An Efficient Partial Data Aggregation Scheme in WSNs

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—Highly accurate event detection makes Wireless Sensor Networks popular for real time monitoring applications. Wireless sensor systems that monitor physical and environmental conditions are expected to be deployed with high density, a situation which leads to spatial correlations and redundancy of collected data. Eliminating these redundancies extends the network lifetime by reducing energy consumption and enhances the velocity of transmitting emergency and periodic messages. In this work, we focus on the scenario where sensors are grouped into clusters. Each Cluster Head (CH) receives samplings from its Cluster Members (CMs), and decides when it should stop sampling, and starts transmitting the resulting packet from the aggregation process in order to reduce the end-to-end delay and ensuring the accuracy of the transmitted data. To this end, we propose a cluster based aggregation scheme which determines, at the CH level, the best timing for achieving a short delay, and provides an efficient buffer management strategy for maintaining low energy consumption. Evaluation results based on simulations show that our scheme achieves a good trade-off between energy consumption and end-to-end delay.

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

I. INTRODUCTION

Wireless Sensor Networks (WSNs) [1] is an emerging technology for event detection and information gathering. Their main feature is to cooperatively sense specified events of interest in the sensor field, and transmit them toward the sink for processing via multi-hop routing protocols. This type of network has become popular due to its low cost and high accuracy in detecting events. It has diverse application domains that include areas such as environment, health, agriculture, and military, where different physical conditions are monitored such as temperature, humidity, vibration, air quality etc.

Sensors are resources constrained devices with limited processing, storage capacity and battery power. A simple solution to this problem could be the replacement of the sensors' battery whenever it is depleted. However, this solution is not convenient, mainly when the sensors are used to monitor hostile or inaccessible areas like volcanos. In addition, sensor nodes density in the network may vary spatially and temporally depending on the application requirements [2]. With high density the reported data is likely to be redundant and correlated. We usually associate the primary energy consumption with communication: it is well known that the communication is often the most expensive activity in terms of energy [3], as the radio communication is the major source of energy consumption [4].

An effective solution for reducing the communication overhead is data aggregation. In-network processing and data 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, energy is conserved by eliminating redundancy and minimizing the transmission of raw data to the sink [5]. Delaying the transmission of data packets in order to wait for
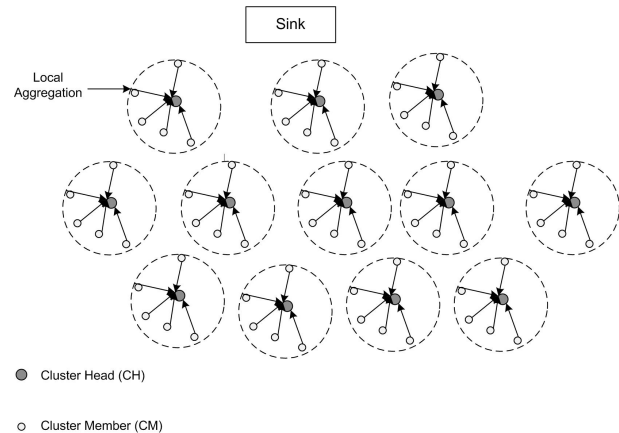

Figure 1: Clustering topology

more packets from neighboring nodes may increase the degree of aggregation [6], however, in real-time monitoring applications, it is compulsory to deliver the gathered readings in a timely manner. For this purpose, the solutions designed for WSNs should consider the reduction of both the end-to-end delay and the energy consumption in their conception.

The new generation sensor nodes have been enhanced with significant energy-efficient storage, processing capabilities and data management abilities [7]. Indeed, sensor nodes can be endowed with energy-efficient storage such as new-generation flash memory with several gigabytes of storage and low-power consumption [8], [9]. This emergence of low cost and high-capacity storage and processing prompts us to design a new cluster based aggregation scheme and buffer management strategy that satisfies the tradeoff between the energy consumption and end-to-end delay, by reducing the number of unnecessary redundant transmissions.

II. SYSTEM DESIGN

We consider in our scheme that sensor nodes are organised into clusters as shown in Figure 2 to make the aggregation and the communication easier and more efficient. Each CH gathers readings from its CMs, and aggregates them before sending the resulting packets toward the sink. We define the gathering time as the Local Waiting Time (WT_{Local}). This time is dynamic and determines when the CHs should aggregate the received readings and forward the resulting packets before the end of the monitoring period. These aggregated readings are then sent onwards towards the sink via multi-hop routing path. We assume that these sensors are equipped with buffer memory and queue as shown in the Figure 2. The CHs' buffer is used for storing the received packets from their CMs. Each CH waits for a minimum number of samples before it aggregates and forwards the aggregated packets. A copy of the aggregated
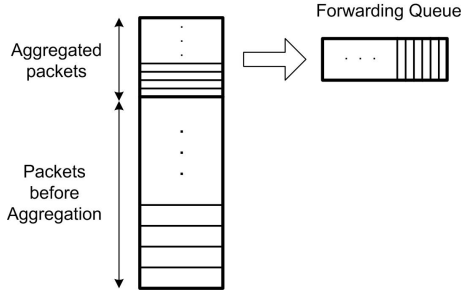
Figure 2: Representation of a Sensor node Buffer

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). The forwarding queue is used for storing the aggregated packets which are now ready to be compared against future packets. The samples stored in the buffer have a limited storage time. Therefore, we define a Time out (T_{out}) for the 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 waiting time and a T_{out} mechanism for the packets in the buffer such that they optimise the energy usage, buffer storage time, and transmission delay of packets, and finally reduce the overall traffic load in the network.

A. Waiting Time Computation

The CMs start periodically measuring given parameters, and forward the reading to their CH in the network. We assume that the CMs start collecting data at the same time (i.e. the beginning of each period). At the end of each monitoring period, the CMs start forwarding their readings (i.e. one reading per CM). The arrival time of the readings at the CH level is random. The aggregation process starts after receiving these first samples.

Our mechanism uses the following key parameters: a minimal number of samples received by the CH, depending on the application, the mean of the received samples values, the calculated standard deviation σ_{cal} and we also define a standard deviation threshold σ_{th} which depend on the application. These parameters are used to select appropriate time to start aggregating the received packets and forward the resulting packets.

When the first samples are received by the CHs, they start the aggregation process, comput the mean and standard deviation (σ_{cal}) of the received values, and compare them with σ_{th}. If the minimal number of required samples is reached, and the threshold condition is satisfied, the samples can be aggregated and sent onwards. If a received sample reports a value exceeding a given upper bound of the measured parameter (e.g. a vehicle out of control on the road), this sample is forwarded immediately as an emergency message.

Once the conditions of minimal number of required samples and σ_{cal} is met, the aggregated packets are forwarded and a copy of these aggregated packets is stored in the CH's buffer for limited period of time T_{out}.

B. Sensor Buffer Management Strategy

Since the buffer capacity and energy consumption are limited in WSN nodes, our second objective is to optimise the energy consumption of sensor nodes (CHs) by keeping the packets in the buffer an optimal time period. In this section we define a scheme for sensor buffer management and processing of the packets, and illustrate the T_{out} calculation, which ensures an optimal trade-off between energy and storage. Our T_{out} calculation is a function of the following parameters: the sensors nodes popularity, the frequency

of receiving the same packets from the same category, their priority level, the residual energy of each CH and the buffer size of the CHs.

$$T_{out} = f(R, F, W, E) \quad (1)$$

Where:
- R: popularity of the CH: computed based on the traffic load transiting trough this CH
- F: frequency of receiving the same packets from the same category
- W: priority level
- E: residual energy of each CH

We assume that the maximum energy and the buffer capacity of the nodes are both known in advance. The popularity of the node is defined by the ratio $R = \frac{PacketTransmitted}{PacketsReceived}$. The closer the nodes are to the sink, the longer they retain their packets in the buffer, since the traffic density is higher closer to the sink.

We assume that:

$$T_{out} = [0, T_{outMax}] \quad (2)$$

The initial $T_{outMax0}$ depends on the energy and the buffer capacity of the sensor nodes. At $t = 0$ (when the network starts running) the buffer is empty and the energy capacity of the node is at its max we have $T_{out} = T_{outMax0}$. In order to give more priority to the packets with high priority level, every time a packet (P_n) is received by the CHs, they check the category of the packet. For each type and category, we adjust the T_{outMax} according to the priority level of the packet received by defining a weighting. The $T_{outMax0}$ is proportional to the level of priority of the packet. If a packet has a high priority level, then the probability to retain the packet longer in the buffer is higher.

- The $T_{outMax0Per1} = T_{outMax0 \times WPer1}$
- The $T_{outMax0Per2} = T_{outMax0 \times WPer2}$
- The $T_{outMax0Per3} = T_{outMax0 \times WPer3}$

Where:

$$W_{PerN} = \frac{Priority_n}{Priority_{max}} \quad (3)$$

Once the buffer is full the sensors execute Algorithm 1. When new packets are received, the CHs have to decide which packets are the most essential for future aggregation. For this, the CHs check the frequency of receiving the same packets from the same category, which will indicate any unusual event. The packet with the smallest T_{out} and frequency and the lowest priority will be deleted first. Next, the CHs calculate the new T_{out} based on the remaining battery power of the CHs. If the same packet is received many times, the T_{out} of this packet will be calculated according to the Average Inter-Arrival time (T_{AIA}) of this packet, for the purpose of extending the storage of this packet for future aggregation.

III. PERFORMANCE EVALUATION

Table I: Summary of simulation parameters

Parameters	Value
Routing Protocol	AODV
Propagation mode	TwoRayGround
Packet Size	64 Bytes
Number of nodes	100
Inter-node distance	10 meters
Monitoring period interval	15 s, 30 s, 60 s, 120 s
Simulation time	3600 seconds
No. of simulation runs	50
Topology	grid

In this section, we evaluate and discuss the performance of our proposed scheme (Partial Aggregation: P_{Agg}) using NS-2.35 [10]

Algorithm 1 Sensor Buffer Management Algorithm

1: At $t = T$ (When the buffer is full)
2: **if** New packet received **then**
3:     Check if the packet already exists in the buffer
4:     **if** Packet-exists **then**
5:         Check T_{AIA}
6:         **if** $T_{outMax} \leq T_{AIA}$ **then**
7:             $T_{outMaxPn} = T_{outMaxPn0}[(\frac{E_{resPn}}{E_{Max}}) * (R)] + T_{AIA}$
8:         **else**
9:             $T_{outMaxPn} = T_{outMaxPn0}[(\frac{E_{resPn}}{E_{Max}}) * (R)]$
10:        **end if**
11:    **end if**
12: **end if**

network simulator. We have compared thP_{Agg} to the Full aggregation (F_{Agg}) where the CHs wait to receive packets from all their CMs, and Aggregation Off (Agg_Off) where the received packets are forward immediately upon their reception. Our solution is static cluster based and the cost of deployment and maintenance of the clusters is not considered. The generation of the packets by the CMs and their values are random during each periodic monitoring interval. Once the first packets are received by the CHs, the aggregation process starts by computing the standard deviation (σ_{cal}) of the received packets value and comparing it with σ_{th}, in order to decide when the aggregated packets should be sent. Once the condition is met the CH aggregates the received packets, forwards them and stores a copy in their buffer.

We summarize in Table I the default parameters used in the simulation. The primary metrics evaluated are: (i) 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 to cross the network and reach the sink. (ii) The Energy Consumption (EC) defined by the ratio of the total energy used when the aggregation is enabled to the total energy used when the aggregation is disabled.

The results plotted in Figure 3 compare the average end-to-end delay of the three schemes. In this scenario we have set the network size to 100 nodes and varied the monitoring period. From these results, we can see that the Agg_{off} achieves negligible E2E delay, this is explained by the immediate forwarding of the received packets from the CMs. The early aggregation in the P_{Agg} leads to a lower E2E delay compared to the F_{Agg}, this is due to the long waiting time scheme used in F_{Agg}. We notice that the difference achieved is equal to 20% on average, and increases to approximately 60% with the increase of the monitoring period. From this we can say that the long waiting time is unnecessary in F_{Agg}.

In Figure 4 we notice that the energy consumption achieved by P_{Agg} is as low as that achieved by F_{Agg}. Also the energy consumption increases with the monitoring period interval. The main objective of our work is to achieve a lower end-to-end delay of the packet transmission, while keeping a low energy consumption in the network. We can conclude from our results that our scheme succeeds in this.

IV. CONCLUSION

We have proposed in this work a new cluster based aggregation scheme for periodic monitoring to deal with the problem of long waiting times in WSNs at the CH level. We have first defined a solution based on a dynamic waiting time which uses an optimal timing for aggregating and forwarding the packets. Next, we proposed a buffer management strategy for processing the stored packets in an efficient way such that the energy power is saved and the optimally aggregated packets are accurate. The performance evaluation results have proven the efficiency of our scheme in terms of end-to-end delay and energy consumption.
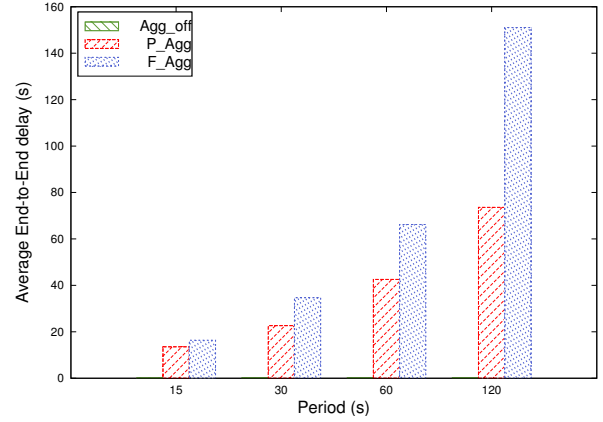

Figure 3: Average End-to-End delay under varying monitoring period interval: Network Size = 100 nodes
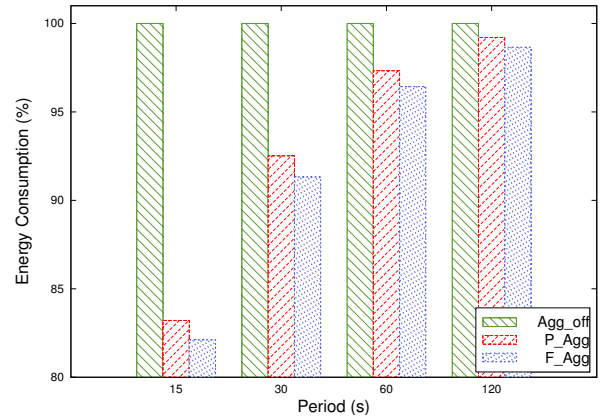

Figure 4: Energy Consumption vs. Monitoring Period Intervals: Network Size = 100 nodes

V. ACKNOWLEDGEMENT

REFERENCES

[1] I. F. Akyildiz, et al., "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
[2] L. A. Villas, et al., "A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks," *Ad Hoc Networks*, 2011.
[3] G. J. Pottie, et al., "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.
[4] A. Somov, et al., "A methodology for power consumption evaluation of wireless sensor networks," in *IEEE ETFA 2009*. IEEE, 2009, pp. 1–8.
[5] E. Fasolo, et al., "In-network aggregation techniques for wireless sensor networks: a survey," *Wireless Communications, IEEE*, vol. 14, no. 2, pp. 70–87, 2007.
[6] H. Yousefi, et al., "Structure-free real-time data aggregation in wireless sensor networks," *Computer Communications*, vol. 35, no. 9, pp. 1132–1140, 2012.

[7] Q. Wang, et al., "Dependable and secure sensor data storage with dynamic integrity assurance," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 1, p. 9, 2011.

[8] G. Mathur, et al., "Ultra-low power data storage for sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 4, p. 33, 2009.

[9] A. Mitra, et al., "High performance, low power sensor platforms featuring gigabyte scale storage," in *SenMetrics 2005, San Diego, CA*, 2005.

[10] "The network simulator ns-2 (http://www.isi.edu/nsnam/ns/)."