QoD Adaptation for Achieving Lifetime Predictability of WSN Nodes Communicating over Satellite Links

Mehdi Amirijoo*, Sang H. Son†, Jörgen Hansson‡
*Department of Computer and Information Science, Linköping University, Sweden, meham@ida.liu.se
†Department of Computer Science, University of Virginia, USA, son@cs.virginia.edu
‡Software Engineering Institute, Carnegie Mellon University, USA

***Abstract*— In this paper we propose an architecture consisting of a particular type of node, namely a communication server that collects and aggregates data, and establishes a link between the users and the sensor nodes through satellite communication. A key challenge with satisfying a lifetime requirement of the communication server is the unpredictability of the sensor data volume arriving at the communication server and the transmission power of the satellite terminal. To provide lifetime predictability we propose an approach that automatically adjusts the quality of the data such that the specified lifetime if achieved. We have shown through an extensive evaluation that the approach manages to provide an actual lifetime within 2% of the specified lifetime despite variations in workload and communication link quality.**

I. Introduction

The use of wireless sensor networks (WSNs) has increased dramatically during the last years due to the development of hardware technologies enabling large-scale deployment of very small devices capable of wireless communication, sensing and computing [1]. In this paper we consider WSNs deployed in remote areas, which are geographically separated from areas having fixed communication infrastructures or may be located near or in an area where the communication infrastructure has been eliminated. WSNs deployed in remote areas may be used for continuous monitoring of the environment, e.g., detection of earthquakes and tsunamis, monitoring of ecosystems, or may be deployed in response to events, e.g., natural disasters. In the latter it is of paramount importance that the WSN is deployed in a timely manner to closely track the event. The latter and deployment in remote areas require that battery-driven satellite terminals must be used to establish communication between a WSN and its users.

We assume the following system architecture for communicating with remote area WSNs, as shown in Figure 1. The physical entities of interest are measured by nodes having sensing and computing capabilities. A set of nodes within a geographic proximity form a sensor patch. The communication with each sensor patch goes through a base station. The sensor nodes either send aggregated or raw data to the base stations. In the latter case the raw data may be aggregated at the base station. Data from the base stations (and the sensor patches) are transmitted to a communication server (CS), which acts as a bridge between the users of the network and the sensor nodes. The CS has the following functionality. It receives queries sent by the users of the WSN and forwards the queries to the base stations for further dispersal. In response to the queries, the nodes send data to the CS where the data is stored in a database (DB) for further processing. The database acts as a data repository for data fusion algorithms, as well as for caching purposes for faster data retrieval from users. The CS also includes a mobile satellite personal communication system (SPCS) terminal, consisting of a satellite antenna and a modem. The CS, i.e., the computer and the satellite terminal, runs on batteries. The satellite system is connected to the Internet, which enables communication with a set of users via a communication central.

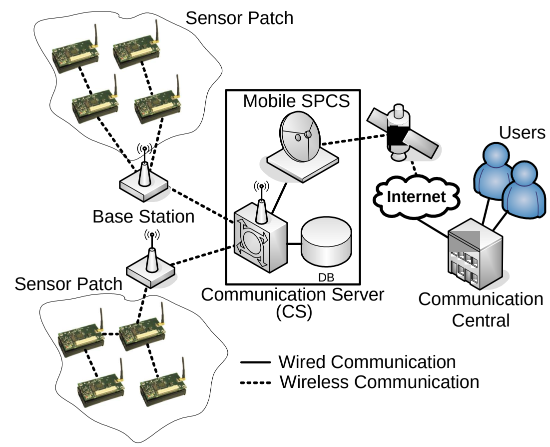


Fig. 1. An architecture for remotely deployed WSNs.

In this paper we focus on WSNs that must satisfy a given lifetime requirement. For example, consider a remote area that needs monitoring due to a natural disaster. Assume that it takes three days until a fixed communication infrastructure is established. Until then we need to setup a temporary WSN that is required to deliver sensed data for three days. Although, it is desirable that all nodes satisfy the lifetime requirement, the importance of individual nodes meeting the lifetime depends on the actual type of node. For example, the CS failing to meet the lifetime has a much more significant impact compared to a sensor failing to meet the lifetime. As such there exists a hierarchy of nodes with respect to criticality of meeting the network lifetime. We have, therefore, in this paper chosen to focus on the lifetime management of the CS, since this is the most critical node. As argued in Section VII, there is a need for developing new approaches for guaranteeing the lifetime of the CS since previous work (e.g. [2]) addressed lifetime management of smaller nodes, e.g., the sensor nodes, which do not hold for the CS.

The main contribution of this paper is an approach for guaranteeing the lifetime of the CS under uncertain workload and satellite communication channel impairments. We start by exploring the relationship between energy consumption in CS and quality of data (QoD), defined in terms of the sampling period of the sensors. We introduce a system specification model enabling a user of the wireless sensor network to declare the required network lifetime, and the desired as well as the lowest QoD. Given the system specification an approach for satisfying the lifetime requirement is introduced, and we show through extensive performance evaluation that the actual lifetime is within 2% of the required lifetime. The approach satisfies the QoD requirement and successfully manages to provide an actual lifetime that is very close to the required lifetime, despite unpredictable workload and satellite channel impairments.

The outline of the paper is as follows. In Section II we give an overview of the current satellite communication technologies and in Section III we present the problem formulation. In Section IV we outline our assumptions on the WSN and its constituents, and this is followed by the approach, which is given in Section V. An extensive evaluation of the approach is presented in Section VI followed related work in Section VII, and conclusions in Section VIII.

II. Overview of Satellite Communication Systems

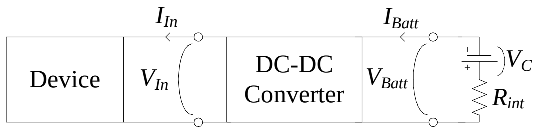
The use of low-earth orbit (LEO) satellite systems has increased due to their ability to provide global coverage including the polar regions. The antennas are omnidirectional, which means that they radiate and receive radio signals from all directions, hence, there is no need to manually direct the antennas. It has been reported that future mobile SPCS will employ LEO satellites due to their appealing characteristics, such as low propagation delay, lower power consumption (due to low transmission power), and smaller antennas, see e.g. [3]. To provide more bandwidth and smaller antennas, the trend points toward communication frequencies in the range of 3-30 GHz. However, attenuation due to rain, fog, and cloud influences significantly for systems above 3 GHz [4]. During severe weather conditions, e.g., rain, sand storm, or when terrestrial landscape blocks the radiowave path, the signal quality (signal to noise ratio) decreases and, consequently, the transmission error increases. Mobile satellite systems, e.g., Globalstar [5], use uplink power control to maintain an acceptable level of transmission error. The transmission power is increased in response to decreasing signal quality.

Studies have shown that forested and suburban areas suffer significantly from channels impairments due to the terrestrial environment, e.g., buildings, bridges, and trees [6]. It was shown that in 90% (or 99%) of the farmland area, the received signal was greater or equal to 0.4 (or 0.06) times the line of sight value, which corresponds to an attenuation of 4 dB (or 12 dB). For forested and suburban areas the situation was even worse. In 90% (or 99%) of the forested and suburban area, the received signal was greater or equal to 0.06 (or 0.008) times the line of sight value, corresponding to an attenuation of 12 dB (or 21 dB). This means that we have to increase the transmission power by 16 times (12 dB) compared to the line of sight power in order to be able to transmit from 90% of the forested and suburban areas and 99% from the farmland areas. The attenuation due to rain is time-variant and depends on the transmission frequency. The International Telecommunications Union (ITU) has developed a procedure for predicting the attenuation caused by rain [7]. For example, the ITU procedure predicts that in 0.10% of the time the rain attenuation in Washington D.C. (USA) exceeds 3 dB, 4 dB, 8 dB, and 15 dB for frequencies 12 GHz, 14 GHz, 20 GHz, and 30 GHz. For a more thorough discussion on satellite systems we refer to [8].

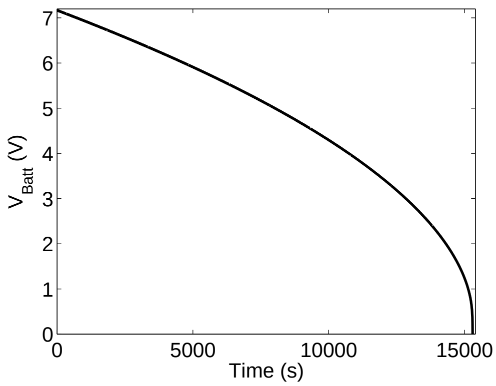
III. Problem Formulation

The energy consumption of the CS increases as the workload applied on the computer, hence, the CPU utilization increases. The CPU utilization increases as the number of queries and data streams increases, and more sophisticated data fusion algorithms are used. Furthermore, the energy consumption of the second component of the CS, i.e., the satellite terminal, increases as the amount of transmitted data increases and the radiowave attenuation increases as discussed in Section II.

In general, QoD increases as the transmission rate and the CPU utilization increases, e.g., the sampling periods decreases and more complex data fusion algorithms are used. If QoD is set too high, then the battery of the CS may be depleted too fast and the lifetime requirements may not be met. On the other hand, if QoD is too low, then the lifetime requirements are satisfied, however, the QoD is unnecessarily low. Given the lifetime requirement for a WSN, determining the right level of QoD a priori to deployment is not possible for the following reasons: (1) the rain attenuation varies over time and is unpredictable as described in Section II, (2) the particular area where the CS is deployed may suffer from radio fading and shadowing due to terrestrial impediments, (3) unexpected events that need additional sensing may occur, (4) the mission and lifetime requirement may change, (5) one or several CSs may fail in the case when there are several CSs deployed for increased fault tolerance and lifetime. As such, the CS needs to determine a suitable QoD when deployed and adjust the QoD depending on the prevailing conditions, e.g., weather and number of events that need to be monitored. The problems that we address in this paper are the following: How can QoD be defined in the context of WSNs? What is the relationship between QoD and energy consumption and how does a certain QoD affect the lifetime of the CS? How can lifetime requirements be met despite unpredictable radiowave channel



(a) A DC-DC converter is used to stabilize the voltage supplied to a device.



(b) The voltage of the battery as a function of time.

Fig. 2.

attenuation, unexpected number of events, and changes of mission and lifetime requirements? In summary, the results obtained in this paper give an insight in how mobile satellite communication can be used for accessing WSNs and retrieving data from them such that lifetime requirements are met.

IV. System Model

A. Sensor Patches

We make the following assumptions regarding the sensor patches and the sensor nodes. Sensor nodes sample data periodically and send raw or aggregated data to the communication server. There is a set of data streams $Stream_1, \ldots, Stream_n$ arriving at the CS. Within each stream, the sensor data arrives periodically with a period p_i. Further, we assume that it is possible to change the arrival rate of the streams. This is done by (i) altering the sampling period of the sensors, or (ii) altering the aggregation window if temporal aggregation [9] is used at the base stations or the sensor nodes.

B. CS - Communication Server

The CS has three constituents, namely, a computer, a satellite terminal, and the battery powering the computer and the satellite terminal. Below we model each of the constituents with respect to their power consumption.

1) Battery: Devices powered by a battery require the DC input voltage V_{In} of the device to be constant. Since, the voltage of the battery decreases over time, a DC-DC converter is used to stabilize and maintain the supply voltage at a constant level, as shown in Figure 2(a). The device discharges the battery with the current $I_{In}(t)$. Hence, the power consumption of a device connected to the battery is $P_{In} = V_{In}(t)I_{In}(t)$. Let $V_C(t)$ be the value of the state of charge, i.e., how much charge the battery holds at time t. The internal resistance of the battery is R_{int} and the efficiency of the DC-DC converter is η. The capacitance of the battery is $C = 3600 \times Capacity$, where $Capacity$ is usually provided in the battery's datasheet. We employ the battery model presented by Benini et al., where the voltage of the battery V_{Batt} is [10],

$$V_{Batt}(t) = V_C(t) - I_{Batt}(t)R_{int}, \tag{1a}$$

$$V_C(t) = V_C(0)\left(1 - \frac{1}{C}\int_0^t I_{Batt}(x)dx\right), \tag{1b}$$

$$I_{Batt}(t) = \frac{V_{In}I_{In}}{\eta V_{Batt}(t)}. \tag{1c}$$

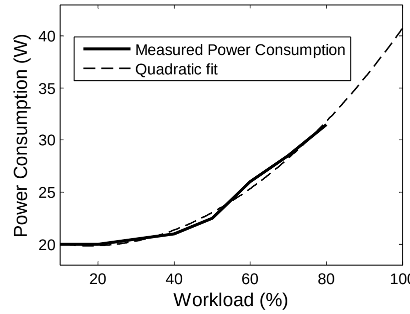


Fig. 3. Measured data from [11] (Figure 34.3, algorithm LEDF3) and corresponding quadratic fit. The power consumption of the LCD display corresponding to 8W is subtracted from the original data.

Figure 2(b) shows how the voltage of the battery evolves as P_{In} is constant. As we can see, V_{Batt} is a nonlinear function over time and decreases significantly at the end.

2) Computer: We assume that the computer is similar to modern energy efficient laptops without any display (to save additional energy). Within a stream $Stream_i$, the arrival of a sensor datum triggers data processing and analysis with an estimated execution time (eet_i). The actual execution time (aet_i) of the data processing and analysis is unpredictable and may deviate from the estimated execution time. The estimated workload generated by the arrival of a sensor datum is $\frac{eet_i}{p_i}$.

We assume that the computer employs digital voltage scaling. We model the power consumption of the computer based on the results obtained by Swaminathan et al., where they measured the power consumption of a laptop with a AMD Mobile Athlon 4 processor as a function of the workload [11]. To model the power consumption, we fitted a quadratic polynomial to the measured data, obtaining a relation between the power consumption of the computer and the workload as shown in Figure 3. We also fitted higher order polynomials, however, the results were less satisfactory.

3) Satellite Terminal: We assume that each data arrival generates a number of bits that need to be transmitted via the satellite link. For simplicity, we assume that the number of bits that need to be transmitted is proportional to the execution time of the data processing and analysis task. This assumption does not affect the approach for guaranteeing lifetime and, hence, we expect our approach to also function satisfactorily when the transmission model is different.

To model the effects of channel impairments we have studied the existing Globalstar GSP-1620 terminal [5] and used available power consumption data for devising a model describing the relationship between power consumption and channel impairments. This gives that during transmission the total power consumption of the terminal varies between approximately 4.6 W and 7.3 W depending on the weather conditions and terrestrial shadowing and blocking. The energy consumption is 0.5 W when the terminal is idle, i.e., not transmitting. Finally, we assume that the maximum bandwidth is 10 kbps (the Globalstar system has a maximum bandwidth of 9.6 kbps). For details regarding the modeling of the energy consumption of the satellite terminal we refer to [8].

V. Approach

In this section we outline the proposed approach for satisfying a given system specification. We start by defining the system specification model followed by a description on the architecture and its constituents.

A. System Specification

In this section we define QoD and provide a system specification model consisting of QoD and lifetime requirements. We define QoD in terms of the sampling period of the sensors. Recall from Section IV-A that $Stream_i$ has a nominal period p_i. The actual period of the streams is altered through period scaling, where the actual period of $Stream_i$ is given by $p'_i = sp_i$. We define QoD in terms of the period scaling factor s and we say that QoD increases as s decreases.

The system specification $< \mathcal{L}, \mathcal{Q} >$ consists of a lifetime specification $\mathcal{L}$ and a QoD specification $\mathcal{Q}$. The lifetime requirement $\mathcal{L} \in Z^+$ gives the required lifetime of the WSN in seconds. The QoD specification $\mathcal{Q}$ is given by, $\mathcal{Q} =< p_1, \ldots, p_n, s_{max} >$, where $p_1, \ldots, p_n$ denote the nominal periods of the data streams $Stream_1, \ldots, Stream_n$ and $s_{max} \geq 1$ denotes the maximum tolerable period scaling factor, i.e., $s \leq s_{max}$. According to above the maximum sampling period of the sensors is $s_{max}p_1, \ldots, s_{max}p_n$. Note that there is no lower bound on the periods as in general the user(s) requires a sampling period as low as possible (such that lifetime requirements are met).

B. Approach for Satisfying the System Specification

Key issues in satisfying the system specification include handling uncertanties in the arrival load, inaccuracies in the power consumption estimates of the computer and the satellite terminal, and unexpected changes in channel impairments (satellite communication link). Using feedback control has shown to be very effective for a large class of computing systems that exhibit unpredictable workload and model inaccuracies [12]. To provide system specification guarantees without a priori knowledge of the workload or accurate system model we apply feedback control, see Figure 4.

The CS stops operating when V_{Batt} reaches a certain cutoff threshold. To meet the system specification, we force V_{Batt} to follow a target trajectory $V_{Batt,r}$. The target trajectory is found by solving the differential equation (1) for an average power $P_{In} = V_{In}I_{In}$, which results in V_{Batt} reaching the

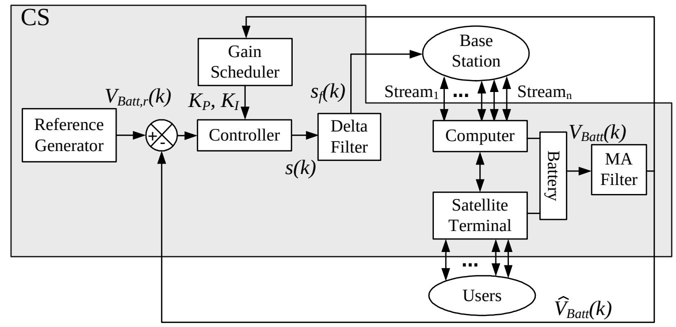


Fig. 4. The feedback architecture for guaranteeing the system specification.

cutoff threshold at time $\mathcal{L}$. We refer to V_{Batt} as the controlled variable and we force V_{Batt} to follow the reference voltage $V_{Batt,r}$ by changing the manipulated variable s, which is computed by the controller.

Let T be the sampling period and $x(k)$ be the value of a variable at time kT. The battery voltage $V_{Batt}(k)$ is periodically measured at each sampling instant $0T, 1T, 2T, \ldots$ and filtered using a moving average (MA) filter to smoothen out large deviations from one sampling period to another. More specifically, the filtered signal $\hat{V}_{Batt}(k)$ is given by, $\hat{V}_{Batt}(k) = \alpha\hat{V}_{Batt}(k-1) + (1-\alpha)V_{Batt}(k)$, where $0 < \alpha < 1$ is the forgetting factor. The filtered voltage $\hat{V}_{Batt}(k)$ is fed back and the difference between the reference and the filtered voltage is formed, which acts as an input to the controller. Using this difference, the controller computes a period scaling factor $s(k)$ such that $V_{Batt}(k)$ converges to $V_{Batt,r}(k)$. The period scaling factor needs to be transmitted to the base stations for further dispersal to the nodes collecting the data. However, since transmitting data over a wireless link is very energy consuming in WSNs, we transmit control packets (containing the period scaling factor) only when the difference between the current scaling factor and the previously transmitted scaling factor is greater than a threshold. We call this technique delta filtering, which is further discussed in Section V-E.

C. Controller

To design controllers there is a need to model the behavior of the so-called controlled system. The controlled system in our case (see Figure 4) consists of the subsystem with $s(k)$ as the input and $V_{Batt}(k)$ as the output, i.e., the delta filter, computer, satellite terminal, and the battery. Once we have an adequate model we can then continue to design the controller.

1) Modeling: To design controllers it is necessary to use models that describe the relationship among the controlled variable $V_{Batt}(k)$ and the manipulated variable $s(k)$ in terms of differential or difference equations. Since computer systems are inherently discrete, below we only address models based on difference equations. The model that we are seeking has the following form,

$$a_1V_{Batt}(k) + \cdots + a_{m+1}V_{Batt}(k-m) = \\ b_1s(k) + \cdots + b_{n+1}s(k-n). \quad (2)$$

This equation basically says that the current voltage is a linear function of the previous battery voltages, and the current and previous period scaling factors. Modeling deals with the problem of choosing the proper model order, i.e., m and n, and the parameters $a_1, \ldots, a_{m+1}$ and $b_1, \ldots, b_{n+1}$, such that the difference between the output predicted by the model and the measured output is minimized. Hence, we would like the model to be as accurate as possible.

The information available to the designer for the purpose of modeling is typically of two kinds. First, there is knowledge about the system being controlled based on equations or laws of science describing the dynamics of the system. Using this approach, a system designer describes the system directly with mathematical equations based on the knowledge of the system dynamics. However, in the case the mathematical function of a system is unknown, or too complicated to derive, the use of models tuned using system profiling and statistical methods have shown to provide good results [13]. In these circumstances the designer turns to data taken from experiments directly conducted to excite the controlled system. Statistical methods are then used to tune the parameters of the model, i.e., $a_1, \ldots, a_{m+1}$ and $b_1, \ldots, b_{n+1}$. Note that the order of the model still has to be determined by the model designer.

Now, let us turn to the controlled system presented in Figure 4. The relationship between $s(k)$ and $V_{Batt}(k)$ is very intricate, since a change in $s(k)$ translates (nonlinearly) into a change in arrival load, which affects (nonlinearly) the power consumption of the computer and the satellite terminal and, consequently, $V_{Batt}(k)$. Also, there may be delays between a change in $s(k)$ and $V_{Batt}(k)$ since $s(k)$ needs to be transmitted to the sensor nodes. Rather than going into the details of each step we model the relationship between $s(k)$ and $V_{Batt}(k)$ by using profiling data and statistical methods, as mentioned above.

We expect $s(k)$ to vary within an envelope of 0.5 and 2.0. To fully excite the system, we apply a binary signal that shifts randomly between 0.5 and 2.0 and we measure $V_{Batt}(k)$. Our studies have shown that the relationship between $s(k)$ and $V_{Batt}(k)$ changes as $V_{Batt}(k)$ decreases, i.e., the variables a_i and b_j in (2) are varying according to $V_{Batt}(k)$. Since $V_{Batt}(k)$ decreases over time, this implies that we have a time-variant system. We have therefore tuned two first order models, G_1 and G_2 that describe the dynamics of the controlled system for voltages 4.5V to 7.2V and 3.0V to 4.5V, respectively.

Now that we have derived a model, the crucial question is whether it is good enough for the intended purpose. Testing if a given model is good enough is known as model validation. This is carried out by checking whether the derived model agrees sufficiently well with observed data from the modeled system. Specifically, the same input is fed into the controlled system and the model of the controlled system. The measured output of the controlled system and the output predicted by the model are then compared and their difference formed. The percentage of the measured output variation explained by the model is given by the metric fit (see, e.g., [13]). The fit, which is between 0% and 100%, increases as the difference between the measurements and predictions decreases.

We have obtained a 94.66% fit for G_1, which implies that the predictions are very close to the actual measured data. Although the fit for G_2 is less (the fit is 65.49%), we have observed that G_2 manages to explain the variations in $V_{Batt}(k)$ significantly better than G_1 for voltages 3.0V to 4.5V. As such, $G_2(z)$ adequately describes $V_{Batt}(k)$ for voltages 3.0V to 4.5V. Now that we have arrived at a model, the next step becomes to tune a suitable controller.

2) Controller Design: Given a model, one can design a controller based on a variety of existing mathematical techniques, such as root locus, frequency response, and state space design [12]. Using mathematical techniques enable us to derive analytic guarantees on how well the controller manages to keep the controlled variable $V_{Batt}(k)$ near its reference $V_{Batt,r}(k)$. Ideally, we would like $V_{Batt}(k)$ to equal $V_{Batt,r}(k)$, however, this is not possible due to variations in energy consumption.

We have found that a proportional integral (PI) controller [12] provides adequate performance. Let the control error, $e(t)$, be the difference between the reference and the controlled variable, i.e., $e(t) = V_{Batt,r}(k) - V_{Batt}(k)$. PI controllers consist of two different parts, namely, the proportional and the integral controller. The proportional (P) controller is given by $K_P e(t)$ where K_P is the proportional gain. We can view the P controller as an amplifier that adjusts the manipulated variable $s(t)$ based on the control error. A P controller may not manage to force $e(k)$ to converge to zero, i.e., we may have a so-called steady-state error. For this reason we add an integral part forming the PI controller,

$$
\begin{aligned}
s(k) &= K_P e(k) + K_I \sum_{j=0}^{k} e(j) && \text{(3a)}\\
&= s(k-1) + K_P(e(k) - e(k-1)) + K_I e(k) && \text{(3b)}
\end{aligned}
$$

where the integral part increases in magnitude forcing $V_{Batt}(k)$ to converge to $V_{Batt,r}(k)$ if the error $e(k)$ persists over time. Note, equations (3a) and (3b) are similar, however, (3b) is computationally lighter than (3a).

Using mathematical analysis techniques we have found that the PI controller (3) is sufficient in forcing $e(k)$ to converge to zero, i.e., we have that $\lim_{k\to\infty} e(k) = 0$. This means that $V_{Batt}(k)$ is guaranteed to converge to $V_{Batt,r}(k)$, even if there are variations in energy consumption of the computer and satellite terminal. We have derived K_P and K_I based on the model tuned in Section V-C.1. Due to space limitation we have not provided a full description of the design and tuning method and the interested reader is referred to [8].

D. Gain Scheduler

In Section V-C.1 we observed that the controlled system is time-variant, meaning that the relationship between $s(k)$ and $V_{Batt}(k)$ changes over time. To handle the time-varying character of the controlled system we have used gain scheduling [12]. Initially, when $7.2V \leq V_{Batt}(k) < 4.5V$ we use controller C_1 that is tuned using model G_1. As soon as $V_{Batt}(k)$ reaches 4.5V we switch to controller C_2, which is tuned using model G_2. We have observed that using gain scheduling results in less variance and oscillations in $V_{Batt}(k)$ for lower battery voltages. This in turn results in an actual lifetime closer to the desired lifetime.

E. Delta Filter

We employ a simple and very effective method for lowering the number of control packets (containing the period scaling factor) that need to be transmitted to the sensor nodes. We say that $s_f(k)$ is the delta filtered signal of $s(k)$. If the difference between the output of the controller $s(k)$ and $s_f(k-1)$ is less than δ_f (i.e., $|s(k) - s_f(k-1)| < \delta_f$), then $s_f(k)$ is set to $s_f(k-1)$. Otherwise we set $s_f(k)$ to $s(k)$ and transmit the new value $s(k)$ to the base stations, which in turn disperse the values to the sensor nodes. During steady-state when $s(k)$ does not change considerably, then we may discard more of the packets containing the period scaling factor. However, when the system is in transient-state, e.g., the workload of the computer and the transmission power vary, then we may be able to discard only a few number of packets since $s(k)$ changes significantly. Hence, the number of packets discarded depends on δ_f and the variance of $s(k)$.

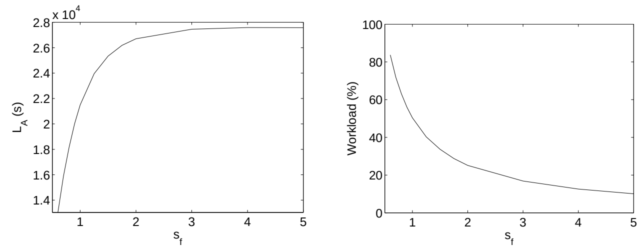
VI. Performance Evaluation

The goal of the experiments are as follows. First, we want to establish the relationship between QoD and lifetime. Second, we want to determine how accurately a given system specification in terms of a lifetime specification $\mathcal{L}$ and a quality of data specification $\mathcal{Q}$ is satisfied when using the approach outlined in Section V. Finally, we want to evaluate the benefits in using delta filtering, namely, we want to show how δ_f affects the difference between the actual and the specified lifetime and the amount of packets that can be discarded.

A. Experiment Setup

In our performance evaluation we have used a simulator, where the underlying characteristics of the battery, computer, and satellite terminal are based on the assumption outlined earlier in Section IV. Within each stream $Stream_i$, the nominal period p_i is uniformly distributed in the range (1s,50s), i.e. $U : (1s, 50s)$. The actual execution time of the data processing and analysis is unpredictable and is given by the normal distribution $N : (eet_i, \sqrt{eet_i})$, where eet_i is uniformly distributed according to $U(5ms, 50ms)$. Note that only eet_i is known to the system, i.e., the actual execution times are unknown. For all experiments the cutoff voltage of the battery is set to 2.0V.

For all the performance data, we present the average of 10 simulation runs. We have derived 95% confidence intervals based on the samples obtained. We denote the average of a time-domain variable $x(k)$ with $\bar{x}$. In addition to monitoring the workload of the computer and $s_f(k)$ we also measure the following performance metrics. Let the lifetime difference $L_\triangle = |L_A - \mathcal{L}|$ denote the absolute difference between the actual lifetime L_A and the specification $\mathcal{L}$. The performance of the approach with respect to satisfying $\mathcal{L}$ increases as



(a) (b)

Fig. 5. QoD vs lifetime and workload.

$L_\triangle$ decreases. Ideally, we would like $L_\triangle$ to be zero. To measure the number of control packets that are discarded using delta filtering we introduce the period scaling discard ratio, $DR = \frac{N_{Discard}}{N_{Rescale}}$, where $N_{Rescale} = \lfloor \frac{L_A}{T} \rfloor + 1$ is the total number of times the manipulated variable $s(k)$ is computed and $N_{Discard}$ is the number of times a computed value of the manipulated variable is discarded.

To the best of our knowledge this is the first paper describing an approach for guaranteeing a given lifetime where there is little or no information about the energy cost related to transmission and data processing. To fully evaluate the approach we devise a baseline as follows. As mentioned in Section VII, a substantial number of papers assume that the voltage of the battery decreases linearly, e.g., [2]. Therefore, in the baseline we assume that the baseline reference voltage $V_{BattBas,r}(k)$ decreases linearly, reaching the cutoff voltage at the lifetime specification, i.e., $V_{BasBatt,r}(t) = -\frac{V_{BasBatt,r}(0)-2.0}{\mathcal{L}}t + V_{BasBatt,r}(0)$, where $V_{BasBatt,r}(0)$ denotes the value of the baseline reference at time 0s.

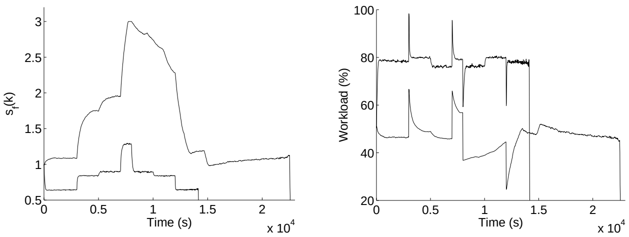
B. Experiment 1: Effects of QoD on Lifetime

The goal of this experiment is to show the effects of QoD on the lifetime of the CS. The setup of the experiment is as follows. We assume that no delta filtering is used, i.e. $\delta_f = 0$ and $s(k) = s_f(k)$. We vary $s_f(k)$ according to 0.60, 0.70, 0.80, 0.90, 1.00, 1.25, 1.50, 1.75, 2.00, 2.50, 3.00, 4.00, and 5.00. The actual lifetime L_A and the workload applied on the computer are measured.

Figure 5(a) shows the actual lifetime L_A as a function of the period scaling factor s_f.[1] We notice that the actual lifetime L_A increases fast for small s_f. However, the slope of L_A decreases for larger s_f and at approximately $s_f = 3$ there is no further significant increase in L_A. We say that L_A becomes saturated at $s_f = 3$, since a change in s_f does not result in a significant change in L_A. Therefore, for the remaining of this paper we set $s_{max} = 3$, since there is no further gain in having $s_f > 3$ or equivalently a lower QoD.

We need to understand the reasons for the saturation of L_A as shown in Figure 5. When nominal periods are chosen, i.e., $s_f = 1$, then the workload arriving to the CS is approximately 50%. Now, there are two reasons why the slope of L_A

[1]Differential equation (1b) requires that a small simulation step is chosen. To keep the overall simulation time within reasonable limits we have chosen a small battery, which results in an actual CS lifetime in the order of hours. We expect, however, that our results also hold when larger batteries are used.



(a) (b)

Fig. 6. Performance of the approach in the time-domain.

decreases with increasing s_f. First, the workload is inversely related to s_f as shown in Figure 5(b) and, therefore, the relationship between L_A and s_f is nonlinear. Second, when s_f is between 3 and 5, then the actual workload arriving at the CS is correspondingly between 16.7% and 10%. As shown in Figure 3 the power consumption of the computer does not change considerably when the load is varied between 10% and 20%. Since the power consumption of the computer accounts for a larger portion of the total power consumption, the difference in lifetime is not significant when altering the load between 10% and 20%. Therefore, the lifetime is not affected noticeably when varying s_f between 3 and 5.

In this experiment we showed that the lifetime is significantly affected by s_f when s_f is small. Experimental data also showed that there is a limit for which a further reduction of QoD does not result in significant lifetime savings. These findings may serve as a guideline for choosing appropriate lower boundaries for QoD.

C. Experiment 2: Satisfying System Specification under Varying Conditions

The goal of this experiment is to show whether a system specification in terms of a lifetime and QoD requirements can be satisfied under varying computer load and satellite channel impairment. The setup is as follows. We evaluate the performance of the approach with respect to L_Δ for the following lifetime requirements (which are chosen arbitrarily): $\mathcal{L}$ = 14440s, 16040s, 17900s, 20120s, 22760s, and 26020s. The QoD specification is given by $s_{max} = 3$. We assume that no delta filtering is used, i.e., $\delta_f = 0$ and $s(k) = s_f(k)$. To change the workload characteristics we carry out the following. We add an additional workload of 20% at times 3000s and 7000s and we remove 20% workload at times 8000s and 12000s. In addition we assume that channel impairments are initially zero and at time 5000s we simulate heavy channel impairments (due to e.g., rain) resulting in increased transmission power. At time 10000s we set the channel impairment to zero. We measure $s_f(k)$, $\bar{s}$, the workload of the computer, and L_Δ.

Figure 6 shows the results of the experiment in the time-domain. From Figure 6(b) we see that the workload increases by 20% at time 3000s. This results in a greater energy consumption, causing $V_{Batt}(k)$ to drop below $V_{Batt,r}(k)$. To keep $V_{Batt}(k)$ at $V_{Batt,r}(k)$, the workload of the computer is reduced by increasing the period scaling factor $s_f(k)$ as shown

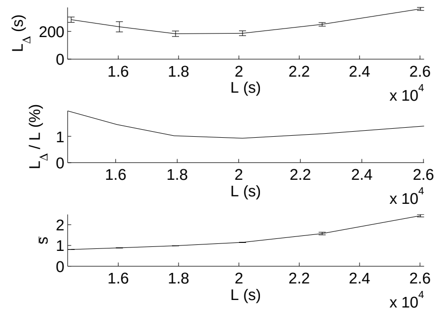


Fig. 7. Average-based performance

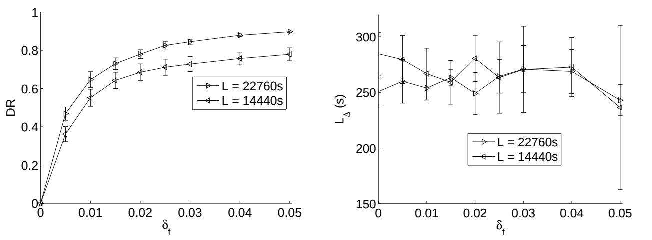
in Figure 6(a) (note that $s(k) = s_f(k)$ in this experiment). At time 5000s the transmission power of the satellite terminal increases resulting in an increase in the overall power. Again, the controller reacts by increasing $s_f(k)$. At time 7000s, the workload of the computer is further increased, which results in an increase in $s_f(k)$. At times 8000s and 12000s the workload decreases and at time 10000s the transmission power of the satellite terminal decreases, resulting in a lower energy consumption. Consequently, the controller decreases the period of the sensor streams, i.e., $s_f(k)$ is reduced, to fully utilize the remaining energy in the battery.

Figure 7 shows the lifetime difference L_Δ and the average $s(k)$, i.e., $\bar{s}$, versus the lifetime requirements. We see from the bottom figure that $\bar{s}$ is monotonically increased in response to increased lifetime requirements. The period scaling factor $\bar{s}$ varies between 0.8025 ± 0.002 and 2.4341 ± 0.0598. The top figure shows that the lifetime difference L_Δ varies from 183.1 ± 19.8s to 362.0 ± 11.2s. This means that we can say with 95% confidence that L_Δ lies between 163.3s and 373.2s. We plot the ratio $\frac{L_\Delta}{L}$ as a function of L and this gives that $\frac{L_\Delta}{L}$ falls within 0.9% and 2.0%.

We have not provided data for the baseline due to space limitation. Our evaluation shows, however, that the baseline fails to provide lifetime guarantees even during stationary conditions when the load of the computer and the transmission power is constant. We showed that during stationary condition $\frac{L_\Delta}{L}$ spans from 5.3% to 16.8%, which is unacceptable in terms of lifetime predictability. In contrast, experimental data presented above show that the approach outlined in Section V manages to guarantee the lifetime within $\pm 2\%$ of the lifetime specification $\mathcal{L}$ even though the workload and transmission channel characteristics vary over time.

D. Experiment 3: Effects of Delta Filtering

In this experiment we determine whether a significant gain in reducing the number of packets can be achieved, while at the same time maintaining acceptable differences in actual lifetime and required lifetime. The setup is as follows. We evaluate the performance of the approach with respect to L_Δ and DR for the following lifetime requirements: $\mathcal{L}$ = 14440s and 22760s. The QoD specification is given by $s_{max} = 3$. We assume that δ_f varies according to 0.000, 0.002, 0.005, 0.010, 0.015,



(a) (b)

Fig. 8. The effects of delta filtering on the lifetime difference and packet discard ratio during transient-state.

0.020, 0.025, 0.030, 0.040, and 0.050. We assume the same experimental setup as in Experiment 2 (Section VI-C), namely we vary the workload and the channel impairment during run-time.

As can be seen from Figure 8(a), we are able to obtain a very high DR meaning that we only send a small fraction of the control packets. Setting δ_f to 0.01 only, results in more than 50% of the control packets to be discarded, which is a substantial improvement in reducing the number of transmitted packets. Figure 8(b) shows the lifetime difference for varying δ_f. As can be seen, the average lifetime difference does not change considerably with varying δ_f meaning that we are able to skip transmitting the period scaling factor and still obtain accurate lifetime guarantees. This means that the lifetime of the network is prolonged while at the same time lifetime guarantees are made for the CS.

VII. Related Work

Numerous reports have mentioned the use of satellite networks for communicating with remote area WSNs, e.g., [1], [14]. However, details regarding the energy supply to the satellite terminals have been omitted. In TinyDB [2], it is possible for the users to explicitly declare the lifetime of the network. The system adjusts the sampling rates of the sensors such that a given lifetime requirement is satisfied. It is assumed that the exact cost of transmitting and processing data is known. Although, these assumptions have shown to be effective for small nodes, the same is not true for larger devices such as the base stations or the CS. For example, the energy consumption for processing data depends on the workload (actual CPU frequency used), cache, and branch prediction. Further, the quality of the satellite communication link varies heavily over time, resulting in the transmission cost to change in an unpredictable manner. In TinyDB, it is assumed that the voltage (which is a measure of remaining energy capacity) decreases linearly, which does not hold for most battery types where a DC-DC converter is used, as discussed in Section IV-B.1. In contrast to the TinyDB approach, in this work we do not assume accurate knowledge of the cost of transmitting over a wireless link and processing data, and we employ a more accurate battery model.

VIII. Conclusions

In this paper we suggest that a particular type of node, namely, the CS to bridge the gap between a WSN and the users through satellite communication. We have focused on the problem of achieving lifetime guarantees for the CS, since this node is the most critical point in the network and previous work on lifetime management does not apply to the CS. Key challenges include uncertainty in the energy consumption of the satellite terminal and the computer. We have proposed an approach based on feedback control where the QoD, defined in terms of the sampling period of the sensors, is continuously adjusted to meet the specified lifetime. The approach is adaptive in that it reacts to unexpected variations in energy consumption and sets the QoD with minimal knowledge of the workload and transmission impairments. We show through extensive performance evaluation that the proposed approach is able to provide an actual lifetime that is within 2% of the specified lifetime. This result implies that the actual lifetime is very close to the specified lifetime even though the workload of the CS and the transmission impairments vary in an unpredictable manner during run-time.

In our future work we plan to investigate whether the approach presented in this paper can be applied to smaller nodes, i.e., the sensor nodes and the base stations. We also intend to incorporate techniques for periodically putting the CS into sleep to save energy.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 38, no. 4, pp. 393–422, 2002.

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

[3] T. Taleb, N. Kato, and Y. Nemoto, "Recent trends in IP/NGEO satellite communication systems: transport, routing, and mobility management concerns," *IEEE Wireless Communications*, vol. 12, no. 5, 2005.

[4] A. D. Panagopoulos, P.-D. M. Arapoglou, and P. G. Cottis, "Satellite communications at Ku, Ka, and V bands: Propagation impairments and mitigation techniques," *IEEE Communications Survey & Tutorials*, vol. 6, no. 3, 2004.

[5] Globalstar, "http://www.globalstar.com/."

[6] J. H. Lodge and M. Moher, *The Communications Handbook*. CRC Press, 1997, ch. Mobile Satellite Systems.

[7] L. J. Ippolito, *The Communications Handbook*. CRC Press, 1997, ch. Satellite Transmission Impairment.

[8] M. Amirijoo, S. H. Son, and J. Hansson, "QoD adaptation for achieving lifetime predictability of WSN nodes communicating over satellite links," Department of Computer and Information Science, Linköping University. www.ida.liu.se/∼meham/publications/satcomm.pdf, Tech. Rep., 2006.

[9] J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek, "Beyond average: toward sophisticated sensing with queries," in *Workshop on Information Processing In Sensor Networks (IPSN), 2003*, 2003.

[10] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "Discrete-time battery models for system-level low-power design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 5, 2001.

[11] V. Swaminathan and K. Chakrabarty, *Distributed Sensor Networks*. Chapman & Hall/CRC, 2005, ch. Operating System Power Management.

[12] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury, *Feedback Control of Computing Systems*. Wiley-IEEE Press, 2004.

[13] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Prentice-Hall, 1999.

[14] R. Szewczyk, J. Polastre, A. Mainwaring, , and D. Culler, "Lessons from a sensor network expedition," in *Proceedings of the First European Workshop on Sensor Networks (EWSN)*, 2004.