A Spatial Correlation Aware Scheme for Efficient Data Aggregation in Wireless Sensor Networks

Imane Horiya Brahmi, Soufiene Djahel, Damien Magoni, John Murphy

To cite this version:

Imane Horiya Brahmi, Soufiene Djahel, Damien Magoni, John Murphy. A Spatial Correlation Aware Scheme for Efficient Data Aggregation in Wireless Sensor Networks. 11th IEEE International Workshop on Performance and Management of Wireless and Mobile Networks, Oct 2015, Clearwater Beach, United States. pp.847-854, 10.1109/LCNW.2015.7365937 . hal-01281869

HAL Id: hal-01281869
https://hal.archives-ouvertes.fr/hal-01281869

Submitted on 3 Mar 2016

A Spatial Correlation Aware Scheme for Efficient Data Aggregation in Wireless Sensor Networks

Imane Horiya Brahmi ∓, Soufiene Djahel ∓, Damien Magoni ‡ and John Murphy ∓
∓ Lero, UCD School of Computer Science and Informatics, Ireland
‡ University of Bordeaux – LaBRI, France
{horiya-imane.brahmi, soufiene.djahel, j.murphy}@ucd.ie, magoni@labri.fr

Abstract—**One of the most popular and efficient methods for conserving energy in Wireless Sensor Networks (WSNs) is data aggregation. This technique usually introduces an additional delay in the transmission of data packets. The inherent trade-off between energy consumption and end-to-end delay imposes an important decision to be made by the nodes, mainly to determine the most appropriate time for aggregating local/transiting packets and forwarding the resulting packet(s) to the next hop towards the sink. Most of the solutions proposed so far are either unable to significantly reduce the overhead caused by the redundant transmissions, or require a long waiting time before aggregating the received packets. To overcome the above limitations, we propose a novel scheme that ensures efficient and fast packets aggregation in WSNs. This scheme defines optimal decision making policies at the cluster head level (i.e. in a cluster based topology) to determine the appropriate waiting time before aggregating the local data sampling, as well as data sampling received from other neighbor cluster heads. The obtained evaluation results confirm the efficiency of our scheme in terms of the achieved end-to-end transmission delay for periodic packets and the reduced overhead. These results reveal also that our scheme outperforms other three literature schemes (no aggregation, randomized waiting and full aggregation) as it ensures the best compromise for aggregation saving and delay reduction.**

Keywords – **WSNs, Data aggregation, Periodic packets transmission, Dynamic waiting time, Spatial correlation.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) [1] involve a large number of sensors that collaboratively collect, process and report various events to fulfil a common task. However, due to the large-scale deployment of such sensors for monitoring purpose, and the resulting overlap of events detection ranges in WSN, the data collected by several sensors could be redundant, with highly correlated or copied data. Moreover, sensor nodes are most likely to be battery-powered, therefore power saving represents a real concern in these networks. The energy consumption is associated primarily with communication; in fact, it is often the most expensive activity in terms of power supply, as stated in [2].

In-network data aggregation represents an efficient way for reducing the energy consumption in WSNs by reducing the communication overhead. In-network aggregation is defined as the process of gathering the data, pre-processing and computing it in the network itself and transmitting the extracted and required data to the sink. Hence, the energy is conserved by eliminating redundancy and minimizing the transmission of raw data to the sink [3]. Deferring the transmission of data packets in order to wait for more information from neighboring nodes increases the degree of aggregation and leads to better data aggregation opportunities. This method has been widely adopted in many data aggregation approaches such as [4] and [5], however, it causes long end to end transmission delays, especially in monitoring applications where the timing strategy is critical.

Nowadays, the new generation sensor nodes have been enhanced with significant energy-efficient storage, processing capabilities, and data management abilities [6]. Hence, sensor nodes can be endowed with energy-efficient storage such as the new-generation flash memory with several gigabytes of storage and low-power consumption [7] [8]. In addition, [9] shows that transmitting data over a radio channel consumes 200 times more energy than storing the same amount of data locally on a sensor node, and radio reception uses 500 times more energy than reading the same amount of data from local sensor buffer [7].

This emergence of low-cost, high-capacity flash storage and processing requires the design of novel data aggregation schemes that leverage those new features to achieve higher efficiency. In this scheme, each intermediate Cluster Head (CH) stores a copy of every local and transiting aggregated sampling for future aggregation purpose based on spatial correlation properties. However, due to the limited storage capacity of sensor nodes, a CH may not be able to store all the received and/or aggregated samples to provide the suitable aggregation level. Therefore, our goal is to design a decision making mechanism that defines the optimal waiting time at a CH level before aggregating a set of packets, sent by spatially correlated CHs, reporting the same parameters.

The proposed mechanism ensures a balanced trade-off between the incurred communication overhead and the achieved end-to-end delay by reducing the number of unnecessary redundant transmissions, while meaningful (i.e. aggregated) data packets are sent as soon as the reported value is confirmed. Our mechanism is suitable for monitoring applications that require real-time information such as military applications, road traffic monitoring applications etc.

The remainder of this paper is organized as follows: in Section II we present the related contributions on data aggregation in WSNs, in structured and structure free topologies, and highlight their limitations. In Section III, we present our scheme in detail and evaluate its effectiveness in Section IV. Finally, we conclude the paper and discuss the future work in Section V.

II. RELATED WORK

In this section, we review the most significant existing works on data aggregation in WSNs that have investigated either structured or structure-free approaches. Data aggregation combines data from different sources (i.e. sensor nodes) and forwards it to the sink using functions such as suppression (i.e. eliminating duplicates), min, max and average [10]. The data aggregation phase presents many advantages in eliminating redundancy from neighbouring nodes due to the high spatial correlation in WSNs, leading to reduction of the overall communication overhead, increase in the network lifetime, and improved utilization of the available bandwidth.

Organizing sensor nodes into clusters has been widely pursued by the research community in order to improve the network performance. Two surveys [3], [11] have explored the most significant contributions dealing with data aggregation based on cluster approaches; such as, [12] [13]. Most of the contributions presented in these two surveys are based on structured approaches, and the most important work among them is Low Energy Adaptive Clustering Hierarchy (LEACH) [12]. In this latter, two main phases are defined: a setup phase during which the network is organized into clusters and the cluster heads are chosen, and a steady phase in which the data is aggregated and transmitted to the sink. HEED [14] is another protocol which extends LEACH by incorporating communication range limits and intra-cluster communication cost information. The initial probability for each node to become a tentative cluster head depends on its residual energy, and final heads are selected according to the cost.

The authors of [15] have proposed EAST algorithm to perform near real-time data collection based on spatial and temporal correlations. In this algorithm, the sensor nodes are grouped into correlation regions according to the spatial correlation property. The CH performs the temporal correlation suppression and forwards the readings to the sink only when they exceed a certain threshold. This algorithm is suitable for aggregating event-driven messages; however the dynamic cluster formation process may represent an issue, in terms of the computation and communication overhead incurred by CHs election process, as it may delay the transmission of emergency messages and increase the energy consumption. Besides, this solution is not appropriate for periodic messages transmissions. For further reading on data aggregation challenges in WSNs, the reader may refer to [16], [17] and [18].

The authors of [19] have proposed the first structure-free data aggregation technique for event detection in WSNs. They use structure-less approach in which the messages are not exchanged explicitly; the nodes do not know where they should send packets to and how long they should wait for aggregation. In this work, two methods were proposed; one for improving spatial correlation and the other for temporal convergence. Indeed, spatial and temporal correlations are key factors to improve the aggregation efficiency in the network. These two factors are leveraged in structured approach based solutions by letting nodes transmit packets to their parents in the aggregation tree, and the parents wait for packets from all their children before transmitting the aggregated packets. In contrast, the proposed scheme in [19] achieves high aggregation efficiency without incurring high overhead caused by the construction and maintenance of structure. However, this scheme does not maintain its high efficiency when the network gets larger. Moreover, the random waiting time adopted in this scheme is not an efficient solution, especially when nodes close to the sink select a small waiting time.

In monitoring applications, the end-to-end latency is required to be as low as possible since it affects the offered QoS. It is well known that an optimal waiting time, before performing the aggragtion, decreases (resp., increases) the data transmission latency and the energy consumption (resp., accuracy). To this end, [20] proposed a dynamic timeout for data aggregation in WSNs by dynamically varying its value according to the amount of currently accumulated data at a given node, achieving an efficient trade-off between energy consumption and latency in the network.

In the same context, the authors of [21] and [22] proposed an algorithm that aims to minimize the latency and ensure high data accuracy based on an efficient selection of the waiting time. In [21], the authors assumed that the aggregation tree already exists before the scheduling process starts. However in this solution, the base-station has to inform the network nodes by flooding a request message, every time there is a change in the frame duration T_{max}. The waiting time at the sensor node is then periodically adjusted depending on the received T_{max} and the number of children is updated.

In [22] the authors defined the data accuracy as the optimal number of data N_{opt} participating in the final aggregation result. The authors assume that the data aggregation tree is already constructed. The base station adjusts the frame duration T_{max} based on the number of the received data at the current and the previous frame, the N_{opt} and a predefined threshold. Once the T_{max} is computed, the base station will broadcast it to the nodes in the network, which, in their turn, compute their waiting time according to the T_{max} value. It is worth mentioning that broadcasting and periodic flooding of messages in the above solutions have a negative impact on the energy consumption and the collision rate in the network.

In contrast to the above discussed solutions, we design, in this work, an original mechanism that ensures an appropriate selection of the waiting time at the CH nodes in order to minimize the end-to-end delay as well as the energy consumption. To that end, this mechanism leverages the content of the received packets along with the spacial correlation properties. Moreover, a copy of resulting packets from the aggregation process at the CH is stored in its buffer to confirm the accuracy of the transmitted data towards the sink.

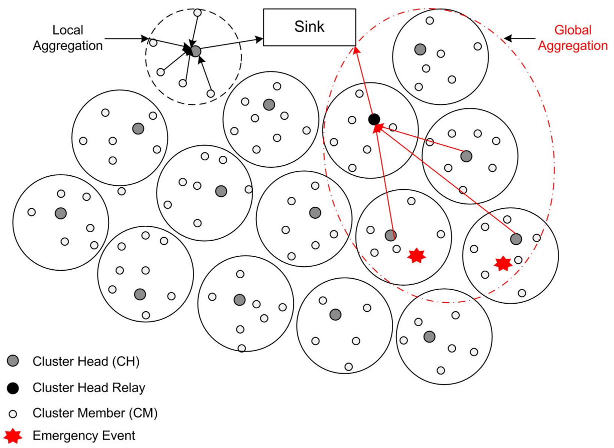

Figure 1: Clustering topology with Local and Global aggregation

III. System design

In our scheme, the sensors measure different parameters and report them periodically to the sink. We classify the collected data into different categories according to their content type and criticality level. For example, in road traffic monitoring scenario, reporting vehicles speed periodically represents the category with the highest priority, while traffic flow density and weather conditions such as temperature, humidity or snow level on the road are less critical compared to the first category. We consider that the sensors are organized into clusters to make the aggregation and the communication easier and more efficient. Each CH is in charge of gathering readings from its Cluster Members (CMs) and/or other CHs, and aggregating them before sending the resulting packet(s) toward the sink. Note that our solution is based on a static clustering approach in which the CHs election process is performed at the network initialization only, mitigating the computation and communication overhead incurred in the legacy schemes.

We define two different types of aggregation: Local and Global as shown in the Figure 1. The local aggregation is performed first by the CHs which gather readings from their CMs as cited in [23]. The CHs gathering time before aggregation is defined by a Local Waiting Time (WT_{Local}) which is a dynamic time value that determines when the CHs should aggregate the received readings and forward the resulting packet (s), before the end of the monitoring period. These aggregated readings are then sent onwards toward the sink via a multi-hop routing path. When the CHs further up the chain receive the aggregated reading, they wait again for a period of time (WT_{Global}) to receive other transiting aggregated readings from neighbouring CHs, based on spatial correlation properties, before they aggregate the data and forward it to the next hop. We can reasonably call this process global aggregation. We assume that these sensors are equipped with buffer memory and queue capability.

A CH buffer is used for storing samples either received from its own CMs or from adjacent CHs. Each CH waits for the required minimum number of samples reporting data from the same category before it starts the aggregation process. The aggregated samples are then stored in the CHs buffer. These copies of aggregated samples are used for aggregating future samples received from other sources (i.e. CMs, or adjacent CHs), according to their content and spatial correlation properties. Next, the forwarding queue is used for storing the aggregated packets which are ready to be sent over the medium to the sink.

The samples stored in the buffer have a limited storage time due to the limited energy of the sensors. Therefore, we define a Time out (T_{out}) for the different packets stored in the buffer. This T_{out} represents the maximum storage time for each packet in the buffer.

Our focus in this work is to design a dynamic local and global waiting time and a T_{out} mechanism for the packets in the buffer such that they optimize the energy usage, buffer storage time, and transmission delay of packets with different priority levels, and finally reduce the overall traffic load in the network.

A. *Local aggregation*

As mentioned above, the local aggregation is carried out by each CH in the network. After network initialization phase, the CMs start measuring given parameters and forward the corresponding readings to the CHs. We assume that the local measurements are periodic and all the CMs start collecting data at the same time (i.e. the beginning of each period). In what follows, we define an algorithm for local aggregation based on a dynamic waiting time.

The local periodic sensing happens during the sampling period as shown in Figure 2. At the end of each sampling period, the CMs start forwarding their readings (e.g. CM_n forward P_n: an aggregated readings). The arrival time of the readings at the CH level and the access to the wireless channel for all the CMs are random, following the CSMA/CA scheme. The aggregation process starts after receiving these first samples. The CHs stop waiting for more samples at every available transmission epoch.

In algorithm 1, we describe the proposed mechanism that defines the appropriate waiting time before aggregating the received packets by each CH. This mechanism uses parameters such as the minimal number of samples received by the CH (M), the mean of the received sample values, and the standard deviation (σ_{th}), to decide about the appropriate time to start aggregating the received packets and forward the resulting packet (s). Once the first samples are received by the CHs, they start the aggregation process, computing the mean and standard deviation (σ_{cal}) of the received values, and comparing them with σ_{th}. If the minimal number of required samples is reached, and σ_{cal} is larger than σ_{th}, then the CH waits for more samples. Otherwise, the samples can be aggregated and the resulting aggregated sample can be sent since the received readings have approximately similar values. On the other hand, if a received sample reports a value exceeding a given upper bound ($Value_{max}$) of the measured parameter (e.g. a vehicle

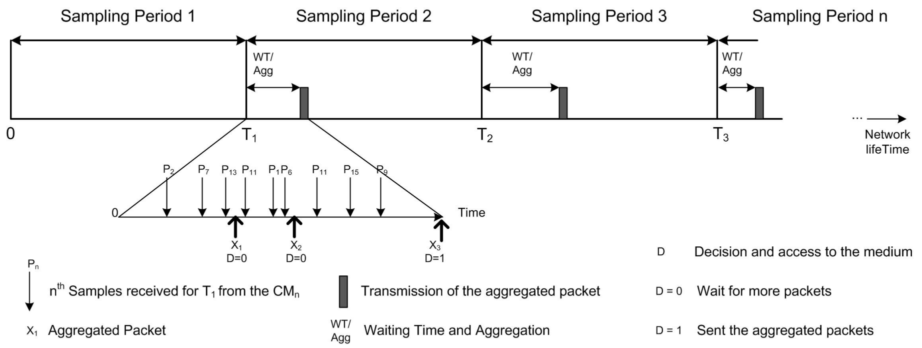

Figure 2: Representation of a sampling period and the decision process for the aggregation

speed exceeding the speed limit in a given road segment), this sample is forwarded immediately as an emergency message. Once the conditions on minimal number of required samples and σ_{cal} are met, the CH waits until the channel is sensed idle and then at the next available transmission epoch (D) forwards the aggregated packet (s), as shown in Figure 2.

B. Global aggregation

Once the CHs aggregate the data locally, this data is then forwarded to the sink via other intermediate CHs in the network. These latter (i.e. CHs relays) are usually located in the core of the network, they are in charge of aggregating the data received from other CHs, as well as the data sent by their local CMs. Notice here that the aggregation of data sent by other CHs is performed according to the spatial correlation among those CHs, meaning that the packets received from CHs spatially correlated are aggregated together to generate the final packet(s) to be forwarded towards the sink.

Whenever a packet is received by the CH relay from one of its children CHs, it checks the cluster ID of the received packet, and based on which the set of neighbouring CHs spatially correlated with the sender of this packet are determined. In our scheme, the CHs deployed in a specific monitored area or within a given distance from each other are more likely to measure similar values of some parameters, therefore they are spatially correlated, as shown in Figure 3. For example: CH_{16} and CH_{12} are both one hop away from CH_4, if the CH_4 receives packets from these two CHs, it will then aggregate them together as they monitor the same area and their reported values are more likely to be redundant.

In our scheme, before the CHs relay forward any packet received from their neighbours (i.e. children CHs), they wait for a period of time WT_{Global}, computed and updated according to Algorithm 2. For example, in a scenario where WSNs are deployed on the road to periodically measure and report the traffic volume and speed of vehicles, each sensor on the road represents a CM, and each lane represents a cluster or a set of clusters. The information sent by sensors deployed on lanes belonging to the same road segment should be aggregated together in order to accurately estimate its state. Moreover, the set of lanes in the same road segment are aggregated by the same CH; which is defined by the global aggregation scheme.

Algorithm 1 Local Waiting Time Algorithm

```
1:  $M$: the required minimum number of samples received by
    the CHs
2:  $\sigma_{cal}$: The computed Standard deviation
3:  $\sigma_{th}$: Standard deviation threshold
4:  $Value_{max}$: threshold of the measured value of a given
    parameter
5:  $P_i$ : packet sent by the $CM_i$
6:  while (the number of received pkts $\leq M$) do
7:    Upon reception of a pkt ($P_i$) do
8:    if (the received value $\geq Value_{max}$) then
9:      CH generates and forwards an emergency pkt
10:   else
11:     CH keeps copy of $P_i$
12:     Update the number of received pkts
13:     Wait for the next packet
14:     CHs compute the mean and the $\sigma_{cal}$
15:   end if
16: end while
17: Compare the $\sigma_{cal}$ with $\sigma_{th}$ to extend the waiting time or
    not
18: if ($\sigma_{cal} < \sigma_{th}$) then
19:   Aggregate the received pkts
20:   Forward the resulting pkt
21:   Keep copy of the forwarded pkt
22:   CHs keep waiting for other CMs' pkt
23: else
24:   if (number of received pkts equals to cluster size) then
25:     Aggregate the received pkts (Algo ended)
26:   else
27:     Wait for the next packet // the CHs wait for more
        packets
28:     Go to step (18)
29:   end if
30: end if
```

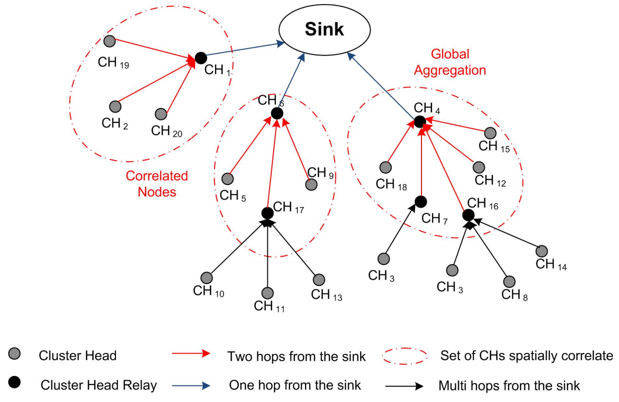

Figure 3: Clustering topology illustrating the Global aggregation, the number of hops separating CHs from the sink

Global Waiting Time Computation: WT_{Global} is another parameter that differs from WT_{Local}, and is calculated by the CHs relay only. Each of those CHs computes the WT_{Global} based on the WT_{Local} of the first received packet for each spatially correlated set of CHs, and updates it according to the WT_{Local} of the subsequently received packets belonging to the same set, as long as the aggregated packet(s) has not been forwarded yet.

In our scheme, each CH child adds the time it has awaited for before aggregating the received readings from its CMs (i.e. WT_i) to the transmitted packet P_i (i.e. the resulting packet from the aggregation) towards its parent CH. Once a CH parent (e.g. CH_5 is the parent of CH_1, CH_2, CH_3 and CH_4 as shown in Figure 4) receives the first packet with its corresponding WT_i, it waits WT_i (i.e. $WT_{Global} = WT_i$) for receiving other packets from its other children belonging to the same correlated set. If another packet P_j from this set is received the CH parent checks the WT_j of the actual packet and compares it with the previous WT_i. If it is greater, the WT_{Global} is updated as per Algorithm 2. Whenever WT_{Global} expires the CH parent aggregates the received packets from the same set, forwards the resulting packet, and stores its copy in its local buffer for future aggregation purposes.

The stored copy is used in case a CH children's transmission has been delayed or its packet is lost, to prevent unnecessary redundant transmissions. After expiration of WT_{Global}, if a delayed packet from a CH children is received, the CH parent compares its reported value with the content of the previously forwarded packet, as a result of the aggregation, and decides if this packet needs to be forwarded or not.

In our scheme, the CHs aggregate the readings sent by their CMs and send the resulting packet (s) towards the sink before the end of the monitoring period; this approach might be beneficial for some applications, such as road traffic monitoring, since this enables early prediction of the traffic jam, for example. Furthermore, in our scheme the aggregator sensors store a copy of the resulting packet from the aggregation process (i.e. the packet forwarded towards the sink), so that it is used to validate the correctness and accuracy of early aggregated packets generated before the end of the monitoring period.

To explain the key principle of our global aggregation strategy, we consider the topology shown in Figure 4 where CH_1, CH_2, CH_3, CH_4 are children of CH_5 belonging to the same spatially correlated set. In this example, CH_5 waits for packets from its children and aggregates them together. Let us further assume that CH_1, CH_4, CH_2 sent their packets with the corresponding WT_1, WT_2, WT_4, respectively. Every time CH_5 receives a packet it updates its WT_5 according to Algorithm 2. Once WT_5 expires, CH_5 aggregates the received packets and forwards the resulting packet towards the sink. The CH_5 will also, as discussed previously, store a copy of the forwarded packet in its buffer, for aggregating future received packet. For instance, when the delayed CH_3 is packet is received at CH_5, this latter compares the received packet's value with the stored copies value. If they are equal then CH_3 s packet is dropped, otherwise, if the difference exceeds a given threshold then this packet is forwarded.

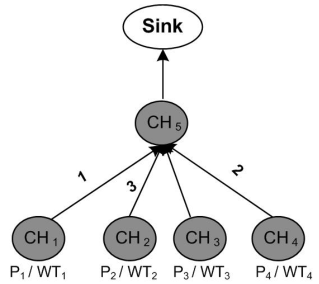

Figure 4: Clustering topology illustrating the Global aggregation and the number of hops separating CHs from the sink

IV. PERFORMANCE EVALUATION

Table I: Summary of simulation parameters

Parameters	Value
Routing Protocol	AODV
Propagation mode	TwoRayGround
Packet Size	64 Bytes
Number of nodes	25, 50,100, 200
Inter-node distance	10 meters
Monitoring period interval	15 s, 30 s, 60 s, 120 s
CsThreshold	25 meters
RxThreshold	12 meters
Simulation time	180 minutes
No. of simulation runs	100
Topology	grid

In this section, we evaluate and discuss the performance of our proposed partial data aggregation scheme. We have compared this scheme (Partial Aggregation: P_{Agg}) to three other schemes: Full Aggregation (F_{Agg}), Randomized Waiting Aggregation [19] (RW_{Agg}), and Aggregation Off (Agg_{Off}). In F_{Agg} scheme, the CHs wait to receive packets from all their CMs or their spatially correlated children CHs, before they aggregate them and forward the resulting packet towards the sink. Unlike the (F_{Agg}) scheme, the CHs in Agg_{Off}

Algorithm 2 Global Waiting Time Algorithm

```
$k$: Number of Children belonging to the same Spatially Correlated Set (SCS)
$Child_j$: j $th$ child of a CH, where $j = 1, ..., k$
$Pkts-recv$: Total number of received packets by the CH from a given set of spatially correlated children
$Time_{remaining}$: The remaining time in the $WT_{Global}$ associated with the SCS to which the sender CH belongs.
Initialization: $pkts_{recv}$ = 0
Upon reception of the first pkt from ($Child_j$) do
$WT_{Global}$ = $WT_j$
$Pkts_{recv}$ ++
while (($WT_{Global} \neq$ 0) && ($Pkts_{recv} < K$)) do
    Upon reception of a pkt from ($Child_j$) do
    $Pkts_{recv}$ ++
    CH updates its $WT_{Global}$ for the corresponding SCS
    if ($WT_j > WT_{Global}$) then
        $WT_{Global}$ = $WT_j$ - $WT_{Global}$ + $Time_{remaining}$
    else
        $WT_{Global}$= $Time_{remaining}$ (1+ $SCF$)
        $SCF$ (Spatial Correlation Factor)= 1- $\frac{Pkts_{recv}}{k}$
    end if
end while
Aggregate the received pkts
Forward the resulting pkt towards the sink
Keep copy of the forwarded pkt
```

scheme disable the aggregation mechanism and forward the received packets immediately upon their reception. In our P_{Agg} scheme, a smart way to aggregate the received packets from CMs or a spatially correlated set of children is designed, as described by Algorithms 1 and 2, in order to ensure faster transmission of the reported values by the sensors, while achieving a good level of aggregation and maintaining the energy consumption usage as low as possible. As opposed to our scheme, in RW_{Agg} scheme, the CHs wait a random time before aggregating the received packets.

We have conducted our simulation using the NS-2.35 network simulator in which we have implemented the four schemes and run simulations for several scenarios. We have also modified and adapted the default implementation of AODV protocol such that routing paths are comprised of CH nodes only so that all transmitted packets will be aggregated at the right node. Since our solution is based on a static clustering approach, the measured evaluation metrics do not include the cost of clusters deployment and maintenance. During each periodic monitoring interval, the CM nodes randomly generate and transmit their packets towards the corresponding CHs. The packets values are selected uniformly from a defined interval and used for aggregating the packets based on the rules defined in Algorithms 1 and 2. We have set the spacial closeness in the global aggregation to 25 meters, the minimum number of samples received by the CHs from its CMs to 20 % of the total number of CMs, and the coefficient of variation $\frac{\sigma}{\mu}$ to 0.25.

Once the CHs receive the first packets; they start the aggregation process, by computing the standard deviation (σ_{cal}) of the received packets values and comparing it with a σ_{th}, in order to decide when the aggregated packets should be sent. Once the conditions described in the above algorithms are met, the CHs forward the resulting packet (RES_{packet}) from the aggregation. At the end of each monitoring period, if one or more packets (from other CMs or other children CHs belonging to the same spatially correlated set of the senders of the aggregated packets) are received after sending the RES_{packet}, with values substantially different from the sent one in RES_{packet}, then a new packet is generated and sent towards the sink to report these values.

We summarize in Table I the default parameters used in the simulation. The primary metrics evaluated are:

(i) The Average Aggregated Ratio (AAR) in the intermediate nodes (i.e. aggregator nodes), defined as the ratio of the number of packets sent by the intermediate nodes to the number of packets that they have received, as described in Eq. 1:

$$AAR = 1 - \frac{\#PktsSent}{\#PktsRecv} \tag{1}$$

(ii) The Aggregation Efficiency (AE) at the sink level, defined as the ratio of the number of packets received by the sink, using an aggregation scheme (i.e. F_{Agg}, P_{Agg}, or RW_{Agg}), to the packets received with no aggregation performed (i.e. Agg_{Off} scheme), as described by Eq.2:

$$AE = 1 - \frac{\#PktsRecv(Agg\_On)}{\#PktsRecv(Agg\_Off)} \tag{2}$$

(iii) The Average End-to-End (E2E) transmission delay of all the packets, which represents the average time needed for a packet sent by the source (CHs/CMs) to cross the network and reach the sink.

(iv) The Energy Saving (ES) is defined as the ratio of the total energy saved (i.e remaining) when the aggregation is enabled to the total energy saved when the aggregation is disabled. It is calculated using the following formula:

$$\begin{aligned} ES &= \frac{Energy_{remaining}(Agg\_On)}{Energy_{remaining}(Agg\_Off)} \\ &= \frac{Energy_I - Energy_{consumed}(Agg\_On)}{Energy_I - Energy_{consumed}(Agg\_Off)} \end{aligned} \tag{3}$$

Where:

- $Energy_I$: initial energy of a sensor node.
- $Energy_{consumed}$: the energy consumed for packets transmission/forwarding.

The results plotted in Figure 5 compare the Average Aggregation Ratio (AAR) under various values of the monitoring period. In this particular scenario we have set the network size to 200 nodes. The results depicted in this Figure reveal that with a partial aggregation we achieve a lower average aggregation ratio compared to the F_{Agg} and RW_{Agg}. This

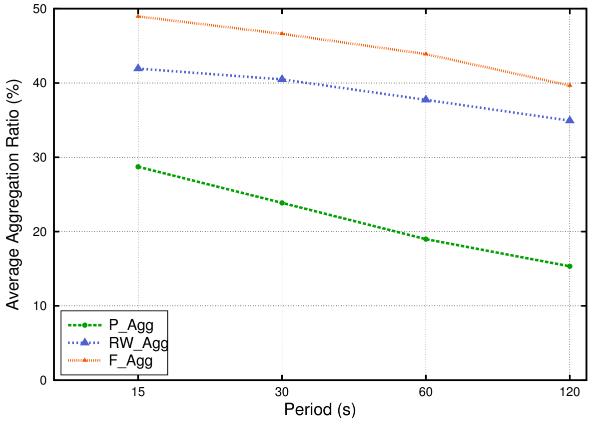

Figure 5: Impact of monitoring period intervals on Average Aggregation Ratio: Network Size = 200

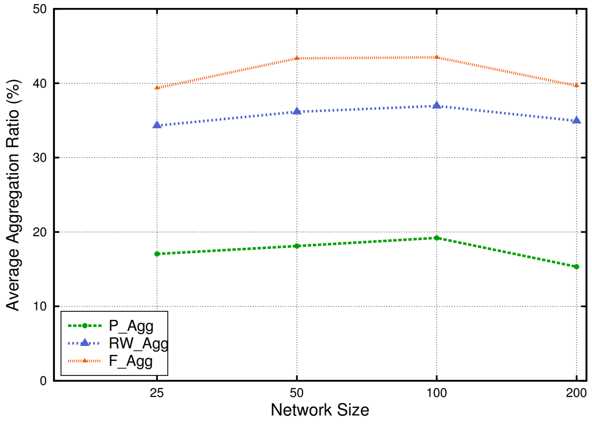

Figure 6: Impact of Network size on Average Aggregation Ratio: monitoring period interval = 60 s

is explained by the early aggregation used in our scheme. However, the difference between the P_{Agg} and the two other schemes is approximately 20% in average, this small difference demonstrates the efficiency of our scheme. We noticed also that the achieved AAR in the three schemes decreases with the increase of the monitoring period interval. When the monitoring period is long, the packets are more likely to be aggregated and sent before the end of this period, which will reduce the aggregation ratio. Also, when the period is short the aggregator nodes will wait till the end to send their packets, leading to an increase of their AAR.

In Figure 6, we varied the network size and set the monitoring period to 60 s. We see similar results to Figure 5, where F_{Agg} and RW_{Agg} schemes perform better than the P_{Agg} scheme in terms of the achieved AAR, this is due to the same reasons explained above. However, we observe that higher network densities have little influence on AAR, in all the schemes.

The results plotted in Figures 7 and 8 show the aggregation efficiency (AE) at the sink level. These results reveal that the F_{Agg} and RW_{Agg} achieve a higher level of AE compared to our scheme. However, the difference is very small and equals to 2% in Figure 7 when we vary the Period. On the other hand, as shown in Figure 8, with 15% difference between the two approaches at a network size of 25 nodes, the gap converges and the three approaches show a very close aggregation efficiency (less than 2%) at a network size of 100 and above.

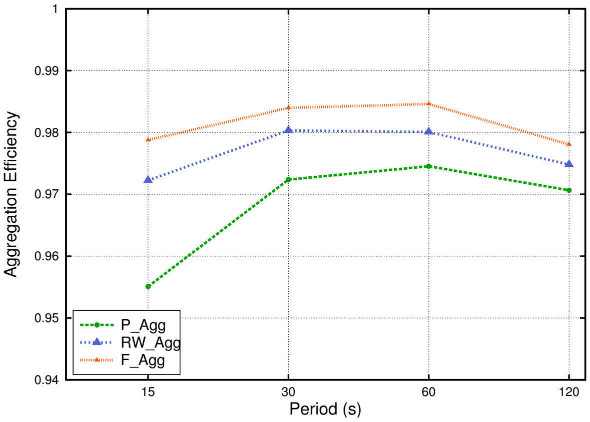

Figure 7: Impact of monitoring period intervals on Aggregation Efficiency in the network: Network Size = 200 nodes

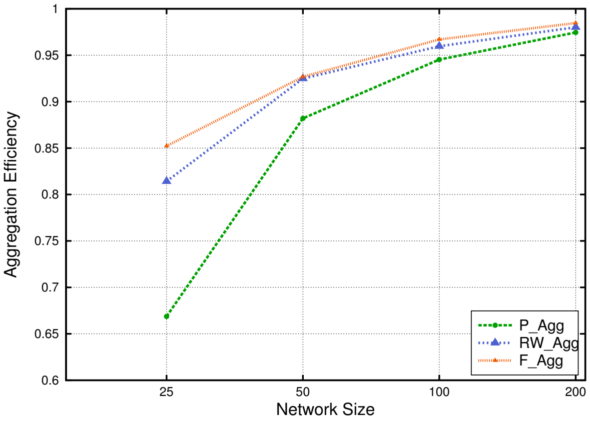

Figure 8: Impact of Network size on Aggregation Efficiency in the network: monitoring period interval = 60 s

We kept the same scenarios described above and ran our simulation for the four algorithms - full aggregation, randomized waiting aggregation, partial aggregation, and no aggregation to evaluate the impact of the monitoring period duration and network size on the achieved average E2E delay. The corresponding results of this scenario are plotted in Figures 9 and 10 respectively, from which we can observe that with no aggregation scheme the E2E delay is very low (in the order of ms) due to the immediate forwarding of the packets by the CHs. However, with P_{Agg}, a short E2E delay of the packets is achieved compared to F_{Agg} and RW_{Agg}, where the E2E delay is high. This results from the early aggregation and transmission of the packets in our scheme. The improvement achieved in some cases exceeds 40% and 50% compared to RW_{Agg} and F_{Agg}, respectively. For example in Figures 5 and 9 where the Period is set to 120 s, the E2E delay is equal to 168.17 s with an aggregation ratio of 39.65 % using the F_{Agg} scheme. On the other hand, with RW_{Agg}, the E2E delay is equal to 136.18 s with 34.92 % aggregation ratio; while our scheme achieves an E2E delay equals to 79.41 s with an aggregation ratio of 15.32 %. Therefore, we notice that our scheme achieves lower aggregation ratio compared to the F_{Agg} and the RW_{Agg} but the packets arrive much earlier.

Under the same conditions used in the previous plots, we have plotted some results showing the Energy Saving (ES) in terms of the total number of packets sent. In these results, we use the Agg_{Off} as a reference to compare the P_{Agg} with the RW_{Agg} and the F_{Agg}.

The results plotted in Figures 11 and 12 show that the achieved ES using the P_{Agg} scheme is almost similar to that achieved in RW_{Agg} and F_{Agg}. However, in Figure 11 the ES decreases with the longer duration of the monitoring period interval, as it is more likely for the aggregator node to receive more packets and aggregate them together. Moreover, the frequency of sending aggregated packets when the period is long is lower than that of shorter monitoring periods. From the Figure 12 we also notice that the ES is proportional to the network size since the AAR and the AE increase when the network gets larger. This explains the impact of the aggregation on the energy usage of the network.

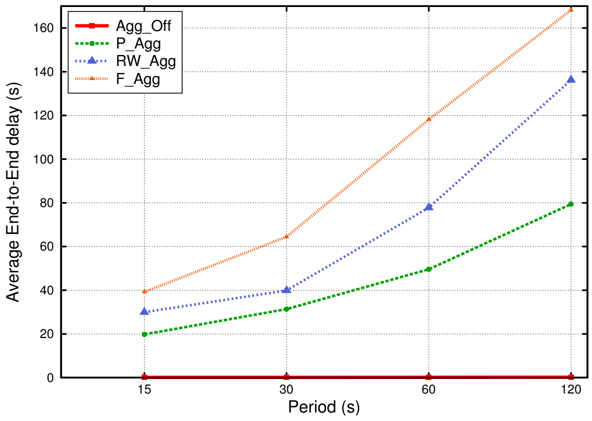

Figure 9: Average End-to-End delay under varying monitoring period interval: Network Size = 200 nodes

Recall that the objective of our work is to achieve a lower E2E delay of the packets transmission, while keeping fair aggregation ratio and energy saving in the network. From the above results, we conclude that F_{Agg} and RW_{Agg} schemes outperform our proposed scheme in terms of the achieved aggregation ratio and aggregation efficiency, but we compensate this with the lower E2E delay that P_{Agg} ensures compared to these two schemes, which is very important metric for real-time monitoring applications. Also our scheme achieves higher E2E delay in comparison with the Agg_{Off} scheme, but ensures the aggregation of the packets and important ES in the network. Finally, it is worth mentioning that our proposed P_{Agg} scheme ensures a good trade-off between the E2E transmission delay, the aggregation ratio in the network, and the energy consumption.

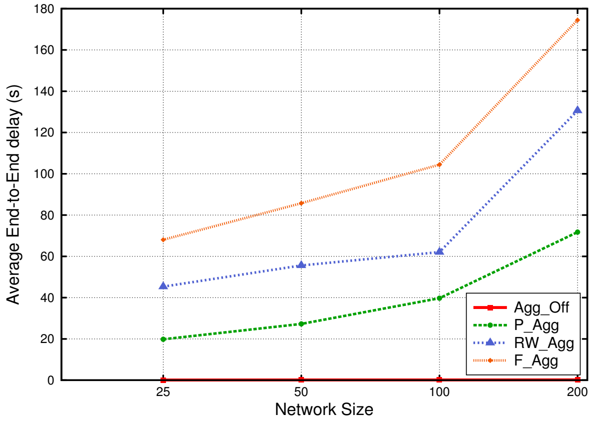

Figure 10: Average End-to-End delay under varying Network size: monitoring period interval = 60 s

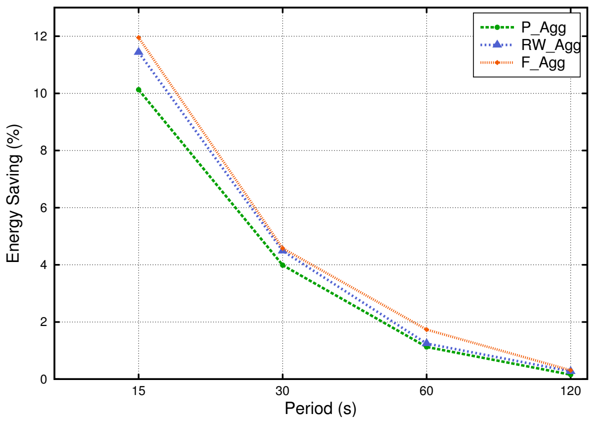

Figure 11: Energy Saving vs. Monitoring Period Intervals: Network Size = 200 nodes

V. CONCLUSION

In this work, we have investigated the problem of long end to end transmission delays incurred by the aggregation of periodic packets in cluster based WSNs. We have proposed a new data aggregation scheme based on a dynamic waiting time, which uses judicious decision making policies at the aggregator nodes level in order to determine the most appropriate time for aggregating the received packets, and forwarding the resulting packet towards the sink. Due to our novel way of setting the waiting time before aggregation (by leveraging the spatial correlation properties), the dissemination delay of periodic packets is reduced by roughly 50% on average as compared to randomized waiting/full aggregation schemes while the energy consumption is almost at the same level as these two schemes, which makes our scheme suitable

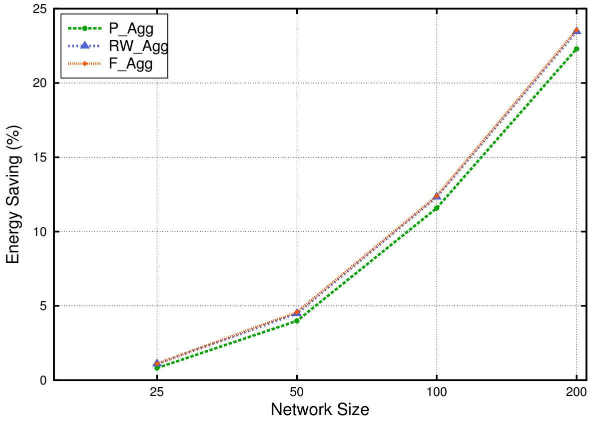

Figure 12: Energy Saving vs. Network Size: Monitoring Period Interval = 60 s

for most of real-time monitoring applications using WSNs technology. The performance evaluation results highlight the effectiveness of our scheme in terms of the achieved end to end delay, energy saving, aggregation ratio and efficiency. The proposed scheme can also address other aspects, for instance, aggregating emergency messages and scheduling packets transmission, which will be the focus of our future work.

VI. ACKNOWLEDGEMENT

This work was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre (www.lero.ie). This work was also supported by the Telecommunications Graduate Initiative (TGI) which is funded by the Higher Education Authority under the Programme for Research in Third-Level Institutions (PRTLI) Cycle 5 and co-funded under the European Regional Development Fund (ERDF).

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[3] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey," *Wireless Communications, IEEE*, vol. 14, no. 2, pp. 70–87, 2007.

[4] H. Yousefi, M. H. Yeganeh, N. Alinaghipour, and A. Movaghar, "Structure-free real-time data aggregation in wireless sensor networks," *Computer Communications*, vol. 35, no. 9, pp. 1132–1140, 2012.

[5] H. Li, K. Lin, and K. Li, "Energy-efficient and high-accuracy secure data aggregation in wireless sensor networks," *Computer Communications*, vol. 34, no. 4, pp. 591–597, 2011.

[6] Q. Wang, K. Ren, S. Yu, and W. Lou, "Dependable and secure sensor data storage with dynamic integrity assurance," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 1, pp. 9:1–9:24.

[7] G. Mathur, P. Desnoyers, P. Chukiu, D. Ganesan, and P. Shenoy, "Ultra-low power data storage for sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 4, pp. 33:1–33:34, 2009.

[8] A. Mitra, A. Banerjee, W. Najjar, D. Zeinalipour-Yazti, V. Kalogeraki, and D. Gunopulos, "High performance, low power sensor platforms featuring gigabyte scale storage," in *Third International Workshop on Measurement, Modeling, and Performance Analysis of Wireless Sensor Networks (SenMetrics 2005), San Diego, CA*, 2005.

[9] N. Bhatnagar and E. L. Miller, "Designing a secure reliable file system for sensor networks," in *Proceedings of the 2007 ACM workshop on Storage security and survivability, StorageSS '07*.

[10] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling data-centric routing in wireless sensor networks," in *IEEE infocom*, vol. 2, 2002, pp. 39–44.

[11] R. Rajagopalan and P. K. Varshney, "Data aggregation techniques in sensor networks: A survey," *Electrical Engineering and Computer Science.*, no. Paper 22, 2006.

[12] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on.* IEEE, 2000, p. 10.

[13] S. Lindsey and C. S. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," in *Aerospace conference proceedings, 2002. IEEE*, vol. 3. IEEE, 2002, pp. 3–1125.

[14] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 4, pp. 366–379, 2004.

[15] L. A. Villas, A. Boukerche, D. L. Guidoni, H. A. De Oliveira, R. B. De Araujo, and A. A. Loureiro, "An energy-aware spatio-temporal correlation mechanism to perform efficient data collection in wireless sensor networks," *Computer Communications*, vol. 36, no. 9, pp. 1054–1066, 2013.

[16] N. Aydin, F. Nait-Abdesselam, V. Pryyma, and D. Turgut, "Overlapping clusters algorithm in ad hoc networks," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE.* IEEE, 2010, pp. 1–5.

[17] A. Boukerche, M. Z. Ahmad, B. Turgut, and D. Turgut, "A taxonomy of routing protocols in sensor networks," *Algorithms Protoc. Wirel. Sens. Networks*, pp. 129–160, 2008.

[18] J. Ai, D. Turgut, and L. Bölöni, "A cluster-based energy balancing scheme in heterogeneous wireless sensor networks," in *Networking-ICN 2005.* Springer, 2005, pp. 467–474.

[19] K.-W. Fan, S. Liu, and P. Sinha, "Structure-free data aggregation in sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 8, pp. 929–942, 2007.

[20] S. Kwon, J. H. Ko, J. Kim, and C. Kim, "Dynamic timeout for data aggregation in wireless sensor networks," *Computer Networks*, vol. 55, no. 3, pp. 650–664, 2011.

[21] J. Y. Choi, J. W. Lee, K. Lee, S. Choi, W. H. Kwon, and H. S. Park, "Aggregation time control algorithm for time constrained data delivery in wireless sensor networks," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, vol. 2. IEEE, 2006, pp. 563–567.

[22] F. Hu and C. May, "Timing control during data aggregation operations of distributed sensor networks: Trade-off between query accuracy and response delay," *Information Technology Journal*, vol. 5, no. 5, pp. 759–778, 2006.

[23] I. H. Brahmi, S. Djahel, D. Magoni, and J. Murphy, "An efficient partial data aggregation scheme in wsns," in *Wireless Days (WD), 2014 IFIP.* IEEE, 2014, pp. 1–3.