ZIZO: A Zoom-In Zoom-Out Mechanism for Minimizing Redundancy and Saving Energy in Wireless Sensor Networks

Ahmad Al Sayyid[a], Hassan Harb[b], Marc Ruiz[a], and Luis Velasco[a]
[a]*Optical Communications Group (OCG), Universitat Politecnica de Catalunya, Barcelona, Spain*
[b]*Computer Science Department, Faculty of Sciences, Lebanese University, Hadath, Lebanon*

***Abstract*—In modern life, there is invisible data being continuously generated, data that if collected and processed can detect risks and changes and enable us to mitigate their effect. Thus, there is an essential need to sense everything around us in order to make better use of it. This lead us to an era known as "sensing-era" in which wireless sensor networks (WSN) play a vital role in the monitoring of natural and artificial environments. Indeed, the collection and transmission of huge amounts of redundant data by sensor nodes will lead to a faster consumption of their limited battery power, which is sometimes difficult to replace or recharge, reducing the overall lifetime of the network. Therefore, an effective way to increase lifetime by saving energy is to reduce the amount of transmitted data by eliminating redundancy along the path to the sink. In this paper, we propose a Zoom-In Zoom-Out (ZIZO) mechanism aimed to minimize data transmission in WSN. ZIZO works on two WSN levels: on the sensor level where we propose a compression method called index-bit-encoding (IBE) in order to aggregate similar readings before sending them to the second network level, e.g. cluster-head (CH). The CH searches then for correlation among node data in order to optimize the sampling rate of the sensors in the cluster through a process called sampling rate adjustment (SRA). We evaluate the performance of our mechanism based on both simulations and experiments while the obtained results are compared to other existing techniques, and we show reduced energy consumption by up to 90% in some cases.**

***Index Terms*—Wireless Sensor Network; Energy Saving; Zoom-In-Zoom-Out; Index-Bit-Encoding; Sampling Rate Adjustment;**

I. Introduction

The great advances in technology have enabled humans to address many threats and risks that face them more effectively, but they also brought in with them new challenges. The planet is changing on such a large scale and faster than can be observed with conventional methods. Modern warfare is becoming increasingly technology-focused and the involvement of humans in it is becoming minimal, creating a great disadvantage to the side with lesser technology. And the wide spread use of antibiotics and other medication has lead to viral mutations that create novel diseases that are resistant and difficult to detect and control. Also, the modern life has become so dependent on large-scale services from the internet to public transportation, that the constant monitoring of hubs providing these services has become essential for early fault detection and maintenance, as the daily flow of human life can be crippled by an even a slight delay in the detection of such a fault , monitoring that is also beyond conventional methods.

The emergence and evolution of WSN in the last two decades has offered effective and low cost systems that allow to remotely monitor the target areas. In these networks, large numbers of sensors collect designated raw data from their surrounding, process them, and transmit them through wireless communication toward the sink for analysis. Furthermore, the rapid development in equipment and communication technologies allowed the production of sensors capable of collecting all sorts of data, setting WSN to make their way into a large number of every-day applications like agriculture, environmental study, military, smart homes, healthcare, industry, etc [1], [2].

Indeed, the design of WSN faces many different challenges and constraints, from the limited resources of the sensor (power, memory, processing speed), to the dense nature of their deployment and the dynamic architecture of the network, to data management and analysis. However, one major long-standing challenge in WSN is the energy consumption because it is strongly related to the network lifetime [3]. Given that the available sensor energy is rapidly consumed during data transmission, an effective approach to save the energy is to minimize the data collection and transmission. That is why adapting sensor sampling frequency approach is becoming fundamental in WSN. The idea behind such and approach is to find the redundancy in data through the study of correlation and adjust sampling frequency to minimize it.

Data collection in WSN is divided into two types: 1) Event Driven: where sensors collect and transmit data based on an event. This event can be a request from the sink, or sensors would be designed to detect events of significance in their surrounding, collect data about them, and transmit that data to the sink. 2) Periodic: where sensors collect and transmit data at fixed intervals of time. Networks that depend on this type of data collection are usually referred to as Periodic Sensor Networks (PSN). Both types have their respective applications. There are different design approaches in WSN, and a very popular one is cluster architecture, where sensors are grouped into clusters with one node called cluster head acting as an intermediate between the nodes and the sink, as it offers many advantages [4].

Such clustered architectures enabling smarter and more efficient WSN management are perfectly encompassed with the deployment of new architectures of the underlying trans-

 DOI 10.1109/JSEN.2020.3025188

port network to fully exploit the possibilities of autonomic networking paradigm [5]. Autonomic networking entails thus the capability to do measurements on the data plane, and generating data records that are collected and analysed to discover patterns (knowledge) from data and implement control loops to tune local elements or trigger decisions that involve elements in different locations [6]–[9]. Recently proposed architectures and tools for monitoring and data analytics in transport networks [10] facilitate the integration of WSNs with the transport infrastructure as a whole, including e2e connectivity (from monitoring data source to customer application and viceversa) and computing resources. This integration is crucial to extend the impact of WSN management beyond its own limits, e.g. allowing coordination between different WSN and allocating/dimensioning data analytics processes that can be shared by various WSN, just to mention a couple.

In this paper, we focus on automatic WSN management and propose an energy-efficient Zoom-In Zoom-Out mechanism based on the cluster network architecture and dedicated to periodic WSN applications. It utilizes two techniques which are applied at sensor level and the CH respectively. The first, called index-bit-encoding, is a data compression method that exploits the similarity in readings collected by the sensor to reduces the size of data that it transmits. The second, called sampling rate adjustment, studies correlation among sensor node data then adapts the sampling frequencies of the sensors in the cluster. We conducted a set of simulations, in addition to experiments using the very popular telosB nodes [11] in order to evaluate the performance of the proposed mechanism.

The rest of the paper is organized as follows: Section II discusses the existing data reduction and energy-efficient techniques proposed in the literature. Section III, introduces the design of the network architecture used in our mechanism. Sections IV and V detail the two techniques proposed at the sensor and CH levels respectively. The results obtained in both simulations and experiments are discussed in section VI. Finally, section VII concludes the paper and gives an onward perspective to our mechanism.

II. Related Work

In the literature, one can find a lot of proposals on data compression, aggregation, sampling frequency adaptation and clustering for WSN [11]–[14]. The main objective of such techniques is to search for redundancy and similarity among data in order to reduce the data transmission thus saving the network energy. The authors of [13], [14] give an overview about various energy-efficient techniques proposed for WSN.

Some works such as [15]–[20] propose data compression techniques. The authors of [15] propose a technique based on the neural network model. It combines feature extraction in convolutional neural networks with variational autoencoder and the restricted Boltzmann machine in order to compress and reconstruct the sensing data. In [16], the authors propose a layered adaptive compression design for efficient data collection (LACD-EDC) in industrial WSN. LACD-EDC is based on the clustering data scheme and it aims to search the spatio-temporal correlation within (e.g. intra) and among (e.g. inter) clusters. Then, a compression method is proposed at the sensor level followed by a recover technique at the sink in order to regenerate the raw data and achieve an approximation of original data. The authors of [17] propose a sequence statistical code based data compression method in order to reduce the packet size and improve the energy efficiency of sensors. The compression process is achieved using first order static code (FOST) and sequence code (SDC) algorithms which give better compression ratio compared to arithmetic coding. In [20], the authors propose a Sequential Lossless Entropy Compression (S-LEC) which organizes the alphabet of integer residues obtained from differential predictor into increased size groups. S-LEC code-word consists of two parts: the entropy code specifying the group and the binary code representing the index in the group.

Other works such as [21]–[26] propose sampling rate adaptation algorithms. The authors of [21] propose three mechanisms that allow sensor to adapt its sampling rate to the variation of monitored environment. The proposed mechanisms are respectively based on similarity functions, distance functions, and analysis of variance with statistical tests. The proposed techniques work on rounds, where each round consists of a set period of time. The sensor adapts its sampling frequency at the end of each round. In [22], the adaptation of sampling rate of the sensor node is based on system-context and application-context levels. On one hand, the availability of harvesting energy represents the system-context to identify the maximum rate of sampling to be assigned to the sensor node. On the other hand, the user request represents the application-context where feedback from a system executing specific rules of user or field scientists is used to set the rates of sensor node sampling in optimal way. The authors of [23] propose two sampling rate adaptation techniques: exponential double smoothing adaptive sampling (EDSAS) and Wiener filter based adaptive sampling (WFAS). Both algorithms search the correlation between current and previous collected data and aims to minimize the sensor sampling rate while a high level of data accuracy. In [26], the authors propose a prefix frequency filtering (PFF) technique based on clustering architecture of the network. Further to a local processing at the sensor node level, PFF uses Jaccard similarity function to allow aggregator nodes to identify similarities between near sensor nodes at each period and integrates their sensed data into one record.

The works in [27]–[31] are dedicated to reduce data transmission in WSN using data clustering techniques. The authors of [27] propose a routing protocol called Gateway Clustering Energy-Efficient Centroid (GCEEC) for WSN. The objective of GCEEC is to improve the load among sensor nodes as well selecting and rotating the CH near the energy centroid position of the cluster. In [28], a robust distributed clustering algorithm based on diffusion moth flame optimization is proposed. The aim of the proposed algorithm is to minimize the intra-cluster distance in order to determine the optimal partition at every sensor node. The authors of [29] propose a cluster-based

data gathering algorithm for WSN called lifetime-enhancing cooperative data gathering and relaying (LCDGRA). Basically, LCDGRA works on three phases: the first phase aims to group the sensor nodes into clusters based on K-means clustering and Huffman coding algorithms. The second phase assigns a set of relay nodes to each CH in order to aggregate data before sending to the sink node. In the last phase, the aggregated data are coded based on random linear coding and then relayed to the base station. The authors in [30] propose a grid-density clustering algorithm that combines grid and density techniques in order to enhance clustering in WSNs. The density technique is used to find arbitrary shaped clusters with noise while the grid technique allows to enhance the clustering quality by eliminating the boundary nodes of grids. Finally, the authors of [32] propose a sink oriented cascading model along with a memetic algorithm to help resist cascading failures via topology optimization. The proposed architecture design a network balancing metric called "sink-oriented betweenness entropy" and apply statistics to identify the correlation between typical network properties and robustness.

Although most of the proposed approaches allow for efficient data reduction and conserve network energy, they suffer from several disadvantages: 1) Most of them are fairly complex, and difficult to implement efficiently due to the limited computational resources of most sensors. 2) Some of them compromise some aspects of sensed data such as temporal information for the sake of energy saving. 3) Most focus on handling data at one of the network (e.g. sensor or CH) only. Our proposed mechanism consists of a set of low complexity techniques that utilize data on two levels of the network before sending it for analysis at the sink.

III. Network Design and Preliminaries

A. Network Design

Network architecture is one of the most important aspects of the deployment of sensor networks. It strongly affects to the performance of any proposed technique, especially int terms of energy consumption and communication overhead. In this paper, we are interested in the cluster-based network architecture due to three main reasons: 1) It supports high network scalability regardless of the number of deployed sensor nodes. 2) It reduces the overhead communication among nodes in the network. 3) It facilitates handling node failure. 4) It reduces energy consumption in the network by reducing the number of hops from the nodes to the sink. Typically, the cluster scheme divides the nodes into a set of clusters and a CH is assigned to each. The CH receives data from the member nodes inside the cluster, performs a certain function with it if needed, then forwards it to the sink node. The CH can be the same type of mote as the nodes, or it can be a more advanced type, better equipped to perform the extra functions needed from it. Another approach would be the dynamic selection of the CH, in which case it would be subject to several metrics like remaining energy of the mote, the distance to the sink, etc.

Fig. 1 presents the cluster scheme that our mechanism is designed to work with best. Data is periodically collected by the sensor and compressed using IBE before being sent to the CH. Data is received from all the sensors at the end of each period by the CH which uses it in the SRA process then forwarded it to the sink.

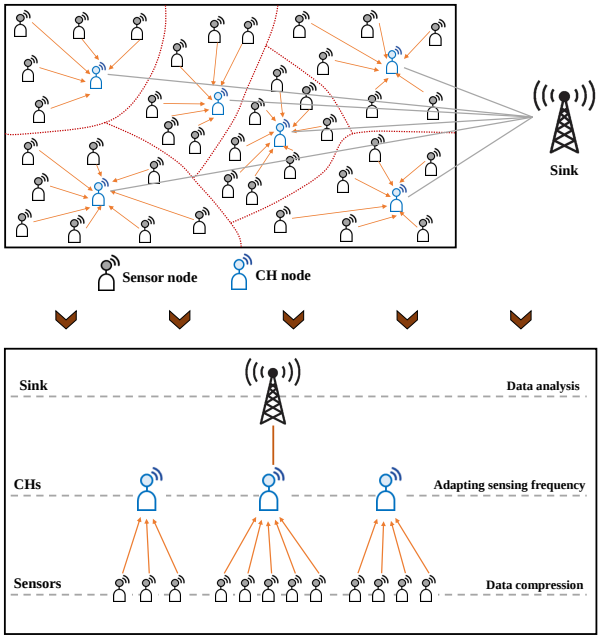


Fig. 1. Network design based on cluster scheme.

B. Notation

In order to maintain the connectivity of the network, sensors are organized into a connected graph $G = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N} = \{N_1, N_2, \ldots, N_n\}$ is the set of the sensor nodes and $\mathcal{A}$ is the set of arcs that connect the nodes. Mathematically, assume a sensor node N_i that collects a set of $\mathcal{T}$ readings during a period p then it forms a vector R_i^p of readings during p as follows: $R_i^p = [r_1^i, r_2^i, \ldots, r_{\mathcal{T}}^i]$.

IV. ZIZO at Sensor Level

In PSN, the similarity in collected readings is inversely correlated to the rate of variation in the conditions of the monitored target. i.e. The bigger the rate of variation in the conditions is, the lower the level of similarity, and vise versa. In other words when the conditions are stable, the rate of variation would be very small and level of similarity will be high, which also means there will be a high level of redundancy in the data. ZIZO aims at reducing the size of raw data sent by using IBE, which allows to aggregate similar readings in each period without compromising the temporal information of the collected data.

Let first define the similarity function (Δ) that each sensor uses to search for similarity between the readings it collected during a period.

Definition 1: (Similarity function, $\Delta_{r_j^i, r_k^i}$). Assume r_j^i and r_k^i are two readings collecting by a sensor node N_i . We consider r_j^i and r_k^i similar if and only if the difference between them is less than a defined threshold ε as follows:

$$\Delta_{r_j^i, r_k^i} = |r_j^i - r_k^i| \ \leq \ \varepsilon \tag{1}$$

where ε is a threshold determined by the application parameters.

By applying the similarity function to the readings of R_i^p collected at the period p sequentially, we get a set of unique readings U_i^p where each reading in R_i^p is similar to one and only one of the values in U_i^p. Then for each reading in U_i^p its weight is calculated using IBE which is explained in the next section. The combination of unique readings and their weights form the compressed set of readings ${R'}_i{}^p$ which is sent to the CH.

A. Index-Bit-Encoding Method

Index-bit-encoding method enables the compression of the set of readings R_i^p while preserving the temporal information of the data. It does this by encoding the indices of the readings in U_i^p in the weight of the corresponding reading. This way, upon decompression, a very close estimate of the original set of readings is generated, with the mean difference between the original set and the decompressed set is less than ϵ. The generation of the weight passes through the intermediate process of generating the code of reading defined in the following:

Definition 2: (Code of a reading $u_k^i \in U_i^p$, c_k^i). c_k^i is a binary code with a size equals to the number of readings in R_i^p. Thus, for a reading u_k^i in U_i^p, for each r_j^i in R_i^p, if u_k^i and r_j^i are similar corresponding bit in c_k^i is set to 1, otherwise it set to 0.

Definition 3: (Code weight, $wgt(c_k^i)$). The code weight of c_k^i is the integer value corresponding to the code c_k^i.

The final form of the set of readings R_i^p would be: ${R'}_i{}^p = [(u_1^i, wgt(c_1^i)), (u_2^i, wgt(c_2^i)), \ldots, (u_k^i, wgt(c_k^i))]$. The size of the set of readings would be optimally set to multiples of 8, as this would correspond to one byte. Given that each decimal value in the set of readings needs 4 bytes to be encoded, and depending on the size of the set of readings R_i^p, which is subject to adapt under ZIZO, the size of the compressed set of readings could be much smaller than the original set.

Fig. 2 shows and illustrative example of the compression and decompression of the set of measurements through IBE using similarity function with threshold $\varepsilon = 0.8$.

B. Sensor Level Algorithm

Algorithm 1 shows the sequential application of the similarity function on the set of readings of a sensor in period R_i^p in order to generate the compressed vector ${R'}_i{}^p$, with a period size $\mathcal{T}$ and similarity threshold ε.

The algorithm can be executed in two ways. Either the R_i^p is collected and at the end of the period the sensor proceeds to compress the vector, like what is demonstrated in IV-B. The algorithm starts at index 1 and proceeds until $\mathcal{T}$, however any possible sequence can be followed in order to obtain the best compression, as the low complexity of the algorithm permits this. The other way is to start the compression at the beginning of the period, with each new reading taken being immediately added to the compressed set ${R'}_i{}^p$ appropriately by applying the similarity function, and adjusting the code of an existing reading or adding a new one. This is even less complex and easier to implement.

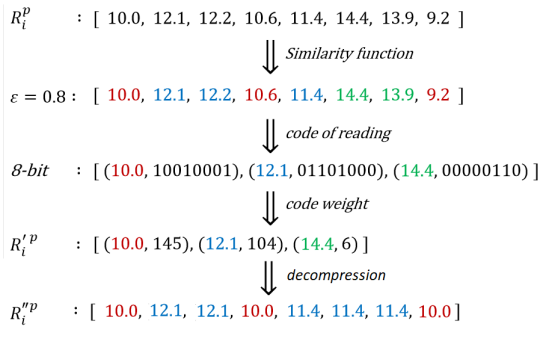


Fig. 2. Illustrative example of data compression and decompression in ZIZO.

The set of unique readings ${U_i}^p$ is started empty (line 1). Then for each reading ${r_k}^i$ this set is checked to see if there a reading ${u_j}^i$ similar to it (line 3). If not, ${r_k}^i$ is added to ${U_i}^p$ with a code that indicates that there is no readings similar to it in the previous indices (line $4-7$). Then the code ${c_j}^i$ of each reading ${u_j}^i$ is appended by 1 if similar to ${r_k}^i$ and 0 otherwise (lines $9-15$). Finally, for each reading ${u_j}^i$ in ${U_i}^p$ the code weight $wgt({c_j}^i$ is calculated from ${c_j}^i$, and they are added to the compressed set ${R'_i}^p$(lines (16−19)) .

Algorithm 1 Sensor Level Algorithm.

Require: period size: $\mathcal{T}$; similarity threshold: ε; set of readings of a sensor in period: R_i^p.
Ensure: compressed readings set of sensor in period: ${R'}_i{}^p$.

1: ${U_i}^p \leftarrow \emptyset$
2: **for** each reading $r_k^i \in R_i^p$ where $k \in [1, \mathcal{T}]$ **do**
3: Sim $\leftarrow$ there exists ${u_j}^p \in {U_i}^p$ where $\Delta_{{r_k}^i, {u_j}^i} \leq \varepsilon$
4: **if** Not Sim **then**
5: $c_j^i \leftarrow \{0, 0, .., k-1\}$
6: ${U_i}^p \leftarrow {U_i}^p \cup \{{r_j}^p, c_j^i\}$
7: **end if**
8: **for** each reading ${u_j}^i \in {U_i}^p$ where $j \in [1, Count({U_i}^p)]$ **do**
9: **if** Sim **then**
10: $c_j^i \leftarrow c_j^i \cup \{1\}$
11: **else**
12: $c_j^i \leftarrow c_j^i \cup \{0\}$
13: **end if**
14: **end for**
15: **end for**
16: ${R'}_i{}^p \leftarrow \emptyset$
17: **for** each reading ${u_j}^i \in {U_i}^p$ where $j \in [1, Count({U_i}^p)]$ **do**
18: $wgt({c_j}^i) \leftarrow CalculateCodeWeight(c_j^i)$
19: ${R'}_i{}^p \leftarrow {R'}_i{}^p \cup \{{u_j}^p, wgt({c_j}^i)\}$

20: **end for**
21: **return** ${R'}_i{}^p$

C. Adapting Index-Bit-Encoding Method to a Large Period Size

Mostly, the performance of any compression technique is evaluated based on the compression ratio. In our mechanism this depends data similarity and on the period size ($\mathcal{T}$), as number of bytes used for the weight of a reading is directly proportional to it. ($\mathcal{T}$) thus becomes a limitation that affects the compression size of IBE. To overcome this limitation we propose two strategies that allow to adapt IBE to a large period size: scaling down and period dividing.

1) Scaling Down: In WSN, readings collected successively are usually similar especially when the slot time is short or the monitored condition varies slowly. That is why it is possible to reduce the number of readings collected during a large period size, by using the scaling down strategy which aggregates every subset of successive readings into one value, e.g. the mean value. The number of aggregated readings is taken according to the scaling down degree, d. The bigger the value of d is, the more the number of the collected readings during a period is reduced, but this could lead to reduced accuracy, and vise versa. Then, IBE is applied over the reduced obtained set of readings . Fig. 3 shows IBE method adapted to scaling down strategy with two degrees $d = 2$ and $d = 3$.

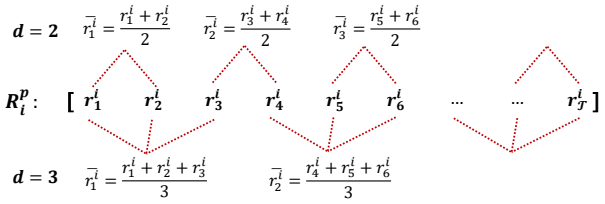


Fig. 3. Scaling down strategy with $d = 2$ and $d = 3$.

2) Period Dividing: In this strategy, we aim to apply IBE over a large period size but without affecting the accuracy of the transmitted data. The intuition is to divide the readings collected during the period into m equal blocks where each block contains t readings. Optimally, t must be a multiple of 8 and $\mathcal{T} = m \times t$. Then, IBE is applied separately over each block of readings. Fig. 4 shows the period dividing strategy applied over a period size of 128 divided into 4 blocks of size 32. Thus, each block needs a 32 bit encoding, i.e. 4 bytes, in order to encode the weight of each readings.

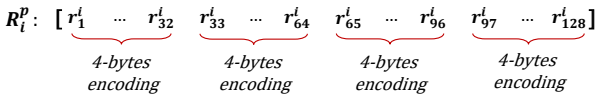


Fig. 4. Period dividing strategy with block size of 32 readings.

V. ZIZO at CH Level

Each period p is divided into $\mathcal{T}$ equal slots of time, e.g. $p = [s_1, s_2, \ldots, s_\mathcal{T}]$. At the end of each period, the CH will receive the sets of readings coming from its member nodes in the cluster. Indeed, data collected by the sensors in the same cluster contain a certain level if redundancy due to the spatial-temporal correlation between them. Thus, in order to reduce the data transmission from the member nodes, the CH would adapt the sensing frequencies of the sensors in order to minimize the redundancy.

Let assume that the CH receives the compressed set of readings ${R'}_i{}^p$ from a sensor node N_i. The first step is to decompress ${R'}_i{}^p$, using the process of reverse index-bit-encoding, index-bit-decoding. This allows the generation the set ${R''}_i{}^p$, which is a very close estimation of ${R_i}^p$. Then, all the decompressed reading sets are vertically stacked to form a matrix $\mathcal{M}^p$ at each period p as follows:

$$\begin{array}{c} \\ {R_1}^p = \\ {R_2}^p = \\ \vdots \\ {R_\mathcal{T}}^p = \end{array} \begin{array}{c} \begin{array}{cccc} s_1 & s_2 & \ldots & s_\mathcal{T} \end{array} \\ \begin{pmatrix} r_1^1 & r_2^1 & \ldots & r_n^1 \\ r_1^2 & r_2^2 & \ldots & r_n^2 \\ \vdots & \vdots & \ldots & \vdots \\ r_1^\mathcal{T} & r_2^\mathcal{T} & \ldots & r_n^\mathcal{T} \end{pmatrix} \end{array} \tag{2}$$

where each row R_k^p for $k \in \mathcal{T}$ consists of a set readings of n sensors at one instant of time τ.

Our objective is to study the redundancy existing in $\mathcal{M}^p$. Then, based on it, to adapt the sensing frequency of all the sensors in the cluster for the next period $p+1$ in order to try to minimize the collection of redundant data. In the next sections, we show how the CH calculates the redundancy level in the data matrix based on the T-test statistic and how it adapts the sensing frequencies of the sensors.

A. Redundancy Rate based on T-test

In our mechanism, we incorporate the temporal information between the collected data in the calculation of the redundancy rate. The CH determines if each pair of sets of readings collected at successive slot times t^i and t^{i+1} are similar, and based on the number of such sets, it calculates the Redundancy Rate for the period (RR_p).

The paired T-test is fitting for this measurement because each pair of successive sets is a set of paired observations by statistical definition. The null hypothesis of the T-test is that the true mean difference between paired observations is zero. The alternative hypothesis being that there is a significant difference.

Lets assume the set of readings collected by all the sensors during a slot time t as $R^t = [r_1^t, r_2^t, \ldots, r_n^t]$, where r_i^t is collected by the sensor N_i and the number of sensors, i.e. the number of readings in R^t is n. Then, the value of the T-test for the pair R^t and R^{t+1} collected at successive time slots t and $t+1$ is calculated as follows:

$$T - test\,(R^t, R^{t+1}) = \frac{\overline{X}^{t,t+1} - \mu}{\frac{\sigma^{t,t+1}}{\sqrt{n}}} \quad (3)$$

where

- $\overline{X}^{t,t+1}$: the mean of the difference set $\{R^{t+1} - R^t\}$
- μ: is the population mean of the whole matrix $\mathcal{M}^p$
- $\sigma^{t,t+1}$: the standard deviation of the difference set $\{R^{t+1} - R^t\}$

Thus, two successive sets of readings R^t and R^{t+1} are considered as having a significant difference if the value of the T-test between them is less than a defined threshold t as follows:

$$T - test\,(R^t, R^{t+1}) \leq t \quad (4)$$

where t is the T-test value corresponding to some desired false-rejection probability risk $\alpha \in [0, 1]$. Sets of readings that do not have a significant difference between them are considered similar.

Finally, the redundancy rate during the period p, i.e. RR_p, can be calculated as the number of similar successive pairs of sets of readings over the total number of possible successive pairs of sets of readings $(\mathcal{T} - 1)$ during the period as follows:

$$RR_p = \frac{\sum_{i=1}^{\mathcal{T}-1} T - test\,(R^t, R^{t+1}) \leq t}{\mathcal{T} - 1} \quad (5)$$

Thus, RR_p will be a number between 0 and 1; with 0 meaning no redundancy and 1 meaning maximum redundancy. The value of α greatly influences the redundancy rate; the higher it is, the greater the tolerance is to the difference between sets. That is why the choice of α is a parameter set depending on the application.

B. Sensing Rate Adjustment Algorithm

After calculating the redundancy rate at each period, the CH adapts the sensing frequencies the sensors in the cluster uniformly in order to try to minimize the collection of redundant data in the next period, within an upper and lower bound. A big level of similarity of the readings of one sensor can be offset by a big level of difference in the readings of another in the cluster, leading to a sampling rate that is not too high nor too low. In addition to redundancy, several metrics that are determined depending on the application can be used, and they can be a predefined set of rules, or dynamic based on the changing nature of the data. The preservation of the temporal data and the fact the a very close estimate of the original set of measurement can be created for each sensor means that trend analysis and time series analysis could be used by the sink, which may have additional resources, and by using these tools a smart decision making system for sensing rate adaption can be implemented, and the sink would then direct the CHs to change the sensing frequencies in their respective clusters accordingly.

VI. Performance Evaluation

The performance of our mechanism is evaluated using simulations and experiments. The first is done using data collected by the Intel Berkeley Research Lab and they are available online. The second is using data are gathered through real experiments using telosB nodes. In the next sections, we detail each of them while discussing the obtained results. The parameters used in our setup for both simulations and experiment are shown in the Table I:

TABLE I
Simulation Application Parameters.

Parameter	Description	Values
$\mathcal{T}$	period size	32 and 64
ε	Similarity threshold	temperature: 0.07, 0.1, 0.2 humidity: 0.2, 0.5, 1 light: 10, 15, 25
C_r	criticality level application	*low* and *high*
t	false-rejection probability risk value	0.2

For our application, we considered there are three levels of redundancy as shown in Table II, Redundancy Rate Table (RRT).

TABLE II
Redundancy Rate Table (RRT).

Redundancy Rate, RR_p	**Description**
$0 \leq RR_p \leq 0.4$	low redundancy rate
$0.4 < RR_p \leq 0.7$	medium redundancy rate
$0.7 < RR_p \leq 1$	high redundancy rate

Also, for SRA, we introduced a parameter that we used along the redundancy rate called application criticality. We considered the application to be either of *low* or *high* criticality. The following Sampling Rate Adjustment Table (SRAT) was created and used a decision mechanism by the CH.

TABLE III
Sensing rate adjustment table (SRAT).

RR_p	$C_r \rightarrow$	***low***	***high***
low		60% of $\mathcal{T}$	$\mathcal{T}$
medium		40% of $\mathcal{T}$	60% of $\mathcal{T}$
high		20% of $\mathcal{T}$	40% of $\mathcal{T}$

A. Simulation Results

In our simulations, we used the real sensor data collected in the Intel Berkeley Research Lab [33]. It was compiled by deploying 54 sensors of type Mica2Dot and collecting weather conditions such as temperature, humidity and light. The data set contains 2.3 million readings collected over the course of about 40 days with a fixed sensor sampling rate of one reading per 31 seconds. For our setup, we assumed that all nodes send their data to a common CH placed at the center of the lab. Fig. 5 shows the distribution of the sensors in the Intel lab. For comparison purposes, we selected those

techniques among all available in the literature that closer fit with the main characteristics of our proposed mechanism, i.e. distributed operation and energy minimization target by means of sensors data collection reduction. Specifically, we selected: 1) S-LEC in [20] as it is a lossless compression technique that compresses data efficiently and robustly, but pays no attention to the redundancy in the data. 2) PFF technique in [26] which is similar to our technique in terms of using aggregation on the part of the sensor to compress data, and finding similarity in the data on the part of the cluster head. However the aggregation method of the technique does not preserver temporal information, and only uses similarity in the data to further remove redundant data before sending the data to the sink.

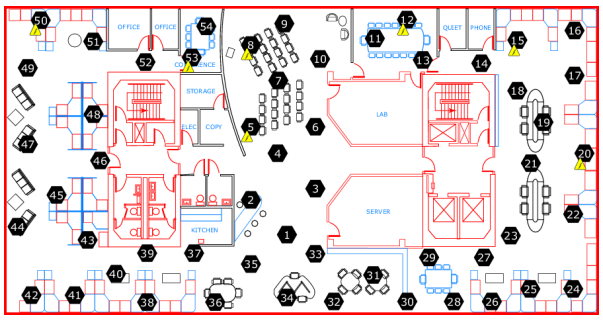
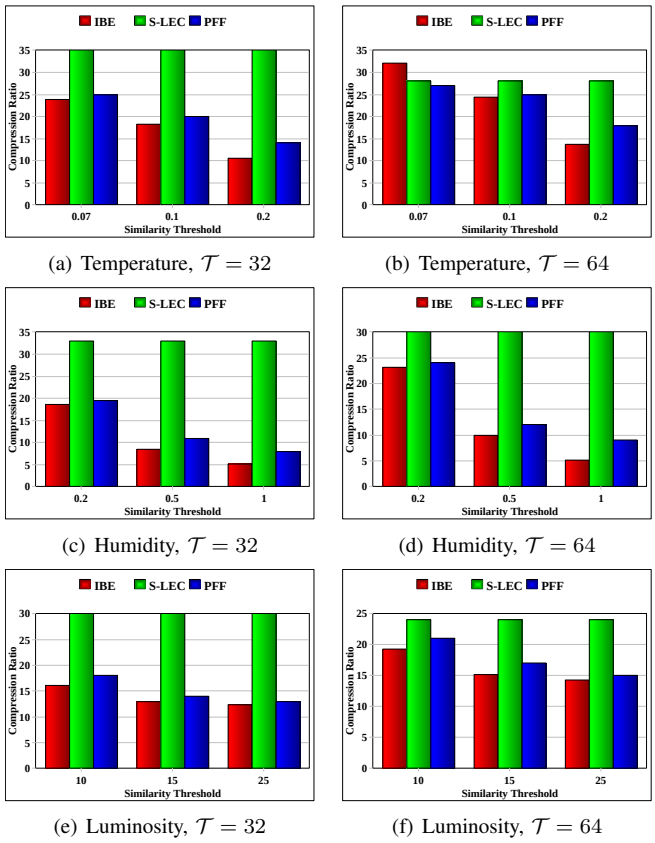


Fig. 5. Distribution map of the sensors in the Intel lab.

1) Compression Ratio Study: First, we study the performance of the IBE method proposed in our mechanism compared to S-LEC and PFF at the sensor level in terms of data compression ratio only i.e the size of data sent after compression relative to the size of the data vie that naive approach. Indeed, the obtained results of IBE is highly related to the period size $\mathcal{T}$ and the Similarity Threshold value ε (Fig. 6). We show that the compression ratio using IBE is less than those obtained using S-LEC in all cases, and than those obtained by PFF in almost all cases, giving a compression ration of 5% in the best case and 32% in the worst case. This is due to the simplicity in which data compression is encoded compared to the other techniques. In addition, the following can be observed: 1) The compression ratio of IBE improves with the decreasing of the period size (Fig. (6(a) and 6(b)) or (6(c) and 6(d)) or (6(e) and 6(f))). This is because when $\mathcal{T}$ decreases, the number of bytes needed to record the temporal information decreases, that is why period dividing becomes an advantage. 2) The compression ratio of IBE improves with increasing the Similarity Threshold (Fig. 6(a) to 6(f)). This is because when the value of ε increases, the greater the difference that can exist between two readings while still considering them similar and aggregating them together. This is why the Similarity Threshold is an important factor in determining both the Compression Ratio and Data Accuracy. 3) IBE gives much better results than S-LEC in the cases where there is high redundancy, such as with the humidity condition as shown in Fig. 6(c) and 6(d)). This is because S-LEC does not aim to reduce redundancy but compresses the data as is, which allows IBE to be more effective in these cases.



(a) Temperature, $\mathcal{T} = 32$ (b) Temperature, $\mathcal{T} = 64$

(c) Humidity, $\mathcal{T} = 32$ (d) Humidity, $\mathcal{T} = 64$

(e) Luminosity, $\mathcal{T} = 32$ (f) Luminosity, $\mathcal{T} = 64$

Fig. 6. Data compression ratio as a function of Similarity Threshold under different Period sizes .

2) Data Redundancy Study: The aim of this section is to show how sampling rate adjustment affected the value of the redundancy rate resulted in the CH at the end of each period. For this aim we fixed three parameters: the period size to 64 readings, the similarity threshold to 0.1 and T-test false-rejection probability risk to 0.2. We also used only temperature data, and ran the simulation under three conditions: Without SRA, with SRA and *low* criticality, with SRA and *high* criticality. Fig. 7 demonstrates how the redundancy level (calculated in equation 5) varied over the course of 20 periods. The results show how the RR_p leads to SRA, which leads to decreasing RR_p in the next round, which is clear in the case of SRA with *low* criticality as compared to No SRA. They also show the effect of the SRA decision parameters, as settings strict SRA decision parameters such as for high criticality applications lead to redundancy rates that sometimes matched the condition where no SRA is used.

3) Adapting Sampling Rate Study: This section uses the same simulation used in the previous section, but aim to highlight how exactly the sampling frequency was adjusted over the course of the periods under the aforementioned conditions. Fig. 8 shows this variation, and demonstrates how strict SRA decision parameters such as with high criticality lead to the sampling frequency remaining equal to period size in much

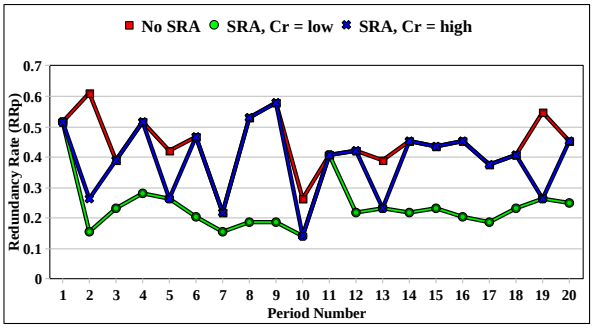


Fig. 7. Redundancy rate variation as a function of period number for temperature data. $\mathcal{T} = 64$, $\varepsilon = 0.1$.

of the simulation, which caused the high levels of RR_p as shown in Fig. 7. While with low criticality the parameters allowed for a bigger decrease in sampling frequency, and lead to lower levels of RR_p. This shows the importance of choosing adequate parameter values in deciding SRA in order to lower redundancy without affecting data accuracy by using a very high Similarity Threshold.

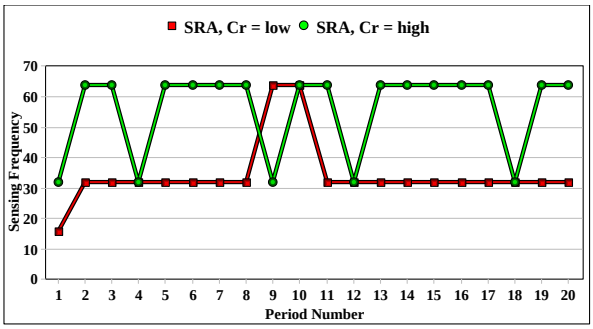
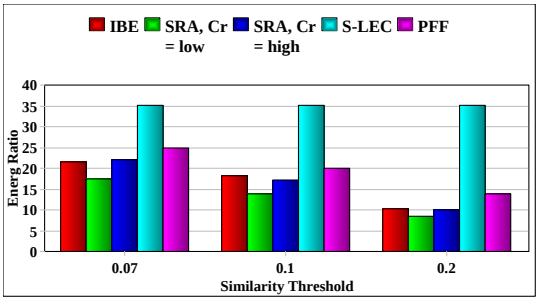


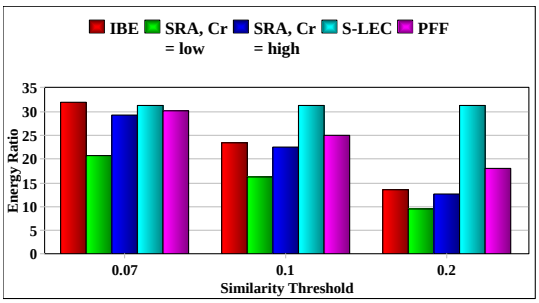
Fig. 8. Variation of sensor sampling frequency as function of period number for temperature data. $\mathcal{T} = 64$, $\varepsilon = 0.1$.

4) Energy Consumption Study: Reducing the energy consumption in the network is one of the most important challenges in WSN. In our simulations, we used the Heinzelman model [34], [35] which is the most popular model for energy consumption estimation to evaluate the energy consumption in the network under different scenarios. In such model, energy consumption is highly dependent on the data transmission and reception in the network while negligent the other factors (sensing, processing, etc..). For each scenario, we calculated the overall energy consumption by all the sensors and the CH over the course of the whole simulation. The scenarios were: using IBE alone, using IBE with SRA with low criticality, using IBE with SRA with high criticality, using S-LEC and PFF, while varying the period size and the Similarity Threshold, along the naive approach.

Fig. 9 displays the energy ratio of these scenarios i.e. the energy consumption in each relative to the energy consumption of the naive approach. It shows that our technique can significantly reduce the energy consumption, as using IBE alone was more energy efficient in all but one case, reducing energy consumption ratio down to 10% compared to 35% and 14% for S-LEC and PFF respectively. This is consistent with the results in Fig. 6 which are similar in terms of compression ratio. Also, we observe that the energy consumption using our mechanism is lower with smaller period size (Fig. 9(a) and 9(b)) and higher Similarity threshold (Fig. 9(a) or 9(b)), which shows a trade-off between energy efficiency and data accuracy. Finally, we see the effect of SRA, as it lead to lowering the energy consumption ratio to 8% in the best case.



(a) $\mathcal{T} = 32$



(b) $\mathcal{T} = 64$

Fig. 9. Energy consumption ratios of the whole network for some algorithms as a function of similarity threshold under different period sizes.

B. Experiments Results

In order to evaluate our mechanism in live scenario, we conducted experiments using 30 Crossbow telosB nodes deployed in our laboratory. The nodes collect data about temperature, humidity and light. The data is collected for about one week and they are sent to a specific CH node called SG1000 [36], which can be connected to a laptop machine for retrieving, then analyzing, the collected data. Due to the limited resources of telosB, we set the period size to 32 readings and the sampling rate of each sensor is fixed to one reading per 30 seconds. Fig. 10 shows the distribution of nodes inside the laboratory. Furthermore, it is important to notice that our mechanism is implemented on the nodes based on the nesC language [37] the standard programming language of tinyOS [38], while a Java code is implemented on the laptop machine to retrieve data from the sink node.

1) Compression Ratio Study: Similar to Fig. 6, Fig. 11 shows the compression ratio of data sent to the CH after applying IBE and S-LEC relative to the naive approach. The obtained results show that IBE outperforms S-LEC in terms of reducing the quantity of data sent from the sensor, achieving

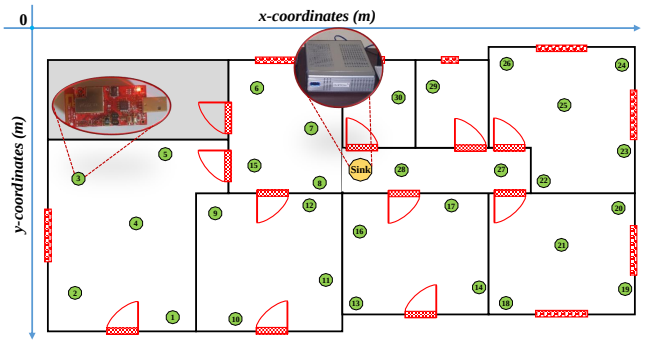
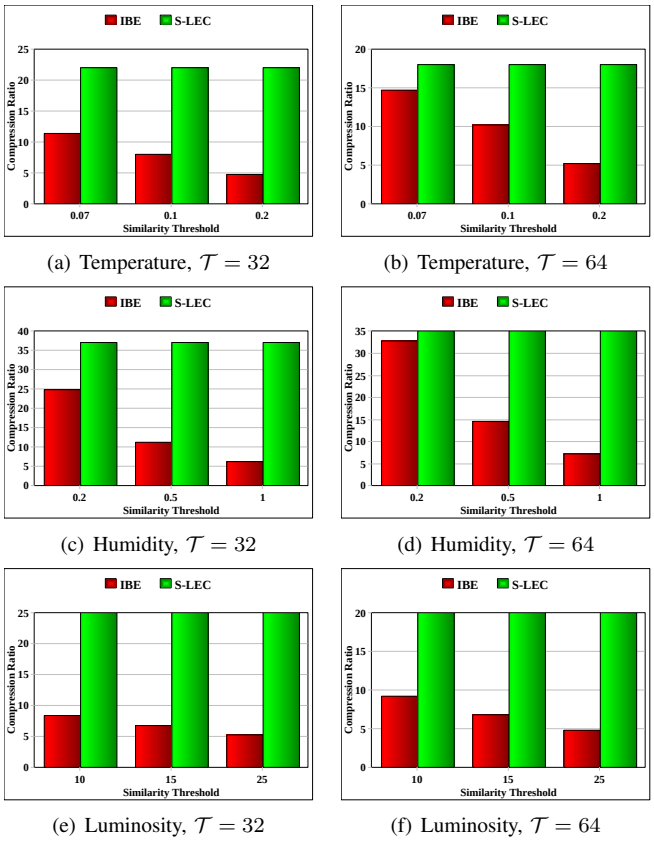


Fig. 10. Distribution map of the sensors in our lab.

a compression ratio of 5% in best case compared to 22% for S-LEC. We also observe that the compression ratio using IBE improves with the decreasing of the period size or the increasing of the Similarity threshold which corroborates the results of the simulation Fig. 6).



(a) Temperature, $\mathcal{T} = 32$ (b) Temperature, $\mathcal{T} = 64$

(c) Humidity, $\mathcal{T} = 32$ (d) Humidity, $\mathcal{T} = 64$

(e) Luminosity, $\mathcal{T} = 32$ (f) Luminosity, $\mathcal{T} = 64$

Fig. 11. Data compression ratio as a function of Similarity Threshold under different periods sizes.

2) Data Redundancy Study: The aim of this section is to replicate the conditions of the data redundancy study of the simulation. Fixing period size at 64 readings, the similarity threshold to 0.1 and T-test false-rejection probability risk to 0.2. We also used only temperature data, and did experiments under three conditions: Without SRA, with SRA and *low* criticality, with SRA and *high* criticality. Our objective is to study the variation of RR_p in the temperature data collected in our laboratory compared to the level obtained in the simulations. The results as shown in Fig. 12 are consistent with those of the experiment, as SRA allowed the RR_p to drop at the end of each round. It is also worth noting that due to the high level of similarity in temperature data, the redundancy results for the high criticality application were closer to those of low criticality application than to No SRA.

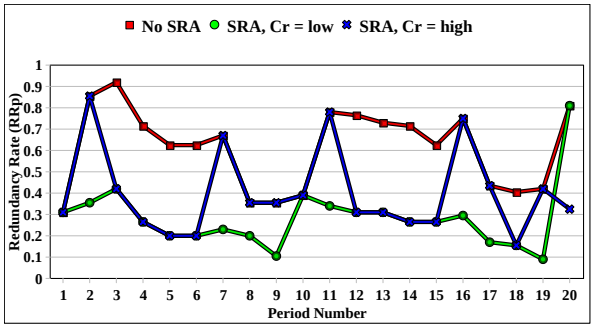


Fig. 12. Redundancy rate variation as a function of period number for temperature data. $\mathcal{T} = 64$, $\varepsilon = 0.1$.

3) Adapting Sampling Rate Study: Relative to the redundancy level studied in Fig. 12, Fig. 13 shows the sensing frequency for a sensor after applying IBE and SRA during a set of rounds. We observe that with the high level of RR_p over the rounds, the high criticality application sensing frequency was lowered to minimize RR_p. This confirms the behavior of the algorithms proposed in our mechanism in terms of reducing the sensing frequency of a sensor when the redundancy level increases.

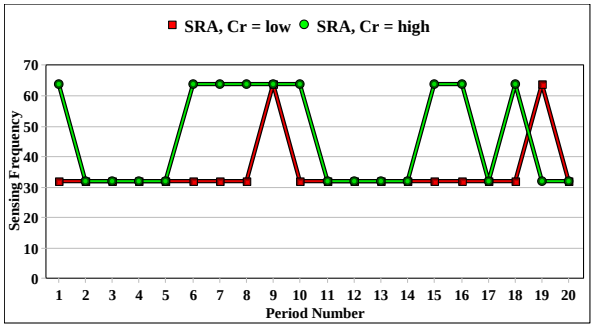
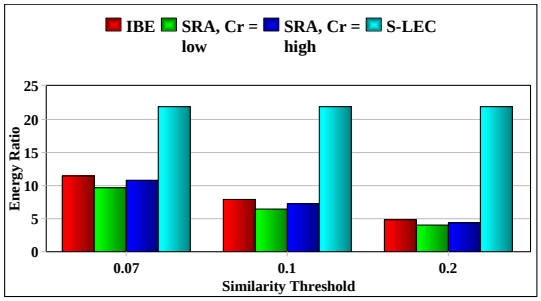


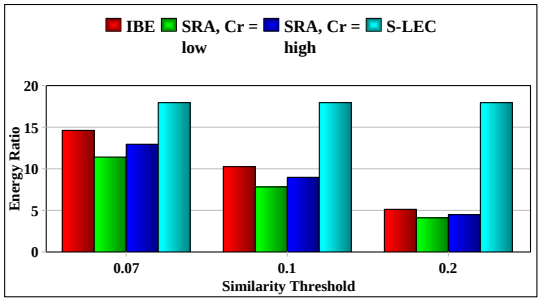
Fig. 13. Variation of sensor sampling frequency as function of period number for temperature data. $\mathcal{T} = 64$, $\varepsilon = 0.1$.

4) Energy Consumption Study: Similar to the energy consumption study in the simulation, we calculated the overall energy consumption of using IBE alone, SRA with low and high criticality, S-LEC and the naive approach. Fig. 14 shows the energy consumption ration in our lab of these scenarios relative to the naive approach. The obtained results show that in the best case scenario, our mechanism optimized the energy consumption ratio to 5%, compared to 22% for S-LEC. The results also confirm the simulation conclusion that energy

consumption decreases with smaller period size and higher Similarity Threshold.



(a) $\mathcal{T} = 32$



(b) $\mathcal{T} = 64$

Fig. 14. Energy consumption ratios of the whole network for some algorithms as a function of similarity threshold under different period sizes.

C. Analytical and Complexity Study

In this section, our objective is to give further considerations of our mechanism by studying its complexity at both sensor and CH levels. Indeed, the complexity is an important metric that must be considered in WSN due to the limited sensor resources and the impact of processing to the data delivery delay when sending the data to the sink [39], [40].

From one hand, each sensor node N_i will form a set R_i^p of $\mathcal{T}$ readings in each period. Due to the index-bit-encoding, the size of this set will be reduced from $\mathcal{T}$ to $|{R'}_i{}^p|$. Therefore, our mechanism has at most $O(|{R'}_i{}^p|^2)$ as a computation complexity at the sensor and it will save at most $(2 \times |{R'}_i{}^p|)$ readings at each period in its memory. These complexities are suitable for the case of sensor node since the collected readings are usually redundant thus, makes $|{R'}_i{}^p|$ small even in the worst case. This fact is showed clearly in both simulation (see Fig. 6) and experiments (see Fig. 11) where the data collected by each sensor in each period is significantly reduced. On the other hand, the complexity of our mechanism at the CH level is dependent on the dimensions of the matrix $\mathcal{M}^p$ which is $\mathcal{T} \times \mathcal{T}$ in the worst case. Thus, after enumerating and comparing every successive reading sets based on the T-test, the complexity of our mechanism will be in order of $O(\mathcal{T}^2)$. Such complexity can be more reduced in case we adapted the scaling down strategy proposed for large period size (cf. section IV.C) or minimized the value of $\mathcal{T}$.

VII. Conclusion and Future Work

Wireless Sensor Networks is one of fields of technology that is and will continue to be a great tool to better monitor and understand natural and artificial environments around us, and avoid the risk rising from them and adapt to changes in them. That is why proposing new data analysis techniques to handle big data in WSN is and will remain a major concern of researchers. In this paper, we have proposed a Zoom-In Zoom-Out (ZIZO) mechanism in order to minimize the data transmission in WSN to extend its lifetime. ZIZO is based on the cluster network architecture and works on the two levels of a WSN: a low complexity, energy efficient data compression technique called Index-Bit-Encoding at the level of the sensor, and a sampling frequency adaptation technique based on statistical similarity study called Sampling Rate Adjustment at the level of the CH. Through both simulations and experiments, we evaluated the performance of our mechanism in terms of minimizing data transmission and energy consumption while comparing them to other techniques, and showed that we can achieve a compression ratio of down to 5%, and an energy consumption ratio of down to 8%.

In terms of future work, ZIZO can be improved in several direction. First, we seek to improve the compression ratio of IBE. Second, at the CH, we aim to reduce the difference between the original set of readings and the decompressed set of readings by applying different filters on the latter, increasing the level of accuracy without sacrificing compression ratio. Third, we aim to use the data collected by the sensors at the sink, in order to implement dynamic clustering, i.e. change the distribution of sensors to the clusters, taking into account factors such as battery levels, and implement a scheduling strategy in order to switch correlated nodes into sleep/active modes.

Acknowledgment

The research leading to these results has received funding from the AEI/FEDER TWINS project (TEC2017-90097-R), and from the Catalan Institution for Research and Advanced Studies (ICREA).

References

[1] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of wireless sensor networks: An up-to-date survey," *Applied System Innovation*, vol. 3, no. 1, p. 14, 2020.

[2] R. A. Khan and A.-S. K. Pathan, "The state-of-the-art wireless body area sensor networks: A survey," *International Journal of Distributed Sensor Networks*, vol. 14, no. 4, p. 23, 2018.

[3] S. Tarannum, "Energy conservation challenges in wireless sensor network: A comprehensive study," *Wireless Sensor Network*, vol. 2, pp. 483–491, 01 2010.

[4] E. Llobet, M. Carlos-Mancilla, E. López-Mellado, and M. Siller, "Wireless sensor networks formation: Approaches and techniques," *Journal of Sensors*, vol. 2016, 2016.

[5] D. Rafique and L. Velasco, "Machine learning for network automation: overview, architecture, and applications [invited tutorial]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D126–D143, 2018.

[6] L. Velasco, B. Shariati, F. Boitier, P. Layec, and M. Ruiz, "Learning life cycle to speed up autonomic optical transmission and networking adoption," *J. Opt. Commun. Netw.*, vol. 11, no. 5, pp. 226–237, May 2019.

[7] M. Ruiz, F. Tabatabaeimehr, and L. Velasco, "Knowledge management in optical networks: architecture, methods, and use cases [invited tutorial]," *J. Opt. Commun. Netw.*, vol. 12, no. 1, pp. A70–A81, Jan 2020.

[8] Y. Luo, Y. Duan, W. Li, P. Pace, and G. Fortino, "Workshop networks integration using mobile intelligence in smart factories," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 68–75, 2018.

[9] ——, "A novel mobile and hierarchical data transmission architecture for smart factories," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3534–3546, 2018.

[10] L. Velasco, A. C. Piat, O. Gonzlez, A. Lord, A. Napoli, P. Layec, D. Rafique, A. D'Errico, D. King, M. Ruiz, F. Cugini, and R. Casellas, "Monitoring and data analytics for optical networking: Benefits, architectures, and use cases," *IEEE Network*, vol. 33, no. 6, pp. 100–108, 2019.

[11] R. P. Narayanan, S. tv, and V. Vineeth, "Survey on motes used in wireless sensor networks: Performance and parametric analysis," *Wireless Sensor Network*, vol. 8, pp. 67–76, 04 2016.

[12] S. Pushpalatha and K. Shivaprakasha, "Energy-efficient communication using data aggregation and data compression techniques in wireless sensor networks: A survey," in *Advances in Communication, Signal Processing, VLSI, and Embedded Systems*. Springer, 2020, pp. 161–179.

[13] H. Fu, Y. Liu, Z. Dong, and Y. Wu, "A data clustering algorithm for detecting selective forwarding attack in cluster-based wireless sensor networks," *Sensors*, vol. 20, no. 1, p. 23, 2020.

[14] F. Chen, S. Xu, Y. Zhao, and H. Zhang, "An adaptive genetic algorithm of adjusting sensor acquisition frequency," *Sensors*, vol. 20, no. 4, p. 990, 2020.

[15] J. Liu, F. Chen, J. Yan, and D. Wang, "Cbn-vae: A data compression model with efficient convolutional structure for wireless sensor networks," *Sensors*, vol. 19, no. 16, p. 3445, 2019.

[16] S. Chen, S. Zhang, X. Zheng, and X. Ruan, "Layered adaptive compression design for efficient data collection in industrial wireless sensor networks," *Journal of Network and Computer Applications*, vol. 129, pp. 37–45, 2019.

[17] S. Jancy and C. Jayakumar, "Sequence statistical code based data compression algorithm for wireless sensor network," *Wireless Personal Communications*, vol. 106, no. 3, pp. 971–985, 2019.

[18] J. Uthayakumar, T. Vengattaraman, and P. Dhavachelvan, "A new lossless neighborhood indexing sequence (nis) algorithm for data compression in wireless sensor networks," *Ad Hoc Networks*, vol. 83, pp. 149–157, 2019.

[19] S. Wang, "Multimedia data compression storage of sensor network based on improved huffman coding algorithm in cloud," *Multimedia Tools and Applications*, pp. 1–14, 2019.

[20] Y. Liang and Y. Li, "An efficient and robust data compression algorithm in wireless sensor networks," *IEEE Communications Letters*, vol. 18, no. 3, pp. 439–442, 2014.

[21] H. Harb and A. Makhoul, "Energy-efficient sensor data collection approach for industrial process monitoring," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 661–672, 2017.

[22] J. Yang, S. Tilak, and T. S. Rosing, "An interactive context-aware power management technique for optimizing sensor network lifetime." in *SENSORNETS*, 2016, pp. 69–76.

[23] M. Başaran, S. Schlupkothen, and G. Ascheid, "Adaptive sampling techniques for autonomous agents in wireless sensor networks," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2019, pp. 1–6.

[24] S. A. Abdul-Wahab, Y. Charabi, S. Osman, K. Yetilmezsoy, and I. I. Osman, "Prediction of optimum sampling rates of air quality monitoring stations using hierarchical fuzzy logic control system," *Atmospheric Pollution Research*, vol. 10, no. 6, pp. 1931–1943, 2019.

[25] Y. Rao, G. Zhao, W. Wang, J. Zhang, Z. Jiang, and R. Wang, "Adaptive data acquisition with energy efficiency and critical-sensing guarantee for wireless sensor networks," *Sensors*, vol. 19, no. 12, p. 2654, 2019.

[26] J. M. Bahi, A. Makhoul, and M. Medlej, "A two tiers data aggregation scheme for periodic sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 21, no. 1-2, pp. 77–100, 2014.

[27] K. N. Qureshi, M. U. Bashir, J. Lloret, and A. Leon, "Optimized cluster-based dynamic energy-aware routing protocol for wireless sensor networks in agriculture precision," *Journal of Sensors*, vol. 2020, 2020.

[28] D. K. Kotary and S. J. Nanda, "Distributed robust data clustering in wireless sensor networks using diffusion moth flame optimization," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103342, 2020.

[29] G. P. Agbulu, G. J. R. Kumar, and A. V. Juliet, "A lifetime-enhancing cooperative data gathering and relaying algorithm for cluster-based wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 16, no. 2, p. 1550147719900111, 2020.

[30] Y. Alghamdi and M. Abdullah, "Improving k-means algorithm by grid-density clustering for distributed wsn data stream," *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, vol. 9, no. 11, pp. 583–588, 2018.

[31] D. Wohwe Sambo, B. O. Yenke, A. Förster, and P. Dayang, "Optimized clustering algorithms for large wireless sensor networks: A review," *Sensors*, vol. 19, no. 2, p. 322, 2019.

[32] X. Fu, P. Pace, G. Aloi, L. Yang, and G. Fortino, "Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm," *Computer Networks*, p. 107327, 2020.

[33] S. Madden *et al.*, "Intel lab data," *http://db.csail.mit.edu/labdata/labdata.html*, 2004.

[34] W. B. Heinzelman, "Application-specific protocol architectures for wireless networks," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[35] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd annual Hawaii international conference on system sciences*. IEEE, 2000, pp. 10–pp.

[36] advanticsys, "Sg1000 hardware," *http://www.advanticsys.com/wiki/index.php?title=SG1000*, 2012.

[37] D. Gay, P. Levis, D. Culler, and E. Brewer, "nesc language manual," *https://github.com/tinyos/nesc/blob/master/doc/ref.pdf?raw=true*, 2009.

[38] J. Griessen, "Tinyos," *http://tinyos-help.10906.n7.nabble.com/Energy-consumption-on-Telosb-td22083.html*, 2012.

[39] X. Fu, G. Fortino, W. Li, P. Pace, and Y. Yang, "Wsns-assisted opportunistic network for low-latency message forwarding in sparse settings," *Future Generation Computer Systems*, vol. 91, pp. 223–237, 2019.

[40] X. Fu, G. Fortino, P. Pace, G. Aloi, and W. Li, "Environment-fusion multipath routing protocol for wireless sensor networks," *Information Fusion*, vol. 53, pp. 4–19, 2020.