Balancing Network Connectivity and the Life-Time of Sensors through Percolation and Consensus

Daehyun Ban and Michael Devetsikiotis
Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC 27695-7911
Email: {dban, mdevets}@ncsu.edu

***Abstract*—Due to replacement infeasibility, methods to extend the life-time of sensors have been an issue in Wireless Sensor Networks (WSNs) and these should consider network connectivity simultaneously. Controlling the sleep/awake of sensors is one simple way to reduce their energy consumption. However, this causes a network connectivity degradation by varying network connection topology. For this reason, we propose a simple and autonomous sensor sleep/awake method to achieve their balance.**

The size of clusters can be a metric to measure network connectivity in that a path exists among any cluster node. From percolation theory, we observe that a cluster size suffers a sharp transition based on edge connection patterns. This allows us to design a sensor sleep/awake algorithm which has an immense simplicity, but still requires global topology information. In many cases, sensors are not aware of the global topology. Further, managing the information becomes challenging under physical topology changes such as sensor add/drop. We show that the global knowledge requirement can be resolved by using a consensus algorithm. Through several graph tests, we show that our method achieves a network balancing between connectivity and life-time with preserving its simplicity. Also, the balancing is autonomous even under physical topology variations.

I. Introduction

As the tool of network construction, the primary purpose of wireless sensors is on the establishment of network connectivity. In contrast to other devices with an external energy support, they are generally operated by battery power which causes the periodic replacement of energy source. Moreover, their installed locations are uncoordinated in many network scenarios and this makes the replacement harder due to uncertain sensor locations. In this respect, we observe that methods to extend network life-time by the energy-saving of sensors receive attentions [1−8]. In the procedure, it is also required to consider network connectivity with them.

In Wireless Sensor Networks (WSNs), the processing capability of sensors has limitations. On the other hand, the suggested balancing methods between *connectivity* and *energy-saving* are becoming more intricate. From the observation of recent energy-saving schemes, the balance is appeared as a general trade-off between complexity and accuracy. For example, [1], [2] propose the power control of sensors which induces their connection topology to be a Minimum Spanning Tree (MST) and this is regarded as a minimal energy consumption topology while maintaining network connectivity. Either maximizing the second smallest eigenvalue [3] or minimizing the sum of effective resistance [4] on graph Laplacian is also in a same vein to find the MST. For sensors, these approaches require not only global topology information, but also to solve an optimization problem. The duty-cycle control of wireless sensors is another approach for energy-savings. [5] suggests the construction of energy-optimal tree by using an opportunistic sleep/awake control and [6] reformulates the duty-cycle problem into a shortest-path finding on dynamic graphs. However, for a sophisticated network management, these go through a reinforcement period with heavy computational burdens. We can additionally observe a multi-path routing for the balanced energy consumption of sensors [7] and the use of mobile sensors for data aggregation [8]. Still, these require either global topology information or a time synchronization among sensors. From these observations, we need to reconsider whether sensors can percept the global information and endure the computational complexity.

The necessity of *simple* and *efficient* energy-saving methods is not invariant to sensor deployment conditions. We frequently observe situations when the sensor deployment is uncoordinated (e.g., military operations which need a rapid infrastructure construction or data collection networks under extreme weather conditions). This restricts the use of above mentioned schemes in that sensors do not know the global topology information. Also, we need to concern the cost for purchasing sensors. Using non-expensive sensors limits to execute complex algorithms due to sensor processing capability. For these reasons, this paper proposes a simple and distributed energy-saving method which resolves above issues.

Percolation theory is widely used to determine clustering behaviors on graphs [9]. Especially, it reveals that the appearance of giant clusters has a phase transition behavior depending on edge connection patterns [10], [11]. We observe that this transition behavior can be related to an energy-saving algorithm if sensors know their global connection topology [12] and the restriction is partly relaxed by utilizing local edge degrees [13]. In contrast to them, we suggest a distributed method to obtain the global information by using a consensus algorithm and show its effectiveness in balancing network qualities between connectivity and energy-savings.

The remainder of this paper is as follows: In section 2, we explain network conditions and define two metrics to estimate network performances. Section 3 reviews the giant-cluster condition in percolation theory and consensus algorithms. Based on them, we propose a sensor sleep/awake control method for network balancing. Under several random graph scenarios, we test the proposed method and compare the results with analytical expectations in section 4. After that, we conclude this paper with a summary and discussion section.

II. Problem Formulation

A. Network Conditions and Assumptions

We consider a WSN with n static sensors and define a graph $\mathcal{G}$ which represents its connection topology. While constructing the network, the sensor deployment is uncoordinated. Therefore, the topology is unknown information for sensors.

Each sensor makes an independent sleep/awake decision to extend its life-time. In a discrete-time system, we set a parameter p which implies the sleep probability of sensors and consider that a sensor energy consumption is proportional to its awake period. Under these, the connection topology is time-varying and the topology graph $\mathcal{G}$ is dynamic (i.e., $\mathcal{G}(t)$).

Due to cost limitation, we assume non-expensive sensors whose functionalities are restricted. Computations are eligible only for light loads due to small memory space. Also, they do not have a power control (i.e., only have an on/off option) and a time synchronization scheme with other sensors.

In the circumstance, the sleep probability p is an important parameter which decides the network performance. We want to find a value which balances the network connectivity and the energy-consumption of sensors. Additionally, the value needs to be adjusted autonomously even if there occur physical topology changes such as sensor add/drop (e.g., additional sensor installations after the first deployment or imbalanced sensor die-outs due to unfair energy consumptions).

B. Performance Measurement Metrics

We use two metrics to estimate the network performance and these are based on packet transmission tests. Each sensor periodically generates a packet whose destination is uniform randomly selected among n–1 other sensors. For those packets, we consider the following metrics (Note. the only distinction is whether a delayed packet delivery is allowed or not):

- **Communicability**: Suppose that a packet generation time is t_k. For each packet, we observe whether there exists a path to its destination under the connection topology $\mathcal{G}(t_k)$. If the topology does not have a path, the packet is dropped. This metric measures the rate of successful path existence at each packet generation instance.
- **Latency**: In contrast to Communicability, a packet is not dropped under no path existence. Instead, it waits until the first path is established on the time-varying $\mathcal{G}(t)$ and we observe the delay for each packet. Latency measures the mean delay of packet transmissions.

Additionally, we estimate the average sleep probability of sensors during tests to measure the degree of energy-saving. When the value is $\bar{p}$, this implies that network sensors save their energy consumption with degree $\bar{p}$.

III. Controlling the Sleep/Awake of Sensors with Percolation Theory and Consensus Algorithm

We show procedures how the giant-cluster condition in percolation theory matches to the sleep/awake control in WSNs and point out the necessity of global topology information. By adding an average consensus algorithm on it, we propose a sensor sleep/awake method to balance network qualities between connectivity and energy-consumption.

A. Matching Giant-Cluster Condition to Sensor Sleep/Awake

Percolation theory investigates the behavior of clusters on graphs and one attractive result is the sharp transition of cluster size according to edge connection patterns [10], [11]:

Theorem 1: **(The Phase-Transition of Cluster-Size [10])** A graph $\mathcal{G}$ has a *giant-cluster* with high probability if each node has connections with at least two other nodes. When a random variable X follows the distribution of local edge degree on graph $\mathcal{G}$, the giant-cluster condition is given by:

$$\Psi(G) = \frac{E[X^2]}{E[X]} \geq 2 \tag{1}$$

The appearance of giant cluster in $G(t)$ improves connectivity as it guarantees a path existence among cluster sensors. In the network, suppose that a sensor makes a sleep decision at time t_k. This simply corresponds to the node removal on a time-varying connectivity graph $\mathcal{G}(t_k)$. For this reason, finding out the feasible fraction of random node removal on $\mathcal{G}$ which satisfies Theorem 1 is equivalent to determine the region of statistical sensor sleep probability p that prevents a severe connectivity loss in the network.

For the graph with p fraction node removals, [10], [11] show procedures to gain the first and second moments of edge degree distribution. When we set $E[X_0]$ and $E[X_0^2]$ as the two moments of $\mathcal{G}$, respectively, and combine them with (1), it gives the feasible range of sensor sleep probability, which conserves the giant-cluster condition, as follows:

$$p \leq 1 - \frac{E[X_0]}{E[X_0^2] - E[X_0]} \tag{2}$$

In WSNs, we observe that similar approaches are used in [12], [13]. However, still the global information (i.e., the first and second moments) is *unknown* for local sensors when their deployment is uncoordinated.

B. Consensus Algorithms

In multi-agent systems, consensus algorithms define methods to reach a value agreement among agents only with local interactions (see the survey [14] and references therein for details). Especially, the sleep/awake formulation (2) lets us focus on an average consensus. Even for a dynamic graph $\mathcal{G}(t)$, we observe that there exist algorithms that achieve a value convergence and choose such an algorithm from [15].

We modify the algorithm for non-weighted graphs. Suppose a graph $\mathcal{G}(V, E(t))$, where $|V| = n$ (i.e., n nodes or agents) and assume that a node i holds its local initial value $x_i(0)$ at $t = 0$, where $i \in V$. Then, each node proceeds the following distributed simple decisions:

$$\dot{x_i}(t) = \sum_{j \in \mathcal{N}_i(t)} (x_j(t) - x_i(t)) \tag{3}$$

where $i \in V$ and $\mathcal{N}_i(t)$ is a i's neighboring node set at time t. This leads an asymptotic value convergence given by:

$$x_1(t) = x_2(t) = \ldots = x_n(t) = \alpha \text{ (Average Consensus)}$$

We note that the sum of all node states (i.e., $\sum_i \dot{x}_i$) is invariant to time. This implies that the sum of initial values $\sum_{i \in V} x_i(0)$ is preserved regardless of time. For all agents, (3) always leads a direction where the value gap with neighbors decreases. These support that the value agreement α reaches $\frac{1}{n}\sum_{i\in V} x_i(0)$ (i.e., an average consensus) in steady-state.

C. Sleep/Awake Control Method for Performance Balancing

In our method suggestion, each sensor only requires four values and a simple computing capability. For sensor i, we set parameters x_{i1}, x_{i2}, x_{i1}^T and x_{i2}^T. First two parameters are used to decide the sleep/awake probability p and the other two are to reflect physical topology variations.

1) Initial Value Estimation: The essence of the average consensus is the equal distribution of the initial value sum $\sum_{i\in V} x_i(0)$ through linear interactions. Hence, an initial value estimation is important to reach a correct value agreement.

Each sensor is required to measure its local edge degree when all sensors are awake. Suppose that $\mathcal{G}_o$ is the connection topology. For sensors, estimating the local degree is not instant as they are utilizing an independent sleep/awake. Hence, they first have a waiting phase to measure an exact local degree.

As the sleep/awake decision is independent, the expected waiting time for a sensor wake up follows a geometric distribution in a discrete time system. For this reason, we set the waiting phase to $\lceil \frac{1}{1-\xi} \rceil$ slots, where ξ is reasonably close to 1. Each sensor uses this local measurement and its square value as $x_{i1}(0)$ and $x_{i2}(0)$, respectively, and joins the following consensus process.

2) Consensus Process and Sleep Probability Decision: Sensors proceed two consensus processes with local neighbors, given by:

$$\begin{aligned} \dot{x}_{i1}(t) &= \textstyle\sum_{j\in\mathcal{N}_i(t)} \beta(x_{j1}(t) - x_{i1}(t)) \\ \dot{x}_{i2}(t) &= \textstyle\sum_{j\in\mathcal{N}_i(t)} \beta(x_{j2}(t) - x_{i2}(t)) \end{aligned} \tag{4}$$

where $i, j \in V$ and $\mathcal{N}_i(t)$ is a neighbor set at time t. β is a small positive constant and affects the speed of convergence.

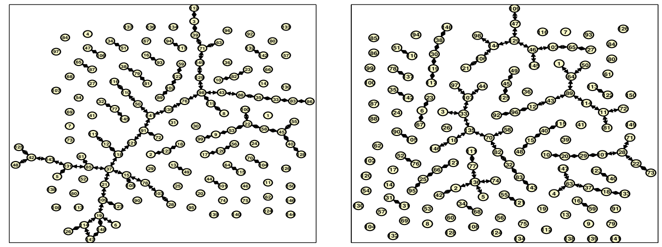
From [15], these processes are guaranteed to reach an average consensus even if the connection topology is dynamic. Hence, in steady-state, sensors have the global information $E[X_0]$ and $E[X_0^2]$ in (2). During the consensus processes, we utilize transient measurements $x_{i1}(t)$ and $x_{i2}(t)$ to set up a sensor sleep probability p as follows:

$$p_i(t) = 1 - \frac{x_{i1}(t)}{x_{i2}(t) - x_{i1}(t)} - \psi \tag{5}$$

We are aware that this sleep probability decision also will arrive at a consensus among sensors as the two independent processes guarantee a convergence. When $\psi = 0$, the sleep probability p locates on the phase-transition boundary in (1). For this reason, we set a small positive constant ψ to prevent a steep connectivity loss.

3) Sleep Probability Adjustment for Topology Variations: During a network operation, the life-time of sensors is uneven. Moreover, we can install additional sensors. These make the connection topology graph $\mathcal{G}$ sparser or denser. In these cases, sensors are required to find a newly balanced sleep probability p for the varied topology. By using the average consensus, an autonomous sleep probability adjustment is capable.

We assume that sensors have a unique ID when they are installed (e.g., i-Mote). This allows sensors to detect their local degree variation when there occurs an add/drop. Then, the issue is how effectively reflect the local topology change into the consensus process for global parameter estimations.



(a) $\Psi(G)=2.0482$ (b) $\Psi(G)=1.9200$

Fig. 1. We generate two random graphs whose Ψ is slightly over and less 2 with 150 nodes and observe a steep decrease on their maximum cluster size.

Once after a sensor participates in the consensus process (4), its parameters x_{i1} and x_{i2} keep varying to the convergence values $E[X_0]$ and $E[X_0^2]$, respectively. Hence, substituting the newly measured local degree and its square values into x_{i1} and x_{i2} during the process does not make correct convergence values for the varied topology.

The property of selected consensus algorithm is the equal distribution of total state sums $\sum_{i\in V} x_{i1}$ and $\sum_{i\in V} x_{i2}$. For this reason, fitting those values according to the topology variation is one way to reach a correct consensus. We suggest a simple recursion by using parameters x_{i1}^T and x_{i2}^T. When sensors join the consensus process, their parameters x_{i1} and x_{i2} are same as x_{i1}^T and x_{i2}^T, respectively. Then, suppose that a sensor i acknowledges that its degree is changed at t and the value is x_i^{new}. By using dummy parameters x_{i1}^T and x_{i2}^T, we consider the following recursions:

$$x_{i1}(t^+) = x_{i1}(t) + \delta_1, \quad x_{i2}(t^+) = x_{i2}(t) + \delta_2 \tag{6}$$

where $\delta_1 = x_i^{\text{new}} - x_{i1}^T$ and $\delta_2 = (x_i^{\text{new}})^2 - x_{i2}^T$. After the changes, the sensor puts x_i^{new} and its square value into x_{i1}^T and x_{i2}^T, respectively.

These recursions reflect the number of added (or subtracted) edges when a topology change occurs and lead the consensus algorithm to be aware of the correct total edge degree sum during operations.

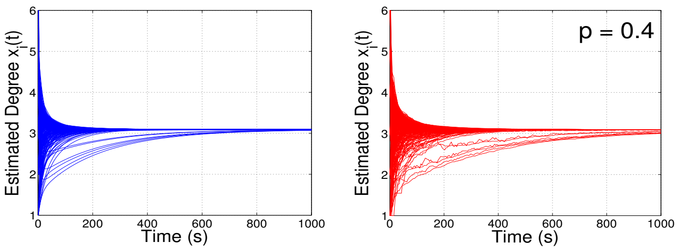
IV. Simulations on Random Graphs

We generate random graphs which have different Ψ and each graph represents a connection topology $\mathcal{G}_o$ when all sensors are awake. In a random graph generation with n nodes, an edge between nodes is established by probability e and we gain different Ψ graphs by varying e. First, we assure the validity of giant-cluster condition and consensus algorithm used in this paper. Then, the effect of the proposed method is tested under several network scenarios.

A. Cluster Size Variation and Average Consensus

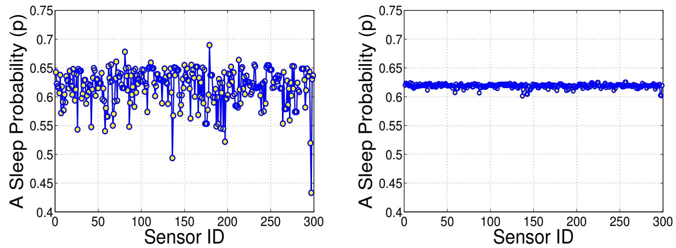
Our sensor sleep/awake method arises from the correctness of the giant-cluster condition and average consensus algorithm. For this reason, we provide simple results to recognize their validity.

1) Phase-transition on the cluster size: Theorem 1 implies that the cluster size on random graphs suffers a shape transition around $\Psi = 2$. In Fig. 1, we plot two random graphs with 150



(a) Convergence on $\mathcal{G}$ (b) Convergence under dynamic reduction

Fig. 2. Convergence test results: $x_i(0)$ is the local edge degree of $\mathcal{G}$ where $i \in V, |V| = 300$. We observe that (3) induces a correct convergence $x_i = 3.184$ under sensor sleep/awake situations only with a speed difference.



(a) Sensor sleep probability at $t = 10$ s (b) Sensor sleep probability at $t = 40$ s

Fig. 3. We observe the sleep probability of sensors at $t = 10$ and 40 s. They show the sleep probability agreement among sensors where $p = 0.619$.

sensors. By varying the edge construction probability e, their Ψ is controlled to be slightly over (2.0482) and less (1.92) than 2, respectively. We estimate their maximum cluster size and observe the occurrence of a steep size reduction (51$\rightarrow$14) around $\Psi = 2$. Except in the boundary case, we observe that the cluster size reduction is gradual with the Ψ decrease (similar results can be found in [16]).

2) Average consensus on dynamic graphs: We generate a connected graph $\mathcal{G}$ with 300 sensors whose graph measurement Ψ is 2.8. By setting their local edge degree as initial values, we test the convergence of algorithm (3). From the estimation of $\mathcal{G}$, the correct average of edge degree is 3.184. Fig. 2 shows the degree variation of sensors under the graph $\mathcal{G}$ and its dynamic node removal with $p = 0.4$ which corresponds to a sensor sleep/awake scenario, respectively. For both cases, sensors reach a convergence into the correct average and the only difference appears in the speed of convergence. This verifies that (3) can be utilized to find a balanced sleep probability p even under sensor sleep/awake scenarios.

B. The Proposed Sensor Sleep/Awake Method Tests

By using two metrics (*Communicability* and *Latency*), we test the performance of proposed sleep/awake method. These metrics are designed to measure network connectivity through packet transmissions. We are aware of the steep transition of cluster size and its occurrence is highly related to the sensor sleep/awake decision. This allows us to expect that the similar performance transitions will appear on communicability and latency measurements.

1) No Sensor Add/Drop Case: We generate a graph $\mathcal{G}$ with 300 sensors whose Ψ is 3.93 and assume that there is no sensor add/drop during tests. From the graph, we measure two local edge degree moments (i.e., $E[X_0]$ and $E[X_0^2]$) and those values are 3.12 and 12.27, respectively. Then, the expected sleep probability p of sensors is 0.619 when a convergence is reached by using (4) with $\psi = 0.04$.

First, we test whether our method leads the expected p. At initial, we set $p = 0.5$ for all sensors. Fig. 3 plots two snapshots for the sleep probability of 300 sensors. At $t = 10$ s, we observe a sleep probability disagreement among sensors. The difference diminishes by using (4) and (5). At $t = 40$ s, the sleep probability snapshot almost shows an agreement with the expected p. This can be a validation that two independent consensus processes in (4) also reach a correct convergence.

By measuring communicability and latency, we investigate the effectiveness of the controlled sleep probability. We assume that sensor generate a packet at every second with a uniform destination. In Fig. 4, we plot the two metric results with varying the sleep probability of sensors. For communicability, it generally decreases with the increase of sleep probability. However, one notable point is the decreasing degree of measurements. We see that communicability shows a steep decrease when the sleep probability of sensors is over the controlled p value. This nature is similarly observed in the latency measure with having a sharper transition behavior.

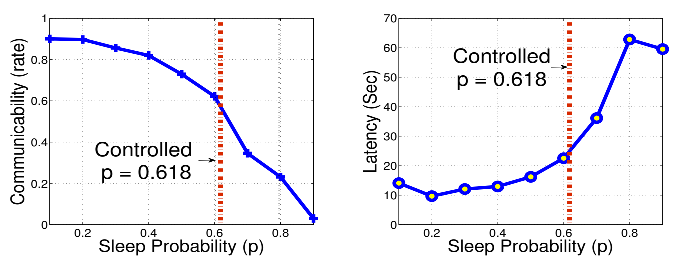
Under the assumption that the energy consumption of sensors is proportional to their awake period, finding out the controlled p has a certain contribution to balance the network performance. For example, from the latency measure Fig. 4(b), it allows sensors to save 60% energy consumption only with a slight delay increase in packet transmissions.

2) Sensor Add/Drop Case: While operating a WSN, the additional installation or uneven die-out of sensors can make changes on $\mathcal{G}$. We test whether the proposed method can autonomously adjust the sleep probability p under those cases.

For consistency, we start from the graph $\mathcal{G}$ in the previous. In there, sensors reach a sleep probability consensus at 40 s with $p = 0.619$. We uniform randomly drop and add 50 sensors on $\mathcal{G}$ at 40 s and define their graphs as $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. The followings are their measurements where $\mathcal{G}(\#$ of sensors, $\Psi, E[X_0], E[X_0^2],$ *Expected* $p)$:

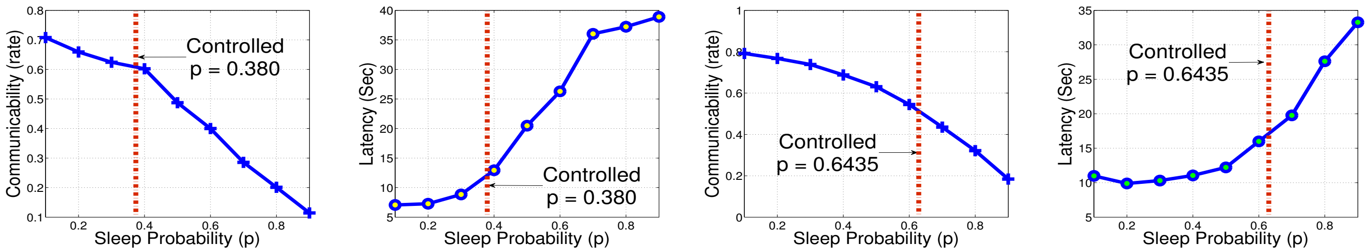
$\mathcal{G}_1(250, 3.06, 1.71, 4.61, 0.38)$, $\mathcal{G}_2(350, 6.19, 3.17, 12.88, 0.64)$

These sensor add (drop) makes $\mathcal{G}$ more dense (sparse) and we observe that the expected p is also increase (decrease). In Fig. 6, we plot the sleep probability of sensors at 80 s. For both



(a) Communicability (b) Latency

Fig. 4. Performance measures under no sensor add/drop: We plot communicability and latency with varying a sleep probability p and observe the steep performance degradation when sensors set their p over the controlled value.



(a) Communicability on $\mathcal{G}_1$ (b) Latency on $\mathcal{G}_1$ (c) Communicability on $\mathcal{G}_2$ (d) Latency on $\mathcal{G}_2$

Fig. 5. Performance measurements under sensor add/drop: $\mathcal{G}_1$ and $\mathcal{G}_2$ are connection graphs for a sensor add and drop case. From measuring communicability and latency with varying a sleep probability p, we observe that their performance suffers a steep degradation after passing the controlled sleep probability.

add and drop cases, we see that p reaches another agreement value that exactly corresponds to our analytical expectation. We note that sensors reflect the topology variation locally by using (6) and the value adjustment is achieved only with distributed local interactions.

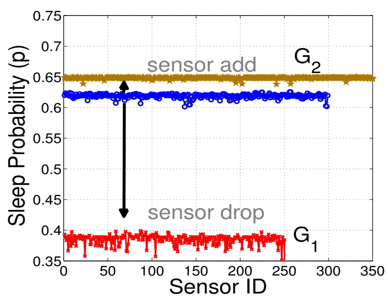


Fig. 6. The sleep probability of sensors for the sensor add case $\mathcal{G}_1$ and drop case $\mathcal{G}_2$ at $t = 80$ s (i.e., 40 s passed after the topology change).

In Fig. 5, we measure communicability and latency with varying the sleep probability p for $\mathcal{G}_1$ and $\mathcal{G}_2$. In there, we observe that the metric performance decreases with the increment of p. Especially, there still exists regions where steep performance decrements occur. We indicate the autonomously controlled sleep probability p, which is attained by using the proposed method. For both communicability and latency, those values locate just before the steep performance reduction.

In Fig. 5(a), the controlled p saves 38% energy consumptions with less than 10% communicability reduction. Also, Fig. 5(d) indicates that almost 65% energy can be saved only with a 5 s delay increase. It is extremely notable that utilizing a consensus algorithm gives an *autonomous* property to find out the balanced p even under topology variations. Also, our test results show that the duration to reach a sleep probability consensus is relatively short.

Our method has an additional advantage in the sense of *fair* energy consumption among sensors. Under the assumption that sensors consume an energy only when they are awake, our consensus approach prevents the imbalanced energy consumption of sensors, dealt in [7].

V. Conclusion

In WSNs, the life-time extension of sensors is required due to their replacement infeasibility and the method must be proceeded under the network connectivity assurance. Controlling the sleep/awake of sensors is one way to extend the life-time. However, it has a trade-off against network connectivity and is challenging to find out their balanced point.

By controlling the sleep/awake degree of sensors, this paper proposed a *simple* and *autonomous* method to balance network connectivity and the life-time of sensors. When global topology information is known, we explained that the giant-cluster condition in percolation theory can be used for sensor energy-saving methods. In addition, by utilizing a consensus algorithm, our method allows sensors to obtain the global information only with distributed interactions. We provided test results under several network topologies. Depending on sensor sleep probability, we observed the expected steep performance degradation region and showed that the proposed method properly achieves a balance between connectivity and energy-consumption. Our method is not only simple and autonomous, but also prevents imbalanced sensor energy consumptions.

References

[1] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized minimum-energy broadcasting in ad-hoc networks," in *INFOCOM*, vol. 3. IEEE, 2003, pp. 2210–2217.

[2] M. Segal and H. Shpungin, "On construction of minimum energy k-fault resistant topologies," *Ad Hoc Networks*, vol. 7, no. 2, 2009.

[3] J. Sun, S. Boyd, L. Xiao, and P. Diaconis, "The fastest mixing markov process on a graph and a connection to a maximum variance unfolding problem," *SIAM review*, vol. 48, no. 4, p. 681, 2006.

[4] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM review*, vol. 50, no. 1, p. 37, 2008.

[5] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 133–144.

[6] F. Wang and J. Liu, "Duty-cycle-aware broadcast in wireless sensor networks," in *INFOCOM*. IEEE, 2009, pp. 468–476.

[7] A. Mei and J. Stefa, "Routing in outer space: fair traffic load in multi-hop wireless networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2008.

[8] M. Bin Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2006, pp. 37–48.

[9] M. Penrose, *Random geometric graphs*. Oxford University Press, 2003.

[10] R. Cohen, K. Erez, D. Ben-Avraham, and S. Havlin, "Resilience of the internet to random breakdowns," *Physical Review Letters*, vol. 85, no. 21, pp. 4626–4628, 2000.

[11] D. Callaway, M. Newman, S. Strogatz, and D. Watts, "Network robustness and fragility: Percolation on random graphs," *Physical Review Letters*, vol. 85, no. 25, pp. 5468–5471, 2000.

[12] O. Dousse, P. Mannersalo, and P. Thiran, "Latency of wireless sensor networks with uncoordinated power saving mechanisms," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2004, pp. 109–120.

[13] Z. Kong and E. Yeh, "Distributed energy management algorithm for large-scale wireless sensor networks," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2007, pp. 209–218.

[14] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[15] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520–1533, 2004.

[16] R. Parshani, S. Buldyrev, and S. Havlin, "Critical effect of dependency groups on the function of networks," *Proceedings of the National Academy of Sciences*, vol. 108, no. 3, p. 1007, 2011.