Efficient Tree-based Aggregation and Processing Time for Wireless Sensor Networks

David Fotue *david.fotue@univ-yaounde1.cm*
Département d'Informatique
Ecole Normale Supérieure de Yaoundé
BP 47, Yaoundé, Cameroun

Houda Labiod *houda.labiod@telecom-paristech.fr*
Informatique Réseaux et Sécurité (INFRES)
Télécom ParisTech
46, Rue Barrault, France

Abstract

Tree-based data aggregation suffers from increased data delivery time because the parents must wait for the data from their leaves. In this paper, we propose an Efficient Tree-based Aggregation and Processing Time (ETAPT) algorithm using Appropriate Data Aggregation and Processing Time (ADAPT) metric. A tree structure is built out from the sink, electing sensors having the highest degree of connectivity as parents; others are considered as leaves. Given the maximum acceptable latency, ETAPT's algorithm takes into account the position of parents, their number of leaves and the depth of the tree, in order to compute an optimal ADAPT time to parents with more leaves, so increasing data aggregation gain and ensuring enough time to process data from leaves. Simulations were performed in order to validate our ETAPT. The results obtained show that our ETAPT provides a higher data aggregation gain, with lower energy consumed and end-to-end delay compared to Aggregation Time Control (ATC) and Data Aggregation Supported by Dynamic Routing (DASDR).

Keywords: Wireless Sensor Networks, Aggregation Time, Aggregation Gain, Parent, Leaf.

1. INTRODUCTION

Recent innovations in micro-electro-mechanical technologies bring significant advantages to the development of low-power, low-cost multifunctional sensors equipped with storage, computing and communication capabilities. Wireless Sensor Networks (WSNs) are ad hoc wireless networks that consist of a large number of small devices, known as sensors, scattered over a particular geographical area [1]. In recent years, WSNs are seen as a reality, due to the potential applications in various domains such as industrial, biological, medical, military, nuclear science, forest fire detection and so on. The lack of a predefined communication infrastructure increases the challenges of designing of communication techniques for these networks, especially in hostile environments, where it is often difficult to replace sensor batteries after deployment and where communication infrastructures are not accessible or available.

In WSNs, all sensors send their data towards the central sink, which is the final recipient of the sensed information. Sensors are equipped with a limited amount of storage capacity, and powered by batteries with a finite life, making power saving an important issue in achieving long-lived wireless multi-hop networks [2]. In WSNs, each sensor covers a defined area, collecting local data and sending it towards the main sink. It may happen that some sensors deployed in the monitored area sense common data. Therefore, much energy will be wasted if all these data are forwarded towards the sink. Data aggregation schemes exploiting in-network processing have been proposed as efficient techniques to conserve energy by locally processing the data as much as possible in order to reduce the amount of data transmitted by each sensor towards the sink [1].

As the sink has to receive the data from sensors in a timely manner, this data aggregation has a relationship with the data aggregation time [3]. We need to determine the data aggregation time that each parent in the tree should spend in aggregating the data sent from its leaves. As the network topology can vary, some parents might have many leaves, making it very expensive for a parent to store all incoming data in its buffer. Failing to account for data aggregation time may lead to a longer waiting time for each parent and increase overall data delivery latency.

We propose an Efficient Tree-based Aggregation and Processing Time (ETAPT) algorithm using Appropriate Data Aggregation and Processing Time (ADAPT) metric. Given the maximum acceptable latency, ETAPT's algorithm takes into account the position of parents, their number of leaves and the depth of the tree, in order to compute an optimal ADAPT time to parents with more leaves, so increasing data aggregation gain and ensuring enough time to process data from leaves. Simulations were performed in order to validate ETAPT. The results obtained show that our ETAPT provides a higher data aggregation gain, with lower energy consumed and end-to-end delay compared to Aggregation Time Control (ATC) [4] and Data Aggregation Supported by Dynamic Routing (DASDR) [5].

The remainder of this paper is organized as follows: Section 2 outlines related work. Section 3 formulates the problem and presents our proposition. Section 4 presents our model and describes notation. Section 5 presents our approach. Section 6 presents performance metrics and comparative results and Section 7 concludes the paper.

2. RELATED WORK

As our WSN focuses on gathered the data from the environment, it is important to forward the data in a timely manner towards the sink. Several approaches have been proposed concerning data aggregation time. Actually, we use tree and cluster structure for data aggregation because they are useful in environment monitoring where the maximum data values received by the sink provide the most useful information [2].

[4] proposes dynamic Aggregation Time Control (ATC) based on the number of leaves of the root node in a tree-structure. ATC adjusts the aggregation time of the sensor.to assign more aggregation time to sensors having more children in order to give more possibilities to aggregate the data. Simulations show that ATC has a high aggregation gain. However, ATC cannot be adopted to the multi hop sensor networks since it requires the global knowledge of the network. In addition, the broadcast scheme used during the construction of the tree needs a high communication overhead and decreases network performance. [5] proposes Data Aggregation Supported by Dynamic Routing (DASDR), which can adapt to different scenarios without incurring much overhead. Sensor nodes that monitor events are concentrated in space as far as possible and data packets flow to the sink along different paths. Dynamic routing constructs a depth potential field, which aims to guarantee that packets will reach the sink eventually and a queue potential field, which makes packets more spatially convergent, making data aggregation more efficient. Simulations show that DASDR improves the data aggregation ratio, saves energy, and scales well with network size. [6] focuses on a real-time data delivery, but do not take the data aggregation and processing into account. [7] proposes a cascading time-out in which sensors schedule their time-outs based on their position in the aggregation tree. A sensor's time-out happens after its leaves' time-outs, so enabling a sensor to aggregate the data from all its children. [8] computes the data aggregation time-out for clustered WSNs. The time-out is calculated for each sub-tree in the cluster taking into account packet transmission and cascading delay, leading to a reduction in aggregation time and energy use. [9] develops an approach which delivers the data to the sink within the deadline. They estimate the time-out of each sensor in the tree, so that the data generated by each sensor is delivered to the sink before the deadline. [10] proposes to construct a centralized and decentralized structure in the network in order to reduce the transmission delay during the collection of data. [11] proposes a Delay-minimized Energy-efficient Data Aggregation (DEDA) algorithm to minimize data aggregation latency. The physical

distance between sensors is takes into account in DEDA to save the transmission energy in order to improve network lifetime.

Our proposal is similar to the one used by [4] and [5]. However we take into account the position of parents, their number of leaves and the depth of the tree, in such a way those parents with more leaves will be dynamically allocated an appropriate aggregation time, so maximizing the data aggregation gain and improving network performance.

3. PROBLEM STATEMENT AND PROPOSITION

In this section, we formulate the problem addressed in this paper and present our proposal.

3.1 Problem Statement

In our context (spatial aggregation), the data gathered by sensors that are close to each other do not vary much over time. Tree-based data aggregation results in increased data delivery time because the parents must wait for the data from their leaves. Since the network topology can be random, as shown in Figure 1, some parents may have many leaves, making it very expensive for a parent to store all incoming data in its buffer. [4] shows that if a parent waits for the data from its leaves for a long time, it collects more data and hence Data Aggregation Gain (DAG) increases. DAG is the ratio of traffic reduction due to aggregation to the total traffic without aggregation [12]. However, this long waiting time means that the data delivery time to the sink

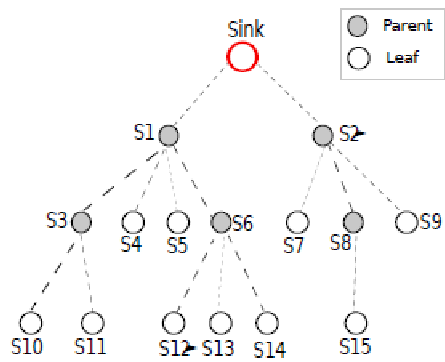

FIGURE 1: WSN: Tree Construction.

may increase. Thus, it is important to consider the time taken by parents to aggregate and process the data, because it takes more time to aggregate and process the data than to transmit the data towards the sink. Lacking of attention to the data aggregation and processing time may increase the overall data delivery latency or reduce the DAG. [13] shows that neglected the data aggregation and processing time may increase the overall data delivery latency or reduce the DAG.

3.2 Proposition : ETAPT Algorithm

We propose an Efficient Tree-based Aggregation and Processing Time (ETAPT) algorithm using the Appropriate Data Aggregation and Processing Time (ADAPT) metric to calculate the data aggregation and processing time for parent nodes as shown in Figure 2. After have been built the tree out from the sink, in order to elect sensors with the highest degree of connectivity as parents and sensors with the lowest degree of connectivity as leaves as shown in Figure 2. Given the maximum acceptable latency, ETAPT's calculation takes into account the position of parents, their number of leaves and the depth of the tree, in order to compute for each parent an optimal ADAPT before aggregating and processing the data from its leaves. So, allocating an appropriate aggregation time (A_{ggTime}) to parents with more leaves in order to increase the DAG, thus ensuring enough time to process the data from leaves.

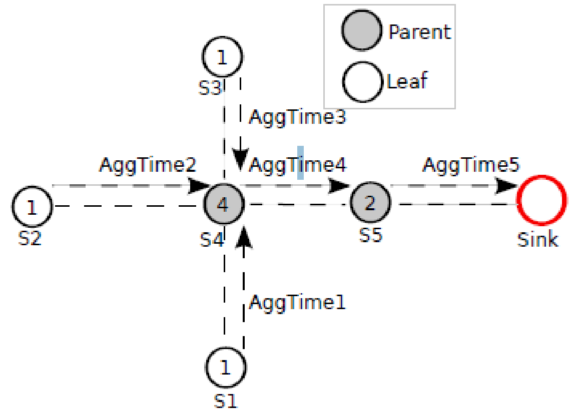

FIGURE 2: Distribution of Aggregation Time.

4. NETWORK MODEL AND NOTATION

4.1 Network Model

The proposed WSN can be modeled as a connected graph G = (S, E), where S is the set of N fixed sensors, and E is the set of wireless links. We use the locality model suggested in [14] to determine network connectivity. The probability of a link between two sensor nodes S_i and S_j is given by:

$$P = \begin{cases} 1 \; if \; D(S_i, S_j) \leqslant R \\ 0 \; if \; D(S_i, S_j) > R \end{cases} \qquad (1)$$

Where $D(S_i, S_j)$ is the Euclidean distance between sensors S_i and S_j, and R is the locality radius.

4.2 Notation

Let s € S. Let Path (s_1, s_k) be the sequence: s_1, s_2...s_k. We define Hop (s_1, s_k) = k-1 as the number of hops from sensor s_1 to s_k. Let d(s) be the degree of sensor s. δ is the minimum transmission time between two sensors of the same Hop in the tree, and ensures that there is a difference in the waiting times at consecutive Hop of the tree. We define:

$$L_{EAF} = s \mid s \in S, d(s) = 1 \qquad (2)$$

as the set on leaves in the tree,

$$M = S - L_{EAF} - sink \qquad (3)$$

as the set on parents in the tree and,

$$Hop_{Distance}(s) = d(s, sink) \qquad (4)$$

as the number of hops of the Path (s, sink). We recall that Hop (sink) = 0. Let the depth of the tree be:

$$Depth = Max_{s \in L_{EAF}}(d(s, sink)) \qquad (5)$$

the number of hops from the sink to the deepest leaf in the tree (the maximum number of hops towards the sink in the tree). We define the weighted length of the Path (s_1, s_k) as:

$$WPath(s_1, s_k) = \sum_{i=1}^{k-1} d(s_i) \qquad (6)$$

the sum of the degrees of the descendant sensors. Let L'_{EAF} be all the leaves in a subtree rooted at sensor s, s € M. We define the maximum weighted depth of the subtree as:

$$MaxWPath(s) = Max_{s_i \in L'_{EAF}}(WPath(s_i, s)) \qquad (7)$$

the maximum degree of all the descendant sensors in L'_{EAF} to root to sensor s in the subtree. For all (s € L_{EAF}), M_{ax}WP(s) = 0. Finally, T_{max} be the maximum acceptable latency.

In the following Section 5, we describe our ETAPT algorithm.

5. EFFICIENT TREE-BASED AGGREGATION AND PROCESSING TIME (ETAPT) ALGORITHM

5.1 Assumptions

We assume in our approach that:

- Sensors are deployed in an area of size L.
- Sensors are homogeneous (same computing, memory...) and fixed.
- Each sensor maintains a list of identifies (Id) of its neighbors.
- Each sensor keeps track of its own degree of connectivity value d(s).
- Each leaf has one parent that is responsible for forwarding the received data towards the sink.
- Leaves can only sense and transmit their data to their parents.
- Aggregation of multiple packets results in one packet.
- A single sink is the final recipient of all the sensed data.
- T_{max} is the maximum acceptable latency.

As shown in Figure 2, the tree is built out from the sink, taking into account the degree of connectivity of sensors d(s). The sensors with the highest degree of connectivity are selected as parents and those with lowest degree of connectivity as leaves. Given T_{max}, ETAPT will determine the ADAPT for each parent based on its position, its number of leaves and the depth of the tree. We assume that every sensor generates a data packet of the same length periodically, and multiple packets can be combined into one packet after the data aggregation process. Any packets arriving after the ADAPT time calculation are discarded. The algorithm consists of two major procedures: M_{ax}WPath, $Hop_{Distance}$, degree of sensor and average waiting and aggregation time's determination.

5.2 M_{ax}WPath, $Hop_{Distance}$ and Degree of Sensor Determination

The first step consists in determining by each sensor in the tree: its degree d(s), M_{ax}WPath(s) and $Hop_{Distance}$(s).The Sink broadcasts a beacon message as a RequestM_{ax}WPath with a $Hop_{Distance}$ field, which is incremented as the beacon travels through the tree as shown in Figure 3(a). Every sensor, on receiving the RequestM_{ax}WPath, adds its $Hop_{Distance}$ value to the beacon, and forwards it to its neighbours. In order to reply to the RequestM_{ax}WPath message, every sensor, starting from the deepest leaf, calculates its own degree and the M_{ax}WPath to its parent, generates a ReplyM_{ax}WPath message and forwards it to its parent as shown in Figure 3(b).

FIGURE 3: Beacon Structure.

Suppose that s € M is a parent. It calculates and saves its own d(s) and M_{ax}WPath (s) based on the ReplyM_{ax}WPath it receives, generates a new ReplyM_{ax}WPath including its own M_{ax}WPath and forwards it to its parent. The ReplyM_{ax}WPath messages are propagated in a cascading manner along the tree towards the sink. When the sink has received all the ReplyM_{ax}WPath messages, it chooses the largest M_{ax}WPath value from among them and sets:

$$MaxWPath(Sink) = Largest(MaxWPath). \qquad (8)$$

5.3 Determination of average waiting and aggregation times

The second phase of ETAPT consists in determining the average waiting time Avg per sensor in order to determine the aggregation Time A_{ggTime} in the tree. The Avg for each sensor (s) is based on T_{max}, M_{ax}WP(s) and Hop (s). When the sink receives a request from an external user specifying Tmax, the sink, based on the information it received in the first step, calculates the Avg per sensor and A_{ggTime} in the tree as follows:

$$Avg_{wait} = \frac{(T_{max} - \delta * Depth)}{MaxWPath(sink)} \qquad (9)$$

We assume that T_{max} > (Depth x δ). After the sink has calculated the Avg, it broadcasts a new beacon message through the network including T_{max} and Avg. Every sensor, on receiving the new beacon message, calculates its A_{ggTime} as follows:

$$Agg_{Time} = Avg_{wait} * MaxWPath(s) + (Depth - Hop_{Distance}(s)) * \delta \qquad (10)$$

δ is the minimum transmission time between two sensors of the same $Hop_{Distance}$ in the tree.

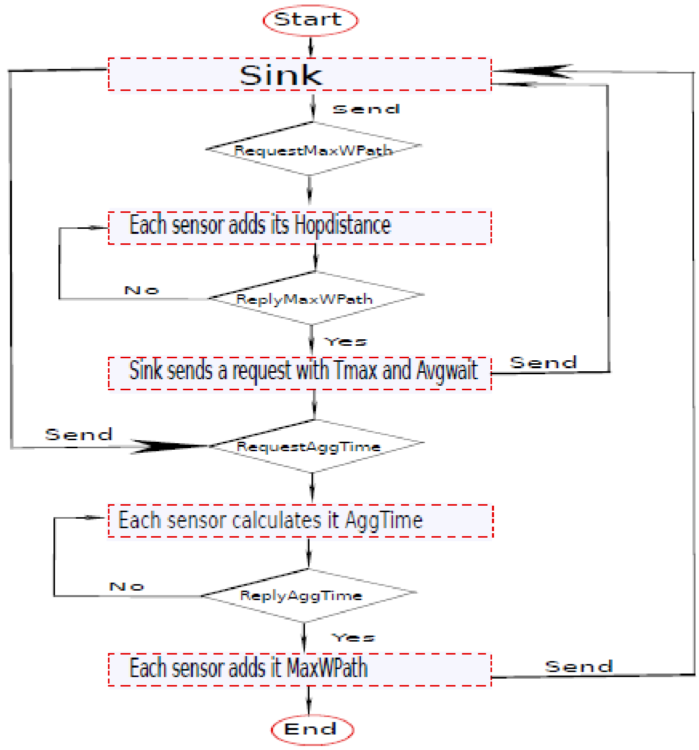

FIGURE 4: ETAPT Algorithm.

5.4 Illustration

Consider a simple topology consisting of 15 sensors as shown in Figure 1. We want to calculate d(s), Hop (s), $M_{ax}WP$ (s) and A_{ggTime} (s) for each sensor in the tree. We suppose that T_{max} = 5s and δ = 0.2s. Taking into account equation (9), the Avg = 0.64s and the Depth = 3. The ADAPT time calculation is summarized in Table 1.

S	d (S)	$Hop_{Distance}$ (S)	$MaxWPath$ (S)	Agg_{Time} (S)
S_1	5	1	7	4.88
S_2	4	1	4	2.96
S_3	3	2	2	1.48
S_4	1	2	0	0.20
S_5	1	2	0	0.20
S_6	4	2	3	2.12
S_7	1	2	0	0.20
S_8	2	2	1	0.84
S_9	1	2	0	0.20
S_{10}	1	3	0	0
S_{11}	1	3	0	0
S_{12}	1	3	0	0
S_{13}	1	3	0	0
S_{14}	1	3	0	0
S_{15}	1	3	0	0

TABLE 1: ADAPT Calculation.

In the following Section 6, we define the performance metrics and present comparative results.

6. PERFORMANCE METRICS AND COMPARATIVE RESULTS
6.1 Performance Metrics
The following metrics are used to evaluate our approach:

- Data Aggregation Gain (DAG)

DAG is defined as the ratio of the benefit of traffic reduction due to aggregation to the total traffic generated without aggregation.

$$DAG = 1 - \frac{P_{Aggregated}}{\sum_{i=1}^{N} P_{Generated_i}} \qquad (11)$$

$P_{Aggregated}$ is the total number of data packets aggregated by parents.

- Aggregation Time (AggTime)

A_{ggTime} is defined as the appropriate time need by a parent to aggregate the data from its leaves.

- End-to-End Delay (E2EDelay)

$Delay_{E2E}$ is the average of the time difference between sensed data leaving a sensor and it being received by the sink.

$P_{Received}$ is the total number of data packets received by the sink. $T_{Received}$ is the reception time at the sink, $T_{Transmission}$ is the transmission time from each sensor. The lower the value, the more promptly is data delivered to the sink.

$$Delay_{E2E} = \frac{\sum_{i=1}^{P_{Received}} (T_{Received_i} - T_{Ransmission_i})}{P_{Received}} \qquad (12)$$

- Energy Consumed (EC)

Often, sensors are deployed in a hostile environment where replacing the batteries is not always possible. A good choice of energy model is essential to optimize sensor network lifetime. Our approach assumes that sensors are usually in the active mode. The energy model used is the same as in [15]. For each pair of sensors (s_i, s_j), the energy consumed when sending a packet of m bits over a distance D can be calculated as:

Sending sensor energy consumption:

$$E_{Ti}(m, D) = E_{elec} * m + E_{amp} * m * D^2 \qquad (13)$$

Receiving sensor energy consumption:

$$E_{Rj}(m) = E_{elec} * m \qquad (14)$$

The total energy consumed by each pair (s_i, s_j) is:

$$E_T(m, D) = E_{Ti}(m, D) + E_{Rj}(m) \qquad (15)$$

E_{Ti} is the energy consumed for the transmission of a packet by the source S_i, E_{Rj} is the energy consumed to receive a packet s_j, E_{elec} is the energy consumed to run the transmitter and receiver, E_{amp} is the energy used by the amplifier and D is the Euclidean distance between s_i and s_j.

6.2 Simulation Set-up

We implemented a simulation of our network topology using QualNet 5.0. A topology is totally described by the number of stationary sensors N belonging to the network and their locations.

Throughout our analysis, we deploy 100 fixed sensor nodes inside a square area L. The sink is placed at the top left corner of L. During the execution of our simulations, a given source and destination pair remains in the evaluated set until communication between them fails due to energy depletion. We repeated the experiments 20 times for the same topology, with the 95% confidence interval of each data. We took the average value of these 20 runs. Initially, each sensor was charged with an energy of 10^4 Joules. In the analysis, we set T_{max} = 3s, 4s, 5s, 6s.

The parameters are described in Table 2.

Parameters	Description	Value
E	Full energy of sensor	10000 Joules (J)
E_{elec}	Energy of trans/receiver	50 (nJ/bit)
E_{amp}	Energy of amplifier	100 (pJ/bit)
L	Simulation area	1000m x 1000m
P_{Length}	Packet length	2 Kbits
Traffic rate	UDP traffic	4 packets/sec
MAC	MAC layer	IEEE 802.11b
T_{max}	Maximum acceptable latency	Between [3, 4, 5, 6]s
B	Bandwidth	128 (kbps)
R	Locality radius (m)	20m
N	Number of sensors	Between [20...100]

TABLE 2: Simulation Parameters

6.3 Comparative Results

We ran simulation to compare our ETAPT strategy vith ATC [4] and DASDR [5] described in Section 2. Figure 5 depicts the evolution of DAG as a function of T_{max}. We can see that as T_{max} increases, the DAG increases for all the three methods. This shows that as T_{max} increases, each parent has enough time to aggregate its data efficiently. ETAPT, with an average DAG of 90%, outperforms DASDR and ATC, which give 84% and 73.5% respectively. This is because, in ETAPT, the Agg_{Time} of a leaf is proportional to MaxWP (leaf). A leaf with a small MaxWP should transmit the data quickly to its parent; only leaves having the same MaxWP value have the same Agg_{Time}. However, DASDR and ATC use a cascading time-out. This means that sensors at the same Hop in the tree have the same A_{ggTime}, consequently increasing the amount of data loss due to congestion at intermediate parents.

We now evaluate the evolution of DAG as a function of the number of sensors, as shown in Figure 6. In the analysis, we set T_{max} = 3s and we observe the evolution. We see a decreasing of DAG from [60-80] sensors, that is due to the fact that some leaves are disconnected to their parents resulting in a tree with disconnected sub-trees. We can see that for all algorithms, as the number of sensors increases, DAG also increases in each algorithm. That means that the three algorithms continue to deliver data accurately towards the sink as the number of sensors increases. ETAPT achieves the best DAG with an average of 86.4%, compared to 78.4% for DASDR and 71.4% for ATC.

After a packet has been sent along a path P_i ($_i$=1....,k), we must perform an energy reduction operation on each sensor along the path except for the sink. Thus, after a packet is sent by a sensor, the energy level of that sensor is decremented by the amount of energy required to send the data packet. A sensor is considered non-functional if its energy level reaches zero. Figure 7 shows the evolution of the total EC for different techniques with a varying value of T_{max}.

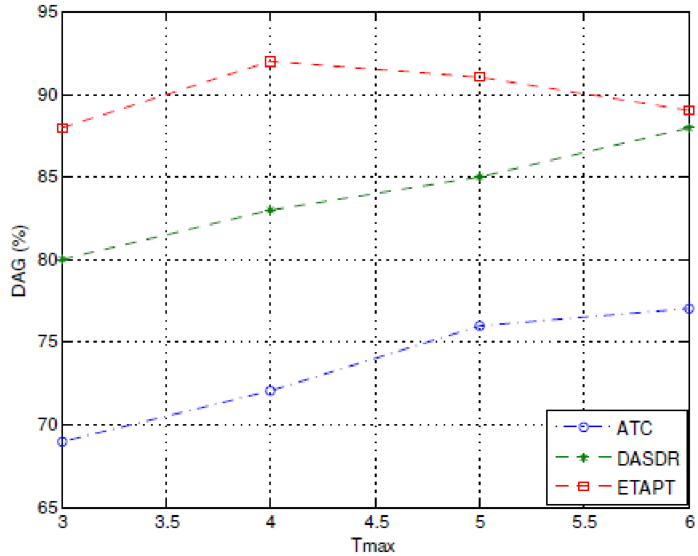

FIGURE 5: Evolution of DAG vs T_{max}.

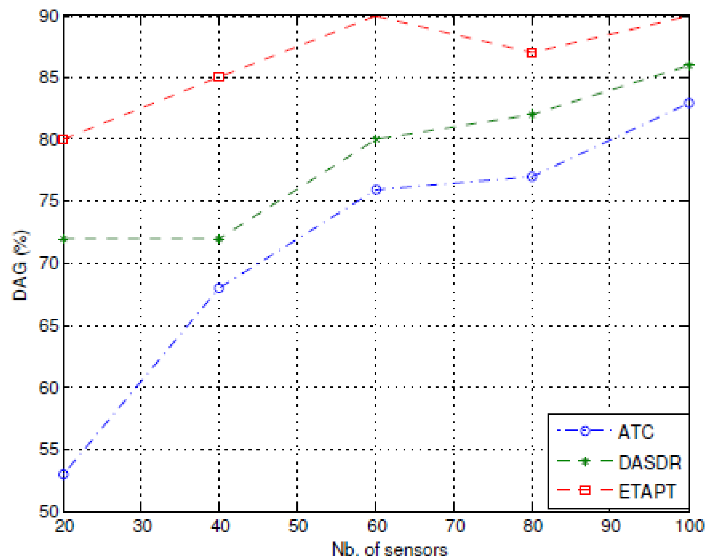

FIGURE 6: Evolution of DAG vs Number of sensors.

We observe that ATC and DASDR have a higher energy consumption than ETAPT. That is due to the fact that in the construction of the tree, we elect sensors having the highest degree of connectivity as parents instead of these with the highest identifier, as in ATC and DASDR. Thus, each sensor has exactly one parent that forwards its data, considerably reducing concurrent transmissions in the network. Our proposal reduces the total EC compared to DASDR and ATC by around 35% and 67% respectively. We evaluated the evolution of the total EC with increasing number of sensors, as shown in Figure 8. We observe a decreasing of EC with increasing number of sensors. That is due to the fact that in dense network, parent nodes might have many leaves which helps by reducing the number of parents necessary to transmit the data in the tree, and hence reduces the EC. The average maximum energy is obtained by ATC with around 45J. An improvement is obtained by DASDR, which uses only around 25J. ETAPT outperforms both, with an average EC of just 16J.

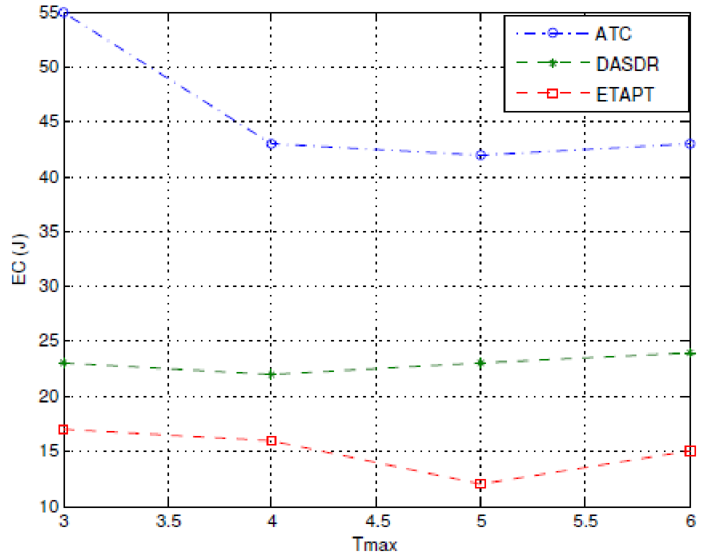

FIGURE 7: Evolution of EC vs T_{max}.

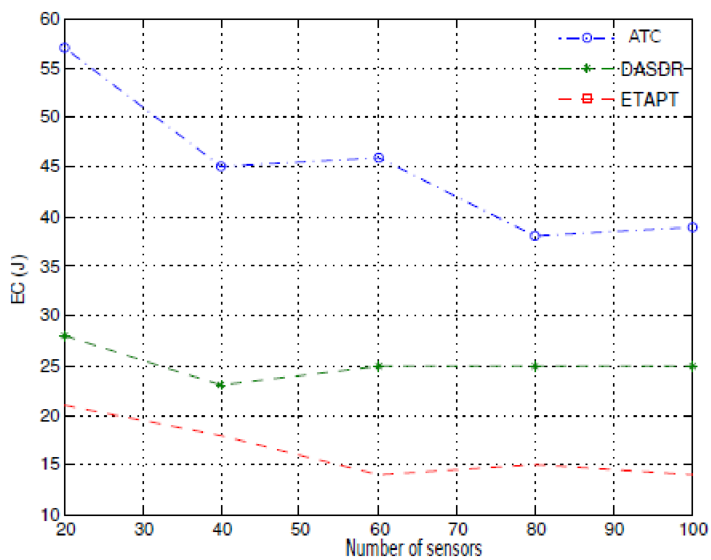

FIGURE 8: Evolution of EC vs Number of sensors.

Figure 9 shows Agg_{Time} vs. the locality radius. In this analysis, we set T_{max} = 6s, and vary the locality radius of sensors among [20, 30, 40, 50, 60] m. We can see that as locality radius increases, the Agg_{Time} decreases in all methods. That is because increasing the locality radius creates a disjoint network in which some sensors are not connected. This decreases the degree of connectivity of parents, and considerably reduces the Agg_{Time} of each parent. ETAPT reduces the Agg_{Time} compared to DASDR and ATC by around 31% and 60% respectively.

Figure 10 depicts the evolution of Agg_{Time} vs. the depth of the network. We set T_{max} = 6s, and vary the depth of the network among [3, 4, 5, 6]. As we have seen in Section 5, Agg_{Time} is a function of the depth of the network. We observe that as the depth of the network increases, Agg_{Time} also

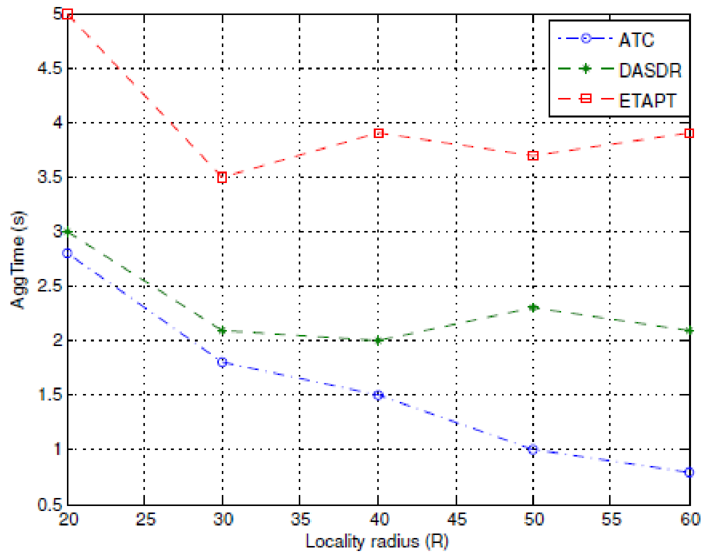

FIGURE 9: Evolution of A_{ggtime} vs Locality radius.

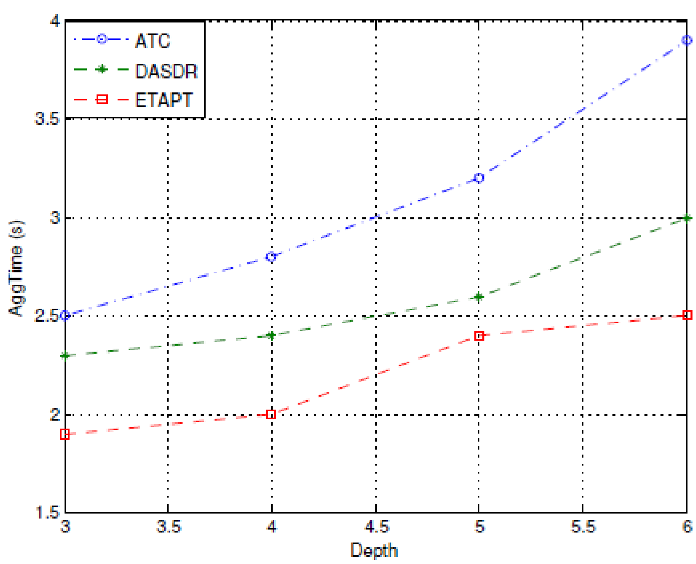

FIGURE 10: Evolution of A_{ggtime} vs Depth.

increases because, the deeper the tree, the more time parents in the tree will need to aggregate the data from leaves. In all three methods, while increasing the Depth, ETAPT reduces the Agg_{Time} compared to DASDR and ATC by around 17% and 40% respectively.

Figure 11 depicts the evolution of $Delay_{E2E}$ vs. the degree of connectivity. We set T_{max} = 6s, and vary the degree of connectivity of the network among [5, 10, 15, 20] with a network consisting of 200 sensors. ETAPT has a smaller $Delay_{E2E}$ compared to DASDR and ATC. This is because there is no need for each parent to synchronize with other parents in the tree before sending data.

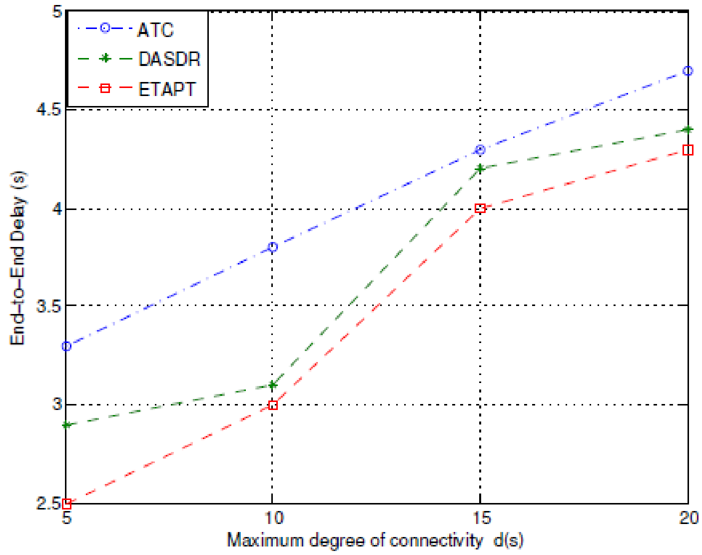

FIGURE 11: Evolution of of $Delay_{E2E}$ vs Degree of connectivity.

7. CONCLUSION

In this paper, we have proposed an efficient ETAPT algorithm using the ADAPT metric. Given the maximum acceptable latency, ETAPT's calculation takes into account the position of each parent, its number of leaves and the depth of the tree, allocating an ADAPT time to parents with more leaves, so increasing the data aggregation gain and ensuring enough time to process data from leaves. The results obtained show that our ETAPT provides a higher data aggregation gain with lower energy consumed, Agg_{Time} and $Delay_{E2E}$ compared to the alternative DASDR and ATC methods. Our suggested ETAPT algorithm is particularly useful in resource-constrained networks, since it does not need synchronization among sensors in the network.

In the future, we will take into account the cost of maintaining the tree in dynamic networks, evaluate the overhead as proposed in [16]. Later, we will study the relationship between waiting time and data aggregation gain in order to make it scalable in more complex WSNs.

8. REFERENCES

[1] D.Fotue, F.Melakessou, H.Labiod and T.~Engel. “Effect of Sink Location on Aggregation Based on Degree of Connectivity for Wireless Sensor Networks.” In proceedings of the 5th IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Jun 30-Jul 02, 2011, pp. 271-276.

[2] E.Fasolo, M.Rossi, J.Widmer and M.Zorzi. “In-Network Aggregation Techniques for Wireless Sensor Networks: A survey.” IEEE Wireless Communications, vol 14, no. 2, pp. 70-87, 2007.

[3] J.Stankovic, T.Abdelzaher, C.Lu, L.Sha and J.~Hou. “Real time Communication and Coordination in Embedded Sensor Networks.” IEEE Journal, vol 91, pp. 1002-1022, 2003.

[4] 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 Proceedings of the 63rd IEEE Vehicular Technology Conference (VTC), May 2006, pp. 563-567.

[5] J.Zhang, Q.Wu, F.Ren, T.He and C.Lin. “Effective Data Aggregation Supported by Dynamic Routing in Wireless Sensor Networks.” In Proceedings of the IEEE International Conference on Communications (ICC), May 2010, pp.1-6.

[6] T. He, J.Stankovic, C.Lu and T.Abdelzaher. “SPEED: a Stateless Protocol for Real-time Communication in Sensor Networks.” In Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS), 2003, pp. 46-55.

[7] I.Solis and K.Obraczka. “The impact of timing in data aggregation for sensor networks.” In Proceedings of the IEEE International Conference on Communications (ICC), 2004.

[8] S.G.Quan and Y.Y.Kim. “Fast Data Aggregation Algorithm for Minimum Delay in Clustered Ubiquitous Sensor Networks.” In Proceedings of the IEEE International Conference on Convergence and Hybrid Information Technology (ICHIT), 2008, pp. 327-333.

[9] A.Sivagami, K.Pavai and D.Sridharan. “Latency Optimized Data Aggregation Timing Model for Wireless Sensor Networks.” International Journal of Computer Science Issues (IJCSI), vol. 7, no 3, 2010.

[10] C.Cheng, C.K.Tse and M.Lau. “A Delay-Aware Data Collection Network Structure for Wireless Sensor Networks.” IEEE Sensors Journal, vol. 11, no 3, pp. 699-710, 2011.

[11] H.N.Le, V.Zalyubovskiy, C.Hyunseung and J.Zhao. “Delay-minimized Energy-efficient Data Aggregation in Wireless Sensor Networks.” In Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Oct 10-12, 2012, pp. 401-407.

[12] U.Roedig, A.M.Barroso and C.J.Sreenan. “Determination of Aggregation Points in Wireless Sensor Networks.” In Proceedings of the 30th International Conference EUROMICRO, Aug 31- Sep 3, 2004, pp. 503-510.

[13] C.Lu, B.Blum, T.Abdelzaher, J.Stankovic and T.He. “RAP: a Real-time Communication Architecture for Large-Scale Wireless Sensor Networks.” In Proceedings of the IEEE Eighth Real-Time and Embedded Technology and Applications Symposium, 2002, pp. 55-66.

[14] E.Zegura, K.Calvert and M.Donahoo. “A Quantitative Comparison of Graph-based Models for Internet Topology.” IEEE Transactions Journal on Networking, 1997.

[15] Z.Cheng, M.Perillo and W.Heinzelman. “General network lifetime and cost models for evaluating sensor network deployment strategies.” IEEE Transactions Journal on Mobile Computing, vol. 7, pp. 484-497, 2008.

[16] D. Fotue “Efficient Data Aggregation and Routing in Wireless Sensor Networks”PhD Thesis, Télécom ParisTech, pp. 1-190, January 31st, Paris, France, 2014.