Autonomic Duty Cycling for Target Tracking in a Bio-Inspired Wireless Sensor Network

Camila H. S. Oliveira and Miguel F. de Castro
Computer Science Dept. / GREat Research Lab / Federal University of Ceará
Email: {camila, miguel}@great.ufc.br

***Abstract*—The evolution of new miniatured devices, increasingly cheaper and more efficient, is enabling the use of Wireless Sensor Networks (WSN) for different application scenarios, such as the Internet of Things. However, even when nodes are deployed with a high degree of redundancy, common in target tracking applications, energy management is still a challenge for research. In order to optimize energy consumption in this dense network scenario, duty cycling mechanisms arise as an efficient solution to improve and maximize the WSN lifetime. In this paper, a new autonomic duty cycling mechanism called BioSched is proposed, aiming at simplicity, robustness and scalability. The mechanism is implemented over Biologically-inspired Optimization for Sensor network Lifetime (BiO4SeL), an autonomic routing protocol based on Ant Colony heuristics designed to optimize WSN lifetime. Based on the routes found by the ants in BiO4SeL, BioSched puts a subset of nodes with less residual energy to sleep while ensuring good delivery rate. Results show that the proposed duty scheduling enhances BiO4SeL in network energy saving and lifetime, and also outruns a related duty cycling algorithm called EC-CKN (Energy Consumed uniformly-Connected K-Neighborhood).**

***Keywords*—Duty Cycling, WSNs, Tracking.**

I. Introduction

WSNs are networks typically made up of thousands of sensor devices able to monitor several types of phenomena and transmit relevant harvested information to one or more base stations. There are numerous application fields to WSN, such as healthcare, security, agriculture, military, smart homes, etc. One of the classic applications of WSN is target tracking, in which the main objective is to search and track a given target within a given area of interest, possibly remote or hostile [1]. Wild life monitoring is an example of its possible scenarios. In general, this application requires sensors closest to the target to keep sending data to the sink, while other nodes sense for a possible target movement, not necessarily sending data to the sink.

For this specific application, the first challenge is to have an efficient routing algorithm in order to enable data harvesting at the sink node. This algorithm must be autonomic and robust, since nodes can be deployed in a field where human intervention is unviable. For the same reason, the routing algorithm must be energy efficient, as nodes' batteries replacement can also be unviable. Intending to overcome these challenges, Castro *et al.* [2] proposed BiO4SeL. BiO4SeL is an autonomic protocol that performs route discovery and maintenance using ants in order to optimize network lifetime. In BiO4SeL, the ants work in a distributed and autonomic way in order to find paths between source and destination. The best paths – not necessarily the shorter ones – are traced through by the ants depositing pheromone at each node in the path, just like biological ants. The pheromone mechanism used by the ants makes it easy to adapt to topology changes and eliminates the need of localization and global information. BiO4SeL uses the residual energy data on a given node in order to deposit and to evaporate the pheromone. The less energy a hop has, the faster its pheromone will dissipate. This will lead to a lower probability of this node to be chosen as a packet relay. Using this parameter along with the probabilistic factor involved on the route decision, the protocol distributes energy consumption among all nodes in the network while optimizing the network lifetime through the decrease of energy variance between the nodes.

BiO4SeL was designed considering a traffic behavior compatible with target tracking application, *i.e.*, one node transmits information to the sink at a given moment (typically the one closest to the target). As shown in [2], BiO4SeL presents good performance when compared to other related protocols: Ad-Hoc On-Demand Distance Vector (AODV) [3], Ant-based Routing Algorithm for Manets (ARAMA) [4], and Energy Aware Routing for Low Energy Ad Hoc Sensor Networks (EAR) [5], and meets its main objective of optimizing the network's lifetime, among others.

Another possible approach that can be associated with BiO4SeL in order to improve network lifetime is to deploy more nodes than necessary to cover the surveilled area (deploy dense network), relying on a duty cycle for these nodes. The work described in this paper takes advantage of the pheromone-based routing scheme provided by BiO4SeL in order to propose a new duty cycling scheme called BioSched. It was designed with two main objectives: (1) to overcome the energy consumption distribution problem of BiO4SeL, and (2) to adapt BiO4SeL to optimize network lifetime when operating in dense networks.

The rest of this paper is organized as follows. In Section II, some related works are discussed. Section III presents an overview of the BiO4SeL protocol. Section IV provides a detailed description of the proposed scheduling mechanism. Section V shows the experiments based on simulations and their numerical results, and Section VI concludes this paper.

 ISBN: 978-1-61208-399-5

II. Related Works

There are different approaches in literature aiming at optimizing network lifetime, such as topology control [6] [7], data aggregation/fusion [8] [9] and routing [10] [11]. In dense networks, node duty cycling is a solution that shows good results. The main idea is to take advantage of the large (more than necessary for full coverage) number of nodes deployed in the environment and coordinate the node operation mode, making them relay their duty cycle in order to save energy. However, the decision about which nodes should change their operation mode needs to take into account network connectivity and coverage requirements [12]. Recently, some techniques have been used to perform node activity scheduling. Some of them deal with the network connectivity problem, others aim at the coverage problem and only very few of them deal with both requirements. Most works that try to ensure 100% coverage and/or connectivity, required for target tracking applications, employ complex methods or have strong assumptions such as network global knowledge or localization awareness.

In [13], a solution based on a graph theory technique called minimum dominating sets was employed to eliminate nodes redundancy and leave in active operation mode a minimum number of nodes needed to ensure network coverage. In theory, these sophisticated methods achieves good results. However, in real WSN, the cost to compute those solutions can be unaffordable. Besides the complexity, the work that uses that kind of technique is in most cases centralized and based on location information. Others examples using complex solutions can be seen in [14] [15].

In [16] [17] [18], some solutions based on routing are presented. In general, the scheduling mechanism integrated in the routing procedure are computationally simpler and do not overload the network. However, it is hard to find works that operate in a simple, autonomous and distributed way. For example, in [17], the nodes have no autonomy to decide when to change their operation mode, and the solution is centralized in a coordinator node.

In [19], authors propose a distributed scheduling algorithm in which nodes are supposed to collaborate to keep a minimum amount of nodes active in order to maintain k-coverage, where k is a parameter. However, the algorithm is based on relatively heavy processing, and it could become an issue when scenario is dynamic.

Another approach by Yuan *et al.* [20], named EC-CKN, selects nodes to sleep based on remaining energy and connectivity, which is a similar approach to ours. The duty cycling is performed in fixed-sized rounds. In each round, the nodes communicate with each others in order to decide which ones will switch to sleep mode. Hence, the decision is not taken by each node autonomously, which can compromise the adaptability in dynamic scenarios.

In order to try to solve the aforementioned deficiencies, this paper proposes a duty cycling mechanism called BioSched based on the routing process provided by the BiO4SeL protocol. The mechanism proposed works in an autonomous and distributed way, such as BiO4SeL, and gives autonomy to nodes to decide when to change its operation mode using local information only (already acquired by BiO4SeL from neighbor nodes), thus providing a simpler solution. In order to evaluate the relative performance of our approach, we compare it with regular BiO4SeL and also with EC-CKN [20].

III. BiO4SeL Protocol

BiO4SeL is a protocol that incorporates the autonomous and distributed behavior of ants and employs a heuristic solution based on the residual energy of nodes to control the pheromone deposition along the paths. The protocol assumes that the nodes do not have prior network knowledge, not even the base station location. It uses ants to set up multiple paths between source and base station, and implements an inverse probabilistic routing table in each node in order to calculate the next hop in the path on the basis of just the node residual energy and the amount of pheromone cumulated in the hop. More details on BiO4SeL can be found in [2].

BiO4SeL operation comprises three phases: Bootstrap, Initial Route Discovery and Data Forwarding. At the Bootstrap phase, each node sends an *iHello* message by broadcast to its neighbors, which is used to initialize the node route tables with neighbors information, including energy.

With all nodes aware of their neighborhood, the protocol initiate the Initial Route Discovery phase. At this phase, the path discovery is done. At the end of this phase, there are paths established between all nodes and the base station.

After that, the Data Forwarding phase takes place. In this phase, data is forwarded from the source node to the base station and announcement ants (*hello*) are broadcasted at regular time interval in order to update the neighborhood with the node status. When a node receives a *hello* ant from a neighbor, the node updates the neighbor residual energy and the expiry time to the neighbor is restarted. At the beginning of the Data Forwarding phase, the best path is the shortest one. However, over time, the procedure of pheromone deposition and evaporation changes the priority to paths where the nodes have more residual energy. In order to encourage exploration of new paths, a probability (*pE*) is used and has its value calculated on the basis of nodes's residual energy.

However, this pheromone update process can concentrate pheromones in a few nodes preventing the choice of different routes. In order to avoid that and to encourage better distribution of energy consumption, BiO4SeL also implements an evaporation procedure. Evaporation takes place when the same hop is chosen several times. As the mentioned solution to decrease the concentration of energy waste takes some time to be effective, the BiO4SeL shows that energy waste still concentrates in the nodes surrounding the shortest paths. Aiming to solve this problem and to increase the network lifetime, the BioSched v1 and v2 presented in the next section show a method to save energy in the nodes most frequently used and in the nodes that waste all their batteries by sending signaling messages.

 ISBN: 978-1-61208-399-5

IV. THE BIOSCHED ALGORITHM

Scheduling node activity results in unquestionable power saving. However, choosing which nodes must be switched to sleep mode may be a critical issue, since it can impact essential requirements for network operation, such as connectivity and coverage, which are crucial for target tracking applications. Aiming to give strict coverage and connectivity guarantees, some works in the literature present their solutions based on restrictive assumptions, what makes their activity impracticable in real life WSN. According to [21], all scheduling mechanisms must provide three basic features: simplicity, scalability and robustness. With these features, a solution can save energy without overloading the network.

In order to fulfill the requirements mentioned above, this paper presents BioSched. The main difference between BioSched and the mechanisms found in the literature is main objectives. BioSched is not interested in switching the maximum possible number of nodes to sleep mode. Instead, it is designed to save energy from the most requested nodes and switching them to sleep mode. Thus, the neighboring nodes can be included in routing and the energy consumption is balanced.

The BioSched proposal includes two versions: the BioSched v1 and BioSched v2. The first one implements a heuristic method that takes into account energy and the workload of each node. In this case, the nodes with a lower battery and higher workload tend to be put to sleep. The second one, in addition to the nodes with high workload, also put to sleep the nodes that have never been required to forward packets. The main feature of BioSched is that the node itself decides whether to be switched to sleep node or not, ensuring that the mechanism is autonomous and simple.

A. BioSched v1

The BioSched v1 was fully integrated into BiO4SeL in order to take advantage of the protocol and to avoid network overload. In order to perform the scheduling activity, BioSched defines three possible states of a node: *sleeping*, *active* and *thinking*. When the nodes are in the *active* and *thinking* states, they are able to perform sensoring and transmission functions. In the *sleeping* state, nodes do not perform any networking function. The state changes are controlled by their own nodes based on their workload. Besides, all information used to make decisions is based on the node local information gathered from BiO4SeL messages.

At the beginning of network operation, all nodes are in the *active* state. When BiO4SeL comes into its second phase, where the routing process starts, the nodes begin to observe their own workload and, based on that, they decide whether to change their state or not. When in the *active* state, if a node decides to change to another state, it only switches to *thinking* state. Once the node is in the *thinking* state, it has two options: go back to *active* state or go to *sleeping* state. This decision is based on the node's workload and on the estimation of how this change can affect network connectivity. The state diagram with the change conditions is shown in Figure 1.

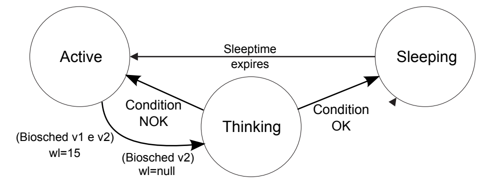


Fig. 1. BioSched state diagram.

The node workload (*wl*) is defined as frequency in that the node is chosen to forward a data packet. Until the source starts sending data to the destination, all nodes have their *wl* set to 0. When the nodes are chosen to transmit data packets, each time it is chosen its *wl* is increased in one unit. The *wl* is the variable observed by the node to trigger the node change of state to *thinking* mode. The state change is triggered off when the *wl* reaches the threshold defined as 15. The *wl* threshold and all the other values set to the variables used in this work are empirically defined by simulation experiments.

Once the node is in the *thinking* state, it has to decide whether it goes to *sleeping* state or to *active* state. The decision is made using only information received from neighbors through the BiO4SeL messages. The node is able to go to sleep when it obeys the set of conditions defined to estimate if the node's state change results in connectivity and coverage problems or not. The following conditions have to be fulfilled to allow nodes to switch to sleep mode: (1) At least ten neighbors in *active* state (*MinNb*); (2) At least one among these ten neighbor nodes fulfills the following conditions: (a) Not being source, destination or previous node in the path; (b) Being closer to base station than the previous node in the path; (c) Being the neighbor of the previous node in the path.

The previous node in this context is the node that sent the last data packet to the node that is in the *thinking* state ($node_{thinking}$). When the $node_{thinking}$ receives the data packet directly from the source, the source is also the previous node. After verifying the conditions, if the $node_{thinking}$ decides to go to *sleeping* state, the node has to calculate how much time it will stay there. After calculating the sleep time, the node's workload (*wl*) is reset and a new count will be started when the node comes back to active state. When all these steps are followed, the node broadcasts an update message to its neighbors and goes to *sleeping* state. At the moment the node wakes up, it goes to the *active* state and does not send any update message. The state of the nodes is updated through the *Hello* message used by BiO4SeL to keep neighbor tables updated. The node calculates the sleep time according to the following equation:

$$T = (\frac{avg\_en}{energy}) \times sp \quad (1)$$

where *T* is the sleep time, *avg_en* is the energy average of all node neighbors, *energy* is the current node energy and *sp* is a constant used as a factor to control the sleep time assigned to the nodes. As the objective of BioSched is to increase the network lifetime, the lower the node's energy compared to its

TABLE I
COVERAGE ENSURED WITH DIFFERENT NEIGHBOR QUANTITY [22].

# of neighbors	% of coverage	# of neighbors	% of coverage
5	91.62%	9	98.85%
7	96.89%	11	99.57%

neighbors' energy, the greater its sleep time. Therefore, the sleep time is directly proportional to the ratio between the the energy average of all node neighbors and the current node's energy.

The constant *sp* was set to 20s after several simulations varying this value between 1s and 30s. The results showed that when simulation set *sp* to 1s, the network get overhead due to the large amount of update message. Thus, a too short *sp* has as a main consequence packets loss and energy waste. On the other hand, when the *sp* is set to 30s, the distribution of energy waste between the nodes is compromised, which may result in energy depletion of just one node, decreasing significantly the network lifetime. This happens when a node is in the *thinking* state, sometimes all node neighbors that fulfill the requirements described above are sleeping. As its neighbors may sleep for a long time, the node energy can run out before it has the opportunity to change its state. The best results were obtained setting *sp* to 20s. This simulation shows that an average time is better because it is enough to save node energy without concentrating energy consumption in the active nodes.

In the case where the node is in the *thinking* state is not able to go to the *sleeping* state, it just comes back to *active* state while keeping its workload. From the first time on, every time the node's workload increases, it goes to *thinking* state to try the chance of going to *sleeping* state.

BioSched v1 ensures network coverage according to data found in [22], where an analytical model was defined and employed to determine the redundant node amount according to a percentage required by the application. Table I presented in [22] defines that if a node has more than 4 neighbors, then it has more than 90% of its range of transmission covered by its neighbors. As the conditions in BioSched is limited to 10 (the value of *MinNb*) it guarantees 98.86% of coverage. However, connectivity is not 100% guaranteed because there is the possibility of losing the state update message. However, results show that BioSched v1 can ensure a high delivery rate.

B. BioSched v2

In BioSched v2, as well as in BioSched v1, the nodes also have the autonomy to decide when to change their states. In this extension, the nodes that have a nill workload also have the opportunity of changing their state. As in BioSched v1 only the overused nodes can go to a sleep state, most nodes in the network remain active while wasting energy to send and receive *hello* messages. BioSched v2 was proposed in order to save energy, by also preserving the less used nodes in routing process.

All states, constant and messages defined in BiO4SeL v2 are also valid in BiO4SeL v3. Indeed, BiO4SeL v3 uses the message *Hello* in order to implement its extension. In this case, each time before sending a *Hello*, a node changes to the *thinking* state and checks if it has its *contPktDados* variable equal zero and at least ten active neighbors. Then, if the two conditions are met, instead of sending a *Hello*, the node sends a *UpdateStatus* message and goes to *sleeping* state. Otherwise, it sends a *Hello* and comes back to the *active* state. The procedure followed to calculate the sleeptime and wake up the nodes are the same used in BiO4SeL v2.

V. SIMULATIONS AND RESULTS

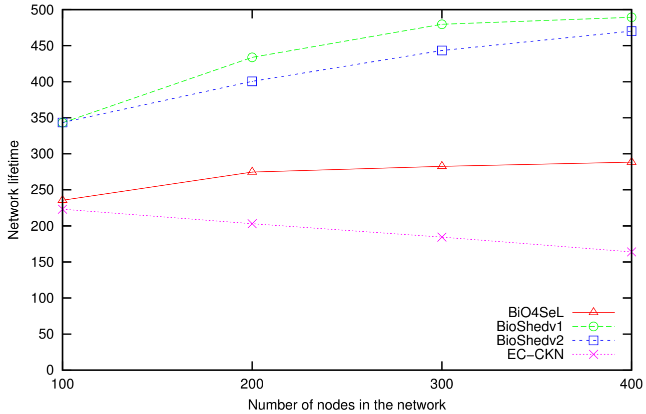
This section describes the experiments carried out using Network Simulator NS-2. As the objective of this work is to perform activity scheduling in dense WSN, the scenarios simulated vary in the number of nodes and in network density. The definition of density employed in this work was found in [23] and claims that a network is dense when a node has more than 10 neighbors. The density level varies from little dense to very dense. As simulations are interested in analyzing the mechanism according to network density, the scenario size was fixed in 50mx50m. Table II shows the simulation parameters and their values, where the density is given by the mean amount of neighbors per node.

TABLE II
SIMULATED SCENARIOS

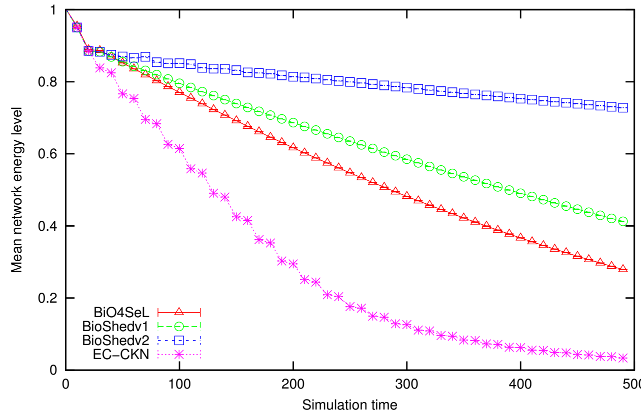
# of Nodes	Density	# of Nodes	Density
100	20	300	60
200	40	400	80

Our solution was implemented using C++ and performed using IEEE 802.15.4 as underlying PHY and MAC layers, with most of the parameters set to default values. We used IEEE 802.15.4 because it is the standard protocol for simulation of WSN in NS-2. The simulation considers a homogeneous and static network with one source and one destination. The positions of the source and destination have been previously assigned. Assuming the scenario size of $50m^2$ fixed to simulation, the source was placed at position (5,25) and the destination at (45,25). Related to the tests, for each number of node shown in Table II, 30 different scenarios were generated. For each scenario, the simulation was repeated 30 times, changing the randomization seeds.

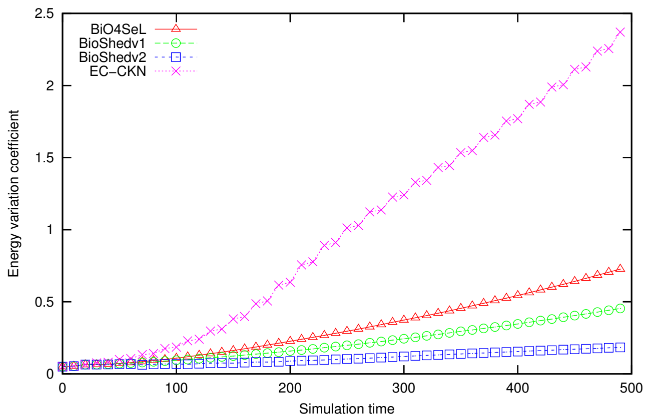
In [2], the total simulation time was set to 200 time units. However, in order to observe energy consumption behavior and the increase in the network lifetime, in this work, the simulation time was increased to 500 time units. The rate of data generation was kept unchanged from [2], sending one 76 KB packet each 0.4 time units. The results presented in this section include a confidence interval considering α = 0.95 and they aim to show the result of an autonomic energy management performed by BioSched v1 and v2, comparing results with original BiO4SeL and EC-CKN. These experiments are shown in the next subsections.



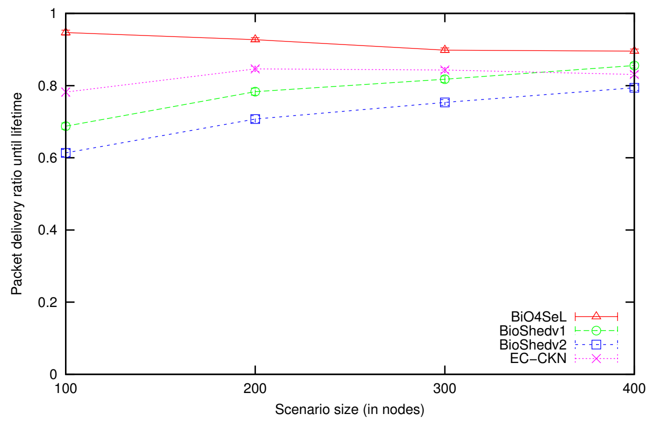
(a) Network lifetime.



(b) Mean network energy level for 400 nodes.



(c) Energy coefficient of variation for 400 nodes



(d) Packet delivery ratio before lifetime

Fig. 2. Simulation results.

A. *Network lifetime*

This experiment evaluated network lifetime, defined as the time in which the first node runs out its battery. Figure 2(a) shows that BioSched v1 and v2 provide an improvement in the network lifetime when compared to basic BiO4SeL and EC-CKN. Furthermore, the experiment suggests that the improvement in lifetime increases as the network density increases.

B. *Network mean battery level*

This experiment evaluated energy consumption throughout the simulation time with a scenario size of 400 nodes. As shown in Figure 2(b), at the beginning, until 30s, the mean energy consumption is the same for all protocols. That is explained by the fact that BioSched v1 and v2 are implemented over BiO4SeL, and EC-CKN also requires an initial stabilizing period. Thus, as the main modifications in BiO4SeL to integrate BioSched v1 and v2 were made in the Data Forwarding phase, the protocols have had the same initial phase behavior. As BioSched v2 puts a larger number of nodes to sleep, it substantially increases its mean battery level when compared to BioSched v1. EC-CKN produced the worst results, as it requires regular negotiation based on signaling messages, which consumes considerable energy. As in BioSched the node is autonomous to take its own decision, the energy expended in signaling is saved. Similar results ware observed with scenario sizes 100, 200 and 300 nodes.

C. *Network energy coefficient of variation*

In this experiment, the Coefficient of Variation (CoV) was used to evaluate how much, on average, node energy differs from the mean network energy level. The aim of this experiment was to show how energy load balancing is performed by BioSched. A lower value for the CoV means that the energy in the network is consumed more evenly among nodes. Figure 2(c) shows the CoV for scenario size of 400 nodes.

This figure shows that BioSched v1 and v2 improved battery consumption by balancing it among nodes, reducing the CoV. The same behavior occurred with other scenario sizes. This happened because even if BiO4SeL did not use all nodes in routing, it still used up the battery sending and receiving *Hello* messages, as the nodes do not sleep. Therefore, as the battery average was smaller and the energy consumption was more concentrated, the energy coefficient of variation tends to increase. As EC-CKN does not have the same routing balancing concern, the shortest path is used to route data packets and the energy consumption by this route is bigger, what increases CoV.

An important fact is that even if BioSched v2 did not perform the best battery consumption distribution, it presented the smaller CoV. This was a consequence of putting most nodes to sleep, thus, saving energy and keeping a higher battery average.

D. Packet delivery ratio until lifetime

The objective of this experiment was to evaluate BioSched v1 and v2 impact in the packet delivery ratio, *i.e.*, the ratio between the amount of packets received by the destination and sent by the source, in the interval $[0; 1]$, when compared to basic BiO4SeL and EC-CKN.

Figure 2(d) shows that, in a very dense scenario, BiO4SeL decreases a little of its delivery ratio. BioSched v1 and v2, in their turn, increase their delivery ratio as the network density rises. That happens because as network density increases, the possibility of BioSched having connectivity problem decreases. As mentioned above, BioSched does not ensure 100% network connectivity, but the results show that, in the very dense network scenario, BioSched tends to work better than BiO4SeL to guarantee a higher delivery ratio. EC-CKN, in its turn, presents a very good delivery ratio before lifetime. However, as its lifetime is the smallest among all protocols, it evinces that the heavy use of the shortest path enhances delivery rate, but this is not a longstanding situation.

VI. Conclusions and Future Works

This paper proposed an autonomic mechanism for BiO4SeL protocol in order to enhance power optimization in dense WSN by means of activity cycling. Aiming to achieve the objective of energy saving, this work developed a simple, autonomous and distributed proposal called BioSched. In this algorithm, the decisions are based only on the node workload and on locally stored neighbor information. The algorithm performs node activity scheduling without generating overheads (other than the overheads already produced by the bio-inspired routing algorithm) in order to improve network lifetime. The tests were based on a traffic scenario compatible with target tracking application, where only one node produces data to be delivered to the sink at a given time.

The first contribution of BioSched v1 and v2 is their ability to perform node activity scheduling in a distributed and autonomous way. This feature, combined with the routing optimization provided by BiO4SeL, outperformed the basic BiO4SeL and also a related work called EC-CKN. Other characteristics of BioSched v1 and v2 are their simplicity, robustness and scalability. As the proposed mechanisms were fully integrated into BiO4SeL, they inherit its characteristics of robustness and scalability. In addition, the simplicity of the mechanisms consists in the way the nodes are chosen to switch to sleep mode. The fact that the nodes do not require information from all network nodes, and each node bases its decision only on its own workload, characterizes the simplicity of the proposed protocols. As a result of these two contributions, results obtained by simulations show that BioSched v1 and v2 can significantly increase the lifetime of the network and improve the distribution of energy consumption when compared to BiO4SeL and EC-CKN.

As future works, we intend to test these algorithms in more heterogeneous scenarios, where power storage, processing and transmission range are differently distributed among nodes in the network. We will also consider evaluating the impact of the introduction of some power harvesting nodes in the network.

References

[1] K. Ramya, K. P. Kumar, and V. S. Rao, “A survey on target tracking techniques in wireless sensor networks,” vol. 3, no. 4, Aug 2012, pp. 93–108.

[2] M. F. De Castro, L. B. Ribeiro, and C. H. S. Oliveira, “An autonomic bio-inspired algorithm for wireless sensor network self-organization and efficient routing,” *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 2003–15, Nov. 2012.

[3] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” Jul 2003, IETF RFC 3561.

[4] O. Hussein, T. Saadawi, and M. J. Lee, “Probability routing algorithm for mobile ad hoc networks’ resources management,” vol. 23, no. 12. Washington, DC, USA: IEEE Computer Society, 2005, pp. 2248–2259.

[5] R. Shah and J. Rabaey, “Energy aware routing for low energy ad hoc sensor networks,” in *IEEE Wireless Communications and Networking Conference (WCNC2002)*, vol. 1, Mar 2002, pp. 350 – 355 vol.1.

[6] J. Yu, E. Noel, and K. Tang, “Degree constrained topology control for very dense wireless sensor networks,” in *GLOBECOM 2010*, Dec 2010, pp. 1 –6.

[7] Z. Luqiao and Z. Qinxin, “Interference and energy aware topology control,” in *Int. Conf. on Consumer Electronics, Communications and Networks (CECNet 2011)*, Apr 2011, pp. 1357 –1359.

[8] A. Avokh and G. Mirjalily, “Dynamic balanced spanning tree (dbst) for data aggregation in wireless sensor networks,” in *Telecommunications (IST), 2010 5th International Symposium on*, Dec 2010, pp. 391 –396.

[9] L.-Y. Sun, W. Cai, and X.-X. Huang, “Data aggregation scheme using neural networks in wireless sensor networks,” in *Intl. Conf. on Future Computer and Communication (ICFCC 2010)*, vol. 1, May 2010, pp. V1–725 –V1–729.

[10] W. Guo, W. Zhang, and G. Lu, “A comprehensive routing protocol in wireless sensor network based on ant colony algorithm,” in *2nd. Intl. Conf. on Networks Security, Wireless Communications and Trusted Computing - Vol. 01*, ser. NSWCTC ’10, 2010, pp. 41–44.

[11] M. Paone, A. Cucinotta, A. L. Minnolo, L. Paladina, A. Puliafito, and A. Zaia, “A bio-inspired distributed routing protocol for wireless sensor networks: Performance evaluation,” in *IEEE 30th Intl. Conf. on Distributed Computing Systems Workshops*, ser. ICDCSW ’10, 2010, pp. 247–255.

[12] S. Mahfoudh and P. Minet, “Survey of energy efficient strategies in wireless ad hoc and sensor networks,” in *Networking, 2008. ICN 2008. Seventh Intl. Conf. on*, Apr 2008, pp. 1 –7.

[13] B. Pazand and A. Datta, “An energy-efficient node-scheduling scheme for wireless sensor networks based on minimum dominating sets,” vol. 19. New York, NY, USA: John Wiley & Sons, Inc., Feb 2009, pp. 75–99.

[14] J. Chen, J. Jia, Y. Wen, D. Zhao, and J. Liu, “Modeling and extending lifetime of wireless sensor networks using genetic algorithm,” in *1st. ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, ser. GEC ’09, 2009, pp. 47–54.

[15] Y. Lin, X.-m. Hu, and J. Zhang, “An ant-colony-system-based activity scheduling method for the lifetime maximization of heterogeneous wireless sensor networks,” in *12th Annual Conf. on Genetic and Evolutionary Computation*, ser. GECCO ’10, 2010, pp. 23–30.

[16] V. Vaidehi, U. Selvan, J. Jayendran, and K. Praveen, “Redundant node deactivation by scheduling in wireless sensor networks,” in *Recent Trends in Information Technology (ICRTIT), 2011 Intl. Conf. on*, Jun 2011, pp. 613 –617.

[17] R. Saravanakumar, S. Susila, and J. Raja, "An energy efficient cluster based node scheduling protocol for wireless sensor networks," in *Solid-State and Integrated Circuit Technology (ICSICT), 2010 10th IEEE Intl. Conf. on*, Nov 2010, pp. 2053 –2057.

[18] A. Swain, R. Hansdah, and V. Chouhan, "An energy aware routing protocol with sleep scheduling for wireless sensor networks," in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE Intl. Conf. on*, Apr 2010, pp. 933 –940.

[19] E. Bulut and I. Korpeoglu, "Sleep scheduling with expected common coverage in wireless sensor networks," vol. 17, no. 1. Secaucus, NJ, USA: Springer-Verlag New York, Inc., Jan. 2011, pp. 19–40.

[20] Z. Yuan, L. Wang, L. Shu, T. Hara, and Z. Qin, "A balanced energy consumption sleep scheduling algorithm in wireless sensor networks," in *Proc. of 7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011)*, 2011, pp. 831–835.

[21] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," vol. 5. New York, NY, USA: ACM, Feb 2009, pp. 5:1–5:39.

[22] K. Wu, Y. Gao, F. Li, and Y. Xiao, "Lightweight deployment-aware scheduling for wireless sensor networks," vol. 10. Hingham, MA, USA: Kluwer Academic Publishers, Dec 2005, pp. 837–852.

[23] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. of the 1st Intl. Conf. on Embedded networked sensor systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 1–13.