International Journal of Informatics and Applied Mathematics
e-ISSN:2667-6990 Vol. 3, No. 1, 22-38

Latency and Energy Efficient Routing-Aware TDMA for Wireless Sensor Networks

Islam Amine Bouchedjera and Lemia Louail

LRSD laboratory, Faculty of Sciences, University Ferhat Abbas Sétif 1
Email: bouchedjera.islam@univ-setif.dz, lemia.louail@univ-setif.dz

Abstract. Simply because the OSI model has been effective for wired networks, the communications within the wireless sensor networks (WSN) since its appearance are ensured by a layer model, inspired by the OSI model. Since, protocols are designed independently in this model, metrics involved in several layers can be affected. Energy consumption and fast data aggregation are among the most important metrics, impacted by both the routing protocol in the network layer and the MAC protocol in the data link layer. Cross-layer, an emerging design that attempts to expand the interactions in the protocol stack has shown an improvement in the overall performance of such networks.

In this context, in order to achieve energy efficiency and fast data aggregation, and since the protocols of the MAC sub-layer and the network layer have a direct effect on these two metrics, we propose Efficient-Depth-ReLO, a centralized cross-layer approach between these two layers. This approach aims to build a TDMA scheduling by using the routing tree information. On the other hand, the proposed approach solves efficiently the hidden node problem. The results of extensive simulations show that the proposed approach performs better than similar existing works in terms of energy consumption and communication latency.

Keywords: WSNs · Cross-Layer · TDMA · Routing tree · Hidden node problem · Energy consumption · Communication latency

1 Introduction

The convergence of Micro-Electro-Mechanical Systems (MEMS) technology, wireless communications and digital electronics have led to the emergence of a new type of wireless networks, which connects the physical and digital environments, called Wireless Sensor Networks (WSN) [2]. This type of networks shares some important features with Ad-Hoc networks, such as self-organization, multi-hop communication, shared radio channel. Besides that, it raises new challenges because of the limited resources allocated to sensor nodes in terms of energy, capture and communication range, bandwidth, data processing and storage capacity. Moreover, the interconnection of WSNs with each other and with the Wide Area Networks (WANs) allowed the emergence of a new concept called the Internet of Things (IoT). The latter has been allowing the increase of the usefulness of these networks and consequently their importance in daily life [3].

Communicatons in this type of networks are ensured through a layered model, inspired from the Open Systems Interconnection (OSI) model. This model is based on the principle of layer separation, where the layers are implemented independently of each other, as each layer is responsible for providing particular functionalities and optimizing certain metrics. The individual decisions at each layer generate a data processing redundancy, and may sometimes result in conflicts between the goals of each layer. They thus lead to an additional cost in terms of energy consumption and a degradation of the QoS of the WSN [6]. Many papers [15], [17], [16] have shown that layers depend on each other and decisions of one affect the decisions of others. For example, the decisions made at the mac and network layers can affect each other, where even though the MAC protocol is designed to minimize some metrics like communication latency, the routing protocol can build high latency paths, because the temporary criterion given by the MAC protocol is not transparent to the network layer. Thus, the paths chosen by the routing protocol to minimize latency can be disrupted by the communications scheduling controlled by the MAC protocol. For consequence, the temporal decisions (taken by the MAC protocol) must benefit from the spatial decisions (taken by the routing protocol).

In order to achieve better performance of the WSN, Cross-layer approaches try to exploit the dependencies between the different layers thus allowing to exert a richer interaction between the different layers of the protocol stack. In other words, this design extends well-defined communications between adjacent layers of the traditional protocol stack to all even non-adjacent layers. It gives great flexibility and freedom, where one protocol can use information from another protocol to achieve its functionalities or two protocols are combined into one new protocol ensuring the functionalities of the two merged protocols.

Low energy consumption is one of the strongest requirements when designing protocols in WSNs. In fact, the main task in WSNs is to capture data from the physical environment and aggregate it to the sink, therefore, fast data aggregation is no less important than energy consumption, especially in real-time applications. Such applications require fast routing of data to the base station to avoid damage.

Since the network layer and the MAC sub-layer play an important role in controlling energy consumption and data routing time, and considering that the cross-layering concept has proved its effectiveness, we aim to present in this paper our cross-layer contribution called Efficient-Depth-ReLO that aims to correlate the decisions of these two layers. Efficient-Depth-ReLO is an approach allowing the construction of TDMA scheduling based on routing protocol information to minimize energy consumption by eliminating sources of energy waste such as: overhearing, idle listening and the frequent switching between the various modes of radio transceivers generated by the various existing works in the literature. On the other hand, it aims to minimize communication latency by resolving efficiently the hidden node problem, to increase the simultaneous transmission of packets in the network.

The remainder of this paper is organized as follows. The next section deals with related works. Section 3 highlights the models used in the proposed approach. Section 4 describes our contribution. Section 5 mentions the scenario description and metrics used to evaluate the proposed approach. Section 6 discusses the results, Finally, section 7 concludes the paper.

2 Related works

In literature, several approaches have been proposed in the context of MAC cross-layer approaches using routing information (Routing-aware MAC) to minimize communication latency and/or energy consumption. We classify these existing approaches in two large classes presented in Figure 1.

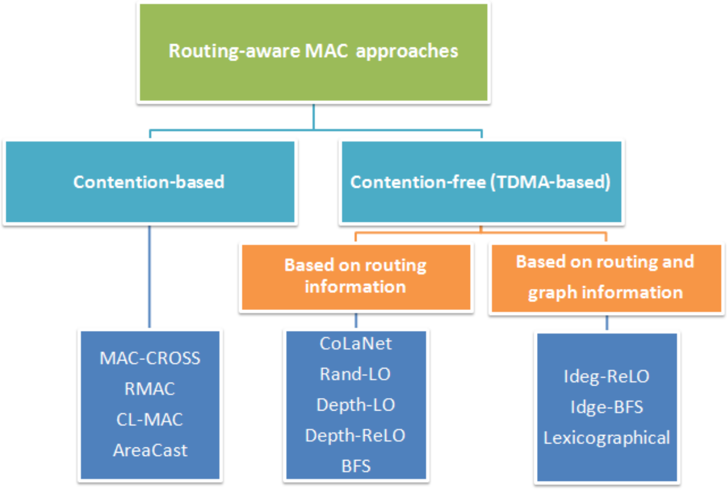

Fig. 1. Classification of routing-aware MAC approaches

In the following, we are interested in works that are based on contention-free MAC approaches and using routing protocol information, which represent the core of our work.

2.1 CoLaNet (Cross-Layer Design of Energy-Efficient Wireless Sensor Networks)

CoLaNet [4], is a TDMA-based cross-layer contention-free MAC approach using routing protocol information to achieve energy efficiency. It represents the first contribution in this class of protocols. CoLaNet operates in two phases: (1) the initial phase (or contention-based phase) carried out by all the nodes except the sink. It aims to build a routing tree called MinDegree. (2) the slot allocation phase achieved by the sink by applying the vertex-coloring algorithm [14] on the MinDegree routing tree to construct the schedule. The vertex-coloring algorithm starts by coloring the node that has the most neighbors in the routing tree. Then the coloring is applied on nodes that have a colored neighbor. At the end of the coloring, the number of colors obtained represents the TDMA length and each color represents the transmission slot of the node to which the color is affected.

2.2 Rand-LO (Random Leaves Ordering)

Rand-LO [10], is an improvement of the CoLaNet cross-layer approach [4], more precisely, an adjustment on the selection of nodes on which the vertex-coloring algorithm starts. The authors of [10] have found that starting coloring from the node that has the most neighbors does not guarantee the optimization of latency, especially this node can be located in any place. Rand-LO privileges the routing tree leaves by starting the scheduling by them, to improve the overall network latency. Because they represent the furthest nodes from the sink, from where they accumulate the latencies of the nodes that are in their paths towards the sink.

2.3 Depth-LO (Depth Leaves Ordering)

The random choice between the routing tree leaves does not use sufficiently the routing tree information as with Rand-LO [10]. Because the length paths of the leaves towards the sink are not equal. For that, the authors of [10] proposed the Depth-LO approach that privileges the farthest leaves from the sink, to color them first (i.e. starting the scheduling by them).

2.4 Depth-ReLO (Depth Remaining Leaves Ordering)

Depth-ReLo [10] is based on the concept, a routing tree can have paths with different lengths, so there are internal nodes of a path that are deeper than some leaves of another path. For this, Depth-ReLo [10] privileges the furthest nodes from the sink, to have the advantage of minimizing the communication latency by accumulating the latencies of the nodes that are in their paths.

2.5 BFS (Breadth First Search)

BFS [9], a top-down traversal based on a breadth-first search of the routing tree, where it starts the scheduling by the nodes belonging to the upper part of the routing tree and continues the allocation of the slots to the nodes that have an already scheduled parent. This allows the TDMA scheduling to start with the closest nodes to the sink because they are included in all the communication paths. Finally, the TDMA scheduling obtained is reversed.

2.6 IDeg-ReLO (Interference Degree Remaining Leaves Order-ing)

Unlike the previous approaches, IDegReLO [9] is based on information from the routing tree and also on additional information from the network graph, which the authors [9] called the interference degree. The interference degree for a node represents the number of nodes in conflict with that node, since it is the sum of the number of its one-hop and two-hop neighbors in the graph (if a node is a one-hop and a two-hop neighbor at the same time it is counted once). Thus, IdegReLO [9] privileges nodes with higher interference degree in the graph.

2.7 IDeg-BFS (Interference Degree BFS)

Similar to BFS, IDeg-BFS [9] starts to allocate slots to the nodes belonging to the upper part of the routing tree, while privileging nodes with higher interference degree. The final TDMA scheduling is reversed, so that the bottom part of the tree can be scheduled earlier.

2.8 Lexicographical

In [13], the authors propose a routing tree traversal based on the Lexicographical method [18]. The latter is considered as an effective method, which can improve the decision making process based on multiple criteria prioritized. The authors of [13] privilege the nodes based on the decision of the lexicographical method [18], where they define three criteria prioritized according to the degree of importance as follows: For each u node in the routing tree. **Distance(u):** represents the distance between the node u and the sink in terms of number of hops. **Fdegree(u):** represents the number of nodes that transmit their data via the node u to the sink. **Cdegree(u):** represents the number of the nodes in conflict with the node u, i.e. the number of one-hop and two-hop neighbors of the node u.

2.9 Limitations of existing TDMA-based contention-free MAC cross-layer approaches using the routing information

These approaches build different TDMA schedules by traversing the routing tree in different ways. However, they present very pessimistic solutions as they do not exploit the routing tree information well, because they deal with the WSNs as Ad-Hoc networks. They keep the same constraints of Ad-Hoc networks when dealing with the hidden node problem i.e. all two-hop neighbors in the graph can not simultaneously transmit their packets.

They do not take advantage of the operation nature of this type of network that aims to collect the data collected by the sensor nodes at the sink, where most of the time the communications are many-to-one. They deal with communications within the WSNs as a point-to-point communication type, from where they cause many sources of energy waste such as; overhearing, idle listening and the frequent switching between the various modes of radio transceivers, such as each node receives data from all its one-hop neighbors, while only the packets of its children are destined for it.

3 Definitions and Models

3.1 Connectivity Graph

We model the wireless sensor network by a non-oriented graph called connectivity graph, as G (V, E), where V = $\{v_1, v_2, ..., v_N\}$ represents the set of vertices of the graph, corresponding to the N nodes in the network and the sink is denoted as v_1, and E = $\{e_{ij}\}_{ij \in \{1,...,N\}}$ represents the set of edges corresponding to direct communicating nodes, where e_{ij} joining vertex i to vertex j. An edge from i to j (i $\longrightarrow$ j) exists if $P_{R_{ij}} \geq \gamma$,where γ is the receiver sensitivity, which represents the minimum acceptable received power. The nodes of the network have the same communication range r that we can model according to the unit-disk graph (UDG) communication model [5].

The density (δ) of nodes represents the average number of neighbors per node in the network, it varies according to the requirements and the application context. According to the UDG communication model, with a random uniformly distributed network in a square area:

$$\delta = \pi \times r^2 \times N/a^2 \tag{1}$$

Where: r is the communication range, N is the network size (number of nodes) and a is the deployment area side length. Figure 2 shows a graph that models a network with 9 sensors (from 2 to 10), with node 1 representing the sink.

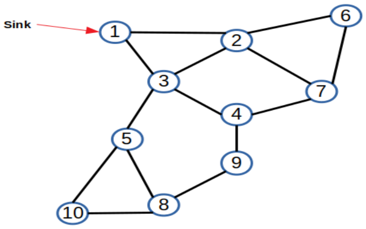

Fig. 2. An example of a graph modeling a network of 10 nodes

3.2 Routing tree

The routing tree is a set of no-cycle edges connecting the nodes of the network to the sink , which is the root of the tree and each node in the tree except the sink has one parent. The latter transmits the data of its children to the sink. The choice of the node's parent depends on the routing protocol. The routing tree is defined by a vector P = $[P_i]_{i\in\{1,\ldots,N\}}$ such that each element i of the vector P contains the node v_i parents identifier in the routing tree. Figure 3 shows an example of a routing tree for the network of Figure 2.

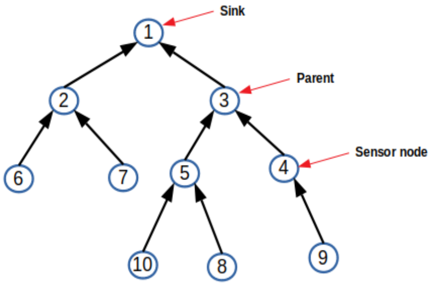

Fig. 3. An example of a routing tree for the network of Figure 2

3.3 TDMA scheduling

The access to a communication channel is done according to a time multiplexing, it is distributed in a number of small-time intervals called slots of the same size. In a TDMA each node uses only one slot to transmit the data to its parent, the other slots are used either to receive the data of its children or to switch into sleep mode, which keeps the energy of the sensor nodes. TDMA-based protocols are considered to be the most powerful approaches to deal with intense traffic and convergecast networks [7], because they exploit efficiently the radio medium, and therefore approach to reach to its theoretical maximum flow. We model the TDMA in a wireless sensor network by an allocation matrix called $Slots_{N\times L}$,

where N is the number of matrix lines, which corresponds to the number of nodes in the network, and L is the number of matrix columns , which corresponds to the TDMA length in number of slots. Each element of the Slots matrix is an integer from 1 to N, is defined as follows:
Slot [i, j] = i, node v_i transmits its data during slot j. For example, in Table 1, Slot[6.4] = 6, node v_6 transmits its data during slot 4.
Slot [i, j] = k, node v_i receives data from one of its children (v_k) during slot j. For example, in Table 1, Slot[2,4] = 6, node v_2 receives data from node v_6 through slot 4.
Slot [i, j] = 0, node v_i does not use slot j neither for transmission nor for reception. For example, in Table 1, Slot[1,2] = 0, node v_1 goes into sleep mode in slot 2.

	Slot1	**Slot2**	**Slot3**	**Slot4**	**Slot5**
Node 1	2	0	3	0	0
Node 2	2	7	3	6	0
Node 3	2	5	3	4	0
Node 4	9	7	3	4	0
Node 5	0	5	3	10	8
Node 6	2	0	0	6	0
Node 7	2	7	0	6	0
Node 8	9	5	0	10	8
Node 9	9	0	0	0	8
Node 10	0	5	0	10	8

Table 1. A TDMA scheduling for the network of Figure 2

3.4 Conflict nodes

The conflict nodes are nodes that must have different transmission slots. Otherwise, they generate collisions in the network. Two nodes are in conflict if one of the two following conditions is verified:

- **Condition(1):** All one-hop neighbors in the network are in conflict, because the communication links are not bidirectional.
- **Condition(2):** All two-hop neighbors connected by an intermediate node are in conflict if and only if the latter is the parent of at least one of these two nodes.

The second condition represents our proposition to solve a hidden node problem. This problem is illustrated in Figure 4. Node B is within the transmission ranges of nodes A and C, while nodes A and C cannot hear each others transmissions. Thus, simultaneous transmissions of nodes A and C can cause collisions at node B. In such a case, nodes A and C are called hidden nodes.

Traditionally [8], TDMA-based approaches solve the hidden nodes problem by assigning exclusive time slots to these nodes, i.e. two-hop neighbors should not transmit simultaneously. Although, these solutions avoid conflict packets and

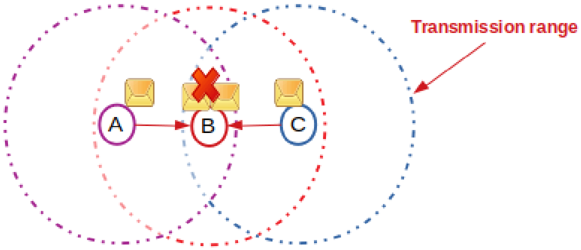

Fig. 4. Exemplary scenario with hidden node problem.

have been effective for point-to-point communication model in ad-hoc networks, there is a wasted opportunity to send a packet simultaneously by two-hop neighbor nodes in WSN, which is using a communication model based on many-to-one (convergecast). Whereas, based on the particularity of the communication model used in WSN and the cross-layer design, the proposed approach in this paper enables to resolve efficiently the hidden node problem, which increases simultaneous transmissions, and thus minimizing communication latency, by applying the second condition.

According to the existing TDMA-based approaches, nodes 4 and 8 in Figure 2 can not transmit their packets simultaneously, because they are two-hop neighbors and node 9 is the intermediate node between them, thus they lead to collision at this node. In contrast, the proposed approach according to the second condition allows nodes 4 and 8 to transmit simultaneously, because the intermediate node between them (node 9) is not the parent of at least one of them in the routing tree. In fact, these simultaneous transmission of packets will generate collision at node 9, which is not important because from the outset these packets are not destined to it.

4 Efficient-Depth-ReLO

Efficient-Depth-ReLO, is a centralized TDMA-based cross-layer contention-free MAC approach that uses routing protocol information to achieve energy efficiency and minimize communication latency. It represents an improved version of the Depth-ReLO approach [10].

The Depth-ReLo approach focuses on minimizing communication latency, its goal has been achieved through its routing tree traversal and slot allocation algorithms, as it generates communication latency lower than that generated by the existing approaches of its class. Unfortunately, it generates a lot of sources of energy waste related to communications such as overhearing and idle listening, which cause an increase in energy consumption.

The Efficient-Depth-ReLO approach is based on two algorithms (1, 2), such as the first algorithm aims to traverse the routing tree, in order to give order according to which the second algorithm assigns the slots:

1. Select nodes by privileging the deepest nodes.
2. If there is more than one node selected in step 1, select among them the node that forwards the most packets to the sink.
3. If there are still more than one node selected in step 2, select among them one node randomly.
4. Find an appropriate slot for each selected node by applying the second algorithm:
   - If the selected node is a leaf, find an appropriate slot by starting with the first slot of the TDMA schedule.
   - If the selected node is not a leaf, find an appropriate slot by starting the search from the highest slot of its children and possibly doing a circular search in the schedule, i.e. look for an appropriate slot till the last slot of the schedule, if no appropriate slot is found, restart from the first slot of the schedule, up to the highest slot of its children.
   - In both cases, if no appropriate slot is found, a new slot is added at the end of the schedule and is given to the node.
5. Repeat from the first step until all the nodes are scheduled.

Algorithm 1: Routing tree traversal

```
Input: G(V,E) /* Connectivity graph                          */
Output: Slots /* Allocation matrix                           */
Local: Q, U /* Node arrays                                   */
/* Initialisation                                            */
for each node $v_i$ in V do
    addNode($v_i$, Q);
end
/* Traversing                                                */
while Q is not empty do
    U ← findDeeperNode(Q);
    if arraySize(U) > 1 then
        U ← findMostTransmittingNode(U);
        if arraySize(U) > 1 then
            U ← findNodeRandomly(U);
        end
    end
    assignSlots($U_{[1]}$);
    removeNoeud($U_{[1]}$,Q);
end
```

An appropriate slot for node u is defined as follows: (1) if it is a free slot; not allocated to any node or (2) if the allocated nodes in this slot do not conflict with node u.

Algorithm 2: Slots Assignment

```
Input: Slots /* Allocation matrix                                      */
       n /* Current node                                               */
       P /* Routing tree                                               */
Output: Slots /* Allocation matrix                                     */
Local: i /* Counter                                                    */
for i = highestSlotChildren(n)+1;
    i = highestSlotChildren(n);
    i = (i+1) mod(TDMALength(Slots)) do
    if appropriateSlot(i, n) then
        Break loop
    end
end
if i = highestSlotChildren(n) then
    addNewSlot(Slots);
    i ← TDMALength(Slots);
end
Slots[n,i] ← n; /* i is the transmitting slot of node n                 */
Slots[P[n],i] ← n;/* i is one of the receiving slots of node n's parent
   */
```

4.1 For communication latency, Efficient-Depth-ReLO:

- privileges the deepest nodes that forward the most packets to the sink. Starting the TDMA scheduling with these nodes to improves the overall latency of the network. These nodes represent the furthest nodes from the sink and they accumulate the latencies of the nodes which are in their paths towards the sink.
- reduces the number of conflict nodes to increase the number of simultaneous transmissions by reducing the number of hidden nodes and thus decreasing the communication latency. Instead of considering that all two-hop neighbors in the network are in conflict, we consider that two nodes connected by an intermediate node are in conflict if and only if the latter is the parent of at least one of these two nodes. Apart that, we do not care if there are contentions at the other nodes, because from the outset these packets are not destined to them.

4.2 For energy consumption, Optimistic-Depth-ReLO:

- eliminates sources of energy waste related to communications for each node to ensure higher overall lifetime for the entire network. Precisely, it eliminates idle listening and overhearing, since each node is in the listening mode only when receiving data from its children in the routing tree, knowing that the radio transceivers are by far the most greedy energy factor.
- reduces the frequent switching between the various modes of radio transceivers such that each node switches from sleep to active mode only to transmit packets to its parent or to receive packets from one of its children.

5 Development environment and evaluation metrics

To evaluate the performance of our approach and compare it with those of existing approaches, we used the JUNG framework (Java Universal Network / Graph Framework) [11]. The latter consists of a set of open source Java libraries providing a common and extensible language for modeling, analysis and visualization of data that can be represented as a graph or network. We note that, extensive simulations are performed on networks generated randomly with size equals 100 nodes, and with different densities. The side length of the square deployment area was changed to obtain different densities (δ) = [4, 20] (the same interval and network size used in [9], [10] and [13]).
Considering,

- The nodes are fixed and randomly deployed in the area of interest has the form of a square and the sink is always placed in one of the corners of this area.
- The nodes operate in single half-duplex mode.
- Upward flow, sensor nodes send packets periodically to the sink and, for each round of the TDMA the nodes always have data captured to send.
- Downward flow is not considered in the proposed approach, but its possible to taking charge of it by assigning the first slot of TDMA to the sink to transmit control packets (e.g. time synchronization) and also by assigning this slot to all nodes in the network as a reception slot.

The evaluating metrics are: the average latency (in number of slots), the average energy consumption (in μjoule), the average duty cycle and the length of the TDMA (in number of slots). These metrics are calculated until the sink receives packets from all the nodes of the network.

5.1 Average latency

Communication latency in a WSN refers to the delay between the moment when a sensor node has a packet to send and when the packet is successfully received at the sink. The latency for one node is computed as the number of slots needed for its packet to arrive to the sink through multi-hop communication according to the routing tree.
Formally, if a packet from node v_i has been sent through the following path:
i $\longrightarrow n_1 \longrightarrow n_2 \longrightarrow ... \longrightarrow n_k \longrightarrow sink$
The latency equals:

$$dt_i = slot_i + nbrSlots(i, n_1) + nbrSlots(n_1, n_2) + ... \\ ... + nbrSlots(n_{k-1}, n_k)$$

We specify that :

$$nbrSlots(i, j) = \begin{cases} (slot_j - slot_i) & \text{if} \quad slot_i < slot_j \\ (l - slot_i + slot_j) \bmod l & \text{if } slot_i > slot_j \end{cases}$$

Where:

- n_k represents one of the sink's neighbors,
- $slot_i$ represents the transmission slot of node v_i,
- $nbrSlots(i, j)$ represents the number of slots between the transmission slot of node v_i and that of its parent (node v_j),
- l represents TDMA length.

Since the sink is not involved in such communication (transmission of the collected data to collect them at the sink, which has ID equals 1), the average communication latency for paquets in a network of size n is defined as:

$$dt = \left(\sum_{i=2}^{n} dt_i \right) / \ (n-1) \tag{2}$$

Where i represents the node that has the identifier i.

5.2 Average energy consumption

Several models of energy consumption have been proposed in the literature to study and evaluate energy consumption [1]. We choose the model that implies the transmission power to calculate the energy consumed on each node i (E_i), because we have represented the transmission range of each node by its transmission power (P_T). According to [12] E_i is expressed by the equation:

$$E_j = E_{Rx} \times n_{children_j} + E_{Tx} + \frac{P_{T_j}}{E_{amp}} \times T_{slot} \tag{3}$$

Where:

- E_{Rx} is the energy consumed when receiving a packet,
- $n_{children_j}$ is the number of children of node v_j,
- E_{Tx} is the energy consumed when transmitting a packet,
- P_{T_j} is the transmission power of the node v_j,
- E_{amp} is the energy used for amplification,
- T_{slot} is the slot duration.

6 Performance evaluation

In this section, we evaluate the performance of Efficient-Depth-ReLO by comparing it with those of CoLaNet [4], Depth-ReLO [10] and Lexicographical [13]. It should be mentioned that each value in the figures below represents the average value of 100 values obtained by 100 simulations. The error bars of the figures represent the 95% confidence intervals, to give greater reliability to the results obtained.

6.1 Average latency

Figure 5 shows that the Efficient-Depth-ReLO approach has better latency than the other approaches, since it generates minimal latency. We find that density increases, latency increases. Logically, because the increase of this parameter generates a large number of the conflict nodes, and thus requires more slots in the TDMA.

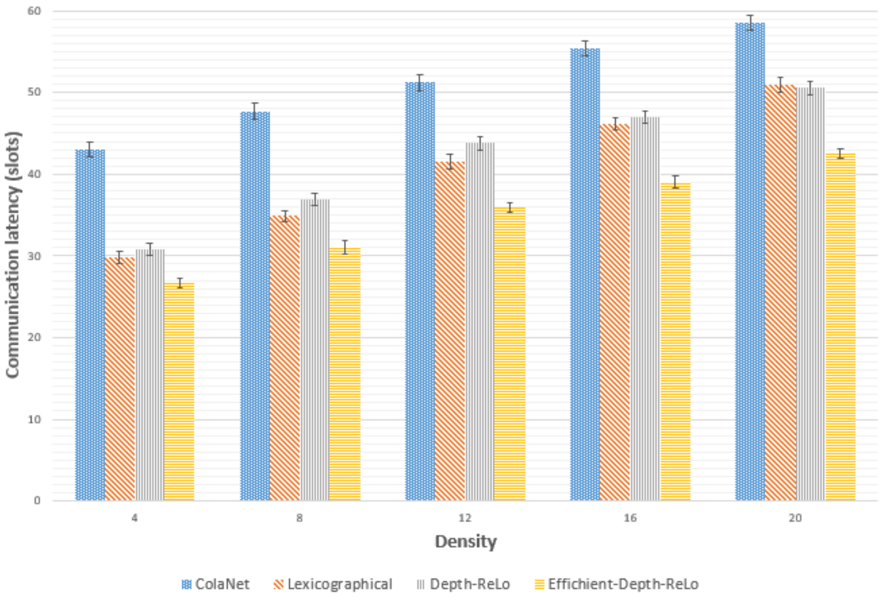

Fig. 5. Average latency based on density for networks of 100 nodes

6.2 TDMA length

Figure 6 shows that CoLaNet generates a TDMA with minimal length, due to its vertex-coloring algorithm. In addition, the performances of CoLaNet shown in Figures 5 and 6 confirm that having a minimal TDMA does not guarantee minimal latency, and that minimal latency requires reasonable allocation of slots.

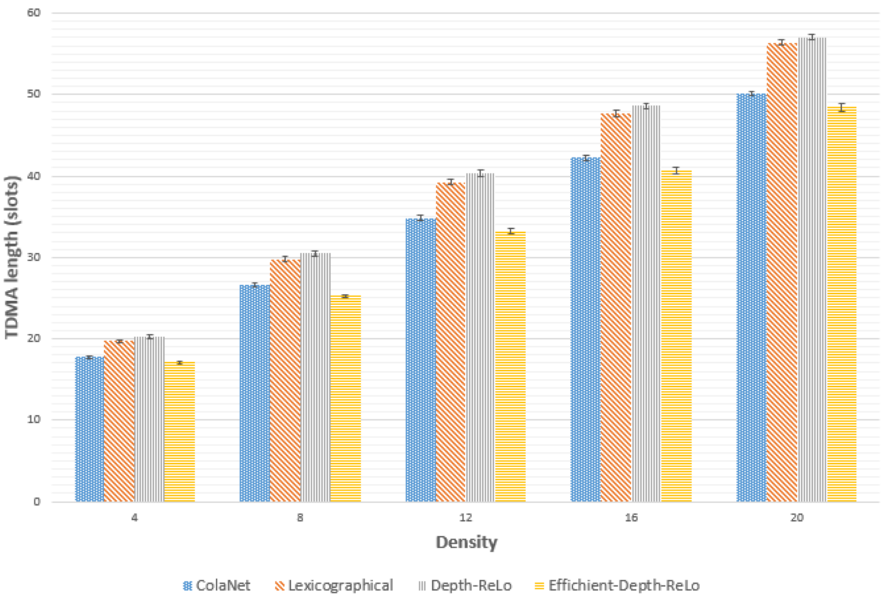

Fig. 6. Average length of TDMA based on density for networks of 100 nodes.

6.3 Average energy consumption

Figure 7 highlights two major advantages of the Efficient-Depth-ReLO approach compared to other approaches in terms of energy efficiency.

Firstly, Efficient-Depth-ReLO generates minimal consumption due to its functionality that aims to leave the sensor nodes in sleep mode as much as possible.

Secondly, Figure 7 shows that for similar approaches, the higher the density, the higher the energy consumption increases, because the density increase means an increase in the number of neighbors for each node and therefore an increase in wastage of neighborhood-related energy at each node. However, in the Efficient-Depth-ReLO approach, the higher the density, the lower the energy consumption, because it increases the number of children in the network, which do not consume a lot of energy. On the other hand, it decreases the lengths paths, from which it reduces the number of packets transmitted in the network.

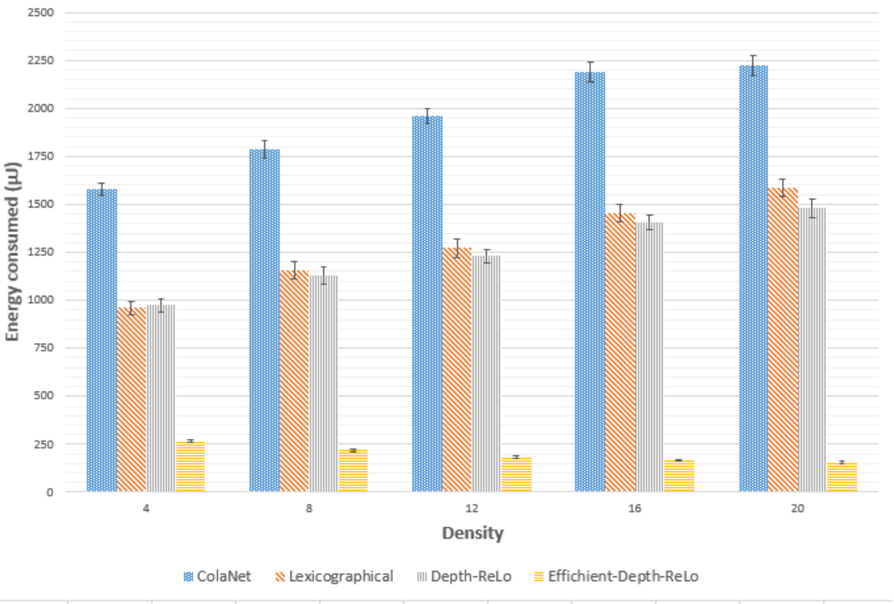

Fig. 7. Average energy consumption based on density for networks of 100 nodes.

7 Conclusion

This paper presents Efficient-Depth-ReLO, a centralized cross-layer contention-free MAC approach, that aims to build TDMA scheduling by using routing protocol information for energy-efficiency and fast data aggregation. Moreover, it solves efficiently the hidden node problem, where instead of considering that all two-hop neighbors in the network are in conflict, it considers that two nodes connected by an intermediate node are in conflict if and only if the latter is the parent in the routing tree of at least one of these two nodes. The comparative study between our approach and similar existing approaches showed that Efficient-Depth-ReLO generates better latency and energy consumption.

References

1. Abo-Zahhad, M., Amin, O., Farrag, M., Ali, A.d.: Survey on energy consumption models in wireless sensor networks. Open Trans. Wirel. Sens. Netw **1**(1) (2014)
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications magazine **40**(8), 102–114 (2002)
3. Bellavista, P., Cardone, G., Corradi, A., Foschini, L.: Convergence of manet and wsn in iot urban scenarios. IEEE Sensors Journal **13**(10), 3558–3567 (2013)
4. Chou, C.F., Chuang, K.T.: Colanet: a cross-layer design of energy-efficient wireless sensor networks. In: 2005 Systems Communications (ICW'05, ICHSN'05, ICMCS'05, SENET'05). pp. 364–369. IEEE (2005)

5. Clark, B., Colboum, C., Johnson, D.: Unit disk graphs, discrete mathmatics (1990)
6. Fu, B., Xiao, Y., Deng, H.J., Zeng, H.: A survey of cross-layer designs in wireless networks. IEEE Communications Surveys & Tutorials **16**(1), 110–126 (2013)
7. Kosek-Szott, K.: A survey of mac layer solutions to the hidden node problem in ad-hoc networks. Ad Hoc Networks **10**(3), 635–660 (2012)
8. Long, J., Dong, M., Ota, K., Liu, A.: A green tdma scheduling algorithm for prolonging lifetime in wireless sensor networks. IEEE Systems Journal **11**(2), 868–877 (2015)
9. Louail, L., Felea, V.: Latency optimization through routing-aware time scheduling protocols for wireless sensor networks. Computers & Electrical Engineering **56**, 418–440 (2016)
10. Louail, L., Felea, V.: Routing-aware time slot allocation heuristics in contention-free sensor networks. In: International Conference on Wired/Wireless Internet Communication. pp. 271–283. Springer (2016)
11. Mijovic, S., Sanguinetti, L., Buratti, C., Debbah, M.: Optimal design of energy-efficient cooperative wsns: How many sensors are needed? In: 2015 IEEE 16th international workshop on signal processing advances in wireless communications (SPAWC). pp. 31–35. IEEE (2015)
12. OMadadhain, J., Fisher, D., Smyth, P., White, S., Boey, Y.B.: Analysis and visualization of network data using jung. Journal of Statistical Software **10**(2), 1–35 (2005)
13. Osamy, W., El-sawy, A.A., Khedr, A.M.: Satc: A simulated annealing based tree construction and scheduling algorithm for minimizing aggregation time in wireless sensor networks. Wireless Personal Communications **108**(2), 921–938 (2019)
14. Pemmaraju, S., Skiena, S.: Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica®. Cambridge university press (2003)
15. Raisinghani, V.T., Iyer, S.: Cross-layer design optimizations in wireless protocol stacks. Computer Communications **27**(8), 720–724 (2004)
16. Sah, D.K., Amgoth, T.: Parametric survey on cross-layer designs for wireless sensor networks. Computer Science Review **27**, 112–134 (2018)
17. Srivastava, V., Motani, M.: Cross-layer design: a survey and the road ahead. IEEE communications magazine **43**(12), 112–119 (2005)
18. Triantaphyllou, E.: Multi-criteria decision making methods. In: Multi-criteria decision making methods: A comparative study, pp. 5–21. Springer (2000)