



HAL
open science

Routing-Aware Time Slot Allocation Heuristics in Contention-Free Sensor Networks

Lemia Louail, Violeta Felea

► **To cite this version:**

Lemia Louail, Violeta Felea. Routing-Aware Time Slot Allocation Heuristics in Contention-Free Sensor Networks. 14th International Conference on Wired/Wireless Internet Communication (WWIC), May 2016, Thessaloniki, Greece. pp.271-283, 10.1007/978-3-319-33936-8_21 . hal-01434858

HAL Id: hal-01434858

<https://inria.hal.science/hal-01434858>

Submitted on 13 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Routing-Aware Time Slot Allocation Heuristics in Contention-Free Sensor Networks

Lemia Louail and Violeta Felea

Femto-ST Institute, University of Bourgogne-Franche-Comté,
Besançon, France

{lemia.louail, violeta.felea}@femto-st.fr

Abstract. Traditionally, in Wireless Sensor Networks, protocols are designed independently in the layered protocol stack, and metrics involved in several layers can be affected. Communication latency is one metric example, impacted by both the routing protocol in the network layer and the MAC protocol in the the data link layer. Better performances can be obtained using cross-layer approaches.

In this paper, we address latency optimizations for communications in sensor networks, based on cross-layer decisions. More particularly, we propose new time slot scheduling methods correlated to routing decisions. Slot allocation for nodes follows particular routing tree traversals, trying to reduce the gap between the slot of a child and that of its parent. Simulations show that latency performance of our contributions improves similar cross-layer approaches from 33% up to 54%. Duty cycle of obtained schedules are also improved from 7% up to 11%.

Keywords: wireless sensor networks, cross-layer approaches, contention-free time slot scheduling, routing protocol

1 Introduction

Wireless Sensor networks (WSN) are an innovating technology used nowadays in many domains (healthcare, military, environment monitoring, etc.). Sensor nodes are deployed in an area to collect information and to relay it to the Base Station (also called Sink) through multi-hop communications. To handle communications in this kind of network, the five layered protocol stack is used [1]. In this stack, the Network layer is responsible for finding paths between sensors and the sink through the routing protocol, and the Data Link layer is responsible for organizing communications to avoid interference and collisions through the MAC protocol. Each of these layers aims to optimize specific metrics as latency or energy consumption.

Generally, designing MAC and Routing protocols separately in the traditional protocol stack gives good performance in terms of metrics related to the Data Link layer or the Network layer, but they are not optimized to improve the overall network performance. For example, when the MAC protocol is trying to minimize the latency by designing a schedule for communications, the routing

protocol can counteract its decisions by adopting a path with a high latency. That is why the two protocols need to communicate with each other to overcome this counteraction.

Recently, cross-layer approaches [2, 3] were adopted, in which two or more layers communicate with each other to achieve better performances. Cross-layer design addressed here concerns protocols, and not transversal information as energy or mobility (as used in cross planes, which are the power management plane, the mobility management plane and the task management plane). For example, the cross-layer design we are interested in concerns situations where the MAC protocol can use information coming from the Routing protocol to design better schedules.

In this context, we propose new TDMA scheduling methods to minimize the latency of communications. The cross-layer aspect is the use of routing information in order to allocate communication time slots for nodes. The routing protocol is supposed to be tree-based. Our main contribution proposes different routing tree traversals to allocate communication time slots in order to reduce communication latency. Each scheduling approach was evaluated with different routing trees to prove the good results of our contribution.

Moreover, an improvement of an existing cross-layer approach, CoLaNet [5], is proposed. The latter constructs a TDMA schedule based on vertex-coloring solutions. To make comparison with our TDMA schedules fair, we include routing tree information in coloring decisions in order to reduce communications latency.

In the next section we present some related works in cross-layering then we present our main contribution in section III. In section IV is presented the configuration of simulations and in section V, the performance evaluation of our approaches compared with the state of the art. Finally, we conclude with the advantages of our contributions.

2 State of the art

Each layer of the sensor network protocol stack is responsible of particular functionalities. For example, the Data Link layer schedules communications between nodes through the MAC protocol and the Network layer finds a path between a sender and a receiver through the routing protocol.

There are two types of MAC protocols. The first type is the collision free protocols (TDMA-like) in which the communications are organized according to a schedule, such that one-hop communications do not generate collisions. The second type is the collision based protocols (CSMA-like) in which no schedule is defined and carrier sense techniques are used to avoid collisions.

In this paper we are interested in the first type, more specifically in TDMA (Time Division Multiple Access) scheduling in which each node has a slot to send information to its neighborhood. The other slots are used to either receive information from the neighborhood or to go to sleep mode. The most important advantage of the TDMA is that it ensures that there will be no collision caused either by the one-hop neighborhood or induced by the hidden terminal problem

during communications and the latency is easily controlled. Moreover, in sleep mode, radio is turned off so energy can be conserved.

Traditionally, this kind of protocol is designed without using any information from the routing protocol [4]. Therefore, resulting schedules could be inadequate in terms of latency because of the path chosen by the routing protocol. Cross-layer approaches are the solution to correlate decisions taken in different layers. We assume that routing information integrated into TDMA-based schedules decisions can improve latency. We use tree-based routing protocols in this work.

CoLaNet [5] is a cross-layer protocol that uses the characteristics of the routing tree to construct a collision-free MAC schedule. Its main idea is to formulate the MAC issue as a vertex-coloring problem. By solving this problem, CoLaNet is able to determine the schedule of each sensor node using an approximation algorithm. CoLaNet is the only TDMA-based cross-layer protocol found in literature, that is why we consider it as a comparison reference.

First, as a routing tree, CoLaNet constructs the MinDegree tree in which the sink is the root, the neighbor of the sink are its children and then each node chooses the node with the fewest children nodes as its parent.

Second, CoLaNet applies the vertex-coloring algorithm in [6] to establish the schedule as follows:

- CoLaNet finds the node with the maximum degree in the routing tree.
- A color is given to this node. In this process, CoLaNet ensures that none of its one-hop or two-hop neighbor in the graph has this color and this to avoid collisions. If not, a new color is generated and is given to the node.
- CoLaNet continues to color vertices that have an already colored neighbor. If two or more vertices have a colored neighbor, no order is applied on them, they are taken randomly one by one.
- When all nodes have a color, the colors are transformed into a schedule where the number of colors represents the length of the schedule, and each color represents a slot. The slot used for transmission is the number of the color assigned to the node. Slots used for reception can be computed based on the color of the neighboring nodes and on the network graph.

The goal of CoLaNet is to provide an allocation of communication slots, for every node, with no collisions. CoLaNet does not generate minimum schedule length, as their goal is not to propose a better approximation algorithm for solving the vertex-coloring problem (the minimum vertex coloring problem is often intractable).

3 Slots allocation based on routing tree traversals

3.1 Motivation

While CoLaNet aims to define a transmission schedule of each sensor node to avoid collisions, it neglects completely the aspect of latency. We noticed that the max-degree node, with which the vertex-coloring begins, can be located anywhere in the routing tree which is not the best way to start the slot allocation in the

schedule. That is why, in this part, we propose a different method to allocate slots than the vertex-coloring algorithm. Three different ways of routing tree traversal are proposed. They all start with the leaves of the routing tree rather than with the node of maximum degree. We first present our general slot allocation method used for all scheduling techniques.

3.2 Slot Allocation Method

The initial schedule length is the maximum node degree incremented (the node with the maximum degree needs a slot for each neighbor to receive information and one slot for itself for the transmission). The initial schedule length is the minimal value but may be increased, during the allocation algorithm, if no free slot is available because of possible collisions in 2-hop neighborhood communications.

This technique is based on the identification of a free slot for a node. A slot is considered free for a node if none of its one-hop or two-hop neighbor has been allocated in the slot (this ensures that the TDMA schedule is collision free).

Considering the routing tree, we suggest the slots allocation to be done as follows.

- Case 1: If the current node is a leaf (in the routing tree), it is given the first free slot starting from the beginning of the schedule.
- Case 2: If the current node is not a leaf, it is given the first free slot starting from the last slot of its children and doing a circular search (look for a free slot up to the last slot of the schedule and possibly, restart from the first slot of the schedule, up to the last slot of the children, if no free slot is found in the first search).
- In both cases (1 and 2), if no free slot is found, a new slot is added at the end of the schedule and it is allocated to the current node.

Case 1 gives to leaves of the routing tree more chances to be scheduled earlier and Case 2 makes internal nodes to be scheduled after all of their children in a circular way.

The schedule for a node contains one slot to send information to its neighborhood and the other slots are used to either receive information from the neighbor or to go to sleep mode. The Slot Allocation Method allows to define the sending slot of each node. The other slots can be deduced by exchanging messages with the neighborhood because the sending slot of a node is a receiving slot for all its one-hop neighbor.

3.3 Tree traversals

Using the Slot Allocation Method, we propose 3 approaches to allocate slots starting from the leaves of the routing tree. The idea is that starting from the bottom of the routing tree allows nodes that are far from the sink and being, therefore, on longer routes, to have earlier slots.

An example of a network composed of 8 sensor nodes with its MinDegree routing tree is given in Figure 1 (node 1 is considered to be the sink node). This example will be used to illustrate the scheduling methods we propose.

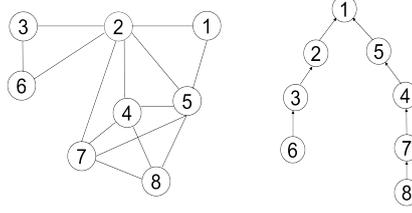


Fig. 1: A sensor network graph with its Min-Degree routing tree

Rand-LO (Random Leaves Ordering) The Rand-LO allocation begins by scheduling leaves of the routing tree in a random order, then it continues to schedule nodes while ascending the routing tree:

- 1- Leaves of the routing tree are selected randomly and each one of them is allocated in the schedule using the previous Slot Allocation Method.
- 2- Parents of the already allocated nodes are selected and are considered for allocation in the same order as their children. According to this order, each parent is allocated in the schedule using the Slot Allocation Method (see subsection 3.2).
- 3- Repeat step 2 while ascending the tree until all nodes are allocated in the schedule.

The traversal order of the routing tree in figure 1, given by the Rand-LO allocation algorithm, is the following: 6 - 8 - 3 - 7 - 2 - 4 - 1 - 5.

Using the Slot Allocation Method, the schedule is constructed as follows:

- Initial schedule length = maximum node degree + 1 = 6 (degree of node 2) + 1 = 7.
- node 6: is a leaf, so it takes the first slot.
- node 8: is a leaf, try to allocate it in the first slot. node 6 and node 8 are not one-hop or two-hop neighbors and thus they can use the same slot. Therefore, node 8 takes the first slot.
- node 3: is parent of node 6, start allocation from slot 2. Slot 2 is free for node 3, so node 3 takes slot 2.
- node 7: is parent of node 8, start allocation from slot 2. Slot 2 is taken by node 3. Node 3 and node 7 are two-hop neighbors, they can not use the same slot. Next slot is free for Node 7, so node 7 takes slot 3.

This procedure continues until all nodes are allocated in the schedule.

The resulting schedule is presented in Figure 2. Every line i gives the slot allocation for node i as a vector. The j th element of the vector for node i , $sched_j$, contains the node concerned by the communication of node i during the slot j . When $i = sched_j$, the slot j is used by the current node i to transmit. Otherwise, when no value is defined, the node does not communicate. When $i \neq sched_j$, the slot j is a slot used by node i to receive information from node $sched_j$. For example, in Figure 2, node 2 uses the 4th slot to transmit information and receives in the next slot (5th) information from node 4.

The latency is computed in number of slots, considering the first slot $sched_1$ of the initial sending node as the first time to send.

For example, when node 6 needs to transmit information (using the routing tree in Figure 1), its latency is of 4 slots (no waiting slots before starting communication, as node 6 has the first slot to transmit, 1 slot for node 6 to transmit, no waiting slots before node 3 transmits, 1 slot for node 3 to transmit, 1 waiting slot for node 2 before transmission and 1 slot for node 2 to transmit). This schedule generates an average latency of 5.714.

	slots →				
nodes					
1			2	1	5
2	6	3	7	2	4
3	6	3	2		
4	8	7	2	4	5
5	8	7	2	4	1
6	6	3	2		
7	8	7	2	4	5
8	8	7	4	5	

Fig. 2: TDMA schedule for the sensor network in Figure 1 using Rand-LO

Depth-LO (Depth Leaves Ordering) The idea is to improve the previous method (Rand-LO) by giving leaves that are far from the sink the privilege to be scheduled earlier than the other leaves:

- 1- Leaves of the routing tree are selected and sorted by their depth in the routing tree, i.e. the leaf with the highest depth is processed first.
- 2- According to the appropriate order, each leaf is allocated in the schedule using the Slot Allocation Method.
- 3- Parents of the already allocated nodes are selected and are processed in the same order as their children.
- 4- According to this order, parents of the already scheduled children are allocated in the schedule using the Slot Allocation Method.
- 5- Repeat step 3 while ascending the routing tree until all nodes are allocated in the schedule.

According to the Depth-LO algorithm, the traversal of the routing tree in Figure 1 is done in this order: 8 - 6 - 7 - 3 - 4 - 2 - 5 - 1. The resulting schedule is presented in Figure 3 which gives an average latency of 5.571 that is lower than the latency obtained with Rand-LO.

	slots →				
nodes					
1				2	5
2	6	7	3	4	2
3	6	3	2		
4	8	7	4	2	5
5	8	7	4	2	1
6	6	3	2		
7	8	7	4	2	5
8	8	7	4	5	

Fig. 3: TDMA schedule for the sensor network in Figure 1 using Depth-LO

Depth-ReLO (Depth Remaining Leaves Ordering) In Depth-ReLO, nodes on long paths in the routing tree are privileged and scheduled earlier than other nodes:

- 1- The leaf with the highest depth in the tree is allocated in the schedule using the Slot Allocation Method.

- 2- This leaf is deleted from the tree.
- 3- If the tree is not empty go to step 1, otherwise the scheduling is done.

According to the Depth-ReLO algorithm, the traversal of the routing tree in Figure 1 is done in this order: 8 - 6 - 7 - 4 - 3 - 5 - 2 - 1. The resulting schedule is presented in Figure 4 which gives an average latency of 5.428 that is lower than the latency obtained by Rand-LO and Depth-LO.

	slots						
nodes							
1					5	2	1
2	6	7	4	3	5	2	1
3	6			3		2	
4	8	7	4		5	2	
5	8	7	4		5	2	1
6	6			3		2	
7	8	7	4		5	2	
8	8	7	4		5		

Fig. 4: TDMA schedule for the sensor network in Figure 1 using Depth-ReLO

4 Simulations configuration

In this section, we present the configuration parameters used in the simulations. We used a Java-based library called JUNG [8] which allows modeling, analysis and visualization of networks as graphs.

We generated 5000 random networks using $n = 100$ nodes, each node having a communication range $r = 25m$. Nodes were deployed in a square area of size a^2 . The connectivity model used is UDG (Unit Disk Graph [7]), in which two nodes are considered neighbor if they are within each other's communication range (r). Based on this model, the density of the generated networks is: $\delta = \pi * r^2 * n / a^2$. The deployment area size was changed to obtain different densities $\delta \in [4, 20]$.

CoLaNet is based on the MinDegree tree which is not the best routing tree in terms of number of hops or energy consumption. That is why, in addition to the MinDegree tree, we evaluated the performance of our contributions using: the Geographic tree (each node chooses as its parent the closest neighbor to the sink, in terms of distance) and the Hop Count tree (each node chooses as its parent the closest neighbor to the sink, in terms of number of hops).

Due to lack of space, we only present the results obtained with the MinDegree tree, to make comparison feasible with CoLaNet. Experiments with the other routing trees give similar results.

We note that, similarly to LEACH cited in [1], an initialization phase is used before the scheduling process. In this phase, the sink processes the routing protocol using the positions of the nodes. After this phase, the sink computes the TDMA and broadcasts the routes and the TDMA in the network.

4.1 Evaluated metrics

Average latency: The average latency was computed as the average of all total communication latencies from each node to the sink. This latency is the number of slots before the source node transmits (the reference slot being the first slot),

and for every node in the routing path, except for the sink, one slot for transmission and the number of slots before its parent can transmit.

Average normalized latency: The average normalized latency is computed as the average of per link latencies for all the nodes in the network. The latency per link for a node is computed as the total latency for the node divided by the number of hops in the routing path for the node to the sink.

Schedule length: The schedule length is computed as the number of slots of the obtained schedule.

Duty cycle: The duty cycle is computed as the ratio of the active period (estimated by the number of slots used either in transmission or in reception) to the total period (the schedule length). A lower duty cycle results in a lower energy consumption [10].

4.2 Improved-CoLaNet (I-CoLaNet)

As already explained, CoLaNet does not consider latency when defining the transmission schedule. Continuing to color all its neighbor in the tree does not reflect the direction of the communications in the routing tree. Whereas, our objective is latency optimization. This difference in objectives makes us propose I-CoLaNet, an improved version of CoLaNet in which an order is given to nodes during the vertex-coloring, in order to improve communication latencies, simultaneously for all sensor nodes.

As in CoLaNet, I-CoLaNet uses the MinDegree tree as a routing tree and a similar vertex-coloring algorithm. The idea of I-CoLaNet is the following:

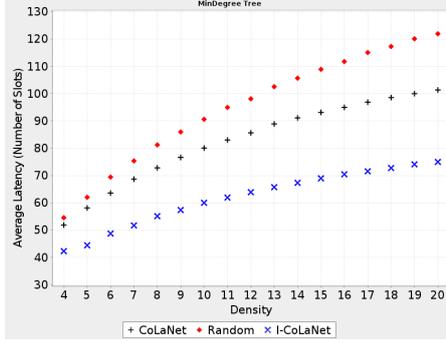
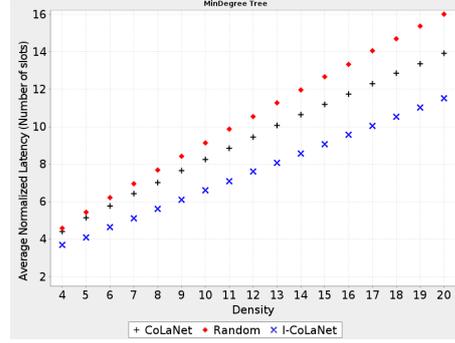
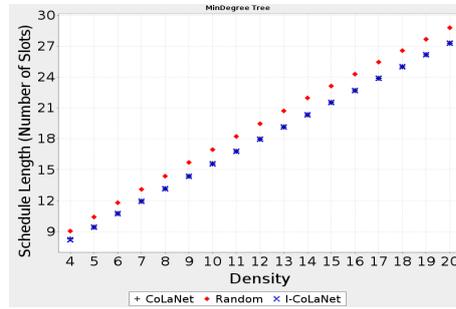
- 1- Color the max-degree node in the routing tree.
- 2- Continue with coloring recursively each node that has an already colored parent node in the tree.
- 3- Restart with step 1, without considering the colored nodes, until all nodes have a color.
- 4- Transform the obtained coloring into a schedule for each node.

In this coloring solution, nodes are colored according to the parent-child relation and the data flow direction given by the routing tree. Therefore, the communication slot of children are generally close to the communication slot of their parents, thus reducing communication latencies.

5 Performance evaluation

This section presents the evaluation of the performances of Rand-LO, Depth-LO, Depth-ReLO and I-CoLaNet obtained by simulations.

First, the performance of I-CoLanet in terms of latency and schedule length was compared with that of CoLaNet and with that of Random TDMA. Random TDMA is a basic protocol (which is not cross-layered) where slots are given to nodes randomly (no specific order is defined) while ensuring that communications are collision free. The computed metrics represent averages of the results for the 5000 randomly generated networks.


 Fig. 5: Average Latency depending on δ

 Fig. 6: Average Normalized Latency depending on δ

 Fig. 7: Schedule Length depending on δ

As shown in Figure 5, CoLanet and I-CoLaNet have better average latency than Random TDMA, because they use information coming from the routing tree in the TDMA schedule. These results show that cross-layer protocols have better performance than traditional layered protocols. Moreover, as I-CoLaNet aims to respect the aspects of parent-children relation and direction in the routing tree, its average latency is better than that of CoLaNet.

Similar to the average latency results, Figure 6 shows that I-CoLaNet has a better average normalized latency because it gives an importance to the order of communication between a parent and its children in the routing tree. This aspect is not dealt with in CoLaNet, that is why its average normalized latency is close to that of Random TDMA where no routing order is respected.

The CoLaNet allocation algorithm is based on an approximation solution for the vertex-coloring problem. Therefore, the obtained schedule length (corresponding to the number of colors) is not optimal. Figure 7 shows that I-CoLaNet has the same schedule length as CoLaNet which gives the same energy consumption. That is because I-CoLaNet and CoLaNet are based on similar vertex-coloring algorithms. Also, both CoLaNet and I-CoLaNet have lower schedule length than the schedule length of Random TDMA, thanks to the vertex-coloring algorithm. Moreover, I-CoLaNet has better performance than CoLaNet because it does not just reduce the length of the schedule, but it also searches opti-

mized allocation for the latency metric. The parent-children relation and the communication direction given by the routing tree is taken into consideration while coloring the nodes. This helps reducing both the average and the average normalized latencies. Since I-CoLaNet has better performances (in terms of latencies) than Random TDMA and CoLaNet, we will compare the scheduling approaches based on three routing tree traversals only with I-CoLaNet.

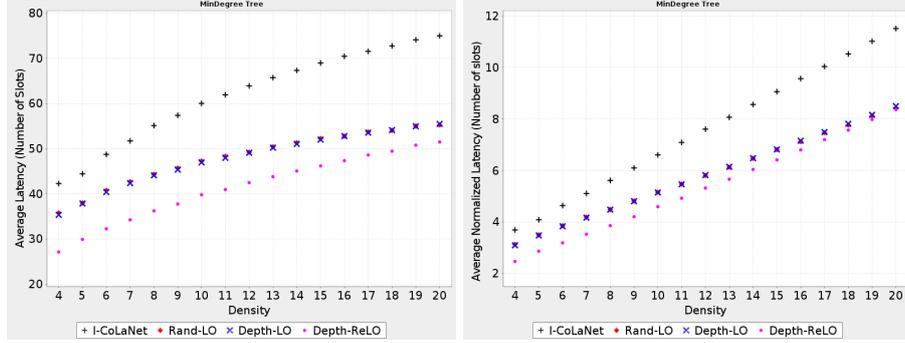


Fig. 8: Average Latency (left side) and Average Normalized Latency (right side)

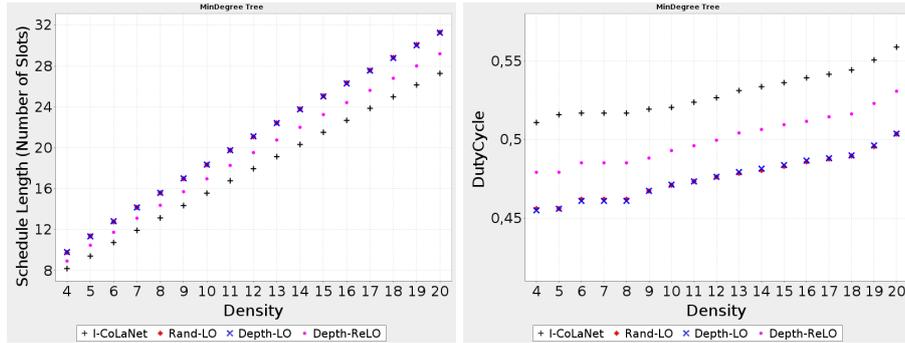


Fig. 9: Schedule Length (left side) and Duty Cycle (right side)

As shown in the left side of Figure 8, it is clear that starting slot allocation with the leaves of the routing tree gives better average latency results than starting with the max-degree node. Starting with the leaves of the tree gives the chance to nodes that are far from the sink to be scheduled first. In Depth-LO, we order leaves by their depth in the routing tree, to give the farthest nodes from the sink the privilege to be scheduled first. This gives the same average latency for MinDegree tree because this tree has few routes and is not balanced (routes with different lengths), so the order does not have great impact. Moreover, eliminating already scheduled nodes from the routing tree before moving to the next step of the allocation (as in Depth-ReLO) allows to give more importance to long routes so their nodes can be given earlier slots, which improves the average latency.

The right side of Figure 8 shows results of average normalized latency when varying the network density. Giving earlier slots to deep leaves and privileging the nodes of long routes, as in Depth-LO and Depth-ReLO, reduces even more

the gap between the slots of a node and its parent in the routing tree and gives better average normalized latency. The difference between Depth-LO and Depth-ReLO is that the first one privileges one node of each route, while the second one privileges more nodes of the same route (if the route is long) thanks to the elimination. That is why Depth-ReLO has the best average normalized latency.

As shown in the left side of Figure 9, I-CoLaNet has the fewest number of slots. Our traversal-based allocation methods generate schedules with one or two slots longer than the schedules of I-CoLaNet. Nevertheless, longer schedules of our approaches are counteracted by better transmission latencies.

The right side of Figure 9 shows that our approaches have lower duty cycle and thus a smaller energy consumption. Rand-LO and Depth-LO have the best duty cycle thanks to their schedule length. Depth-ReLO comes in second place in terms of duty cycle because its schedule length is smaller than the first ones. Unlike I-CoLaNet, which reduces the schedule length with the vertex-coloring algorithm but most of the slots are active (either in transmission or in reception) and this increments the duty cycle and therefore, the energy consumption. We remind that nodes can turn off their radio during the inactive slots.

All the previous results are averages and do not reflect data individually. We analyze the extent of variability of the obtained values in relation to the correspondent mean, on the basis of the coefficient of variation (Relative Standard Deviation). It is expressed in percentage as $\frac{\sigma}{\mu}$ where σ is the standard deviation and μ is the mean. We remind that the size of the analyzed population is 5000. Table 1 shows the intervals of the coefficient of variation, irrespective of density, for all the three metrics, depending on the analyzed allocation methods. All the obtained coefficients of variation are inferior to 11.5% which proves that the absolute values are not very scattered.

	Latency	Normalized Latency	Schedule Length
Random	[8.9 ; 11.2]%	[5.8 ; 7.2]%	[3.1 ; 5.2]%
CoLaNet	[8.8 ; 9.5]%	[5.5 ; 7.6]%	[3.6 ; 4.8]%
I-CoLaNet	[7.4 ; 8.3]%	[6.6 ; 8.5]%	[3.6 ; 5.0]%
Rand-LO	[10.2 ; 11.5]%	[7.4 ; 9.2]%	[3.6 ; 5.4]%
Depth-LO	[9.4 ; 11.5]%	[7.4 ; 9.0]%	[3.6 ; 4.6]%
Depth-ReLO	[10.3 ; 11.0]%	[7.6 ; 9.8]%	[3.3 ; 5.9]%

Table 1: Intervals of the coefficient of variation irrespective of density

Synthesis of simulations CoLaNet is a cross-layer MAC protocol in which the scheduling is constructed applying a vertex-coloring algorithm on the MinDegree routing tree. It is not oriented to optimize the latencies of communications.

I-CoLaNet is also based on the vertex-coloring algorithm but respecting the parent-child order; it reduces the latency by 19% compared with CoLaNet.

The main goal of the Slot Allocation Method used in our scheduling approaches is to reduce the gap between the slot of a child and the slot of its parent in the routing tree. Therefore, when considering communications on routing paths, latency is reduced. Compared with CoLaNet, Rand-LO improves latency by 33%, Depth-LO improves it by 35% and Depth-ReLO by 54%. Even though all our approaches generate longer schedule lengths than I-CoLaNet, duty-cycle is improved of 7% for Depth-ReLO and of 11% for Rand-LO and Depth-LO. Better

duty-cycle helps minimizing energy consumption in collision-free MAC protocols because inactive nodes may be switched to sleep mode.

6 Conclusion and future work

This paper presents new scheduling methods to reduce latency of communications simultaneously for all nodes of the sensor network, based on the routing information. We present a different slot allocation method associated with several traversals of the routing tree, beginning with its leaves, that are Rand-LO, Depth-LO and Depth-ReLO. Results of simulations show that these techniques have better performance than the state of the art and improve latency up to 54%. Even if the obtained schedules are longer than the schedules of CoLaNet by 1 or 2 slots on average, the duty cycle is improved up to 11% which results in less energy consumption. These results show the importance of the traversal of the routing tree in the slot scheduling method and reveal the need to study more extensively the effect of nodes enumeration of the network's graph on the slot allocation method.

References

1. I. F. Akyildiz and W. Su and Y. Sankarasubramaniam and E. Cayirci, Wireless sensor networks: a survey, *Computer Networks*, 2002, volume 38, pages 393–422
2. D.P. Mendes Lucas and J.P.C. Rodrigues Joel, A survey on cross-layer solutions for wireless sensor networks, *Journal of Network and Computer Applications*, volume 34, number 2, March, 2011, pages 523–534
3. T. Melodia and M. C. Vuran and D. Pompili, The state of the art in cross-layer design for wireless sensor networks, *Wireless Systems and Network Architectures in Next Generation Internet*, Springer, 2006, pages 78–92
4. M. Brzozowski and H. Salomon and P. Langendoerfer, Support for a long lifetime and short end-to-end delays with TDMA protocols in sensor networks, *Int. Journal of Distributed Sensor Networks*, 2012, Hindawi Publishing Corporation
5. C. Cheng-Fu and C. Kwang-Ting, CoLaNet: a cross-layer design of energy-efficient wireless sensor networks, *IEEE Systems Communications*, 2005, pages 364–369
6. S. Pemmaraju and S. Skiena, *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, Cambridge University Press, 2003
7. S. Schmid and R. Wattenhofer, Algorithmic models for sensor networks, 14th WP-DRTS, 2006, pages 450–459
8. J. O'Madadhain and D. Fisher and P. Smyth and S. White and Y. Boey, Analysis and visualization of network data using JUNG, *Journal of Statistical Software*, volume 10, number 2, pages 1–35, 2005
9. F. Atay Onat and I. Stojmenovic, Generating Random Graphs for Wireless Acuator Networks, *IEEE WoWMoM*, pages 1–12, 2007
10. Z. Shuguo and S. Ye-Qiong and W. Zhi and W. Zhibo, Queue-MAC: A queue-length aware hybrid CSMA/TDMA MAC protocol for providing dynamic adaptation to traffic and duty-cycle variation in wireless sensor networks, 9th IEEE WFCS, 2012, pages 105–114