Cinzia Cappiello, Fabio A. Schreiber

# Experiments and analysis of quality and Energy-aware data aggregation approaches in WSNs

QDB 2012

10th International Workshop on Quality in Databases
In conjunction with VLDB 2012,

August 27th, 2012,

Instanbul, Turkey

pp. 1-8

http://www.purdue.edu/discoverypark/cyber/qdb2012/papers/7data%20aggregation.pdf

# Experiments and analysis of quality- and energy-aware data aggregation approaches in WSNs

Cinzia Cappiello
Politecnico di Milano, Dipartimento di Elettronica e Informazione
Piazza Leonardo da Vinci 32
20133 Milano
cappiell@elet.polimi.it

Fabio A. Schreiber
Politecnico di Milano, Dipartimento di Elettronica e Informazione
Piazza Leonardo da Vinci 32
20133 Milano
schreibe@elet.polimi.it

## ABSTRACT

A wireless sensor network consists of autonomous devices able to collect various data from the area that surrounds them. However, the resources associated with sensors are limited and, thus, in order to guarantee a longer life of all the network components, it is necessary to adopt energy-savings methods. This paper, considering that the transmission phase is the main cause of energy dissipation, presents an approach aimed to save energy by capturing and aggregating signals instead of sending them in raw form. Anyway, aggregation should not imply the loss of useful data. For this reason, information about possible outliers is preserved and the aggregated values have to satisfy data quality (i.e., accuracy, precision, and timeliness) requirements. In order to show the correctness and validity of the proposed method, it has been tested on a real case study and its performance has been compared with two other consolidated approaches.

## 1. INTRODUCTION

Sensors are getting more complex and they must be treated as equal partners in future distributed database systems as they can store, manipulate and communicate information. In fact, each sensor produces a continuous stream of data which flows from the sensor node itself to the consumer node - usually one or more base stations - possibly by multi-hop transmission.

However, a sensor node is not a long-life product: its small size implies limitations on the associated resources. In particular, it is necessary to deal with two main technological issues, such as memory and power amount. On the one hand, memory can only store few data for a limited time span and it is necessary to periodically transfer data to a larger storage device. On the other hand, the life of on-board batteries is limited and transmission is the most power consuming function. These constraints conflict with each other since the need of transmitting data in order to free the sensor local memory requires frequent transmissions of long data sequences which are highly power consuming. Thus, we need to compact incoming data in order to optimize the local storage and transmit only few value-added data to the parent nodes. Data amount reduction is one effective method to use limited resources of WSNs. Anyway, it is also necessary to consider that these methods negatively impact on the quality of transmistted information. In fact, data aggregation implies the loss of values and thus the loss of precision. This could be a relevant issue especially in contexts in which also small signal variations are important to understand the phenomenon. A good aggregation method should be able to deal with the energy saving/data quality trade-off. It is important to contemporarily satisfy data quality requirements and to maintain the error introduced in reducing data below a specified threshold.

In this paper, we illustrate an adaptive data aggregation algorithm that tries to satisfy both data quality and energy saving requirements. The initial description of the algorithm has been introduced in [3]. In this paper, we perform a step forward by improving the algorithm performance and providing: (i) a validation of the algorithm by using a real case study; (ii) a comparison with other similar algorithms. The paper is organized as follows. Section 2 describes the main contributions about data aggregation in sensor networks in order to show the novel aspects of our approach. Section 3 introduces the main data quality criteria to be considered for data aggregation. Section 4.1 defines the context in which our algorithm is proposed. Details about the algorithm and its performance are provided in Section 4.2, Section 5 discusses the experimental testbed that has been setup for the validation of the approach. Finally, Section 6 discusses some conclusive remarks.

## 2. RELATED WORK

*Data compression*, and in particular data reduction, is a well-established research field, but sensor networks present a context in which new design issues have to be addressed [17]. In fact, the small code and data memories, and the primary focus on energy, call for new approaches [11]. In this paper we focus on a specific data reduction technique: *data aggregation*. Data aggregation is the process in which information is gathered and expressed in a summary form. In the literature, a large variety of aggregation algorithms have been proposed. Most of existing data aggregation algorithms are, however, not feasible for WSNs owning to their size and complexity. Within the WSNs community, there

are several contributions (e.g., [6]), where authors address the analysis of high spatial correlation in data from fixed sensors in dense networks. Here, the context is specific and the addressed problems have particular characteristics and criticisms. Our approach aims to handle heterogeneous data sources and to aggregate any data stream characterized by various and unexpected trends.

Data reduction has been mostly studied to enable in-network processing. In-network processing is the general term used for techniques that process data on a node or group of nodes before forwarding it to the user. The goal of in-network processing of data streams is to select and give priority to reporting the most relevant data gathered [10]. Here, spatial and temporal aggregation techniques are widely used. *Spatial aggregation* deals with data redundancy in a same physical area. Contributions in this field propose models to discover similar values and to aggregate them by using specific functions [20]. In [8], similar values compose the base signal used to forecast and evaluate the collected data. In spatial compression analysis, the contributions about research on sensors' communication paradigms are extremely relevant (e.g., [13]).

*Temporal aggregation* is suitable for all the contexts in which the main goal is to detect data changes over time. In this scenario, one of the main contribution is a lightweight linear approach [4] [21]. The linear aggregation algorithm provides a good balance between maximizing compression and minimizing processing complexity for each node. The approach just considers different measures taken at different time instants. Each value is compared with the previous one and it is transmitted only if the measure is significantly different. This algorithm is suitable for all the contexts in which phenomena are quite stable and data are characterized by linear trends. In fact, in case of unstable data, the approach would support the communication of all the measured values. The algorithm proposed in Section 4 aims to detect data changes over time as the linear aggregation algorithms, but it maximizes the compression ratio even when the data trend changes frequently. Therefore, it is not so strictly dependent on the phenomenon characterization.

The proposed aggregation algorithm is also based on the concept of time series as [5] [25] [18]. In [5], the authors propose to perform on-line regression analysis over time series on data streams. Autoregressive models built on each sensors are instead used in [25] to forecast time series and approximate the value of sensors readings. Lazaridis and Mehrotra [18] also propose to fit models to time series, but they try to improve system performance, rather than doing regression analysis. We refer to this work, since our model is focused on both quality requirements satisfaction and energy saving. Our model divides the time series in windows and introduces the concept of *continuity interval* in order to detect permanent data trend changes. Furthermore, our model deals with all types of trends and not only with a limited set as the algorithm proposed in [18]. We also propose an adaptive mechanism to change the measure frequency in case of very irregular trends, so increasing the system bandwidth. The adaptation in data stream management is driven by data quality requirements. Several contributions in the literature adopt a similar approach by considering a different set of dimensions; for example, [24] monitors the processing delay to assure data freshness. The total response time is also checked in [12] to optimize the overall QoS performance according to the network condition and work load on-line. Furthermore, quality has been often analysed together with costs; typically, for WSNs the most important component of cost typically is the energy consumed in providing the requested data. In turn, this is dominated by the energy required to transport messages through the sensor field. This cost versus quality (e.g., data accuracy) trade-off has been thoroughly analysed in in-network aggregation research area [26][1][23]. Only some contributions deal with data quality analysis on a single sensor. Most of them use data quality dimensions to clean data streams [16], while only a few papers consider quality as an important factor in data aggregation. A relevant framework is proposed in [15] in which the precision dimension is used to filter data by using the Kalman filter. In this way they build a flexible system that is able to automatically adapt the reference model to the real-world signal. Authors discard outliers and try to detect the value trend. As for the adaptability to a variety of different types of signals, the contribution in [15] aims to achieve a similar level of flexibility as proposed in this paper. Anyway, differently from [15][16], in our approach, outliers are important elements of the data stream to consider and store since scientific researchers deem they can be very useful in studying and interpreting natural phenomena. In addition, we consider other data quality dimensions (i.e., accuracy and timeliness) to improve the efficiency of the algorithm and further reduce the need for transmitting data to the base station.

# 3. DATA AGGREGATION: QUALITY REQUIREMENTS

The quality of the data provided by a sensing application is a combination of accuracy and delay [26]. Our approach is based on the idea that data quality measures can be used by the base station as the main driver for the selection of the most relevant and thus useful sensors' data and also for the evaluation of the reliability of the received data. In fact, on the one hand the base station could define quality requirements to influence the behaviour of the sensors involved in the network: each sensor node just collects and sends data in order to satisfy all the quality requests. On the other hand, the base station could evaluate the trustworthiness of the different sensors by assessing the correctness and updateness of the incoming data.

The most relevant data quality dimensions in the WSN scenario are *Accuracy*, *Precision* and *Timeliness*. *Accuracy* is usually defined as the degree of conformity of a measured or computed quantity to its actual (true) value. Accuracy is related to *precision* that is the degree to which repeated measurements show the same or similar results [14]. The impact of these two dimensions on data stream management is discussed in Section 4.1. *Timeliness* is defined as the property of information to arrive early or at the right time. Timeliness is usually measured as a function of two elementary variables: currency and volatility [9][19]:

$$Timeliness = max(1 - Currency/Volatility; 0)^s$$

where the exponent $s$ is a parameter necessary to control the sensitivity of timeliness to the currency-volatility ratio. In the analysed context, *currency* can be defined as the interval from the time the value was sampled to the time instant at which data are sent to the base station. *Volatility* is a static information that indicates the amount of time units

(e.g., seconds) during which data remain valid. Volatility is usually associated with the type of phenomena that the system has to monitor and depends on the change frequency. Timeliness constraints are one of the main drivers for data processing and transmission. In fact, timeliness constraints could limit the time validity of sensors' data and can force the transmission of data before the scheduled time instant. When users submit queries, they have to define their quality requirements. As an example, the PERLA language [22] allows a conditional execution of operations on the basis of quality parameters.

Moreover, it is necessary to consider that the design of a sensor network and of the related algorithms are tailored for a particular type of applications and thus for the type of expected signal. WSNs can be used for data collection purposes in situations such as environmental monitoring, habitat monitoring surveillance, equipment diagnostics, disaster management, and emergency response [7]. The idea behind the algorithm presented in this paper is to use data quality dimensions to design an adaptive aggregation algorithm able to work effectively for any type of signal.

# 4. THE DATA AGGREGATION ALGORITHM

## 4.1 The system and the data structure

The input data stream can be seen as a continuous flow of real-time data tuples of the form <*sensor-id, time stamp, value*> coming to the sensor's input buffer. As in many real-time systems, we can suppose that the Input Buffer (IB) is actually split into two separate storage areas (i.e., $IB_1$ and $IB_2$). Data are fed to $IB_1$ until either it is full or timeliness requirements force data processing, then input is switched to $IB_2$ while data in $IB_1$ are transferred to the compression engine and then to the output buffer. The switching process is repeated and data are processed from $IB_2$.

In this way, the potentially infinite data stream is reduced, with a windowing approach, to a sequence of finite time-ordered data sets on each of which the compression algorithm can work. In our approach, the main window can be further partitioned into smaller sub-windows (see Figure 1) in which values that are considered *similar* can be aggregated by computing their average.

In particular, each sensor has a sampling period that defines the time instant $t_i$ in which data are acquired. Considering these time instants, we define a value series $V =< v[1], v[2], ...v[n] >$ as a collection of values observed in subsequent $n$ time instants. The maximum number of measure points $N$ coincides with the cardinality of data in the input buffer. Sub-windows are characterized by their width $W$ and height $H$. The former coincides with the number of points in a sub-windows, it expresses the compression factor and depends on the data trend variability and/or timeliness requirements. The larger the number of points in the window, the larger the compression we get, but also the larger the transmission delay of time sensitive data. Notice that a window with a single point does not compress data and thus a reduction of the window's width is tantamount to increase the measure frequency (system bandwidth) in order to catch sudden changes. The height $H$ expresses the accuracy that is the biggest difference between two measure points in a window and controls the measure and the robustness in finding outlier values. An outlier is a value which departs from the normal trend (see for example the first window in the Figure 1a).

The aggregation algorithm permits to transfer only the average values and the outliers to the base station. The average values are sent together within the time stamp of the last received value while the outliers are associated with the time stamp in which they are received. Note that, using this approach, the information about the time stamp is relevant in order to monitor the timeliness dimension and also to enable the re-building of the incoming signal when aggregated data are received by the base station. Furthermore, in some monitoring applications (e.g., earthquakes) the relative timing of the data that are detected from different sensors becomes fundamental since the correctness of the data synchronization directly impacts on the diagnose effectiveness.

## 4.2 Description of the algorithm

The algorithm is based on the observation that an outlier could mean either measurement errors (the circled point the first window in Figure 1a) or a change in the measured windows as in the last two sub-windows in Figure 1a. The two cases can be automatically distinguished by considering the precision value. Indeed, if the values are not accurate, but precise, it means that values are not similar to the reference value but they are characterized by a small standard deviation. In this case a change in the measured system has occurred. It is also possible to consider the case in which data values are not accurate nor precise and this occurs when the trend in the measured system is very irregular. A measurement error, on the other hand, is an occasional event and the values are still judged as accurate and precise. It is possible to distinguish different situations along the values of the two dimensions ($\varepsilon_{acc}$, $\varepsilon_{prec}$) (see Figure 1):

1) *Expected trend*: the trend can be defined as regular since values are precise and accurate (see the first two windows in Figure 1a);

2) *Slow change*: the trend is characterized by an unexpected, but lasting and small variation. Values are still precise, but not accurate (see the third and fourth windows in Figure 1a);

3) *Irregular trend*: the trend is characterized by unexpected and discontinuous variations. Values are not precise when the variation occurs, but they can be both accurate or inaccurate. The irregular trend could be further classified as (i) *Oscillatory/bursty trend* (see the second window in Figure 1b) if the values are not precise also in the subsequent time instants; (ii)*Step change* if the trend is characterized by an unexpected, but lasting and large variation in a way that precision requirements are satisfied again sometimes after the instant at which the variation occurs (see the third and fourth windows in Figure 1b).

Note that any data stream can be described as the combination of the described trends and of outlier values. Outliers are identifiable in all cases in which unexpected and not lasting variations occur. In this case, either accuracy or precision requirements are not satisfied when the trend change occurs, but after that the trend turns back to a stable behaviour. Outliers should not be ignored since they

could hide significant information. In fact, analysing historical series, outliers sometimes reveal periodical- and thus systematic-irregularities that can not be inferred from a local and limited analysis. The identification and analysis of

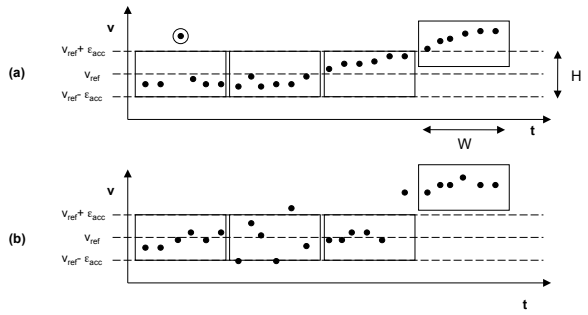


**Figure 1: Possible value trends**

the trends that compose the data stream is the first step of the aggregation algorithm briefly described in this paper. More details of the algorithm can be found in [3]. Aggregation operations aim to identify a set of values that can be considered a good approximation of the sensed data stream.

The algorithm is based on the evaluation of the similarity of the incoming data with the previous data values acquired in the sensing activity. Similarity degree depends on the precision and accuracy assessment. Checking these two dimensions, the algorithm is able to recognize three different situations described in the following also with excerpts of the corresponding pseudocode:

- Data follow the *expected trend*: the average of all the stored values except for outliers is calculated.

---

1. CASE ($< \varepsilon_{acc}, < \varepsilon_{prec}$)
2. IF number of analysed values=W AND Number of outliers $< L$ THEN t[z]=AVG(Acceptable values); Increment z
3. ELSE IF Number of outliers $> L$
4. THEN T= $\langle t_1, t_2, ...t_z \rangle$= V= $\langle v_1, v_2, ...v_w \rangle$
5. ELSE Analyse new value and GO TO (1)

---

- Data undergo a *slow change*: when the algorithm detects an outlier, it controls if it is associated with a permanent or transient data trend change. Such evaluation is performed on the basis of a specific parameter called *continuity interval* (C). This parameter specifies the number of values that the algorithm analyzes in order to define the kind of trend. If the values contained in C are precise and not accurate, the aggregation algorithm classifies the trend as "slow change" and calculates the average of the values stored before the exception and recalculates the expected value $v_{ref}$ along the last inaccurate values. Otherwise, inaccurate values are transmitted to the base station as outliers.

---

1. CASE ($> \varepsilon_{acc}, < \varepsilon_{prec}$)
2. Variable initializations: the number of unexpected values, the time instant in which the exception occurs ($T_e$)
3. O[j]=$v_w$ *storage of the outlier*
4. Indata[w]= $v_{w+1}$
5. WHILE accuracy $> \varepsilon_{acc}$ AND precision $< \varepsilon_{prec}$
6. INCREMENT the number of unexpected value and the number of outliers
7. O[j]=$v_w$
8. IF number of analysed values=W AND number of subsequent unexpected values = C
9. THEN t[z]=AVG(Acceptable values arrived before $T_e$); Increment z ;
10. t[z]=AVG(Acceptable values arrived after $T_e$); Increment z;
11. $v_{ref}$=AVG(Acceptable values arrived after $T_e$)
12. ELSE IF number of analysed values=W AND number of subsequent unexpected values<C
13. THEN t[z]=AVG(Acceptable values); Increment z;
14. ELSE IF number of subsequent unexpected values = C
15. THEN t[z]=AVG(Acceptable values arrived before $T_e$); Increment z;
16. $v_{ref}$=AVG(Acceptable values arrived after $T_e$)
17. Delete outliers from O[j] to O[j-C]
18. GO TO 1
19. else Indata[w]=$v_{w+1}$ and GO TO 1

---

- Data are characterized by an *oscillatory trend or bursts*: the algorithm recognizes that the number of outliers is greater than a specific threshold (L) and classifies the trend as very irregular. Therefore, all the values are transmitted, thus saving the additional computation energy. Moreover, in such a case, the measurement frequency of the sensor should be increased (and the window width consequently reduced) in order to identify the small data variations. In fact, on the basis of the received data, the base station must reconsider the context conditions and increase the measurement frequency for the considered sensor.

---

1. CASE ($< \varepsilon_{acc}, > \varepsilon_{prec}$) OR ($> \varepsilon_{acc}, > \varepsilon_{prec}$)
2. O[j]=$v_w$
3. IF number of analysed values=W AND Number of outliers $< L$ THEN t[z]=AVG(Acceptable values); Increment z;
4. ELSE IF Number of outliers $> L$
5. THEN T= $\langle t_1, t_2, ...t_z \rangle$= V= $\langle v_1, v_2, ...v_w \rangle$
6. ELSE Analyse new value and GO TO (1)

---

## 4.3 Energy evaluation

In sensors, energy drain is caused when using any of the sensor equipment, including (i) powering its memory, (ii) using its CPU, (iii) sending/receiving data. The rates of energy consumption for these operations are sensor-specific. Communication is often the major cause of energy drain in sensors and hence, in the interest of extending the sensor's life, communication must be limited. In details, each sensor is characterized by different energy factors: a) $e_t$:energy

consumption for the transmission of one byte; b) $e_e$*:* energy consumption for processing one instruction; c) $E_t$*:* energy consumption for data transmission to the base station; d) $E_e$*:* energy consumption for processing analysis and aggregation algorithms; e) $E_{tot}$*:* total energy consumption of the sensor node, calculated as $E_t + E_e$. The model presented in the following aims to minimize the total energy consumption by considering that $e_t \gg e_e$. In fact, the algorithm analyses the values in the data stream in order to define if it is possible to communicate to the base station only the aggregate values and the outliers. By considering Z aggregate values and J outliers, the algorithm is efficient if the output is composed by (Z+J) values instead of N where (Z+J)$\ll$ N:

$$e_t \cdot N > E_e + e_t \cdot Z + e_t \cdot J$$

## 4.4 Cost-quality tradeoffs

The outputs of the algorithm, and thus the number of values transmitted at the base station and the energy consumed strictly depend on the input parameters. *Precision*, *accuracy*, *timeliness* and the *continuity interval* values can be modified in order to obtain higher or lower quality. In fact, high precision and accuracy requirements are associated with a higher number of values transmitted. Therefore the higher the quality required the higher the energy consumed. Quality and timeliness requirements have to be properly designed along with the analysed context. In fact, stable and not critical phenomena need not a high quality level, increasing energy saving benefits. Irregular and critical contexts, on the other hand, require high quality results since even the smallest system changes should be detected and transmitted.

Note that it is also possible to influence the number of outliers and the accuracy of compression by defining a suitable continuity interval (C) value. In fact, a high C value increases the probability of classifying values as outliers instead of trend changes. Thus, the probability to send all the N values increases as well.

Figure 2 shows the list of input parameters. The possibility of choosing these values gives our algorithm a great flexibility in adapting to different application environments. Obviously their choice must be made, in each case, on the basis of a previous knowledge of the application and its operating environment or of experimental tests. A further development can be the usage of learning techniques to automatically adapt the input parameters set-point values.

# 5. TESTBED EXPERIMENTS

An experimental testbed has been setup in order to evaluate the performance of the proposed algorithm in terms of functional features and energy consumption properties; results have been evaluated by implementing the algorithm on a sensor and measuring the energy consumed during the elaboration phase. We have used two real data sets, named $A$ and $B$, in which the former represents the absorption spectrum of the acetylene C2N2 measured by means of a laser spectrograph method set up to 1.54 $\mu$m and the latter describes the absorption spectrum in which the measures are expressed in frequency modulation. Mica2 sensors have been used. These sensors are powered by two AA alkaline batteries and are built around an Atmel Atmega 128L microcontroller circuit and the CC1000 integrated radio circuit. Mica2 sensors use the operating system TinyOS that is programmed by using the event-oriented language nesC.

In order to experimentally measure and record the electrical energy absorbed by the sensor over time, we used a DAS (Data Acquisition System) capable of A/D conversion and recording of two voltage signals during the sensor operation: a first voltage signal $(v_1)$[1] is taken directly at the sensors power supply input pins; a second $(v_2)$ voltage signal is the one across a series $r = 10\Omega$ resistance, connected between the power supply and the Mica2 "load", used to convert the current supplying the sensor into a corresponding easy to acquire voltage signal. Even if one considers $V_{batt}$ changing with the circuit absorption, the electrical power and energy consumed by the Mica2 sensor can be properly measured by the product of current and voltage over the sensor and by its time integral, respectively. Henceforth, we did measure the circuit power consumption as $P(t) = i(t) \cdot v_1(t)$ and the corresponding electric energy consumption:

$$E(t) = \int_{t_0=0}^{t} [v_2(t')/r] \cdot v_1(t')dt'$$

where $t$' is just the integration variable, $t_0 = 0$ is an arbitrary starting time for measurement/integration and $t$ is either the current time or the whole/final observation time.

The algorithm code is composed of two main phases: data acquisition and evaluation and data transmission. The former phase gets the input data and evaluate them in order to define the type of trend that characterizes the data stream. Aggregate values and outliers are stored in an output buffer until the transmission phase has to be performed. As described in the previous sections, the transmission is required when either the input buffer is complete or certain timeliness requirements are needed. In the experiments the output buffer has been physically realized, and aggregated data and/or outliers have been sent at the end of the window. However, the values have not been grouped in one transmission packet, but we used a packet for each tuple. The algorithm has been implemented and tested in order to check its correctness and efficiency; its footprint - 11 kByte of RAM and 1 kByte of ROM - is rather small and this property is one of the significant factors for storage management in small sensors, but scarcely considered in the literature.

## 5.1 Experimental results

For the evaluation of the data set $A$, the values shown in Figure 2 have been used as parameter input data. Parameter values have been selected on the basis of the trend characteristics. In particular, $v_{ref}$ has been defined by calculating the average of the first subset of values in the analysed trend. The variability of the trend has been instead considered to determine suitable values of $\varepsilon_{acc}$ and $\varepsilon_{prec}$. The window width $W$ has been established in order to have a significant number of windows to work with. The definition of the window width influenced the setting of the continuity interval $C$ that includes a small number of values in order to increase the effectiveness of the algorithm. In fact, the smaller the continuity interval the lower is the sensitivity of the algorithm in the trend changes and outliers detection.

Figure 3 represents the values of data set $A$ together with the values transmitted by our algorithm (dotted line). Note

[1]Unlike in the rest of the paper, in this subsection $v_i$ denotes an electric voltage value

| Name | Value |
|---|---|
| Number of values | 150 |
| $v_{ref}$ | 0.18 |
| $\varepsilon_{acc}$ | 0.02 |
| $\varepsilon_{prec}$ | 1 |
| W | 30 |
| C | 4 |
| L | W/2 |

**Figure 2: Experimental Input data for data set *A***

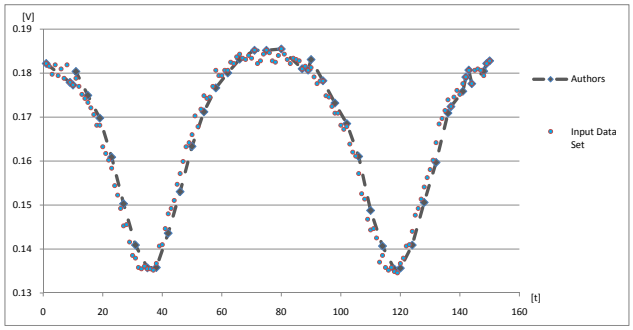


**Figure 3: Values of the data set *A* compared with the values produced by the proposed algorithm**

that our trend is calculated on the basis of the transmitted aggregate values and the outliers. Figure 4 describes the absorbed current and the energy consumed. The energy spent by the algorithm in data processing and transmission is respectively shown by the slope changes in the lower diagram, the leftmost being the energy spent in downloading the program into the sensor.

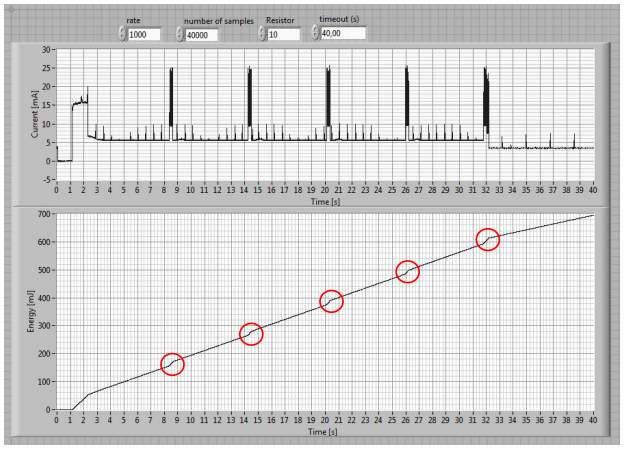


**Figure 4: Energy consumption by the algorithm**

The algorithm performance has been compared with other two approaches proposed in the literature and discussed in Section 2, namely [18] and [21]. Both approaches are based on the satisfaction of accuracy constraints and work quite well in the analysis of phenomena that are quite stable or data which are characterized by a linear trend.

The comparisona between algorithms have been based on three main criteria:

- *Compression rate*: the degree with which data have been aggregated. This dimension can be assessed as the ratio between the number of values transmitted and the number of data received. Note that the higher the compression rate the higher the probability to loose information about the original trend.

- *Energy savings*: the degree with which the aggregation allows sensors to save energy with respect to the case in which all the original values are sent to the base station.

- *Correctness*: the degree with which the aggregated data allow the base station to retrieve the original trend. In order to evaluate the correctness, we have evaluated all the values $v'_n$ that the base station can retrieve by using a linear interpolation between the received aggregated data. On the basis of these estimated values, it is possible to assess the achieved accuracy level in terms of Mean Absolute Error (MAE) and the related Mean Absolute Percentage Error (MA%E) that consider the average error calculated as the difference between the computed values and the real ones.

Figure 5 describes the comparison of the three algorithms on the basis of the energy savings and the compression rate while Figure 6 describes the comparison of the three algorithms on the basis of the correctness criteria.

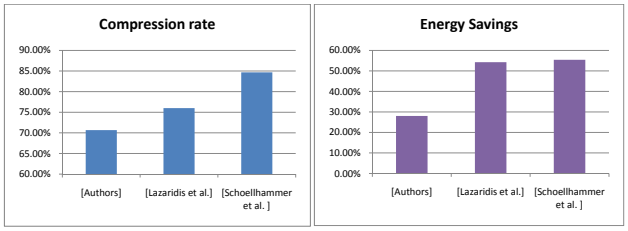


**Figure 5: Comparison among the three aggregation algorithms**

| | MAE | MA%Er |
|---|---|---|
| Authors | 0.0021 | 1.30% |
| Lazaridis et al. | 0.0030 | 1.88% |
| Schoellhammer et al. | 0.0010 | 0.63% |

**Figure 6: Comparison of correctness results**

Since our algorithm aims to improve both energy saving and data quality aspects, we can state that it has better performance than the algorithm by Lazadiris et al. [18]. Furthermore at a first sight, the algorithm by Schoellhammer [21] has a better performance than ours, but a deeper analysis shows that in case of non linear input data our algorithm provides results characterized by a higher quality. In fact, we have conducted a deeper and accurate analysis to confirm the performances of our algorithm. We have set the algorithm with the parameters in Figure 2 and we have calculated the error in the intervals in which data are characterized by a higher variance. Three areas can be identified: (a) the set of values arrived between t=1 and t=10; (b) the set of values arrived between t=65 and t=86; (c) the set of values arrived between t=139 and t=150. The evaluation of the correctness in these three intervals provides the results in Figure 7.

| | Authors | Schoellhammer et al. |
|---|---|---|
| (a) | 0.0008 | 0.0009 |
| (b) | 0.0011 | 0.0014 |
| (c) | 0.0008 | 0.0009 |

**Figure 7: Comparison of MAEs in case of non linear trends**

For the evaluation of the data set $B$, the values shown in Figure 8 have been used as parameters input data. The parameters have been defined following the same method used to define the parameters of the experiments based on the first data set (see Figure 2).

| Name | Value |
|---|---|
| Number of values | 150 |
| $v_{ref}$ | 0 |
| $\varepsilon_{acc}$ | 0.3 |
| $\varepsilon_{prec}$ | 1 |
| C | 4 |
| W | 30 |
| L | W/2 |

**Figure 8: Experimental Input data for data set $B$**

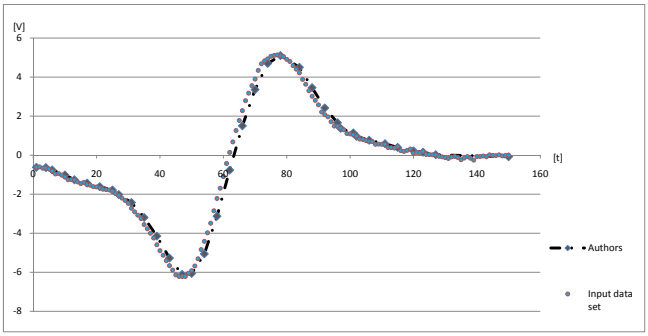


**Figure 9: Values of the second data set**

In Figure 9, the values of data set B are represented together with the values transmitted by our algorithm (dotted line). Data set B is characterized by a higher variance than the previous case and our algorithm improves its own performance compared with the other two considered algorithms.

Figure 10 describes the comparison of the three algorithms on the basis of the energy savings and the compression rate while Figure 11 describes the comparison of the three algorithms on the basis of the correctness criteria.

In this case, the higher variance that characterizes the trend makes the algorithm proposed in this paper and the algorithm proposed by Schoellhammer et al. almost equivalent. This experiment allows us to state that our algorithm is able to guarantee the quality of the output in linear and non linear trend and irregularities do not worsen its performance.

Looking at the results obtained by running the three algorithms on the first and the second data sets, it is possible to notice that the MAE differences in our algorithm appear to be mainly affected from inaccuracies in the horizontal dimension (i.e., time) while for the Schoellhammer et al. algorithm, the MAE differences appear to be mainly affected from inaccuracies in the vertical dimension (i.e., voltage).

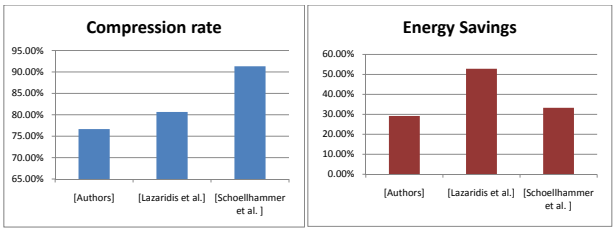


**Figure 10: Comparison among the three aggregation algorithms**

| | MAE | MA%E |
|---|---|---|
| Authors | 0.2234 | 48.20% |
| Lazaridis et al. | 0.3956 | 144.06% |
| Schoellhammer et al. | 0.1725 | 55.96% |

**Figure 11: Comparison of correctness results**

This is due to the fact that our approach transmits the aggregated values at the end of a time window and this implies delays in the transmission. This also confirms that the used aggregation function provides accurate values while using the Schoellhammer et al. algorithm the errors are mainly due to the aggregation evaluation and the transmission is timely, but less accurate.

It is possible to summarize the results obtained by the previous analysis by metrics proposed in Figure 12. From these

| | Compression rate/ Energy saving | MA%E/ Compression rate | MA%E/ Energy saving |
|---|---|---|---|
| Authors | 2.5 | 0.64 | **1.6** |
| Lazaridis et al. | **1.55** | 1.77 | 2.77 |
| Schoellhammer et al. | 2.84 | **0.615** | 1.75 |

**Figure 12: Summary of the comparison results**

results, it is possible to infer that each algorithm has some advantages on the others. [18] has a good trade-off between compression rate and energy saving while [21] presents the best results in terms of errors and compression rate. The algorithm presented in this paper is characterized by the best trade-off between errors and energy saving. This means that we succeeded in guaranteeing a defined quality degree saving energy.

## 6. CONCLUSIONS AND FUTURE WORK

Data management in WSNs must take care of energy drain by optimizing transmission operations among sensors, which are the most energy consuming, while keeping their quality as high as possible. In this paper, we have presented a data aggregation algorithm characterized by adaptivity features based on quality and energy saving requirements. Furthermore, the algorithm is designed to improve the capabilities of a single sensor to mine the input data and decrease its dependency on the base station. Experimental results discussed in Section 5 show the correctness of the algorithm and its efficiency. The comparison with other two aggregation algorithms proposed in [18] and [21] shows that the algorithm proposed in this paper is more effective in case of non-linear and irregular trend. In fact, especially in case several outliers occur, the proposed approach gives importance to all the changes by sending all the information about

unexpected behaviours to the base station. This is an innovative aspect with respect to other algorithms proposed in the literature in which the aggregation performs quite well with linear trends, but it worsens when outliers or non linear/sudden changes occur.

The energy savings in our approach could be further optimized by considering that so far, we simulate the separate transmission of the aggregate values and outliers. Actually, the transmission could be performed at the end of the window processing by using packet based protocols. The sensor stores the values to be transmitted and sends them in one or more packets at the end of the analysed window. In this case higher compression could be obtained, possibly limited by the consideration of timeliness-energy tradeoffs [2]. As to this issue, we have already performed some experiments using TinyOS and we have initially observed that, in terms of energy, it is better to use a packet for each tuple instead of bigger packets that include more tuples since, in the latter case, more computation and a longer transmission time is needed. Anyway, further experiments are to be performed.

Further work will also improve the algorithm by focusing on the definition of an optimization model for the maximization of energy savings through the automatic definition of the controllable algorithm parameters (e.g., accuracy, precision, continuity interval).

## 7. ACKNOWLEDGMENTS

A special thank you goes to Federico Rossini e Matteo Rugginenti who contributed to perform the experiments to validate the work. This work has been partially funded by the project Industria2015 SENSORI and by the ERC project 227977 SMScom at Politecnico di Milano.

## 8. REFERENCES

[1] A. Boulis, S. Ganeriwal, and M. B. Srivastava. Aggregation in Sensor Networks: an Energy-accuracy Trade-off. *Ad Hoc Networks*, 1(2-3):317–331, 2003.

[2] M. Busse, T. Haenselmann, and W. Effelsberg. Energy-efficient forwarding in Wireless Sensor Networks. *Pervasive and Mobile Computing*, 4(1):3–32, 2008.

[3] C. Cappiello and F. A. Schreiber. Quality- and energy-aware data compression by aggregation in wsn data streams. In *PerCom Workshops*, pages 1–6, 2009.

[4] R. Cardell-Oliver. ROPE: a Reactive, Opportunistic Protocol for Environment Monitoring Sensor Networks. In *EmNetS-II*, May 2005.

[5] Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang. Multi-Dimensional Regression Analysis of Time-Series Data Streams. In *VLDB*, pages 323–334, 2002.

[6] J. Chou, D. Petrovic, and K. Ramchandran. A Distributed and Adaptive Signal Processing Approach to Reducing Energy Consumption in Sensor Networks. In *INFOCOM*, 2003.

[7] D. E. Culler, D. Estrin, and M. B. Srivastava. Guest Editors' Introduction: Overview of Sensor Networks. *IEEE Computer*, 37(8):41–49, 2004.

[8] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos. Compressing Historical Information in Sensor Networks. In *SIGMOD Conference*, pages 527–538, 2004.

[9] H. P. D.P. Ballou, R.Y. Wang and G. Tayi. Modelling Information Manufacturing Systems to determine Information Product Quality. *Management Science*, 44(4), 1998.

[10] J. Gama and M. M. Gaber. *Learning from Data Streams*. Springer, 2007.

[11] L. Golab and M. T. Özsu. Issues in data stream management. *SIGMOD Record*, 32(2):5–14, 2003.

[12] H. Hu, C.-H. Jiang, K.-Y. Cai, and W. E. Wong. A Control-Theoretic Approach to QoS Adaptation in Data Stream Management Systems Design. In *COMPSAC (2)*, pages 237–248, 2007.

[13] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: a Scalable and Robust Communication Paradigm for Sensor Networks. In *MobiCom*, pages 56–67, 2000.

[14] ISO. *ISO/IEC Guide 99-12:2007 International Vocabulary of Metrology, Basic and General Concepts and Associated Terms*, 2007.

[15] A. Jain, E. Y. Chang, and Y.-F. Wang. Adaptive Stream Resource Management Using Kalman Filters. In *SIGMOD Conference*, pages 11–22, 2004.

[16] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom. Declarative support for sensor data cleaning. In *Proceedings of the 4th international conference on Pervasive Computing*, PERVASIVE'06, pages 83–100, 2006.

[17] N. Kimura and S. Latifi. A survey on data compression in wireless sensor networks. In *ITCC*, pages 8–13, 2005.

[18] I. Lazaridis and S. Mehrotra. Capturing Sensor-Generated Time Series with Quality Guarantees. In *ICDE*, pages 429–439, 2003.

[19] R. S. M. Bovee and B. Mak. A Conceptual Framework and belief- function Approach to assessing overall Information Quality. In *ICIQ*, 2001.

[20] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks. In *OSDI*, 2002.

[21] T. Schoellhammer, E. Osterweil, B. Greenstein, M. Wimbrow, and D. Estrin. Lightweight Temporal Compression of Microclimate Datasets. In *LCN*, pages 516–524, 2004.

[22] F. A. Schreiber, R. Camplani, M. Fortunato, M. Marelli, and G. Rota. Perla: A language and middleware architecture for data management and integration in pervasive information systems. *IEEE Trans. Software Eng.*, 38(2):478–496, 2012.

[23] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis. Balancing Energy Efficiency and Quality of Aggregate Data in Sensor Networks. *VLDB J.*, 13(4):384–403, 2004.

[24] Y.-C. Tu, M. Hefeeda, Y. Xia, S. Prabhakar, and S. Liu. Control-Based Quality Adaptation in Data Stream Management Systems. In *DEXA*, pages 746–755, 2005.

[25] D. Tulone and S. Madden. PAQ: Time Series Forecasting for Approximate Query Answering in Sensor Networks. In *EWSN*, pages 21–37, 2006.

[26] D. J. Yates, E. M. Nahum, J. F. Kurose, and P. J. Shenoy. Data quality and query cost in pervasive sensing systems. *Pervasive and Mobile Computing*, 4(6):851–870, 2008.