Work in Progress: Two ID-free Distributed Distance-2 Edge Coloring Algorithms for WSNs

André C. Pinho[1,2], Alexandre A. Santos[1,2], Daniel R. Figueiredo[1], and Felipe M. G. França[1]

[1] PESC/COPPE, Universidade Federal Rio de Janeiro, Rio de Janeiro, 21941-972, Brazil {felipe,daniel}@cos.ufrj.br

[2] Information Technology Division, Centro Tecnológico do Exército, Rio de Janeiro, 23020-470, Brazil {apinho,alexandre}@ctex.eb.br

Abstract. One of the most important problems for Wireless Sensor Networks (WSNs) is energy consumption since it ultimately determines the lifetime of the system. Medium Access Control (MAC) protocols based on schedules (e.g., TDMA) play an important role, since collisions and idle listening can be avoided, effectively reducing energy consumption. The problem of determining good transmission schedules for WSNs can be mapped to the *distance-2* edge coloring problem in graphs, where edge colors represent slots in a TDMA-based MAC protocol, for example. In this paper, we propose and evaluate two new probabilistic and distributed *distance-2* edge coloring algorithms that require no global node identifiers. We obtain analytical results for the worst-case convergence time. Moreover, we use simulations to evaluate the performance of the algorithms with respect to several metrics. Our findings indicate a tradeoff between convergence time and message overhead versus number of colors used.

Keywords: WSN, MAC, algorithms, distance-2

1 Introduction

1.1 Wireless Sensor Network

Wireless Sensor Networks (WSNs) are composed of a large number of low cost, low power, multi function, small sensor nodes capable of sensing, processing and communicating [1]. Among the characteristics that distinguish WSNs from other wireless networks, one of most important is limited access to energy or even finite energy. Since the radio communication subsystem of sensor nodes is greatly responsible for the consumption of energy, the Medium Access Control (MAC) protocol plays an important role, as it determines when the radio transmits and listens for transmissions. In particular, energy efficiency is ranked as one of the most important attributes of MAC protocols for WSNs, leaving behind attributes such as fairness, latency and channel utilization [4].

It is known that MAC related energy consumption in WSN is mainly due to packet collisions caused by hidden terminals and continuous idle listening [4]. The hidden terminal problem can be illustrated using the communication graph shown in Figure 1. In this graph, vertices correspond to sensors and edges indicate that two sensors are within direct communication range of each other. The hidden terminal problem occurs among vertices A, B and C, since nodes A and C are not aware of each other. Thus, when the transmissions of the two nodes overlap in time, a packet collision will occur in node B, and a retransmission will be required. The second concern is inherent to the characteristics of short-range radios commonly used WSN applications. The power consumption of listening in these radios has the same order of magnitude as power consumption of receiving or transmitting packets [3]. Therefore the energy wasted from continuous idle listening periods can become a significant burden for obtaining a prolonged network lifetime. Thus, schedule-based protocols emerge as a promising approach for developing energy efficient MAC protocols. It is worth noticing that, in the same spirit of what was presented in [12], our work addresses the issues of communication scheduling and sensing scheduling in a fully decoupled manner. In this work, we focus solely on the communication scheduling.

1.2 Distance-2 Coloring

The problem of developing a schedule-based MAC protocol for WSNs can be well illustrated using the communication graph, as shown in Figure 1. In particular, given the communication graph, one can determine a collision-free schedule among the nodes that avoids hidden terminals and allows for selective listening. For example, communication links can be assigned a time slot, such as in a TDMA-based MAC protocol, which would then determine when nodes should transmit and listen for transmissions.

The problem of determining a collision-free schedule for nodes of WSNs that avoids hidden terminals can be mapped to the *distance-2* edge coloring problem in graphs. In this mapping, edge colors represent time slots in a TDMA-based MAC protocol, for example. Let G be an undirected graph, representing the communication graph. We say that two edges of G are within distance-2 of each other if either they are adjacent or there is some other edge that is adjacent to both of them. A distance-2 edge coloring of G is an assignment of colors to edges so that any two edges within distance-2 of each other have distinct colors. There are two equivalent ways to view the coloring constraint: *(i)* any two edges in any 3-edge path cannot have the same color; *(ii)* any two neighboring nodes cannot have any adjacent edge with the same color. Note that such coloring would clearly avoid the hidden terminal problem, if colors correspond to a transmission schedule.

The *distance-2* edge coloring problem can be mapped to the *distance-2* **vertex** coloring problem in graphs. We say two vertices of G are within distance-2 of each other if they are adjacent or if they are adjacent to a common vertex (i.e., have a neighbor in common). A distance-2 vertex coloring of G is an assignment of colors to vertices so that any two vertices within distance-2 of each other have distinct colors. Given a *distance-2* vertex coloring of G we can obtain a *distance-2* edge coloring by associating with each edge the non-ordered pair of the colors of the nodes incident to the edge. Note that this mapping is clearly a *distance-2* edge coloring of the graph, since any two neighboring nodes will not have edges with repeated colors. Thus, the *distance-2* edge coloring problem can be indirectly solved via the *distance-2* vertex coloring of the graph.

1.3 Our Contribution

In this paper we propose and evaluate two new probabilistic distributed *distance-2* edge coloring algorithms that require no global identifiers on the nodes. The first algorithm presented, known as $Edge^3 - Sched$, performs a *distance-2* edge coloring directly. The second algorithm, known as $Node^2 - Sched$, performs a *distance-2* vertex coloring, which is then mapped to a an edge coloring. In this paper, both algorithms are considered with single purpose of defining a collision-free link scheduling. Our numerical evaluation of the two algorithms indicates a trade off in their performance, in particular, with respect to convergence time and number of colors used. Our objective is not to obtain the optimal (minimum) number of colors required to produce a *distance-2* edge coloring. Note that an optimal color assignment, despite having potentially very long convergence time, does not necessarily produce the best schedule for a given application. We thus aim at algorithms that can efficiently produce an edge coloring in the lack of global node identifiers.

The remainder of the paper is organized as follows. Some related work is presented in Section 2. Section 3 describes the two new distributed algorithms in detail. Subsequently, the simulation setup is reported in Section 4, with evaluation and results being presented in Section 5. Finally we present our conclusions and future work in Section 6.

2 Related work

The problem of developing schedule-based collision-free MAC protocols for WSNs has been extensively studied in the recent years. Trigoni et al. [7], propose a new protocol that carefully schedules message transmissions to avoid packet collisions. Their protocol assumes nodes are organized in a grid structure and a collision-free scheduling is established through simultaneous link activation according to predefined distances between the nodes and transmission direction. Under this scenario, sensors' radios can work in a deterministic duty cycle saving energy by turning off and avoiding idle listening periods.

Kulkarni et al.[9] proposed a self-stabilizing, deterministic algorithm for TDMA-based MAC in WSNs where a sensor node is aware only of its neighbors. The proposed algorithm relies on the initial slot assignment that is dynamically determined by the nodes through periodically diffusion of messages sent by a base station.

Ergen et al. [10] proposed a distributed algorithm based on distributed coloring of nodes. This work consider a network comprising a single access point (AP) and several sensor nodes that use the same AP for transferring their messages. They also obtained an upper bound for these schedules as a function of total number of packets generated in the network.

Pantazis et al. [12] proposed a TDMA-based scheduling scheme that balances energy-saving and end-to-end delay. In their approach, sensor nodes send neighboring and routing information to a gateway, which constructs the schedule. The schedule is then sent to the sensor nodes using flooding techniques.

Gandham et al. [6] proposed a distributed algorithm for obtaining a TDMA-based MAC schedule based on the edge-coloring problem. Their work aims in minimizing the number of colors produced and considers that nodes have a global identifiers.

Balakrishnan et al. [5] considers the 2-distance edge coloring problem to determine the maximum capacity of WSNs. Their algorithm assumed nodes have a unique global identifier and are running globally synchronized clocks.

Our work differs from the above in following aspects: (i) our algorithms are fully decentralized (there is no central node); (ii) no global identifiers are needed across the set of sensor nodes, a strong requirement for scalable WSNs as mentioned in [1]; (iii) our probabilistic algorithms are fully independent of network topology, including the case of disconnected networks (i.e., communication graph is composed of multiple connected components). It is worth noting that lack of global node identifiers necessarily requires a non-deterministic approach to break the symmetry (i.e., color edges), as demonstrated in [2].

3 Distributed Algorithms

The two distributed algorithms proposed in this paper use an election mechanism to determine which nodes, on each step, can color their adjacent edges or themselves. In summary, the algorithms work as follows. Each node generates a random *sort* number which is then sent to their h-neighborhood in the communication graph (the h-neighborhood of node v is the set of nodes that are at distance at most h from v). This message is sent through multi-hop communication. The nodes that have the highest sort number among the sort numbers it receives (known as *winners*) then color the edges adjacent to them. Once a node is a winner, it ceases to participate in future elections, performing only message retransmissions (i.e., aiding multi-hop communication) afterwards. The process continues until all edges (or nodes) have been colored. The parameter h will be defined in following subsections for each algorithm proposed.

3.1 $\boldsymbol{Edge^3 - Sched}$

As briefly discussed in Section 1.3, the $Edge^3 - Sched$ algorithm performs a distance-2 edge coloring of the communication graph directly. Observing Figure 1, and assuming edges *(C,D)* and *(C,F)* have been colored, note that when

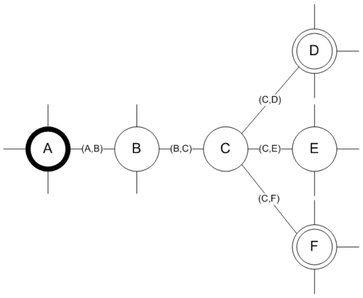

Fig. 1. $Edge^3 - Sched$ constraints

node A attempts to color its edges, in particular edge *(A,B)*, it cannot use the colors already assigned to edges *(C,D)* and *(C,F)*. Moreover, note that edges *(B,C)* and *(C,E)* cannot be colored at the same time because there could be a conflict. In this way the election mechanism needs to reach a 3-neighborhood (nodes B, C and E) and information about colored edges needs to reach a 2-neighborhood (nodes B and C). A node stops when its 1-neighborhood and 2-neighborhood has all its edges colored.

Before describing the details of this algorithm, it is necessary to define some terminology. *winners*: nodes elected to color edges incident to them; *node status*: active or inactive, as the nodes participate or not in the election process; *constraints*: colors already used to color edges or nodes; *sort number*: a random number chosen uniformly in the the interval from 0 to 999 used to determine the winners.

Basically the $Edge^3 - Sched$ functions as follows: all nodes in the network starts in the *active status* exchanging information about their *constraints*, *sort number* and *status*. The *constraints* and status are transmitted to the 2-hop neighborhood and the *sort number* to the 3-hop neighborhood. Once a node becomes a winner, it color its incident edges respecting the constraints it receives. It then changes status, going to the *inactive* status. An *inactive* node becomes only a relay aiding multi-hop communication. The process terminates when all nodes in the 2-hop neighborhood become *inactive*.

3.2 $Node^2 - Sched$

Differently from the $Edge^3 - Sched$, the $Node^2 - Sched$ algorithm achieves a distance-2 edge coloring indirectly, via a distance-2 vertex coloring of the communication graph. It is worth reminding that, although $Node^2 - Sched$ performs a vertex coloring, the main goal is to establish a distance-2 edge coloring that will lead to a link scheduling. The Figure 2 shows an example of distance-2 node coloring and its corresponding distance-2 edge coloring. The value within the nodes in the figure represents their color. An edge is colored using the non-ordered pair of the color of the nodes incident to the edge. We note that this indirect edge coloring mechanism obeys the restriction imposed by distance-2

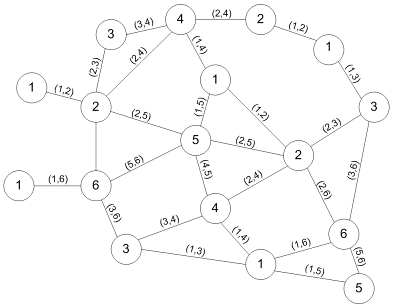

Fig. 2. Edge Coloring by Node Coloring

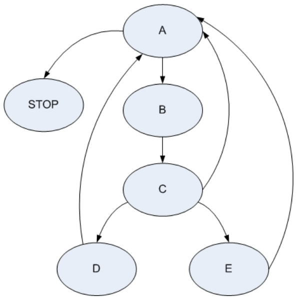

Fig. 3. $Node^2 - Sched$ state machine

edge coloring, namely, that any two neighbors cannot have edges with the same color.

The detailed state machine of $Node^2 - Sched$ algorithm is depicted in Figure 3 and its functionality is described below. Note that the terminology used in the $Edge^3 - Sched$ is perfectly compatible with the description of this algorithm.

$\boldsymbol{Node^2 - Sched}$ State Machine

- **state A:** all nodes send to their 1-hop neighbors the randomly chosen *sort number* and their *status*. Nodes receive these messages and store the respective sort numbers. The inactive nodes go to state F and stop. All others go to state B;
- **state B:** all active nodes select the highest sort number among the numbers received and its own. Afterwards the nodes send the maximum sort number to all active 1-hop neighbors. The active nodes receive these messages and compare the values with their own sort number. Nodes that have a sort number greater than any other received become winners. All nodes then go to state C;
- **state C:** all nodes send to their active neighbors their condition: *winner* or not. This procedure aims in establishing a distributed knowledge of nodes' conditions in the communication graph. The *winner* nodes go to state E; the *winner* neighbors go to state D; the other nodes return to state A;
- **state D:** the *winner* neighbors send to their *winners* their own color and the colors of their neighbors. These are the *constraints* that the *winners* will take into consideration when coloring themselves. After this, *winner* neighbors wait for messages containing the new color of nodes. Afterwards, nodes update their *status*. In particular, a node becomes *inactive* if the node has already colored itself and all its neighbors have also colored themselves. Nodes then return to state A;
- **state E:** Nodes wait to receive the *constraints* from their neighbors in *state E* and then color themselves. Afterwards the nodes update their status (nodes in this state become inactive if all their neighbors have been colored), send its own color to their neighbors and return to state A;

3.3 Complexity Analysis

We generalize the proof presented in [13] by developing the following complexity analysis. Let $G = (N, E)$ be the communication graph for which $distance - 2$ edge (or vertex) coloring will be performed using one of the algorithms presented. During the execution nodes can color incident edges (or itself) while others wait for a chance to do the same. Therefore, we define nodes that continue executing the algorithm and have incident edges not colored (or itself) as probabilistic nodes, while the nodes that have all its incident edges (or itself) colored as deterministic nodes.

Considering h as the hop distance (h-neighborhood) to be considered for a node to become a winner and n the distance to be considered so that a node finishes its execution, we have the following configuration for the two presented algorithms: $Edge^3 - Sched : h = 3, n = 2$ and $Node^2 - Sched : h = 2, n = 1$.

1. Execution
   (a) Every node sends its sort number and constraints to all its neighbors;
   (b) Every node, upon receiving a message re-sends it $(h - 1)$ hops;
   (c) Every node that has superiority on the h-neighborhood colors its edges (or itself) and cease to be a probabilistic node, performing only retransmissions;
   (d) Every node stops executing when there is no probabilistic nodes on its n-neighborhood.
2. Correctness
   The algorithms are correct if they avoid the hidden terminal problem and if they do not present any deadlock condition. Therefore, we can start the analysis as follows:
   (a) The nodes compete to be a winner in their h-neighborhood using the sort number (probabilistic nodes). After an specific node is declared a winner, this node will no longer compete (becoming a deterministic node). Thereby every node will be a winner at some point in the execution of the algorithm. After becoming a deterministic node, a node waits all other nodes in its n-neighborhood to become deterministic nodes. This stop condition is sufficient to ensure that the algorithm will never suffer deadlock or starvation;
   (b) Since the algorithm ensures the association of time slots considering the h-neighborhood constraints then there remains whether the synchronization mechanism will ensure correct use of the links. As the algorithm ensures that every pair of neighbors in the network has distinct link colors, we have that collisions cannot occur.

3. Notations and Analysis
   Let $N_k \subseteq N$ denote the set of probabilistic nodes after k algorithm steps for k integer and positive. Let $viz_{N_k}(v)$ denote the set of probabilistic neighbors in the h-hop neighborhood of v. Let $G_k = (N_k, E_k)$ denote the induced subgraph of G after k steps. Let d_i^k be the sort number generated by the

node v_i in the k. Nodes that have the largest d_i^k in their h-neighborhood will be the winners.

(a) Let S_i^k be the event that corresponds to node v_i being a winner in step k;

(b) Let S^k be the event that corresponds to some node being a winner in step k;

(c) Let $B_j^k(v_i, \alpha)$ be an event that occurs whenever $d_i^k = \alpha > d_j^k$ for some $v_j \in viz_{N_k}(v_i)$, that means that v_i wins some probabilistic node in h-neighborhood. It is important to note that the $B_j^k(v_i, \alpha)$ represents independents events for all $v_j \in viz_{N_k}(v_i)$.

(d) Considering that the sort number was generated from a die with f faces, it defines the following probability:

$$Pr(d_i^k = \alpha) = \frac{1}{f} \Rightarrow Pr(B_j^k(v_i, \alpha)) = \frac{\alpha}{f}; \tag{1}$$

(e) Thus we can continue the complexity development. Now consider the probability that node v_i is a winner in step k:

$$Pr(S_i^k) = \sum_{\alpha=0}^{f-1} Pr(d_i^k = \alpha) * Pr(S_i^k / d_i^k = \alpha) \tag{2}$$

$$Pr(S_i^k / d_i^k) = Pr(\bigcap_{v_j \in viz_{N_k}(v_i)} B_j^k(v_i, \alpha)) = \prod_{v_j \in viz_{N_k}(v_i)} (\frac{\alpha}{f}) \geq (\frac{\alpha}{f})^{\Delta_{kh}} \tag{3}$$

where Δ_{kh} is the number of probabilistic nodes in h-neighborhood in step k.

(f) Replacing 3 in 2 and observing that $Pr(d_i^k = \alpha) = \frac{1}{f}$:

$$Pr(S_i^k) \geq \sum_{\alpha=0}^{f-1} (\frac{1}{f})(\frac{\alpha}{f})^{\Delta_{kh}} = (\frac{1}{f^{\Delta_{kh}+1}}) \sum_{\alpha=0}^{f-1} \alpha^{\Delta_{kh}}, \forall v_i \in N_k \tag{4}$$

(g) Using lower limit presented in [14]:

$$\sum_{\alpha=0}^{f-1} \alpha^{\Delta_{kh}} \geq (\frac{(f-1)^{\Delta_{kh}+1}}{\Delta_{kh}+1} + \frac{(f-1)^{\Delta_{kh}}}{2}) \quad , \quad \text{where } \Delta_{kh} \geq 1 \tag{5}$$

(h) Finally replacing 5 in 4 :

$$Pr(S_i^k) \geq (\frac{1}{\Delta_{kh}+1})(1 - \frac{1}{f})^{\Delta_{kh}+1} + (\frac{1}{2f})(1 - \frac{1}{f})^{\Delta_{kh}} \tag{6}$$

(i) Considering a complete graph with n nodes as the worst case, and therefore $\Delta_{kh} = n - 1$:

$$Pr(S_i^0) \geq (\frac{1}{n})(1 - \frac{1}{f})^n + (\frac{1}{2f})(1 - \frac{1}{f})^{n-1} = (\frac{2(f-1)+n}{2fn})(1 - \frac{1}{f})^{n-1} \tag{7}$$

$$Pr(S^k) = Pr(\bigcup S_i^k) \geq Pr(\bigcup_{i=1}^{n} S_i^0) = \sum_{i=1}^{n} Pr(S_i^0) \tag{8}$$

$$Pr(S^k) = (\frac{2(f-1)+n}{2f})(1 - \frac{1}{f})^{n-1} \tag{9}$$

(j) We can now determine the average attempts required for a node to be a winner in step k

$$T(n) = (\frac{2fn}{2(f-1)+n})(\frac{f}{f-1})^{n-1} \tag{10}$$

(k) Thus, the worst case convergence complexity is:

$$O(f(\frac{f}{f-1})^{n-1}) \tag{11}$$

4 Experimental Setup

We have developed a simulation environment to evaluate the two proposed algorithms. The underlying communication network was implemented using a multi-threaded java simulation, where each sensor node corresponds to the instance of an unique thread. Each thread independently operates the state machine described in Section 3. Threads communicate with other threads in accordance with the underlying communication network, which determines the direct neighbors of a node. We adopt an unicast communication model where all communication is fully reliable.

We consider two types of topologies to evaluate the algorithms: a regular grid and a random node placement. In the grid topology, we consider networks with 25, 49, 100, 200 and 400 nodes, regularly distributed on a square grid, such that radio coverage creates links only between the closest neighbors (maximum degree is 4). In the random node placement, nodes are randomly (uniformly) distributed in dimensionless 200x200 square area. We consider scenarios with 50, 100, 150, 200 and 400 nodes and radio coverage radius of 30. All nodes within distance 30 of a given node correspond to direct neighbors in the communication graph. Each scenario was simulated 100 times and we report sample averages of all metrics.

The following performance metrics were considered: convergence time, number of messages exchanged, number of bits transmitted (i.e., message sizes), and number of colors used by the algorithm. The convergence time is measured in steps, which correspond to transitions in the state machine, and a node is said to have converged once it enters the stop state.

5 Evaluation and Results

5.1 Grid Topology

Figures 4, 5, 6 and 7 show a direct comparison of the performance of the two algorithms on convergence time, messages exchanged, bits transmitted, and number of colors used, respectively. We first note that $Node^2 - Sched$ converges much faster than $Edge^3 - Sched$, and for larger topologies its at least 4 times faster. Consequently, the number of messages exchanged by $Node^2 - Sched$ is much smaller, reaching a factor of 6 for larger topologies. This also reflects on number of bits transmitted, which for larger topologies is around 10 times smaller. On the other hand, the $Node^2 - Sched$ algorithm uses more colors than its counterpart to color the edges. In particular, $Node^2 - Sched$ uses around 2 times more colors than $Edge^3 - Sched$ for larger topologies. The results clearly indicate a tradeoff between convergence time and message efficiency versus number of colors used when considering the two proposed algorithms.

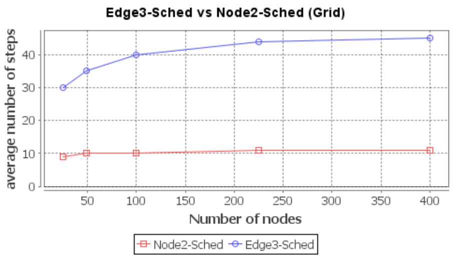

Fig. 4. Edge vs Node (steps)

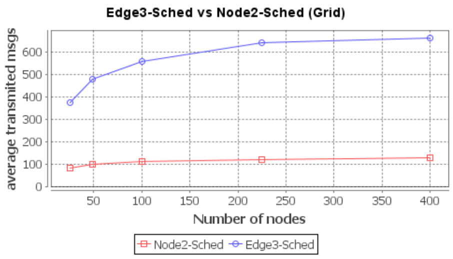

Fig. 5. Edge vs Node (messages)

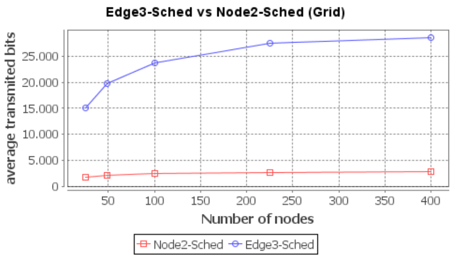

Fig. 6. Edge vs Node (bits)

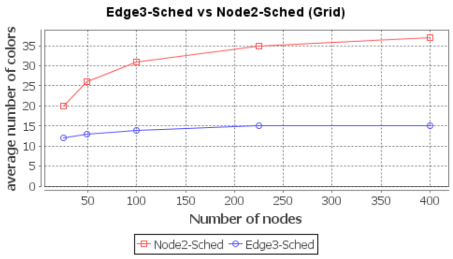

Fig. 7. Edge vs Node (colors)

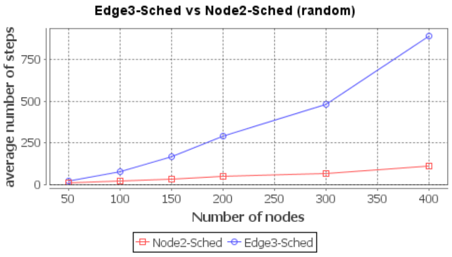

Fig. 8. Edge vs Node (steps)

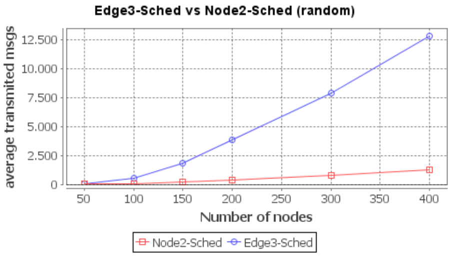

Fig. 9. Edge vs Node (messages)

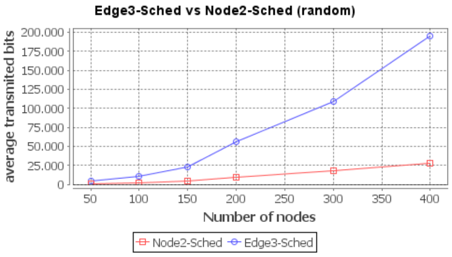

Fig. 10. Edge vs Node (bits)

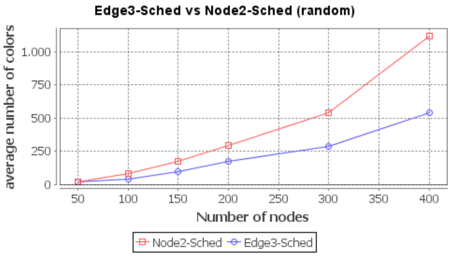

Fig. 11. Edge vs Node (colors)

5.2 Random Node Placement

We now consider the networks formed by random placement of node, as described in Section 4. Figures 8, 9, 10 and 11 present a direct comparison of the performance of the two algorithms on convergence time, messages exchanged, bits transmitted, and number of colors used, respectively. Similarly to the results obtained in the grid topology, we note that $Node^2 - Sched$ converges much faster than $Edge^3 - Sched$ and requires many fewer messages and bits transmitted. However, such performance differences have become more pronounced, specially with higher number of nodes. Interestingly, though, the number of colors used by the $Node^2 - Sched$ did not increased when compared to $Edge^3 - Sched$. Again, as with the grid topologies, we observe a trade-off between convergence time and overhead (in messages and bits) versus number of colors used by the algorithms. It is worth mentioning that, except for the number of colors used, $Node^2 - Sched$ has superior performance when compared to the link-scheduling algorithm presented in [6], which corresponds to a very similar scenario.

6 Conclusion

We have proposed and evaluated the performance of two distance-2 edge coloring algorithms for WSNs. To the best of our knowledge, these are the first probabilistic, distributed and global ID-free algorithms that can fully yield support to decentralized TDMA-based MAC protocols. Simulations have shown that $Edge^3 - Sched$, a direct edge coloring algorithm, has a higher overhead in terms of convergence time and message exchanged, while using less colors to color the edges. In the other hand, the $Node^2 - Sched$ algorithm has a much smaller overhead while requiring more colors in its coloring. We note that either algorithm could serve as the basis for a TDMA-based MAC protocol, depending on the application requirement. Note that minimizing the number of colors in the coloring may not necessarily lead to the optimal schedule, in terms of energy efficiency, and will depend on the application requirements.

References

1. Akyildiz, I. F., Su, Sankarasubramaniam, W., Y., and Cayirci, E., A Survey on Sensor Networks, *IEEE Communications Magazine* 40(8) (2002) 102–114.
2. Angluin, D., Local and Global Properties in Networks of Processors, *Proc. of 12th ACM Symposium on Theory of Computing* (1980) 82–93.
3. Reason, J. and Rabaey, J. M., A Study of Energy Consumption and Reliability in a Multi-hop Sensor Network, *ACM Mobile Comp. and Comm. Rev.* 8(1) (2004) 84–97.
4. Ye, W. and Heidemann, J., Medium Access Control in Wireless Sensor Networks, *USC/ISI Tech. Rep.* ISI-TR-580, Oct. (2003).
5. Balakrishnan et al., The Distance-2 Matching Problem and its Relationship to the MAC-layer Capacity of Ad hoc Wireless Networks, *IEEE J. on Selected Areas in Communications* 22(6) (2004) 1069–1079.
6. Gandham, S. Dawande and M. Prakash, R., Link Scheduling in Sensor Networks: Distributed Edge Coloring Revisited, *J. of Par. and Distr. Comp.* 68(8) (2008) 1122–1134.
7. Trigoni, N., Yao, Y., Demers, A., Gehrke, J., Rajaraman, R., WaveScheduling: energy-efficient data dissemination for sensor networks, *Proc. of the 1st Inter. Work. on Data Management for Sensor Networks* (2004) 48–57.
8. Arumugam, M. and Kulkarni, S., Self-Stabilizing deterministic TDMA for Sensor Networks, *Proc. of 2nd Inter. Conf. on Distr. Comp. and Internet Tech. (ICDCIT)*, LNCS 3816 (2006) 22–24.
9. Kulkarni, S and Arumugam, M., SS-TDMA: A self-stabilizing MAC for sensor networks, in: *Sensor Network Operations*, Wiley-IEEE Press (2006).
10. Ergen, S. C. and Varaja, P., TDMA scheduling algorithms for sensor networks, Tech. Rep., University of California, Berkley (2005).
11. Cui, S., Madan, R., Goldsmith, A., Lall, S., Joint routing, Mac, and link layer optimization in sensor networks with energy constraints, *Proc. of IEEE ICC 2005*, Vol. 2, (2005) 725–729.
12. Pantazis, N. A., Vergados, D. J., Vergados, D. D., Douligeris, C., Energy Efficiency in Wireless Sensor Networks using Sleep Mode TDMA Scheduling, *Ad Hoc Networks* 7(2), (2009) 322–343.
13. Arantes Jr., M., França, F. M. G., Martinhon, C. A., Randomized Generation of Acyclic Orientations upon Anonymous Distributed Systems, *J. of Par. and Distr. Comp.* 69(3) (2009) 239–246.
14. Spiegel, M. R., *Mathematical Handbook of Formulas and Tables*, Makron, MacGraw-Hill, (1992).