Adaptive Data Update Management in Sensor Networks

C. M. Krishna
ECE Department, University of Massachusetts at Amherst, Amherst, U.S.A.

Keywords: Sensor Networks, Energy-aware Computing, Energy Harvesting, Accuracy Balancing.

Abstract: Transmitting messages is by far the most energy-intensive thing that most sensors do. We consider the problem of a sensor which regularly senses some parameter in its operating environment. Based on the value it knows to have been estimated at the base station (or other central information collation station) for that parameter, the actual sensed value, its remaining energy levels, and other quantities such as the time-to-go in the mission (if limited) or the anticipated energy inflow (if energy harvesting is used), it decides whether that sensed value is worth transmitting. We present heuristics to make this decision and evaluate their performance.

1 INTRODUCTION

We address the following problem. We have a single sensor, which samples the value of some environmental parameter at periodic intervals. It then has to decide whether or not it is worth the expense of transmitting this information. We explore heuristics for doing so in two cases: *Case 1:* There is a fixed overall energy store, and *Case 2:* The network harvests energy from the operating environment.

The problem arises from the fact that broadcasting is by far the most energy-expensive thing that most sensor nodes can do. By comparison with the energy it takes to send a message, the act of sensing, i.e., of obtaining the information to be transmitted, takes almost negligible energy in most instances, especially since the node can sleep between sampling epochs.

The decision as to whether or not to transmit is based on the following considerations: (a) The value of the sensed parameter, (b) the value that would be estimated by the user if this parameter were to be suppressed at the sensor rather than being reported to the user, (c) the current energy state at the sensor, and (d) the residual mission lifetime (if this is limited). If energy reserves are replenished by harvesting from the operating environment, an additional factor is the anticipated near-term inflows of energy from such harvesting.

Our contributions in this paper are to develop lightweight adaptive algorithms to decide whether or not to transmit for the two cases. Our adaptive thresholding algorithms do not require any information about the operating environment. Rather than keep the sensor value within some specified bounds, their aim is to provide some means to do "accuracy balancing" over time, i.e., keep the minimum accuracy of the information at the base station at roughly the same level over the period of operation. Such an approach would be useful when the priority is to keep the network functioning over a given period under the assumption that the deleterious impact of inaccuracy is the same at any point in time. It would be useful when the cost of inaccuracy is a roughly linear function of the inaccuracy with which the sensor information is known.

Our MDT-based algorithm imposes a greater computational load, but that problem can be circumvented by carrying out most of these calculations offline and storing the appropriate action in a lookup table.

2 RELATED PRIOR WORK

2.1 Adaptive Data Aggregation and Reporting

A number of authors have reported work on adaptive data aggregation and reporting in sensor networks. We list here a representative sample of them.

One of the first contributions in this area was the approach of Goel and Imielinski (2001). They borrowed from the compression techniques used for MPEG video. The field of data generated from spatially distributed sensors can be visualized as intensity values in an image. Existing MPEG spatial and temporal compression techniques can then be used to exploit the spatio-temporal correlation that exists between neigbouring sensor values.

Deshpande, et al. (2004) follow the idea of learn-

M. Krishna C..
Adaptive Data Update Management in Sensor Networks.
DOI: 10.5220/0004034404760481
In *Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2012), pages 476-481
ISBN: 978-989-8565-21-1

ing the spatial and temporal correlations of sensor data. Santini and Romer (2006) use a Least-Mean-Square (LMS) adaptive algorithm for making predictions given a data stream. Han, et al. (2007) consider sensor transmissions that are either source-triggered or consumer-triggered (the consumers are the queries that come into the system, asking for parameter information). In all these cases, the sensor transmission is suppressed if it would not add sufficient value.

Ahmadi and Abdelzaher (2009) take reliability into consideration. In particular, they take into account the fact that wireless networks are often noisy and drop packets.

2.2 Energy Harvesting

Energy harvesting has been the focus of increasing interest. Surveys of energy harvesting techniques can be found in Chalasani and Conrad (2008) and Park, et al. (2007). Related power management techniques are studied in Kansal, et al. (2007) and Sharma, et al. (2010). These include migrating tasks to nodes depending on their energy levels. Task scheduling in energy harvesting real-time systems is considered in Moser, et al. (2007); their algorithms take into account both the prevailing energy and time constraints, rather than simply the task timing constraints.

3 MODELS

3.1 Environmental Model

The sensed environment has a behaviour as projected by the user. For example, if we have a sensor measuring outdoor temperature, there could be a formula that uses the current time-of-day and the last few reported observations to estimate the current temperature at the sensor. Since there are stochastic aspects to the sensed environment (if the environment were not stochastic there would be no need to use a sensor), the actual parameter value can vary from that estimated. The amount of variation from the estimated value obviously depends on the age of the information that is used to make the estimate: the temperature at 10:05 AM is likely to be estimated very accurately if the temperature at 10:00 AM was reported; by contrast, if the last temperature report was at 8:00 AM, the estimate is likely to be of poorer accuracy.

In our environmental model, we do *not* model the actual value of the sensed variable. Instead, we model the *difference*, δ, between the actual value and the projected value. It is in this difference that all the stochastic nature of the environment is captured. For example, if the sensed variable is treated as falling in some set of discrete quantities (which is the case in finite word-length machines even if the underlying sensed variable is continuous), we use the probability mass function (pmf), $\pi_\delta(\Delta)$, of the additional deviation, Δ, of the actual, from the projected (or modelled) value, arising from the passage of one sampling period. That is, if the state variable was last reported at sampling point n, the divergence from the projected value would be $\Delta_1 + \cdots + \Delta_m$ at sampling point $n+m$, where the Δ_i follow the pmf $\pi_\delta(\cdot)$.

It is important to recognise that our algorithm does not *require* a prior model of the environment to be available. If one such is available, it can be used to project into the future, the value of the next parameter sample. A simple case would be where, for instance, the Δ_i can be modeled as i.i.d. random variables. For example, in our numerical examples, we assume for concreteness that $\pi_\delta(\cdot)$ is a geometrically distributed random variable truncated at some maximum deviation: $\pi_\delta(i) = K\alpha^{|i|}$ if $-D_{max} \leq i \leq D_{max}$ and 0 otherwise, where α is a constant characterizing the random process, K is a normalization constant and D_{max} is some given truncation point.

However, if such a prior model is not available, we can simply use some extrapolation techniques based on recent observations to project what the next sample value will be.

Regardless of whether we use an environmental model, extrapolation, or a combination of the two, the sensor is able to replicate, without communication, the value that the base station would project (based on its prior transmissions) in the absence of a report of its current value. In other words, the sensor can calculate what the base station would estimate for the current value of the parameter based on the prior sensor reports. Since it has the actual measured value of the current parameter value, it knows the divergence between these two quantities.

3.2 Cost Measure

Our cost measure is the sum of the absolute errors as a result of not reporting the value of every sample that is measured. That is, let B_i be the broadcast indicator function,

$$B_i = \begin{cases} 1 & \text{if sampling point } i \text{ is reported} \\ 0 & \text{otherwise} \end{cases}$$

Define L_i as the last sampling point prior to i whose value was reported. Then, the aggregate cost incurred up to (and including) some sampling point i is given by

$$\Theta(i) = \sum_{j=1}^{i} (1 - B_i) \left[\sum_{k=1}^{L_i - 1} |\Delta_k| \right]$$

where the Δ_k are, as specified before.

This cost measure allows us to carry out approximate accuracy balancing. That is, for a given amount of available energy, we try to ensure that the accuracy of the best value at the sink node (base station) is roughly balanced over the lifetime of the system.

Our algorithms are by no means limited to this cost measure. They can, in fact, take any other cost function that may be more appropriate to the application.

4 FIXED ENERGY BUDGET AND FINITE MISSION LIFETIME

We start by considering the problem of a fixed energy budget (provided, for example, by a battery) and a specified finite mission lifetime. Generally, sensing consumes very little energy and so the number of broadcasts that the sensor still has energy for adequately characterizing the amount of energy available. (It is not difficult to relax this assumption by slightly inflating the energy required for transmission; we include it because it simplifies our description). We will therefore define the energy state at any moment as the number of broadcasts the node is still able to make.

We will assume a simple energy model in which battery leakage and fading are considered negligible. Also, we assume that the time between sampling points is sufficient for the battery to recover from the (heavy) power draw associated with transmission. If this is not the case, one can use battery models to capture this effect: in prior work, Krishna (2011) has shown how this can be done.

The baseline algorithm against which to compare our lightweight heuristics is to space the reporting instances as evenly as possible. More concretely, suppose the operating lifetime consists of N sampling points and we have enough energy only for r transmissions. If $N = \ell r + \rho$ for some integer $0 \leq \rho < r$, we have $r - \rho$ instances where we report every ℓ samples and ρ instances where we report every $\ell + 1$ samples.

4.1 Known Parameter Statistics

When the statistics of the parameter being sensed are known, we can analytically obtain an appropriate thresholding scheme. If the estimated value according to this model diverges from the actual value (as sensed) by more a specified threshold, the sensor transmits. Suppose the interval between successive samples is τ and the target lifetime of the network is T. Denote the threshold by θ and, as before, r as the number of transmissions for which we have energy. To ensure accuracy balancing, we use our knowledge of the parameter statistics to set θ so that the average inter-transmission duration is approximately T/r.

Example: Suppose our knowledge of the sensed parameter is such that the error (between the estimated and the last reported measurement) can be modeled as a Wiener process with zero drift and variance σ^2. We obtain θ as follows. Consider a Weiner process, whose initial value is 0. This represents the divergence from the sensed value at the last time a sample was taken and transmitted. Set up absorbing boundaries for this random walk at θ and $-\theta$. Calculate the expected first passage time from the initial state to one of the boundaries. This is given by $E[t_{firstPassage}] = \theta^2\sigma^{-2}$ ((Domine 1995)). Now, select θ so that the $E[t_{firstPassage}] = T/r$. This yields $\theta = \sigma\sqrt{T/r}$.

In general, let $g_n(\Delta_1, \Delta_2, \cdots, \Delta_n)$ be the joint density function of the deviations of the parameter measured $\tau, 2\tau, \cdots, n\tau$ seconds after the previous measurement. Denote the time between successive transmissions by t_s. Define $S_i = \Delta_1 + \cdots + \Delta_i$. Then, from our knowledge of the system statistics, we can calculate the following terms:

$$
\begin{aligned}
Prob(t_s > \tau) &= Prob(S_1 < \theta) \\
Prob(t_s > 2\tau) &= Prob(S_1 < \theta; S_2 < \theta) \\
&\vdots \\
Prob(t_s > n\tau) &= Prob(S_1 < \theta; \cdots; S_n < \theta)
\end{aligned}
$$

The mean time between transmissions is given by $\xi = \tau\sum_{n=1}^{\infty} P(t_s > n\tau)$. Now, set $\xi = T/r$ and solve (either numerically or analytically depending on the complexity of the expressions) this equation for θ.

4.2 Unknown Parameter Statistics

When the behaviour of the parameter being sensed is not known, we can proceed in one of two ways. First, we can start with some default setting of the threshold and use Bayesian or other methods to learn the dynamics of the parameter being sensed. This does, however, impose an overhead on the system, of calculating and maintaining posterior distributions representing current knowledge. In this section, we look at a much simpler method that bypasses the need to measure parameter dynamics.

In particular, we now present and evaluate a simple adaptive thresholding heuristic. As mentioned before, the sensor can assess the current value as estimated by the base station, in the absence of this current transmission. If the current error, defined as the absolute deviation of the actual current value (as known to the sensor) from this estimated value at the base station is not less than the threshold, the node transmits. The threshold is adaptively increased or decreased depending on the ratio of the number of transmissions that are still possible to the number of sampling epochs to go to the end of the mission. The pseudocode is provided in Figure 1.

```
actual_ratio=no_of_trans_still_possible/no_of_sampling_epochs_left
target_ratio= no_of_trans_initially_possible/total_no_of_sampling_epochs
if (actual_ratio ≥ 1)
        adaptive_threshold=1
else if (actual_ratio>target_ratio)
        adaptive_threshold++;
else if (actual_ratio<target_ratio)
        adaptive_threshold--;

if (current_error≥adaptive_threshold)
        transmit
```

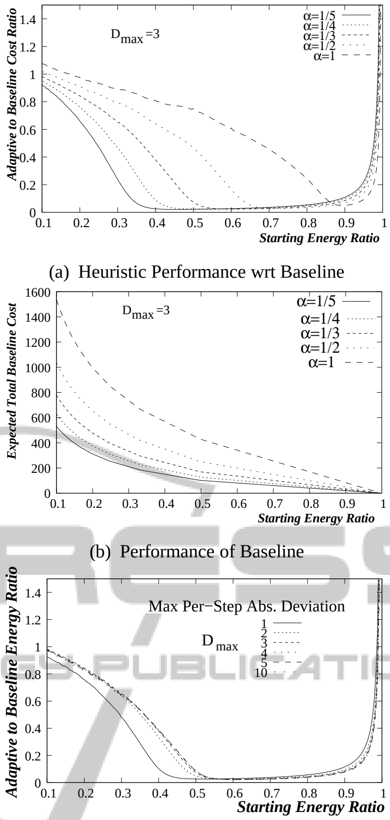
Figure 1: Pseudocode for Adaptive Thresholding Heuristic.

4.2.1 Simulation Results

As mentioned earlier, in our simulations for this paper, we assume (unless stated otherwise) that the deviation from the projected value is geometrically distributed, i.e., the pmf of the deviations is given by

$$\pi_\delta(i) = \begin{cases} K\alpha^{|i|} & \text{if } -D_{max} \leq i \leq D_{max} \\ 0 & \text{otherwise} \end{cases}$$

Figure 2 provides some performance results. The Starting Energy Ratio (SER) is the ratio of transmissions that are possible to the number of sampling epochs. For low SER, the adaptive thresholding algorithm offers no meaningful savings over the baseline algorithm; indeed, for some extremely low values, it may actually perform a little worse. The reason is that it takes a few adjustments for the threshold to settle down to an appropriate value; if no more than a handful of transmissions is possible, the system may spend most of them while the threshold is still adjusting significantly. For other regions, the adaptive thresholding algorithm significantly outperforms the baseline until SER approaches 1. At this point, we have so much energy that the baseline performs very well; the adaptive thresholding algorithm now performs *worse* than the baseline because it was initially much too parsimonious while still in its initial phase of adjusting for the correct threshold. In other words, when SER is very low, the heuristic loses out (in relative terms) because it was initially too profligate with transmissions; when SER approaches 1, because it was initially too stingy. This can obviously be dealt with by setting the initial value of the threshold based on our knowledge of how much energy is



(a) Heuristic Performance wrt Baseline

(b) Performance of Baseline

(c) Impact of D_{max}

Total number of Sampling Epochs = 500; initial threshold = 5

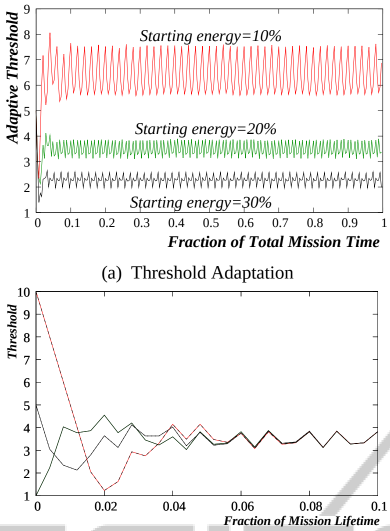
Figure 2: Heuristic Performance: Impact of α.

available. However, note that this is the effect of what is essentially a startup transient; its relative effect on the overall performance will therefore dwindle as the system lifetime goes up.

For smaller values of α, the chances are higher that the deviation will be small. Figures 2(a) and (b) show the deterioration that happens with increasing α. When $\alpha = 1$, the the variation is uniformly distributed from $-D_{max}$ to D_{max}. Note that when $SER = 1$, the baseline algorithm has enough energy to transmit each sample, so that the base station has all the samples and therefore zero cost.

Figure 2c shows the impact of the deviation amplitude, D_{max}, in each sampling epoch. As might be expected, when D_{max} is just 1, there is not much variation and the heuristic performs quite well. As D_{max} increases, the variation per step increases, and the heuristic deteriorates slightly. However, beyond $D_{max} = 2$, the relative performance of the heuristic is insensitive to the value of D_{max}. Henceforth, unless otherwise stated, all numerical results are provided for $D_{max} = 3$.

Figure 3 shows the way in which the threshold



(a) Threshold Adaptation

(b) Impact of Initial Threshold Value

Figure 3: Threshold Variation with Time.

varies with time. After an initial transient, depending on the initial value of the threshold (see Figure 3b), the threshold settles into a fairly narrow range. An obvious question from this figure is whether low-pass filtering the thresholds will have a positive effect on performance by damping down on the variations. Figure 4 indicates that any such gains will be minimal: here, we carry out low-pass filtering by using as the actual threshold the average of the past five threshold values.

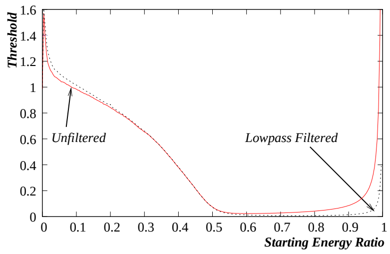


Figure 4: Impact of Low-Pass Filtering of Threshold.

5 ENERGY HARVESTING

If sensor networks must work indefinitely, they require some means to harvest energy from the operating environment. Various approaches have been studied for this. The most commonly suggested method for outdoor networks is to use solar cells feeding into a rechargeable battery or a supercapacitor. Other approaches include using wind energy and energy from vibrations, using a piezoelectric transducer.

Any system that harvests energy from the environment must be prepared to put up with the inherent variability in power inflow. For example, solar harvesting is obviously subject to the diurnal cycle; in addition, there is the incidence of clouds and dust. To smooth out these effects, we require an energy store that can be used to smooth out these variabilities. The size of the energy store is a design issue; if it is too small, the system will be highly vulnerable to variations in power inflow; if it is too large, it will be very expensive.

The sensing model is as described previously. The energy model is as follows. The sensor has a repository of energy which is replenished continuously by energy harvesting and depleted by message transmission. As before, to simplify our description, we assume that transmission is the dominant energy consumer; computation and sensing are negligible by comparison. Energy inflow is stochastic; the probabilistic laws governing it are assumed to be known to the user. At each sampling instant, the sensor determines whether or not its datum is worth transmitting based on the energy available to it and the error that would result at the base station from not transmitting.

We present here an adaptive thresholding algorithm and compare it against a baseline greedy algorithm. The parameter statistics are assumed to be unknown. We continue to study adaptive thresholding algorithms when the parameter statistics are unknown. Perhaps the simplest adaptive thresholding algorithm is to increment the threshold at a sampling point whenever the energy store is empty (defined as being too small to support even one data transmission), and to decrement it when the energy store is full. Such an approach does not require one to keep track of energy inflows or outflows: only the amount of energy stored at any given moment (which can easily be measured).

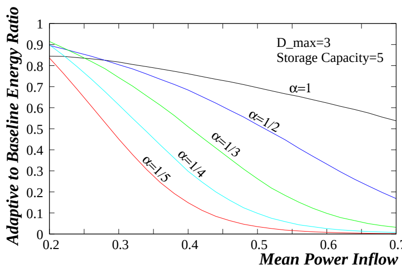


Figure 5: Performance of Thresholding Algorithm.

Figure 5 shows the performance of this algorithm relative to the baseline. The power inflow is in units of transmission energies per sampling interval; we assume in this simulation that the inflow is normally distributed with standard deviation equal to the mean, and conditioned on falling in the interval $[0, 2\mu]$ where

μ is the mean.

For very small power inflows, there is no real advantage over the baseline algorithm: the system is energy-starved. As power inflows increase, the performance improves markedly as compared to the baseline. The algorithm becomes more effective compared to the baseline as the deviations of the underlying sampled variable are more clustered around 0 (i.e., small values of α) and less effective when the deviations are uniformly distributed over $\pm D_{max}$ ($\alpha = 1$).

6 DISCUSSION

We have developed lightweight algorithms to keep the minimum accuracy at the base station as balanced as possible over the given period of operation. The aim is to pace the sensor transmissions appropriately given the energy constraints. We have considered two models: one in which there is a fixed energy budget and another in which there is an energy store (a rechargeable battery or a supercapacitor are the most convenient options) that is replenished by means of energy harvesting. Such an approach is likely to be useful in applications involving long-term environmental monitoring.

ACKNOWLEDGEMENTS

This research was partially supported by the National Science Foundation under grant CNS-0931035. The author thanks the reviewers for their helpful comments.

REFERENCES

H. Ahmadi and T. Abdelzaher, (2009) "An Adaptive-Reliability Cyber-Physical Transport Protocol for Spatio-Temporal Data," *IEEE Real-Time Systems Symposium*.

S. Chalasani and J. M. Conrad, (2008). "A Survey of Energy Harvesting Sources for Embedded Systems," *IEEE SoutheastCon*, pp. 442–447.

M. H. DeGroot, (2004). *Optimal Statistical Decisions*, Wiley.

A. Deshpande, C. Guestrin, S.R. Madden, J.M. Hellerstein, and W. Hong, (2004). "Model-Driven Data Acquisition in Sensor Networks," *30th VLDB Conference*.

M. Domine, (1995). "Moments of the First Passage Time of a Wiener Process with Drift Between Two Elastic Barriers," *Journal of Applied Probability*, Vol. 32, No. 4, pp. 1007–1013.

S. Goel and T. Imielinski, (2001). "Prediction-Based Monitoring in Sensor Networks," *ACM Computer Communications Review*, Vol. 31, No. 5.

Q. Han, S. Mehrotra, and N. Venkatasubramanian, (2007). "Application-aware Integration of Data Collection and Power Management in Wireless Sensor Networks," *Journal of Parallel and Distributed Computing*, Vol. 67, pp. 992–1006.

A. Kansal, J. Hsu, S. Zahedi, and M.B. Srivastava, (2007). "Power Management in Energy Harvesting Sensor Networks," *ACM Transactions on Embedded Computing Systems*, Vol. 6, No. 4, September 2007, Article 32.

C. M. Krishna, (2011). "Managing Battery and Supercapacitor Resources for Real-Time Sporadic Workloads," *IEEE Embedded Systems Letters*, Vol. 3, No. 1, pp. 32–36.

C. Moser, D. Brunelli, L. Thiele, and L. Benini, (2007). "Real-Time Scheduling for Energy Harvesting Sensor Nodes," *Real Time Systems*, Vol. 37, pp. 233–260.

G. Park, C. R. Farrar, M. D. Todd, W. Hodgkiss, and T. Rosing, (2007). "Energy Harvesting for Structural Health Monitoring Sensor Networks," *Los Alamos National Labs*, LA-14314-M5, February 2007.

S. M. Ross, (1970). *Applied Probability Models with Optimization Applications*, New York: Dover Publications, 1992.

S. Santini and K. Romer, (2006). "An Adaptive Strategy for Quality-Based Data Reduction in Wireless Sensor Networks," *International Conference on Networked Sensing Systems, pp. 29–36.*

M. A. Sharaf, J. Beaver, A. Labrinidis, and P.K. Chrysanthis, (2004). "Balancing Energy Efficiency and Quality of Aggregate Data in Sensor Networks," VLDB Journal, *Vol. 13, No. 4, December 2004, pp. 384–403.*

V. Sharma, U. Mukherjee, and V. Joseph, (2010). "Optimal Energy Management Policies for Energy Harvesting Sensor Nodes," IEEE Transactions on Wireless Communications, *Vol. 9, No. 4, p. 1326.*