

Event-Driven Field Estimation for Wireless Sensor Networks

Daniel de O. Cunha^{1,2}, Otto Carlos M. B. Duarte¹, and Guy Pujolle² *

¹Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro - UFRJ
Rio de Janeiro, Brazil

²Laboratoire d'Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie - Paris VI
Paris, France

Summary. This paper introduces and analyzes a field estimation scheme for wireless sensor networks. Our scheme imitates the response of living beings to the surrounding events. The sensors define their periphery of attention based on their own readings. Readings differing from the expected behavior are considered events of interest and trigger the data transmission to the sink. The presented scheme is evaluated with real-site-collected data and the tradeoff between the amount of data sent to the sink and the reconstruction error is analyzed. Results show that significant reduction in the data transmission and, as a consequence, in the energy consumption of the network is achievable while keeping low the average reconstruction error.

1 Introduction

Recent advances in MEMS technology and wireless communications made possible to embed sensing, processing, and communication capabilities in low-cost sensor nodes. As a consequence, the use of hundreds or thousands of nodes organized in a network [1] becomes an alternative to conventional sensing techniques. The resulting sensor network has the advantage of being closer to the sensed process, being able to acquire more detailed data.

Field estimation is an important application of wireless sensor networks. This type of application deploys sensor nodes in a specific region to remotely sense space-temporally variable processes, such as temperature or UV exposure. It is a continuous data-delivery application [2] and the simplest approach of such a system is based on deploying the sensors in the area of interest and requiring all the nodes to transmit data to the sink at a prespecified rate. The quality of the estimation depends of the spatial and temporal frequencies of sampling.

* This work has been supported by CNPq, CAPES, FAPERJ, FINEP, RNP and FUNTTEL.

Please use the following format when citing this chapter:

de O. Cunha, D., Duarte, O.C.M.B., Pujolle, G., 2006, in IFIP International Federation for Information Processing, Volume 211, ed. Pujolle, G., Mobile and Wireless Communication Networks, (Boston: Springer), pp. 89–98.

These frequencies must be high enough to avoid the aliasing problem [3]. Nevertheless, higher frequencies generate a larger amount of data to transmit. As the data transmission is the most consuming task of a sensor node [4], it is interesting to trade some data transmission for local processing.

The spatial sampling frequency is related to the number and position of the nodes. A higher spatial sampling frequency results in a larger number of nodes transmitting data to the sink. Without any prior analysis, it is necessary to deploy the nodes in such a way that the entire field is sampled with the highest required spatial frequency in the field. This required frequency is a function of the process and the field. As a consequence, the typical sensing field is not uniform. It is composed by areas where the sensed process varies smoothly and areas where the process varies sharply from one position to other. Usually, sharp-variation areas are *borders* between different smooth-variation areas. Some works aim at identifying such smooth-variation areas and deactivate some of the nodes in these regions [5, 6, 7]. The main idea is to reduce the amount of transmitted data by avoiding the transmission of redundant information. Nevertheless, these approaches are unable to save energy at the borders, which remain with a high spatial density of active nodes. The temporal sampling frequency is related to the period between consecutive samples. The required temporal sampling frequency is dictated by the sensed process and reducing this frequency may result in the lost of important information about the process. Typical solutions to reduce transmission time of a node to send the amount of data is the codification of the resulting temporal series [8, 9] in a more efficient way.

We have propose an event-driven field estimation scheme for wireless sensor networks [10]. Differing from the above discussed approaches, our scheme reduces the amount of transmitted data by sending only part of the samples. The assumption behind our scheme is that although we have to sample the process with its required temporal frequency to avoid losing important data, not all the samples will bring interesting information. Hence, the proposed scheme exploits specific features of the monitored processes in order to reduce the amount of data transmitted to the sink. Each sensor node collects the samples and decides to only send to the ones considered an event of interest to itself. This mimics an event-driven system over a continuous-data transmission application.

In this paper we evaluate the proposed scheme with real-site-collected data. Furthermore, we take a new metric, the average reconstruction error, into account and analyze the tradeoff between the average reconstruction error and the total amount of data transmitted.

The rest of the paper is structured in the following way. In Section 2, we introduce the proposed field-estimation scheme. Then, the simulations are then presented in Section 3, showing the efficiency of the scheme. Finally, conclusions and future research work are discussed in Section 4.

2 The Field Estimation Scheme

Our field-estimation scheme is bio-inspired, based on how living beings respond to the surrounding events. People and animals are continually receiving *stimuli*; however, it is impossible to handle consciously all these *stimuli*. The organisms develop the notion of periphery and center of attention [11]. While the periphery is handled in a sub-conscious manner, the center of attention is the event consciously treated. Generally, an event migrates from the periphery to the center of attention when it differs much from the periphery as a whole.

In practice, the sensor identifies a recurrent pattern in the process and defines an expected behavior, or the periphery of attention, for the next readings. Based on this expected behavior, the sensor decides whether a sample is important or not. If it decides that the sample will aggregate useful information, it sends the sample to the sink. These samples sent to ensure the quality of the estimation are called *refining samples*. Otherwise, the sample is still used to calculate the expected behavior of the process in the future, but is not transmitted. As we discussed earlier, the data transmission is the most energy consuming task in a sensor network and the reduction in the number of transmitted samples impacts significantly the energy consumption of the network.

Originally, we proposed two different ways for implementing the proposed scheme, a sample-bounded algorithm and an error-bounded algorithm [10]. The sample bounded algorithm limits the maximum number of samples transmitted to the number of samples collected by the sensor, no matter how badly the scheme is configured. On the other hand, the error-bounded algorithm ensures that no event of interest will be lost by limiting the maximum reconstruction error at the sink. However, this algorithm must be well tuned or may result in the transmission of more samples than the originally collected by the sensor. As we want to ensure a correct estimation of process all the time, we will only consider the error-bounded algorithm in this paper. The first step is to determine the periodicity of the recurrent pattern. The algorithm must run once at the end of each period, after the collection of all samples. The physical process used in our analysis is the temperature, which clearly has a daily periodicity. It presents a regular behavior where the temperature is low in the morning and rises near noon. The temperature falls in the afternoon and reaches low values again at night. Temperature also presents annual periodicity, but our analysis is based only on daily periodicity. Thus, the sensor nodes must identify a daily-expected behavior, updated every day. The decision about refining samples is made based on this behavior. Additionally, the sink must have enough information to successfully reconstruct the remotely sensed process. Therefore, the sink needs to know an expected behavior, which is assumed to occur when no refining samples are received. The node must periodically send an updated expected behavior to the sink. The sink then assumes the process behaves exactly like the most recent expected-behavior vector, whenever no refining samples are received. These refining samples replace part of the expected behavior as informed by the sensor. The sensor node must decide on sending or not refining

samples based on the last expected behavior vector sent to the sink. This procedure maintains the consistency between the measured and the reconstructed information. Therefore, the sensor verifies whether the measured value differs above a certain threshold from the corresponding sample of the last expected-behavior vector sent. If this difference is higher than the configured threshold, the sensor sends the refining sample to the sink.

Assuming that the daily periodicity is already known, Fig. 1 shows the daily procedure, where DB_j is the vector with the expected behavior during day j , D_i the vector with the measurements of day i , $last_update$ is the vector with the most recent expected behavior sent to the sink, and the notation $X(k)$ is used to represent the k -th element of vector X .

```

DB i = α D i + (1-α) DB i-1
If update time
  last_update = DB i
  Send last_update
For all k samples in D i do
  If |D i(k) - last_update(k)| > |last_update(k)| * configured_error
    Send D i(k)

```

Fig. 1. Daily procedure.

As we can see in Fig. 1, the algorithm has three important parameters: the α factor, the *update* specification, and the *configured_error* limitation. The α factor weights the importance of the recent samples in the generation of the expected-behavior vector. The *update* parameter specifies the interval between the transmission of expected-behavior vectors to the sink. The *configured_error* is used to limit the reconstruction error at the sink. The performed simulations are detailed in the next section.

3 Simulations

We analyze the proposed scheme by simulating the local processing of one sensor node. The field-estimation scheme is evaluated considering two distinct metrics: the reduction in the total number of samples sent to the sink and the average reconstruction error at the sink. We use the percentage samples sent as an index to estimate the energy conservation. This preserves the generalization of the results by avoiding specific-MAC-layer biases. The average reconstruction error is used to evaluate how well the scheme performs the field estimation and is calculated as

$$AE = \frac{\sum_{i=1}^N \frac{sample(i)' - sample(i)}{sample(i)} \cdot 100}{N}, \quad (1)$$

where $sample(i)$ is the value sensed by the sensor for a given sample, $sample(i)'$ is the value for this sample after the reconstruction at the sink, and N is the total number of samples collected.

The simulations are performed with different configurations of the scheme. These configurations results from the variation of the three parameters highlighted in Section 2: the α factor, the *update* specification, and the *configured.error* limitation. Furthermore, the scheme is evaluated with real-site-collected data. We analyze the performance of the scheme in a temperature monitoring application based on the measurements of a Brazilian meteorological station. The next section details the data preprocessing.

3.1 Data Treatment

The field-estimation scheme is analyzed based on data available at the web page of the Department of Basic Sciences of the Universidade de São Paulo, Brazil [12]. The web-site maintains a history of meteorological data from the last eight years. The temperature measurements used in this paper presents the evolution of the temperature at an interval of 15 minutes, which results in 96 samples per day. We performed a preprocessing of the data to avoid the use of corrupted data. A small part of the daily files skipped one or two temperature measurements. A few files had three or more measurements missing. In the cases where only one measurement was missing, we replaced the missing measure by the linear interpolation between the preceding measurement and the measurement immediately after the missing measurement. Files with more than one measure missing were discarded.

After the elimination of corrupted data, the measurements of each day were arranged in a single vector with 96 elements. These daily-measurements vectors were concatenated to form a large vector with all the measurements available from the last eight years. In the cases where the measurements of a day were discarded due to corruption, we just skipped that day and concatenated the day before the corrupted data has appeared and the day after.

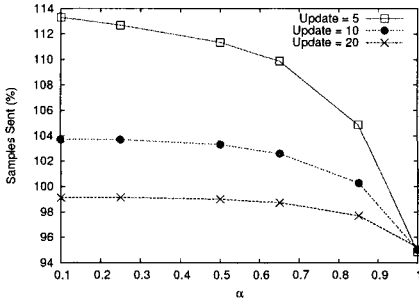
This data processing resulted in one single vector with the information of 2,880 days chronologically ordered. Thus, the simulations were performed with a data vector with 276,480 elements.

3.2 Results

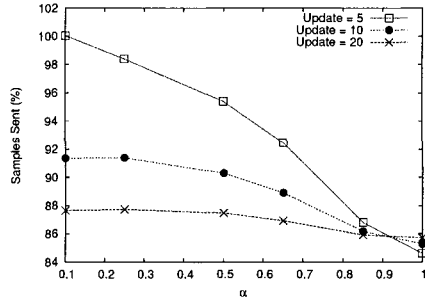
The simulations are based on the data vector generated as described in Section 3.1, which is used to represent the readings of a sensor. The proposed scheme is applied to this data set and the fraction of the total samples that must be actually sent to the sink and the average reconstruction error (Eq. 1) are obtained. The smaller the fraction of samples sent is, the better is the proposed scheme efficiency. Furthermore, a small average reconstruction error means a better estimation of the sensed process. The simulation assumes the daily (96 samples) periodicity has already been defined. The identification of the

period of the regular behavior of the process is independent of the parameters used to configure the scheme and only depends on the data set used.

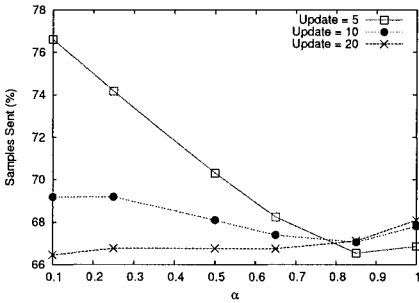
The three parameters previously discussed are varied during the simulation in order to better understand their effects. In all simulations, the update frequency is one expected-behavior vector sent at each *update* days. Thus, higher values of *update* means lower update frequencies. The maximum tolerated sample error is equal to the parameter *configured_error* times the expected behavior of the specific hour. The α factor is bounded to 1.



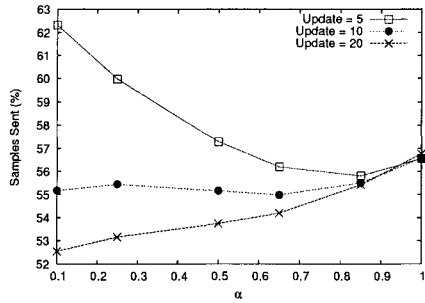
(a) Maximum error 1%.



(b) Maximum error 3%.



(c) Maximum error 7%.



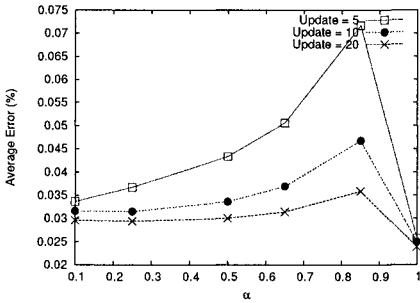
(d) Maximum error 10%.

Fig. 2. Amount of data sent.

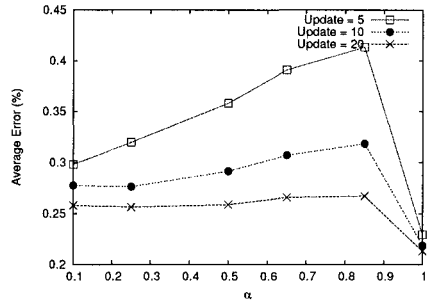
Fig. 2 shows the percentage of samples sent for different tolerated reconstruction errors. As we can see in Fig. 2(a), for very low tolerated errors, the scheme does not reduce much the percentage of samples sent. Moreover, it is possible to see that a bad tuning of the scheme can result in the transmission of more data than the original samples. Nevertheless, as the tolerated error is

relaxed, the scheme can reduce significantly the percentage of samples sent. For a maximum tolerated error of 3% (Fig. 2(b)) it is possible to reduce the amount of data in 15%. A larger tolerated error, such as shown in Fig. 2(d), reduces the amount of data in almost 50%. It is worth noting that a large tolerated error does not necessarily mean a poor estimation, as we will show later while analyzing the average reconstruction error.

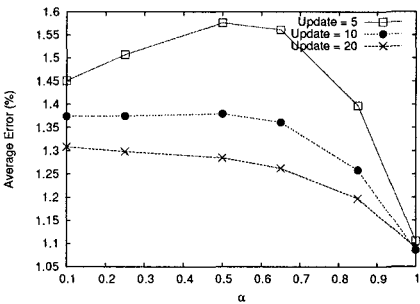
Analyzing Fig. 2, we notice that the configuration of the α and *update* parameters affects the results differently as the maximum tolerated error varies. For very small maximum tolerated errors (Figs. 2(a) and 2(b)), the scheme sends less samples when α is close to 1 and the parameter *update* is low. As we increase the maximum tolerated error, better results are achieved with a low α and a high *update*.



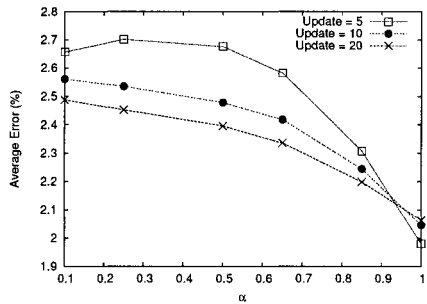
(a) Maximum error 1%.



(b) Maximum error 3%.



(c) Maximum error 7%.



(d) Maximum error 10%.

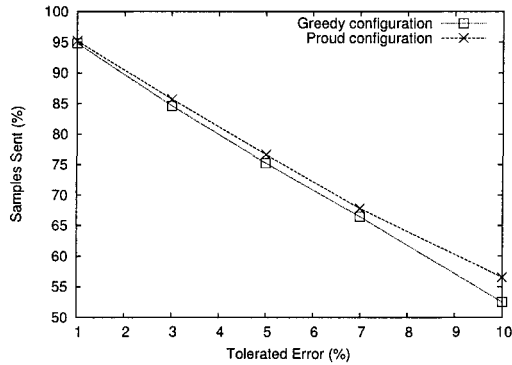
Fig. 3. Average reconstruction error.

Although a higher maximum error allows the transmission of less samples, this maximum error must be controlled in order to maintain the average reconstruction error satisfactory. Fig. 3 shows the average reconstruction error for different tolerated reconstruction errors. As can be seen, the average reconstruction error is significantly smaller than the maximum tolerated error. This means that we have a certain flexibility to define the maximum tolerated error for the scheme. Moreover, the average error presents some interesting behaviors. First of all, we see that for very small maximum tolerated errors, (Fig. 3(a)) the average error grows rapidly with the increase in α , until α reaches 1, when the average error falls sharply. It occurs because an α equals to 1 results in sending the exact readings of the update day as the expected-behavior vector for the next days. As the tolerated error grows, the increase of the average error according to α gets smoother (Fig. 3(b)) and in certain cases the error is reduced with the increase of α (Fig. 3(d)).

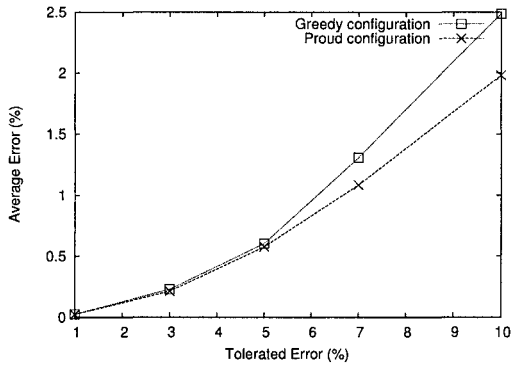
Fig. 3 also shows that, as it happened for the fraction of samples sent, the α and *update* configurations affects differently the average error as the maximum tolerated error grows. When the maximum tolerated error is low, the lowest average error is achieved with a higher *update* (Figs. 3(a) and 3(b)). As the maximum tolerated error grows, smaller values of *update* achieve better results (Figs. 2(c) and 2(d)).

The results shown in Figs. 2 and 3, highlight the importance of a correct configuration of the scheme in order to achieve the best possible results. Moreover, according to the metric we decide to optimize, we can achieve very different results. One possible configuration is to decide to always transmit as few samples as possible for every configuration of the maximum tolerated error. We will call this the *Greedy* configuration, and it could be implemented using α equals to 1 and a low *update* value, whenever the maximum tolerated error is low. When the maximum tolerated error grows we adopt a low α and a high *update* value. Another option, named *Proud* configuration, is to always try to minimize the average reconstruction error. It can be implemented using α always equal to 1 and assigning high values for the *update* parameter, when the maximum tolerated error is low. For higher maximum tolerated errors the *update* parameter must be low. Fig. 4 shows the results obtained for these two configurations as a function of the tolerated error.

Fig. 4(a) shows that for both configurations the number of samples sent decreases almost linearly with the increase in the maximum tolerated error. The difference in the results increase as the maximum tolerated error increases, but for the error range analyzed this difference is always lower than 10%. As it can be seen in Fig. 4(b), the average reconstruction error grows faster with the increase of the tolerated reconstruction error. When we use a larger tolerated reconstruction error, the *Proud* configuration results in a reconstruction error 20% lower than the error resulting from the use of the *Greedy* configuration, in the expense of sending a few more samples. This suggests that it may be interesting to balance the two metrics in the configuration of the field-estimation scheme.



(a) Amount of samples sent.



(b) Average Reconstruction Error.

Fig. 4. Results from the ideal configurations.

4 Conclusion

This paper introduces and analyzes an event-driven field-estimation scheme. The scheme exploits the fact that not all the collected samples result in useful information. Therefore, we reduce the number of samples sent to the sink and, as a consequence, the energy consumption in the network.

The field-estimation scheme is analyzed based on real-site-collected data from a meteorological station. We take two metrics into account while evaluating the scheme: the amount of data sent to the sink and the average reconstruction error. The results show that the configuration of the scheme severely impacts the results, suggesting an automated configuration procedure. Moreover, the

scheme configuration must take into account the two metrics simultaneously. The results show that a significant increase in the estimation quality can be achieved in expense of a slightly smaller gain. A configuration aiming to optimize the average reconstruction error results in much smaller errors, achieving a gain a little smaller than the gain achieved by a Greedy configuration.

In the future, we intend to analyze the impact of the network losses on the results and to develop an adaptive configuration mechanism to ensure the achievement of the best possible results.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.
2. S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *ACM Mobile Computing and Communications Review (MC2R)*, 2002.
3. A. Kumar, P. Ishwar, and K. Ramchandran, "On distributed sampling of smooth non-bandlimited fields," in *Information Processing In Sensor Networks - IPSN'04*, apr 2004, pp. 89–98.
4. G. P. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, may 2000.
5. R. Willett, A. Martin, and R. Nowak, "Backcasting: adaptive sampling for sensor networks," in *Information Processing In Sensor Networks - IPSN'04*, apr 2004, pp. 124 – 133.
6. M. Rahimi, R. Pon, W. J. Kaiser, G. S. Sukhatme, D. Estrin, and M. Sirivastava, "Adaptive sampling for environmental robotics," in *IEEE International Conference on Robotics & Automation*, apr 2004, pp. 3537–3544.
7. M. A. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G. Sukhatme, W. Kaiser, M. Hansen, G. J. Pottie, M. Srivastava, and D. Estrin, "Towards event-aware adaptive sampling using static and mobile nodes," Center for Embedded Networked Sensing - CENS, Tech. Rep. 38, 2004.
8. I. Lazaridis and S. Mehrotra, "Capturing sensor-generated time series with quality guarantees," in *International Conference on Data Engineering (ICDE'03)*, mar 2003.
9. H. Chen, J. Li, and P. Mohapatra, "Race: Time series compression with rate adaptivity and error bound for sensor networks," in *IEEE International Conference on Mobile Ad-hoc and Sensor Systems - MASS 2004*, oct 2004.
10. D. O. Cunha, R. P. Laufer, I. M. Moraes, M. D. D. Bicudo, P. B. Velloso, and O. C. M. B. Duarte, "A bio-inspired field estimation scheme for wireless sensor networks," *Annals of Telecommunications*, vol. 60, no. 7-8, 2005.
11. M. Weiser and J. S. Brown, "The coming age of calm technology," in *Beyond calculation: the next fifty years*. Copernicus, 1997, pp. 75 – 85.
12. Universidade de São Paulo, *Departamento de Ciências Exatas*, LCE - ESALQ - USP, 2005, <http://www.lce.esalq.usp.br/indexn.html> - visited in Feb. 2005.