33] Contiki. [cited 2017 Nov 15]. Available from: http://www.contiki-os.org/.

[34] Raspberry P. raspberry pi is a trademark of the raspberry pi foundation. [cited 2017 Nov 15]. Available from: http://www.raspberrypi.org/.

[35] ADC. [cited 2018 Jun 01]. Available from: http://www.abelectronics.co.uk/products/3/Raspberry-Pi/39/ADC-DAC-Pi-Raspberry-Pi-ADC-and-DAC-expansion-board.

[36] Nazir S, Hamdoun H, Verdicchio F, et al. Autonomous monitoring of critical architectures, ICCNEEE2015 The International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering Sudan; 2015 Sep

[37] Lilypad microcontroller. [cited 2018 Jun 01]. Available from: http://arduino.cc/en/Main/arduinoBoardLilyPad.

[38] Lawrence MG. The relationship between relative humidity and the dewpoint temperature in moist air: A simple conversion and applications. Bull Amer Meteor Soc 2005. 2005;86:225–233.