We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Performance Analysis of a Compression Scheme for Highly Dense Cluster-Based Wireless Sensor Network

Sofiane Moad, Mario E. Rivero-Angeles, Nizar Bouabdallah,
Bruno Sericola and Yassine Hadjadj Aoul

Additional information is available at the end of the chapter

1. Introduction

Wireless Sensor Networks (WSNs) consist of spatially distributed autonomous devices, called sensors, that communicate in a wireless manner. Sensors cooperate together to monitor physical or environmental conditions such as temperature, sound, vibration or pressure [1]. This technology has been originally developed for military applications such as battlefield surveillance [2]. Recently, WSNs have been deployed in many other civilian application areas. It includes environment applications such as fire detection in forest monitoring and health-care applications like monitoring the patient status.

Sensor nodes are expected to be of tiny size. Therefore, the size of every sensor's components, such as the power source, processing and data storing memory, are expected to be also very limited. In addition of the physical characteristics, a large number of sensors are often deployed in hostile environments, where the human intervention is difficult if not impossible, for example inside a volcano. Hence, in these networks it is not practical to perform maintenance operations, such as changing batteries on deployed sensor nodes. This requires sensors to be able to self-organize, self-configure and should optimize the energy consumption to maximize the network's lifetime. The network lifetime in WSNs refers to the period of time from the deployment of the sensor nodes to the instant when the network is considered unusable [3].

In terms of energy consumption, sensors consume energy for three main reasons: data sensing, data processing and wireless data communicating. Wireless communication refers to data transmission and reception. Among these three operations, the most power-consuming task is data transmission. Approximatively 80% of power consumed in each sensor node

is used to transmit data [4]. One field of research, aimed at extending the lifetime for WSNs, is data compression. In general, by applying a suitable compression scheme, power consumption can be reduced during the transmission and processing stages thus extending network lifetime. Also, by reducing the data packet size, less bandwidth is required to send and receive data [4]. In addition, to further enhance energy efficiency, cluster-based communication schemes are widely used in WSNs [5]. In view of this, we propose to take advantage of both data compression and clustering in order to further reduce energy consumption in the network. Indeed, the compression and clustering strategies are not supposed to work independently but rather the network design has to consider both simultaneously. Therefore, by enabling both compression and clustering schemes in WSNs, energy consumption would be greatly enhanced.

In this chapter, we propose a complete analysis of a new Compression Cluster-based scheme in a Spatially-Correlated Region (CC_SCR) for event-driven applications in WSNs. As WSNs are typically densely-deployed over a sensor field [6], sensor nodes are typically very close to each other. Contrary to continuous monitoring applications, in Event Detection-Driven (EDD) applications, active nodes are concentrated in a relatively small area. Therefore, readings from these nodes are expected to be quite similar. Building on this, we propose a clustering scheme that exploits the spatial correlation of sensed data among the nodes to reduce the size of the data packets that will be sent. Specifically, in the proposed scheme, Cluster Members (CMs) send only the differences between their readings and a reference data which corresponds to the value sensed by the selected Cluster Head (CH). As such, one of the proposal's main issue is to select as CH the node that reduce the average packet size in the cluster. Note that, in this chapter, we complement our previous work published in [7] with several simulation results and analytical analysis.

The main contributions to this chapter are:

1. The CC_SCR compression cluster-based protocol for WSNs is proposed. It exploits the spatial correlation of the sensed data in order to reduce energy consumption.

2. An analytical energy consumption model for comparison to both classical and single hop schemes is developed.

3. The CC_SCR is implemented in TinyOS [8] for simulation analysis, in order to prove the potential benefits of CC_SCR in future applications of real WSNs deployments.

The remainder of this chapter is organized as follows. Section 2 presents a background of research related to this work, while Section 3 exposes the network model. Section 4 shows the analytical results regarding the energy consumption and network lifetime, while Section 5 shows the simulation results. Finally, the chapter concludes in Section 6 with a summary of the main advantages of the proposed scheme.

2. Related work

We review some of the related works regarding the compression and the clustering schemes in Subsections 2.1 and 2.2, respectively.

2.1. Compression schemes

In the literature, there has been an increased interest in studying compression algorithms for WSNs. On the other hand, many of these compression algorithms have been proposed for classical networks, which are not suitable to be deployed in the WSNs context [9, 10]. The main reason is the limited memory size of sensor nodes, for example, the size of bzip2 is $219KB$ and the size of LZO is $220KB$. Another reason is the limited processor speed of sensor nodes which is around $4 - 8MHz$. Thus, embedding classical data compression schemes in these tiny nodes is very difficult and it is necessary to design a low-complexity and small-size data-compression algorithm for sensors. We review in the following some of these compression techniques. A more detailed description of compression methods can be found in [11].

One compression algorithm for WSNs is the coding-by-ordering data-compression technique [12]. In this algorithm, when data is combined at an aggregation node, some of these data are implicitly transmitted. The main idea behind this technique is to replace the data transmission of certain nodes by the order in which the aggregated packets are placed by the aggregator. For example, consider five nodes (n_1, n_2, n_3, n_4, n_5) that send data to their aggregator node, n_a and suppose that data value of each node can be any integer from a range of 0 to 23. When the aggregator node receives a value from each node n_1, n_2, n_3, n_4, n_5, the order of transmission of first four nodes n_1, n_2, n_3, n_4 determines the value of n_5 implicitly. Indeed, there are $4! = 24$ possible ways of ordering these data packets.

The pipelined in-network compression algorithm is discussed in [13]. The main idea behind this technique is to combine data in order to make them smaller than the original size. After the aggregator collects data from differents nodes, it is stored for a certain amount of time. Data packets are then combined into one packet to minimize data transmission. For example, consider that each data packet has the following form: < *data value*, *node ID*, *timestamp* >. Then, the compressed data packet has the following form: < *shared prefix*, *suffix list*, *node ID list*, *timestamp list*>. The *shared prefix* field, i.e. the most significant bits, is the same for all the measured values. The suffix list field expresses the list of measured values excluding the *shared prefix* part. The *node ID list* is the list of node identifiers and the *timestamp list* is the list of timestamps. One advantage of this simple compression scheme is that the shared prefix system can be used for both *node ID* and *timestamp* fields. By doing so, more data compression can be achieved.

The distributed compression scheme proposed in [14] uses a side information to encode a source information. For example, if there are two data sources: X and Y, which are correlated and chosen from a discrete alphabet, then X can be compressed at the theoretical rate of its conditional entropy $H(X|Y)$. The receiving node maintains the cosets and can decode X knowing Y's codevector, and with partial information from the source X.

The main advantage of the *CC_SCR* algorithm, compared to those previously mentioned is that *CC_SCR* takes into account the physical characteristics of the sensed data in order to compress data and also considers clustering communication in order to reduce energy consumption.

2.2. Clustering schemes

In addition to the compression techniques, there has been an increased interest in studying energy efficient clustering algorithms and extensive clustering algorithms have been proposed for WSNs. Hereafter, we briefly review the most relevant energy efficient clustering algorithms. For more details, the reader can review [15], [16], [5], and [17].

Hybrid Energy-Efficient Distributed clustering (HEED) [18] protocol operates in two main phases: the set-up phase where clusters are formed and the steady phase where the sensor nodes transmit their data using the Time Division Medium Access (TDMA) frames. HEED set-up phase operates in three sub-phases. The first sub-phase is the initializing. Nodes exchange *hello* messages to discover their neighborhoods. The second sub-phase consists of a competition process. The third sub-phase is the finalizing and it allows nodes to join their corresponding CH based on the connectivity degree.

Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [19] steady phase consists of a formation of chains instead of clusters. In the chain formation, the Base Station (BS) and sensor nodes are connected via a chain using a greedy algorithm. One of the nodes, in the chain, is selected by turns to represent the head. In data gathering phase, each node delivers the sensing data to the nearest neighbor node until the data reach the head node which aggregates and delivers the sensing data to the BS.

In [20], the authors proposed an Energy Efficient Clustering Scheme (EECS) protocol. In this protocol, CH candidates compete for the ability to elevate to a CH with a certain probability. This competition involves candidates broadcasting their residual energy to neighboring candidates. If a given node does not find a node with more residual energy, it becomes a CH.

The main difference between the aforementioned clustering algorithms and the CC_SCR algorithm is that nodes use the physical characteristic of the sensed data to elects CHs. The benefits of our proposed scheme is that nodes use the compression scheme to reduce the energy consumption.

In this chapter, we combine the benefits of using both the clustering and the compression techniques to reduce the energy consumption in the network. Indeed our proposal scheme takes into account the characteristics of the physical surveilled event and also takes advantage of the energy unconstrained of the Base Station (BS) which participates into the CH selection. Specifically, our proposal scheme takes advantage of the fact that the nodes that sense a certain event are usually very close to each other (which entails a high correlation between sensed data) in order to reduce the size of data packets communicated through the network. The BS then selects an efficient CH that minimize the data transmission over the network. To the best of knowledge this is the first clustering protocol, that takes into consideration the physical characteristics of the environment to elect energy efficient clusters and therefore implement the compression scheme to reduce the energy consumption.

3. Network model

We consider an event-driven WSN consisting of N sensors deployed over a vast field as in Figure 1. We denote the *i-th* sensor node as n_i and the corresponding sensor node set as $\{n_1,$

$n_2,...,n_N\}$. Some assumptions concerning the sensor nodes and the underlying network model are now presented:

- Nodes are uniformly distributed in an $A \times A$ area with (x, y) coordinates. Nodes are homogenous and all have the same capabilities. A unique identifier ID is assigned to each node.

- Sensor nodes and the sink node are all stationary after deployment.

- Nodes have two power controls to vary transmission power, which depends on the distance to the receiver [21]. Each node, n_i, can reach any other node with a transmission range, R_c. The sink can be reached with a transmission range, $R_t > R_c$.

- CHs use the average operation as the aggregation technique in order to eliminate data redundancy.

We consider event detection driven wireless sensor applications. The center of the event is located in a random uniformly-distributed point with coordinates (x_{event}, y_{event}) within the network's area. The range of the event, i.e., the area range where sensors can detect the event, is R_event meters, where $R\_event \in [1, A]$. We also suppose that an event has a duration of T_event seconds. In addition, only sensor nodes within R_event range are considered as active nodes in the network and they are the only nodes performing as the source of the detected event. The rest of the nodes in the system are not considered in our analysis as they do not participate in data reporting. Additionally, in our model, only one event can be active inside the system area and the data value C at the center of the event is constant, i.e., the stationary model in which the measured data do not change during the T_event seconds that the event is active.
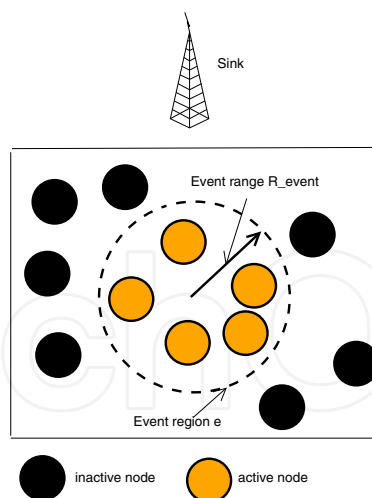

Figure 1. Event-driven application in WSNs.

A cluster-based WSN is considered where only one CH is elected for each event. The clustering process is triggered whenever an event is sensed by the nodes inside the event area.

The spatial correlation of the data from the different active nodes has been modeled on previous works in the area according to the following models:

1. Diffusion propriety [22].

2. Data is jointly Gaussian, with the correlation being a function of the distance [23].

3. Data is a function for their joint entropy [24].

4. Correlation is calculated from realistic environmental monitoring and testbeds [25].

In this chapter, we use diffusion property to model spatially-correlated data [26]. The model considered in this chapter is the same as in [22] in which the data reading at a distance d from the center of the event is $D = C/(d+1)^\alpha$, where C is a constant representing the value at the center of the event, and α is the diffusion parameter, which depends on the particular environment and phenomenon surveyed, e.g., for light $\alpha = 2$, heat $= \alpha \simeq 1$.

Figure 2 shows the data reading using the aforementioned model, with different values for α, and $C = 250$. On one hand, when $\alpha \geq 1$, we observe a relatively big difference between the value sensed at the center of the event and the values observed at distance d far away from the center of the event. On the other hand, when $\alpha < 1$ ($\alpha = 0.1, 0.01$ and 0.001), the data readings away from the center of the event are very similar. In our study, we are interested, specifically, in the types of event where data values are highly correlated.
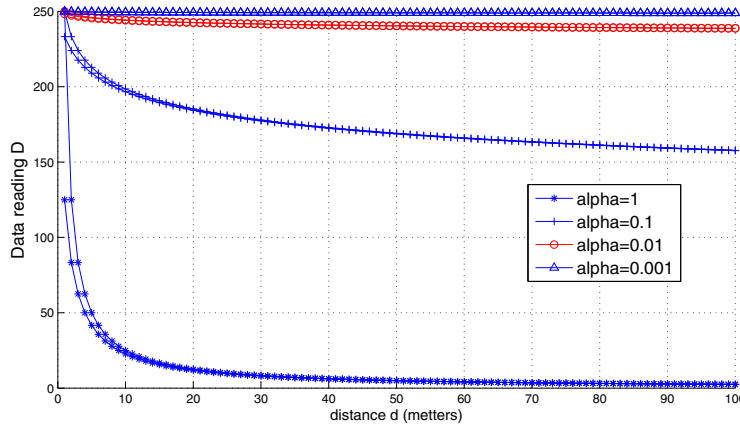
Figure 2. Variation in data reading with distance from the center of the event.

We use Henizelman's energy consumption model [27]. Specifically, the energy consumed to transmit a message at a distance d is given by

$$E_{tx}(sz, d) = \begin{cases} sz \times E_{elec} + sz \times E_{fs} \times d^2, & \text{if } d \leq d_0. \\ sz \times E_{elec} + sz \times E_{mp} \times d^4, & \text{otherwise.} \end{cases} \tag{1}$$

where sz is the data packet size in bits, E_{elec} is the energy consumed due to the transmitter/receiver circuitry, E_{mp} is the energy consumed by the transmitter amplifier and $d_0 = \sqrt{E_{fs}/E_{mp}}$ is the distance threshold between the transmitter and the receiver over which the multi-path fading channel model is used. The energy consumed to receive a message is $E_{rx}(sz) = sz \times E_{elec}$.

3.1. Classical clustering protocol

A classical clustering process is composed of two phases: the set-up phase and the steady-state phase, as depicted in Figure 3. When an event occurs in a random (uniformly-distributed) point of the network, nodes inside the event area wake up and start the clustering process. At the beginning of this phase, active nodes compete among each other to become a CH. Specifically, active nodes transmit their control packet to the sink node according to the specified random medium access protocol. In this chapter, CSMA control protocol is used, which is the default MAC protocol for the Mica platform. The control packet only comprises the node's ID and no data is transmitted at this point. The first node that successfully transmits this packet becomes the CH. All nodes involved in event-reporting immediately send their signaling message to the sink node. Therefore, the sink node selects the first node that successfully transmits a signaling message and then it broadcasts a signaling message over the network for a CH notification. Thus, the rest of the nodes become CMs. In the steady phase, CMs send their data in a scheduled fashion using a Time Division Multiple Access (TDMA) protocol. Note that the CH assigns slots to its CMs to accomplish the successful data transmission. Then, the CH aggregates the data values received from its CMs with its own, and sends the resulting data to the sink node.

3.2. Proposed compression clustering protocol

The proposed clustering Compression Cluster-based scheme in Spatially-Correlated Regions (CC_SCR), process is also composed of the same two phases, namely: the set-up phase and the steady-state phase. As in classical protocol, the set-up phase happens whenever an event occurs in a region of the network. However, in CC_SCR, the active nodes send their first measured data value to the sink node, i.e., they no longer send just their control packet. Instead, active nodes send a data packet. The reason for this is that this sensed data is used in CH selection procedure. Indeed, this entails an extra energy consumption at the set-up phase, compared to classical protocol. However, this first data transmission allows important energy savings in the steady-state phase.

It is important to note that CC_SCR is best suited for environments where event conditions are fairly stable during of the duration of the event. This is due to the fact that the CH is chosen according to the first sensed data. Hence, if event conditions suffer a high variation, the originally-selected CH may no longer render acceptable energy savings. An example of such an application is a fire surveillance forest, in which, when a fire occurs in a region, the temperature remains stationary for the duration of the fire in the region. Another example includes target-tracking. In this kind of application, the target is the source of the measured data at sensor nodes, such as light or temperature. Here, the measured data remains the same whenever the target stays in the same place and hence the sensor nodes sense the same measured data during the presence of the target. Next, we describe the set-up and the steady-state phases of the proposed algorithm.

- In the set-up phase, after receiving the first data packets from all the active nodes, the sink node calculates the difference between the data from node n_i and those from node n_j, $i \neq j$. Next, these differences are summed over. We call this sum of the difference between data values S_i. Then, the sink selects as CH the node which minimizes the total difference,

calculated value S_i, between each node n_i and node n_j, $i \neq j$. Finally, the sink broadcasts a control message to the active nodes in order to notify the node selected as CH. Therefore, the rest of the nodes consider themselves as CMs. Note that there is no need for the CMs to send any extra packets since the sink already knows which nodes are active.

- In the steady-state phase, the CMs send the difference between their sensed data and the CH's data value, which corresponds to a compressed value, called Δ_i, rather than the complete data packet, $value\_CM_i$. Therefore, $\Delta_i = |value\_CM_i - value\_CH|$ represents the difference between the i-th CM's data value $value\_CM_i$, and the corresponding CH data value $value\_CH$. In order to perform this compression, the CH sends its complete sample data value to the CMs at the beginning of each event occurrence. Therefore, CMs send only the Δ_i to the CH. The main advantage of the proposed scheme is that the S_i calculation is made at the sink node, which is not energy or memory-constrained.
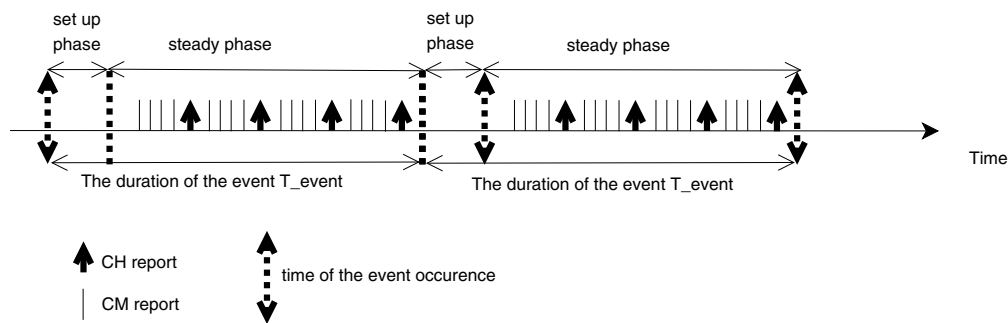
Figure 3. System operation.

3.2.1. Example

To illustrate the protocol's operation, let us consider the following example as presented in Figure 4(a) (Figure 4(b) shows the case of a classical scheme). We consider five active nodes n_1, n_2, n_3, n_4 and n_5 in the event region e which covers a region of range R_event as in Figure 1. In this example, we consider temperature as the sensed measurement value. Nodes n_1, n_2, n_3, n_4 and n_5 sense the value of $20°$, $22°$, $19°$, $20°$ and $15°$, respectively, and they send the values to the sink. When the sink receives the data values, it calculates the S_i value for each node n_i. The node which has the smallest S_i is considered as the CH.

The following calculation is done at the sink. For node n_1:
$|20 - 22| = 2$
$|20 - 19| = 1$
$|20 - 20| = 0$
$|20 - 15| = 5$
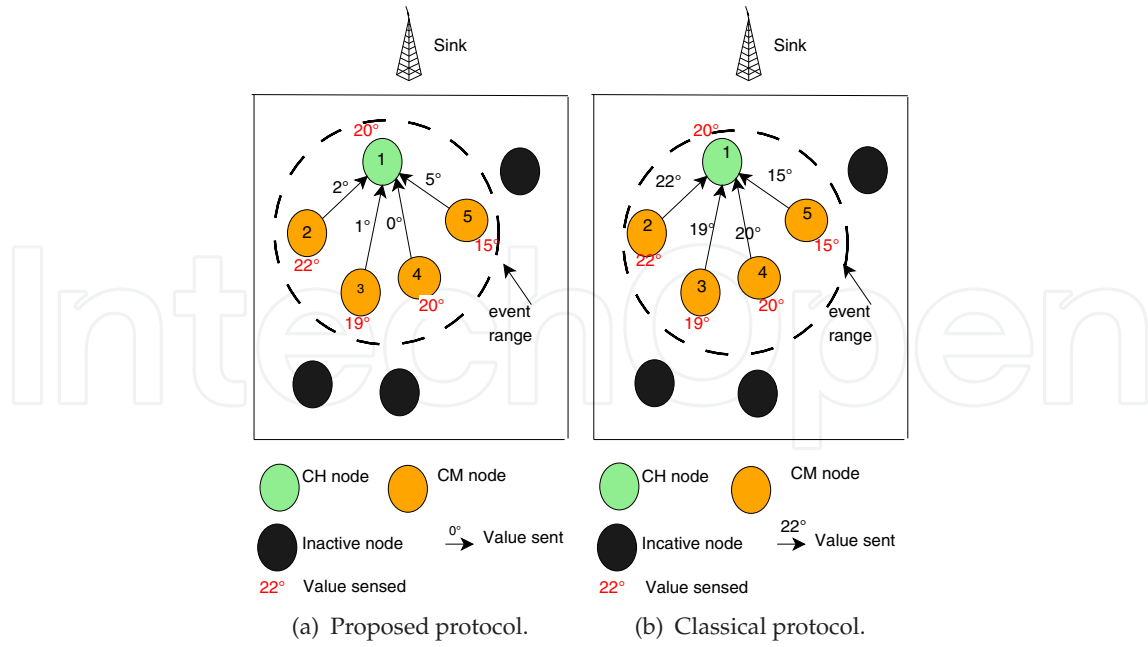The sink node calculates $S_1 = |20 - 22| + |20 - 19| + |20 - 20| + |20 - 15| = 8$
For node n_2:
$|22 - 20| = 2$
$|22 - 19| = 3$
$|22 - 20| = 2$
$|22 - 15| = 7$
The sink node calculates $S_2 = |22 - 20| + |22 - 19| + |22 - 20| + |22 - 15| = 14$

Figure 4. Example of the system operation.

For node n_3:

$$|19 - 20| = 1$$

$$|19 - 22| = 3$$

$$|19 - 20| = 1$$

$$|19 - 15| = 4$$

The sink node calculates $S_3 = |19 - 20| + |19 - 22| + |19 - 20| + |19 - 15| = 9$

The calculation for node n_4 is the same as node n_1.

For node n_5:

$$|15 - 20| = 5$$

$$|15 - 22| = 7$$

$$|15 - 19| = 4$$

$$|15 - 20| = 5$$

The sink calculates $S_5 = |15 - 20| + |15 - 22| + |15 - 19| + |15 - 20| = 21$.

Finally, it selects either node n_4 or node n_1 as a CH. Both nodes minimize the total difference value measured. The other nodes become CMs. During the steady phase, CM nodes send their Δ_i value to the CH rather than their complete value. In this example, n_3 sends the sample value of 2 rather than the complete sample value of 22. Note that the compression data sent in our scheme involves sending less coded bits compared to a complete data that is sent in the classical scheme. Therefore, considerable energy saving is achieved in our scheme, as can be seen in Sections 4 and 5. Note that this election algorithm can be achieved at nodes in a distributed manner. Distributed scheme here means that nodes first exchange data and then the calculations specified at the sink node will run at the level of individual nodes. However,

in this scheme nodes will receive a considerable amount of data, which may complicate the election process as nodes have a limited local memory.

4. Analytical results

In this section, the mathematical model of classical, single hop and CC_SCR protocols are described. The total energy consumed in the network, E_{total}, for the duration of an event, can be calculated as follows:

$$E_{total} = E_{competing} + E_{reporting} \qquad (2)$$

where $E_{competing}$ is the energy consumed during the cluster formation phase and $E_{reporting}$ is the energy consumed during the steady-state phase. We calculate in the following $\mathbb{E}[E_{total}]$ as the expected energy consumed through the network for a single hop protocol and for both the classical and CC_SCR protocols.

$$\mathbb{E}[E_{total}] = \mathbb{E}[E_{competing}] + \mathbb{E}[E_{reporting}] \qquad (3)$$

4.1. Classical protocol

For classical protocol, where no data compression is enabled, we first calculate $\mathbb{E}[E_{competing}]$. The energy consumed in the cluster formation phase is due to the signaling packet transmission of the active nodes in the event area directly to the sink plus the reception of the signaling packet from the sink to the active nodes, then:

$$\mathbb{E}[E_{competing}] = m \times [E_{tx}(S, R_t) + E_{rx}(S)] \qquad (4)$$

where $m = N\pi R\_event^2/A^2$ is the expected number of active nodes in the range R_event when the disk is totally included in the area $A \times A$ and N is the total number of nodes in the network. $S = 24$ bits is the size of signaling message, and

- $m \times E_{tx}(S, R_t)$ is the energy consumed to send m competing messages to the sink.
- $m \times E_{rx}(S)$ is the energy consumed by the resulting competing messages sent from the sink through the network.

$$\begin{aligned}\mathbb{E}[E_{reporting}] = Nb_r \times [&E_{tx}(S, R_c) + (m-1) \times E_{rx}(S) + (m-1) \times E_{tx}(fixe, R_c) \\ &+ (m-1) \times E_{rx}(fixe) + E_{DA} \times fixe + E_{tx}(fixe, R_t)]\end{aligned} \qquad (5)$$

where $fixe$ bits is the size of the full data packet, Nb_r is the number of packets sent during the steady phase, and

- $E_{tx}(S, R_c)$ is the energy consumed from a signaling message sent by the CH to its CMs in order to send their data.
- $(m-1) \times E_{rx}(S)$ is the energy consumed by CMs to receive the signaling message.
- $(m-1) \times E_{tx}(fixe, R_c)$ is the energy consumed by CMs to send data to the CH.
- $(m-1) \times E_{rx}(fixe)$ is the energy consumed by the CH to receive data sent by the CMs.

- $E_{DA} \times fixe$ is the energy consumed by the CH due to data aggregation.
- $E_{tx}(fixe, R_t)$ is the energy consumed by the CH to send the aggregated data to the sink.

In the simulation, we take $fixe = 32$ and $Nb_r = 29$.

4.2. Single hop protocol

In the single hop protocol, there is no energy consumed during the set-up phase. Nodes start sending data packets directly to the sink, then:

$$\mathbb{E}[E_{total}] = Nb_r \times m \times E_{tx}(fixe, R_t) \tag{6}$$

where

- Nb_r is the number of reports sent to the sink during T_event sec.
- m is the number of nodes involved in data reporting.
- $E_{tx}(fixe, R_t)$ is the energy required to send a full data packet to the sink.

The interest of analyzing this case is to have an insight into the benefits of clustering schemes in event-driven WSNs.

4.3. CC_SCR

We now consider the case where the CC_SCR strategy is enabled. It is to be noted that the proposed scheme, CC_SCR, behaves in the same manner in the cluster formation phase as classical protocol, with the important difference being that nodes transmit the data packet instead of the signaling packet, then:

$$\mathbb{E}[E_{competing}] = m \times [E_{tx}(fixe, R_t) + E_{rx}(S)] \tag{7}$$

where

- $m \times E_{tx}(fixe, R_t)$ is the energy consumed to send m data packets to the sink.
- $m \times E_{rx}(S)$ is the energy consumed by the resulting compete message sent from the sink through the network.

In the steady-state phase, energy consumption is found as follows:

$$\begin{aligned} \mathbb{E}[E_{reporting}] = &E_{tx}(fixe, R_c) + (m-1) \times E_{rx}(fixe) + Nb_r \times [E_{tx}(S, R_c) \\ &+ (m-1) \times E_{rx}(S) + (m-1) \times E_{tx}(S + log_2(\mathbb{E}[\Delta_i]), R_c) \\ &+ (m-1) \times E_{rx}(S + log_2(\mathbb{E}[\Delta_i])) + fixe \times E_{DA} + E_{tx}(fixe, R_t)] \end{aligned} \tag{8}$$

where

- $E_{tx}(fixe, R_c)$ is the energy consumed to send CH data packets to the CMs.

- $(m-1) \times E_{rx}(fixe)$ is the energy consumed by CMs in order to receive data sent by the CH.

- $E_{tx}(S, R_c)$ is the energy consumed from a signaling message sent by the CH to its CMs in order to send their data.

- $(m-1) \times E_{rx}(S)$ is the energy consumed by CMs in order to receive the signaling message.

- $(m-1) \times E_{tx}(S + log_2(E[\Delta_i]), R_c)$ is the energy consumed by the CMs to send the compressed data to the CH.

- $(m-1) \times E_{rx}(S + log_2(E[\Delta_i]))$ is the energy consumed by the CH to receive the compressed data from the CMs.

- $E_{DA} \times fixe$ is the energy consumed by the CH due to data aggregation.

- $E_{tx}(fixe, R_t)$ is the energy consumed by the CH to send the aggregated data to the sink.

where $\mathbb{E}[\Delta_i]$ is the average data packet size which corresponds to the difference between the CMs' data and the CH's data. It is worth noting that when considering an uniform node distribution with a large N, the node that minimizes the distance in the R_event region will be located at the center of R_event. Therefore, to calculate $\mathbb{E}[\Delta_i]$, let us first calculate the average distance between active nodes and the CH, $\mathbb{E}[d_{toCH}]$. We denote by $\mathcal{D}$ the disk of radius R_event, i.e. $\mathcal{D} = \{(x,y) \mid x^2 + y^2 \leq R\_event^2\}$. Since active nodes are uniformly distributed in $\mathcal{D}$, we have

$$\mathbb{E}[d_{toCH}] = \int \int_{\mathcal{D}} \sqrt{x^2 + y^2} dxdy = \frac{1}{\pi R\_event^2} \int_{r=0}^{r=R\_event} \int_0^{2\pi} r^2 drd\theta = \frac{2R\_event}{3}. \quad (9)$$

We then calculate $\mathbb{E}[\Delta_i]$, the average data difference between the data at the CM and the maximum value at the CH C, considering that the CH is located at the center of the cluster. Indeed, in a highly dense WSN, such the one considered in this work, it is reasonable to consider that the CH is located at the center of the cluster, i.e., very close to the event origin.

$$\mathbb{E}[\Delta_i] = |C - \frac{C}{(\mathbb{E}[d_{toCH}] + 1)^\alpha}| = C \times |(1 - \frac{1}{(\frac{2R\_event}{3} + 1)^\alpha})| \quad (10)$$

Note that the previous model is an approximation of reality, in which an ideal channel is considered, i.e., there is no consideration of packet loss. According to the previous models, Figure 5(a) and 5(b) show the average energy consumed in the network of $N = 1000$ for different values of R_t and R_c, respectively.

In Figure 5(a), we set $R_c = 100\ m$ and $Nb_r = 29$, and in Figure 5(b), we set $R_t = 150\ m$, $Nb_r = 29$. These results show that the CC_SCR strategy is suitable when R_t is less than around 180 meters and R_c is greater than around 50 meters.

Exceeding these thresholds makes the competing process of the proposed protocol very costly in energy due to the full data packet sent to the sink during the set-up phase. Remember that classical protocol only transmits a control packet in this phase. Therefore, CC_SCR has the highest energy consumption when the distance from the cluster to the sink becomes considerable.

In addition, Figure 5(a) demonstrates that the single hop scheme achieves the greatest energy consumption as 1) its transmissions depend directly on R_t, and 2) it sends the full data packets. Figure 5(b) shows a steady energy consumption for the single hop scheme as it does not use a CH to shorten its transmission range, therefore all nodes use a costly direct-transmission R_t, which dramatically decreases network lifetime.

Figure 5(c) shows the average energy consumed in the network when varying the Nb_r parameter. Here, R_t and R_c are set to $300m$ and $100m$, respectively. The results show that significative energy savings can be achieved when increasing the number of reports sent from the CMs to the CH. Building on from these observations, in Figure 5(c) we observe that the single hop scheme achieves the greatest energy consumption as its transmission depends directly on Nb_r.

The point of intersections, R_t_inter, R_c_inter, and Nb_r_inter of Figure 5(a), 5(b) and 5(c), respectively, are calculated as follows.

$$R_t\_inter = \sqrt[4]{\frac{fixe \times (m \times E_{elec} + E_{mp} \times R_c^4) - Nb_r \times (m-1) \times \Delta \times [2 \times E_{elec} + E_{mp} \times R_c^4] - (S-fixe) \times E_{elec}}{(S - fixe) \times E_{mp}}}$$
(11)

$$R_c\_inter = \sqrt{\frac{m \times fixe \times E_{elec} - m \times (S-fixe) \times E_{elec} - m \times (S-fixe) \times E_{mp} \times R_c^4 - 2 \times Nb_r \times (m-1) \times \Delta \times E_{elec}}{-fixe \times E_{fs} + Nb_r \times (m-1) \times \Delta \times E_{fs}}}$$
(12)

$$Nb_r\_inter = \frac{-m \times (S-fixe) \times E_{elec} - m \times (S-fixe) \times E_{mp} \times R_t^4 + m \times fixe \times E_{elec} + fixe \times E_{mp} \times R_c^4}{2 \times (m-1) \times \Delta \times E_{elec} + \Delta \times E_{mp} \times R_c^4}$$
(13)

where $\Delta = fixe - S - \log_2 \mathbb{E}[\Delta_i]$.

We conclude from Figure 5(a), 5(b) and 5(c) that 1) using a clustering scheme saves more energy than a single hop scheme, and 2) the application depends on R_t (referring to how far the sink node is from the area-sensed field), R_c (referring to the size of the area field), and Nb_r (referring to the number of data updating to the CH). More specifically, we conclude that for a fixed R_c, the higher R_t results a poor performance of CC_SCR concerning energy consumption. However, increasing R_c or Nb_r gives CC_SCR a better performance compared to the classical scheme. In addition, we conclude that both CC_SCR and the classical scheme outperforms the single hop scheme.

In the following, we analyze the network lifetime. Based on [28], the general definition of the average network lifetime, $lifetime$, can be expressed as follows:

$$lifetime = \frac{\mathbb{E}[E_0\_total]}{\mathbb{E}[E_{total}]}$$
(14)

where $\mathbb{E}[E_0\_total] = m \times E_0$ is the average total residual energy in the area of R_event range, E_0 is the initial energy of a node and $\mathbb{E}[E_{total}]$ is the average energy consumed per unit of time (i.e., during T_event seconds).
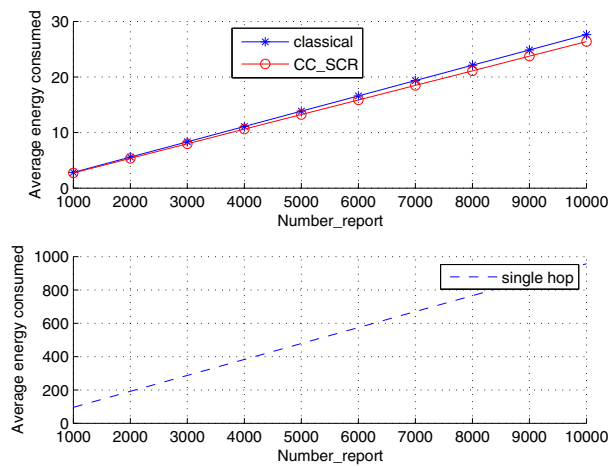
Using the same parameter settings as in the energy consumption analysis described above, Figure 6(a), Figure 6(b) and Figure 6(c) show network lifetime for different values of R_t,
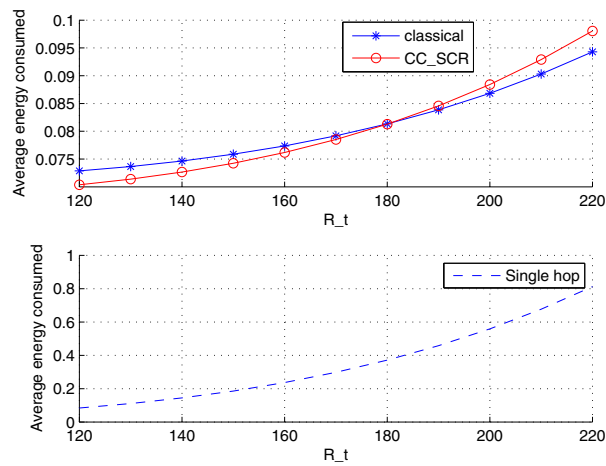
(a) Varying R_t.
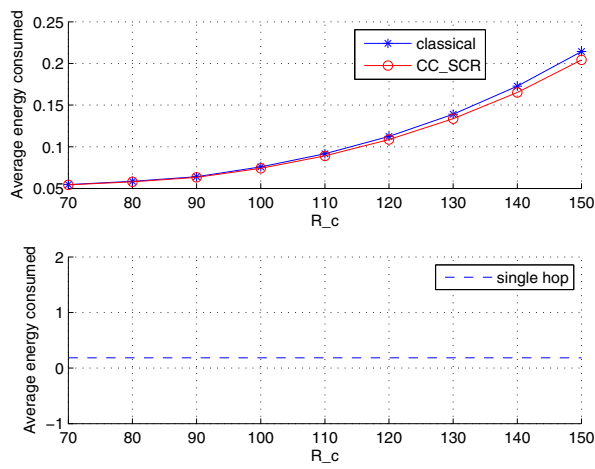
(b) Varying R_c.

(c) Varying Nb_r.

Figure 5. Analytical results of the energy consumption.

R_c and Nb_r, respectively.   From Figure 6(a), we observe that CC_SCR outperforms the classical scheme when R_t is lower than around $180m$. The result of the single hop scheme, in Figure 6(a), shows that network lifetime decreases faster than in CC_SCR and in the classical schemes (Figure 7(a) shows the ratio gain of network lifetime).

From Figure 6(b), we observe that the more increase of R_t, the less network lifetime gain in CC_SCR, compared to the classical scheme. We also notice the steady network lifetime for the single hop scheme, but with a shorter value than for both the proposed and the classical schemes (Figure 7(b) shows the ratio gain of network lifetime).

From Figure 6(c), we observe that the higher the Nb_r, the longer the network lifetime, concerning the CC_SCR strategy compared to the classical scheme. We also observe that the network lifetime decreases faster in the single hop scheme, than compared to both CC_SCR and the classical schemes (Figure 7(c) shows the ratio gain of network lifetime).

5. Simulation results

We use TinyOS [8] as a simulation tool.   The parameters used for this set of results are presented in Table 1.

| Parameter	value
E_{fs}	$10pJ/bit/m^2$
E_{mp}	$0.0013nJ/bit/m^4$
E_{elec}	$50nJ/bit$
E_{DA}	$5nJ/bit$
Signaling packet length S	$24bit$
Data value at the center of the event C	$250°$
Initial energy per node E_0	$10J$
T_event	$200\ seconds$
R_event	$60\ m$
R_c	$100\ m$
R_t	$400\ m$
Area A	$(100 \times 100)m^2$

Table 1. Simulation Parameters.

Figure 8 shows the average energy consumed in the network per unit of time for different concentrations of nodes. In this case, there are twenty simulated events. The results clearly demonstrate that our proposal outperforms the classical scheme. It can be seen that, as the number of nodes in the system increases, also the energy consumption increases. Indeed, when there is a high number of nodes in the network, there is also a high number of nodes that sense the event. Hence, the number of packet transmissions (both control and data packets) is much higher than in the case where just a few nodes are active per event. The main reason for the better performance of the proposed protocol is that only the difference, Δ_i, is transmitted rather than the complete data packet, during the steady-state. Note that this difference between classical and the proposed protocol increases concerning higher densities networks. The rationale behind this is that, in high density networks, nodes are closer to each
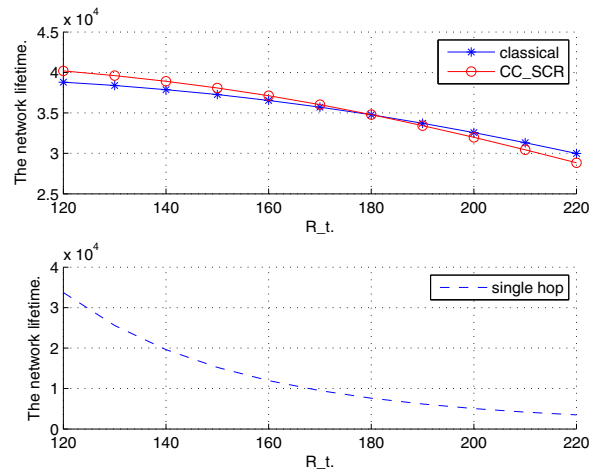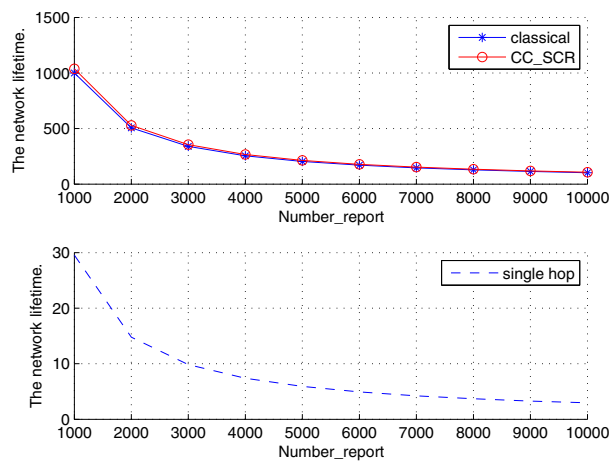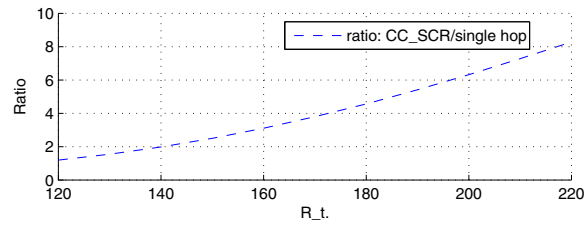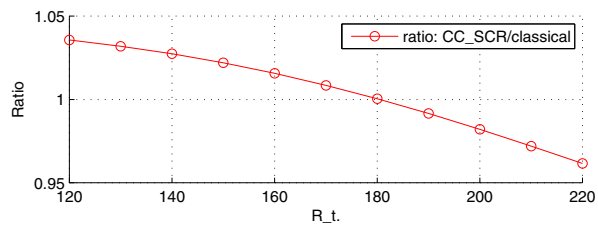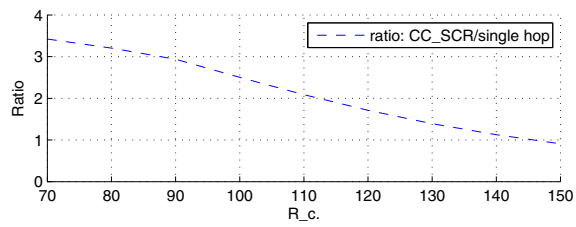
(a) Varying R_t.
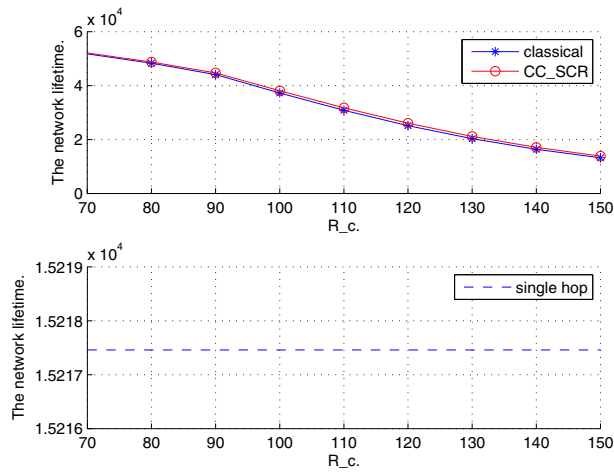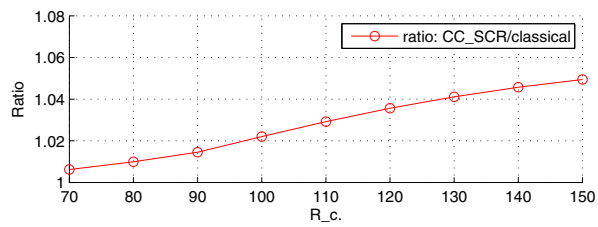

(b) Varying R_c.

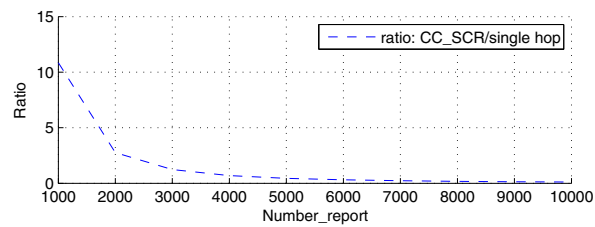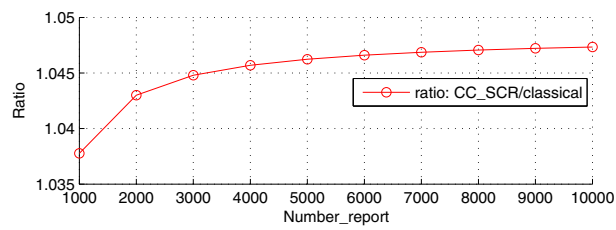(c) Varying Nb_r.

Figure 6. Analytical results of network lifetime.

(a) Varying R_t.

(b) Varying R_c.

(c) Varying Nb_r.

**Figure 7.** Ratio gain in network lifetime.

other, which in turn entails a higher correlation degree among the sensed data. This in turn renders a smaller packet size. Conversely, in the classical scheme, since the packet size is fixed, a higher density network only increases the number of packets transmitted, consuming a lot of energy.
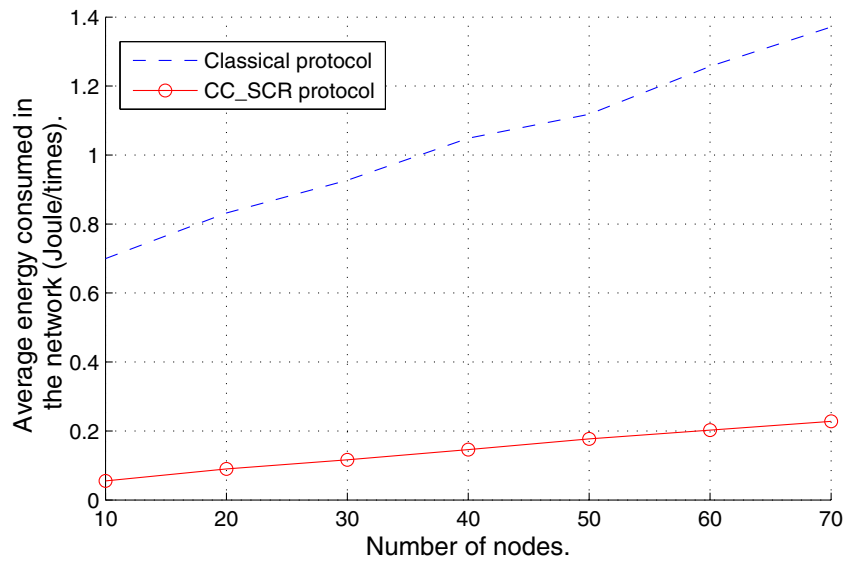

Figure 8. Average energy consumed per unit of time vs number of nodes.

Figure 9 shows the average energy consumed over time for 60 nodes in the classical, CC_SCR protocols, and also for one single hop to reach the sink. In order to explore the benefits of clustering architecture, a scenario where all nodes transmit directly to the sink is presented. For the proposed scheme, all active nodes transmit their initial packet to the sink in order to choose the reference node (note that in this case there is no CH). Then, the sink selects the node that minimizes the data difference, as explained in the previous section, and then transmits a control packet indicating the ID of the reference node. Then, for data transmission, active nodes only transmit their difference, Δ_i, directly to the sink. The results demonstrate clearly that CC_SCR conserves more energy compared to the classical scheme. Also, it is clear that the choice of clustering scheme offers more energy savings than the single hop scheme. The ratio gain presented in Figure 10 may reach up to 11 times more energy conservation than the classical scheme, and up to 119 times more energy conservation than the single hop scheme, which are considerable results.
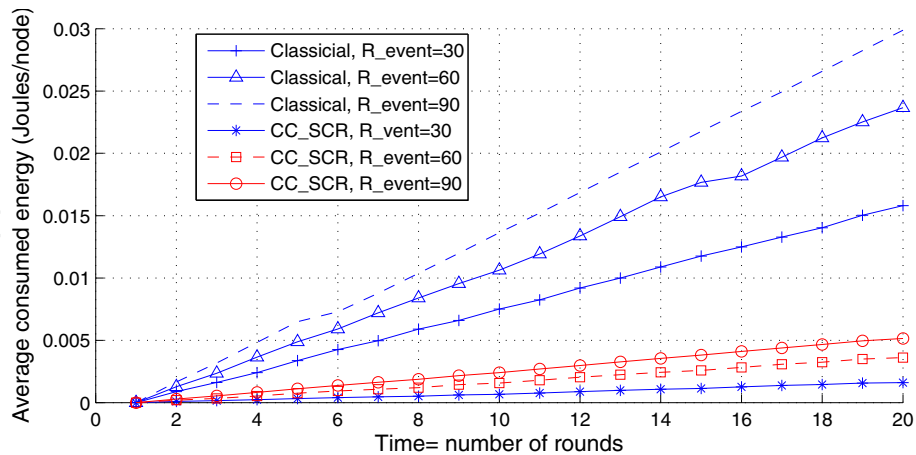
Figure 11 shows the average energy consumed for different values of R_event region. When R_event is varied, also the number of active nodes per event is modified accordingly. Figure 12 shows the number of active nodes per event. It can be seen that the average number of active nodes for both the classical and proposed scheme is approximately the same. Indeed, the proposed mechanism has no impact on the number of active nodes. Note that, by increasing the number of active nodes, energy consumption also increases. Observe, for instance, that energy consumption, when $R\_event = 30$ meters, is less than the consumption when $R\_event = 60$ meters and 90 meters. In each scenario, we observe that, by enabling our compression scheme, energy consumption over the network is reduced, therefore extending network lifetime.
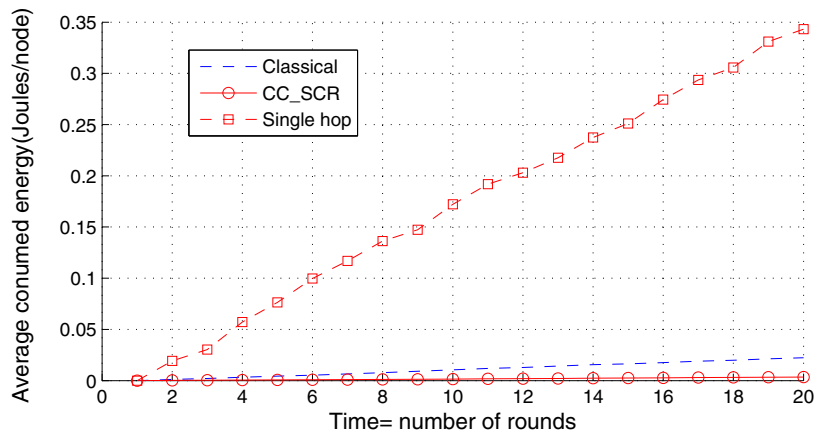
Figure 9. Average energy consumed over time.
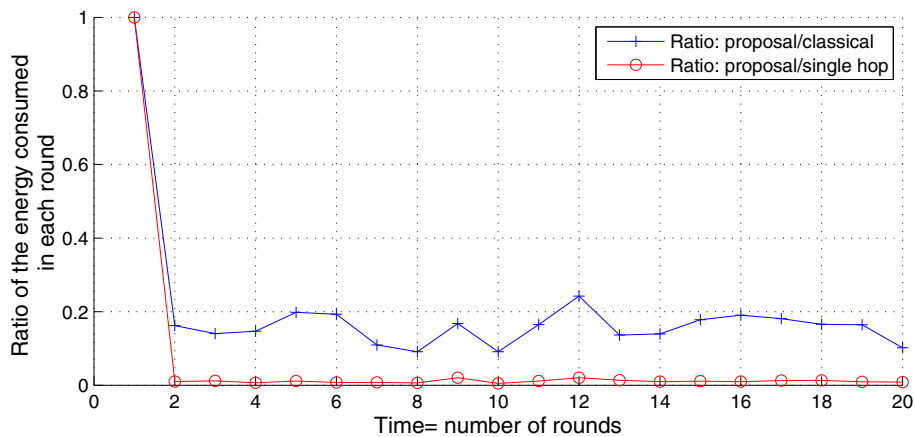
Figure 10. The ratio energy gain for each event occurrence.

Figure 11. Average energy consumed while varying the *R_event* region.

Figure 13 shows the average energy consumed for different values of *T_event* period. Increasing *T_event* also increases the period of the steady-state phase and the number of data reported, thereby it can be seen as an increase in the energy consumption. That explains why the energy consumed by *T_event* = 200 seconds is less than that of *T_event* = 300 and 400 seconds. In each scenario, we observe that enabling our compression scheme reduces energy
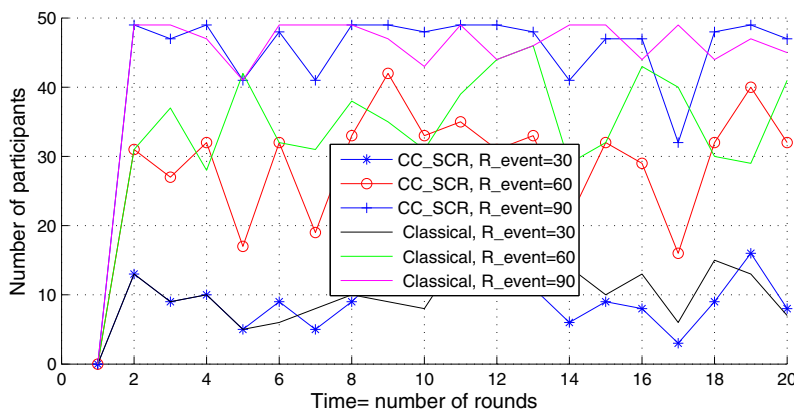
Figure 12. Number of active nodes for each event occurrence.

consumption over the network and thereby extends network lifetime. It is important to note that the proposed mechanism is particularly energy-efficient for high-event duration times. This is due to the fact that, as event duration increases, CMs in the classical scheme transmit many full-length packets while, in the proposed scheme, CMs also transmit many packets but with a much smaller length. This results in a slight increase in energy consumption for the proposed scheme while for the classical scheme there is a significant increase in energy consumption when the event duration increases.
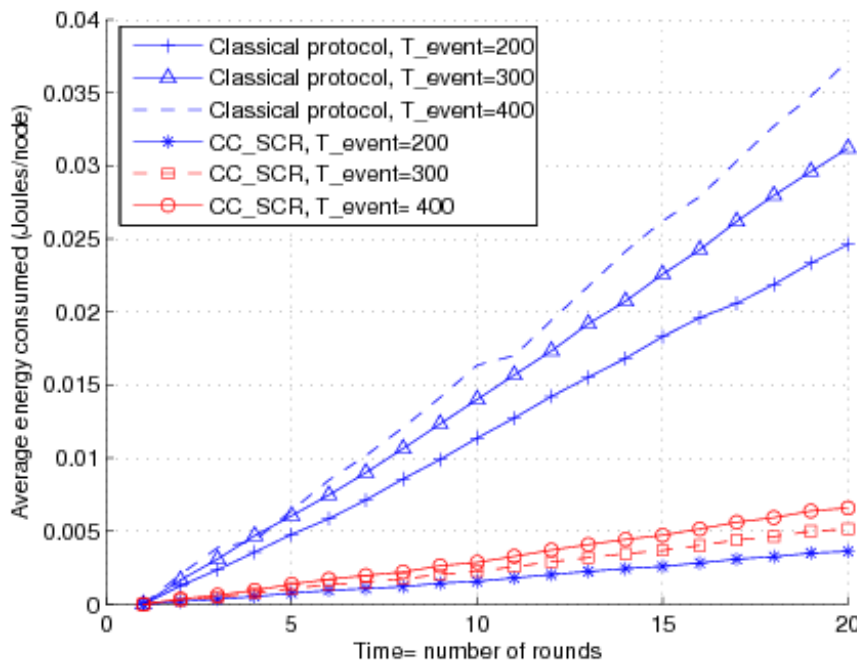

Figure 13. Average energy consumption vs number of rounds of length *T_event*.

Figure 14 shows the average energy consumption in *CC_SCR* when the aggregation technique is enabled at the CH, compared to the case where no aggregation is performed. The results clearly show that the aggregation technique conserves more energy (Figure 15 shows the ratio of the gain). The result is expected because when the CH aggregates the data of its CMs, the

CH only transmits one single packet to the sink, unlike the case when no aggregation is used, where the CH transmits each data value received from the CMs.
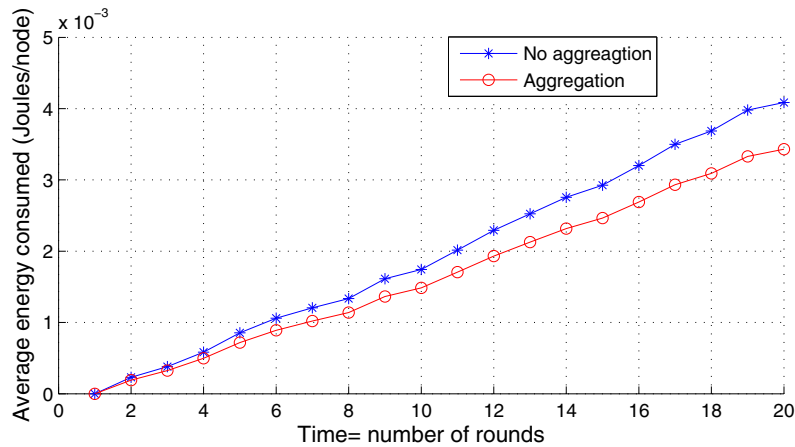

Figure 14. Energy consumed with aggregation and without aggregation.
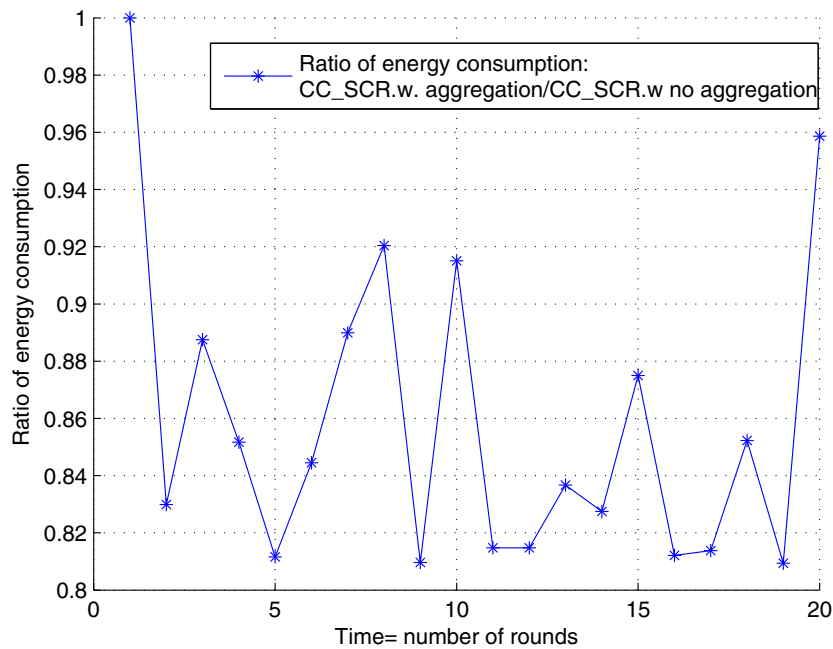

Figure 15. Ratio of energy consumed.

6. Conclusion and future work

In this chapter, we have proposed a novel clustering scheme, namely, *CC_SCR* protocol, which uses a compression technique for event-driven applications in WSNs. The proposed clustering scheme is based on selecting the node that reduces the packet size among all the active nodes in the system. The sink selects the node which minimizes the total amount of data as a CH, therefore increasing the efficiency of the compression technique by sending only the difference, rather than the complete data value, to the CH. To analyze the performance of the proposed scheme compared to both single hop (i.e., direct transmission to

the sink) and classical schemes, an approximate mathematical model for energy consumption was developed. In addition, we implemented the *CC_SCR* in TinyOS, and for different system parameters, simulation results conclude that, considering the spatial correlation in the communication of WSNs, achieves significant energy conservation compared to a classical clustering scheme. The ratio benefit may be up to 11 times that of the classical scheme. As such, the proposed scheme greatly extends network lifetime. In future work, we aim to investigate a generalization of the clustering scheme in order to consider a higher number of events that can occur simultaneously in within the network, as it is the case in some environment monitoring applications. These results can be verified and deployed in our testbed. We also aim to include a mobility aspect to a certain number of nodes in the network, and consider this propriety in the clustering process in order to achieve considerable energy savings.

Author details

Sofiane Moad, Nizar Bouabdallah, Bruno Sericola and Yassine Hadjadj Aoul
IRISA/INRIA Rennes - Bretagne Atlantique, France

Mario E. Rivero-Angeles
National Polytechnique Institute (IPN), Mexico

7. References

[1] J.N. Al-Karaki and A.E. Kamal. Routing Techniques in Wireless Sensor Setworks: A Survey. IEEE Wireless Communications, 11(6):6–28, 2004.

[2] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks.* Wiley-Interscience, 2007.

[3] I. Dietrich and F. Dressler. On the Lifetime of Wireless Sensor Networks. ACM Transactions on Sensor Networks (TOSN), 5(1):1–39, 2009.

[4] N. Kimura and S. Latifi. A Survey on Data Compression in Wireless Sensor Networks. IEEE International Conference on Information Technology: Coding and Computing (ITCC), *Nevada, USA*, 2:8–13, April 2005.

[5] A.A. Abbasi and M. Younis. A Survey on Clustering Algorithms for Wireless Sensor Networks. Elsevier Computer Communications, 30(14-15):2826–2841, 2007.

[6] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A Survey on Sensor Networks. IEEE Communications Magazine, 40(8):102–114, 2002.

[7] S. Moad, M. Rivero, N. Bouabdallah, and R. Langar. *CC_SCR*: Compression Cluster-based scheme in a Spatial Correlated Region for Wireless Sensor Networks. International Conference on Communication (ICC), *Kyoto, Japan*, June 2011.

[8] P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. ACM International Conference on Embedded Networked Sensor Systems (SenSys), *California, USA*, pages 126–137, November 2003.

[9] K.C. Barr and K. Asanović. Energy-aware Lossless Data Compression. ACM Transactions on Computer Systems (TOCS), 24(3):250–291, 2006.

[10] C.M. Sadler and M. Martonosi. Data Compression Algorithms for Energy-Constrained Devices in Delay Tolerant Networks. ACM International Conference on Embedded Networked Sensor Systems (SenSys), *California, USA*, pages 265–278, November 2006.

[11] F. Marcelloni and M. Vecchio. Enabling Energy-Efficient and Lossy-aware Data Compression in Wireless Sensor Networks by Multi-Objective Evolutionary Optimization. Elsevier Information Sciences, 180(10):1924–1941, 2010.

[12] D. Petrovic, R.C. Shah, K. Ramchandran, and J. Rabaey. Data funneling: Routing with Aggregation and Compression for Wireless Sensor Networks. IEEE International Workshop on Sensor Network Protocols and Applications (SNPA), *Alaska, USA*, pages 156–162, May 2003.

[13] T. Arici, B. Gedik, Y. Altunbasak, and L. Liu. PINCO: A Pipelined in-Network Compression Scheme for Sata Collection in Wireless Sensor Networks. IEEE International Conference on Computer Communications and Networks (ICCCN), *Dallas, USA*, pages 539–544, October 2003.

[14] S.S. Pradhan, J. Kusuma, and K. Ramchandran. Distributed Compression in a Dense Microsensor Network. IEEE Signal Processing Magazine, 19(2):51–60, 2002.

[15] D. Wei, S. Kaplan, and H.A. Chan. Energy Efficient Clustering Algorithms for Wireless Sensor Networks. IEEE Workshops of IEEE International Conference on Communications (ICC), *Bejing, China*, pages 236–240, May 2008.

[16] O. Younis, M. Krunz, and S. Ramasubramanian. Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges. IEEE Network, 20(3):20–25, 2006.

[17] D. Amaxilatis, I. Chatzigiannakis, C. Koninis, and A. Pyrgelis. Component Based Clustering in Wireless Sensor Networks. ACM Computing Research Repository, 2011.

[18] O. Younis and S. Fahmy. HEED: a Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad hoc Sensor Networks. IEEE Transactions on Mobile Computing (TMC), pages 366–379, 2004.

[19] S. Lindsey and C.S. Raghavendra. PEGASIS: Power-Efficient Gathering in Sensor Information Systems. IEEE Aerospace Conference, *Montana, USA*, 3:3–1125, 2002.

[20] M. Ye, C. Li, G. Chen, and J. Wu. EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks. IEEE International Performance, Computing, and Communications Conference (IPCCC), *Louisiana, USA*, pages 535–540, April 2007.

[21] B. Kusy, C. Richter, W. Hu, M. Afanasyev, R. Jurdak, M. Brunig, D. Abbott, C. Huynh, and D. Ostry. Radio Diversity for Reliable Communication in WSNs. ACM/IEEE International Symposium on Information Processing in Sensor Networks (IPSN), *Chicago, USA*, pages 270–281, April 2011.

[22] J. Faruque and A. Helmy. Rugged: Routing on Fingerprint Gradients in Sensor Networks. IEEE/ACS International Conference on Pervasive Services (ICPS), *Beirut, Lebanon*, pages 179–188, July 2004.

[23] M.C. Vuran and I.F. Akyildiz. Spatial Correlation-based Collaborative Medium Access Control in Wireless Sensor Networks. IEEE/ACM Transactions On Networking (TON), 14(2):316–329, 2006.

[24] S. Pattem, B. Krishnamachari, and R. Govindan. The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks. ACM/IEEE International

Symposium on Information Processing in Sensor Networks (IPSN), *California, USA*, pages 28–35, April 2004.

[25] G. F. Smoot et al. Description of COBE DMR. IEEE Astrophysical Journal Letters (ApJ), pages 360–685, 1990.

[26] A. Jindal and K. Psounis. Modelling Spatially-Correlated Sensor Network Data. IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), *Santa Clara, Canada*, pages 162–171, October 2004.

[27] W.R. Heinzelman, A. Sinha, A. Wang, and A.P. Chandrakasan. Energy-Scalable Algorithms and Protocols for Wireless Microsensor Networks. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), *Istanbul, Turkey*, pages 3722–3725, June 2000.

[28] Y. Chen and Q. Zhao. On the Lifetime of Wireless Sensor Networks. IEEE Communications Letters, 9(11):976–978, 2005.