

ON THE STOCHASTIC MODELING OF DESYNCHRONIZATION CONVERGENCE IN WIRELESS SENSOR NETWORKS

Dujdow Buranapanichkit

Prince of Songkla University
Electrical Engineering Dept.
Hat Yai, Songkla, 90112 Thailand

Nikos Deligiannis, Yiannis Andreopoulos

University College London
Electronic and Electrical Engineering Dept.
Torrington Place, London, WC1E 7JE, U.K.

ABSTRACT

Desynchronization is a fundamental approach in wireless sensor networks that allows for convergence to time-division multiple access (TDMA) of the medium without the need for clock synchronization and centralized coordination. The method is based on the concept of reactive listening of periodic fire message broadcasts between nodes sharing the given spectrum. We propose a novel framework to estimate the required iterations for convergence to fair TDMA scheduling. Unlike previous conjectures or bounds found in the literature, our estimation framework is based on a stochastic modeling approach. Experiments via imote2 TinyOS nodes and simulations demonstrate that the proposed estimates characterize the experimental desynchronization convergence iterations significantly better than existing conjectures or bounds.

Index Terms—wireless sensor networks, desynchronization, stochastic modeling, TDMA.

1. INTRODUCTION

Efficient usage of shared spectrum is key in wireless sensor networks (WSNs), where packet losses because of collisions translate to wasting battery resources [1–8]. Desynchronization is a primitive leading to fair time-division multiple access (TDMA) scheduling without requiring clock synchronization among sensors or a coordinating node [3, 4, 6, 7, 9–13]. The key concept behind desynchronization is *reactive listening*, according to which, nodes periodically broadcast fire messages and then update their next broadcast time based on the reception of fire messages from other nodes. After a number of iterations, where nodes adjust their firing times, a *steady-state* (SState) period is achieved. In SState, nodes have converged into fair TDMA scheduling and fire messages are sent in periodic intervals of T seconds, followed by data packets.

1.1. Relation to Prior Work

In all desynchronization methods for WSNs [3, 4, 12], the number of convergence iterations required until the steady

state plays a crucial role in latency, energy and bandwidth efficiency of WSN deployments. Moreover, deriving estimates for the convergence iterations forms a crucial step in the analytic understanding of how the system evolves from a random setup to the steady state [12].

Deriving *closed-form* estimates for the required convergence iterations to SState is hard [2, 4, 9, 12] due to the non-deterministic aspects of the desynchronization process, namely, the random initial condition of the phases and the random perturbations in the firing order of nodes due to noise. Hence, existing works focus on *order-of-convergence* [4] or *lower bounds of convergence iterations* of desynchronization, proven or conjectured via experimentation [3, 9, 10, 12]. Moreover, these works consider only the noise-free case.

While order-of-convergence estimates provide a coarse characterization of the convergence, they do not predict the expected number of iterations required for desynchronization to converge to the SState. On the other hand, the existing *lower bounds* on the desynchronization convergence iterations are currently given without a characterization on their tightness to real-world experiments or simulations.

Instead, following a *stochastic approach* yields an analytic understanding of the behavior of the convergence. Particularly, it leads to closed-form expectations for the iterations that should be a close match to experiments and simulations. Such an approach was only followed in the work of Leidenfrost and Elmenreich [2]. Their model, however, is applicable to *synchronization*, and the required differences (i.e., different phase update, reachback response and pre-emptive message staggering [2]) do not permit a direct mapping of their convergence estimates to desynchronization systems.

1.2. Contribution

In this paper, we address this issue proposing *stochastic* (instead of deterministic) estimates for the convergence iterations, focusing on the DESYNC primitive of Degesys *et al.* [4, 9]. Specifically, we propose a novel analytic estimate for the number of iterations until desynchronization is *expected to have converged* to SState within a predetermined threshold. We validate our results based on a real WSN deployment, as

This work was funded by EPSRC grant EP/K033166/1.

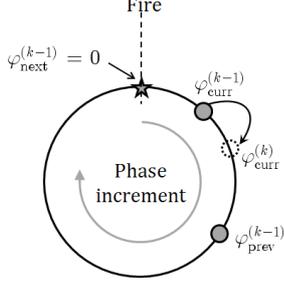


Fig. 1. The k th phase update of node “curr” when the next fire message takes place in DESYNC.

well as under a simulation environment, and demonstrate the superiority of the proposed stochastic estimates against the existing convergence bounds in the literature [3, 4, 10, 12].

2. REVIEW OF DESYNCHRONIZATION

Consider a WSN with W fully-meshed nodes. Each node performs a task with a period of T seconds [4]. Initially, each node joins the network by broadcasting an initial *fire message* at a random time between $[0, T]$ seconds. Upon the completion of its cycle (after T seconds), each node broadcasts a fire message anew. For each node, the percentage of the way through its cycle at a given time $t \in [0, T]$ is denoted as the node’s *phase*, $\varphi \in [0, 1)$ [2–4, 14–17]. A *firing cycle* comprises W consecutive fire messages, each stemming from an individual node in the WSN.

The fire messages’ phase values can be imagined as beads moving clockwise on a ring with period $T = 1$ s [4]. When a node phase reaches unity, a fire message is broadcast and the corresponding node’s phase is reset to zero. Fig. 1 illustrates three nodes that fire consecutively (*phase neighbors*¹). According to DESYNC [4, 9], the phase of node “curr” is updated towards the middle of the interval between the phases of fire messages from nodes “prev” and “next”, i.e.,

$$\varphi_{\text{curr}}^{(k)} = (1 - \alpha) \varphi_{\text{curr}}^{(k-1)} + \frac{\alpha}{2} (\varphi_{\text{prev}}^{(k-1)} + \varphi_{\text{next}}^{(k-1)}) \pmod{1}, \quad (1)$$

where $\alpha \in (0, 1)$ is the *phase-coupling constant* that controls the speed of the phase adaptation. This approach disperses all fire message broadcasts at intervals of $\frac{T}{W}$ within each periodic firing cycle. Thus, after k_{SSstate} iterations of (1), DESYNC leads to fair TDMA scheduling, where all fire messages are periodic and the phase update of (1) leads to convergence to SState, expressed by

$$\left| \varphi_{\text{curr}}^{(k_{\text{SSstate}})} - \varphi_{\text{curr}}^{(k_{\text{SSstate}}-1)} \right| \leq b_{\text{thres}}, \quad (2)$$

with b_{thres} the preset convergence threshold e.g. $b_{\text{thres}} = 0.001$.

¹The notion of phase neighbors indicates temporal adjacency of fire messages and is independent of the nodes’ identity and physical location.

Assuming negligible propagation delay and error-free detection of messages, it has been conjectured [3, 4] that convergence requires iterations of the order

$$k_{\text{DESYNC},[3][4]} \sim O\left(\frac{1}{\alpha} W^2 \ln \frac{1}{b_{\text{thres}}}\right). \quad (3)$$

This order estimate gives a coarse characterization for the convergence iterations, but it cannot provide the expected number of iterations until convergence to SState.

3. PROPOSED STOCHASTIC MODEL

Definition 1 (Phase Model): Every node’s initial phase (e.g., $\varphi_{\text{curr}}^{(0)}, \varphi_{\text{prev}}^{(0)}, \varphi_{\text{next}}^{(0)}$), is modeled by an independent random variable, $\Phi^{(0)}$, that is uniformly distributed in $[0, 1)$. Namely, $\forall \text{fire} \in \{\dots, \text{prev}, \text{curr}, \text{next}\} : \Phi_{\text{fire}}^{(0)} \sim P_{\Phi_{\text{fire}}^{(0)}}$, with²

$$P_{\Phi_{\text{fire}}^{(0)}} = \mathcal{U}\left(\mu_{\Phi_{\text{fire}}^{(0)}}, \sigma_{\Phi_{\text{fire}}^{(0)}}\right) \pmod{1}, \quad (4)$$

where $\mathcal{U}(\cdot)$ denotes the uniform probability density function (PDF), with mean μ_{Φ} and standard deviation σ_{Φ} . We define the mean times of successive phase updates to be equidistant, i.e., $\mu_{\Phi_{\text{prev}}^{(0)}} - \mu_{\Phi_{\text{curr}}^{(0)}} = \frac{1}{W}$, $\mu_{\Phi_{\text{curr}}^{(0)}} - \mu_{\Phi_{\text{next}}^{(0)}} = \frac{1}{W}$. In the beginning, all fire message broadcasts are completely uncoordinated (random), i.e., $\sigma_{\Phi^{(0)}} = \frac{1}{\sqrt{12}}$. ■

There is no loss of generality from the assumption of equidistant means as the modulo operator in (4) ensures that the PDF wraps around one such that $\Phi_{\text{curr}}^{(0)}$ is always uniformly distributed between $[0, 1)$ irrespective of the assumed mean values. However, we opt for the use of (4) as this facilitates the mathematical exposition of the proposed estimates.

Definition 2 (Measurement Noise): Each phase in (1) is assumed to be contaminated by additive white noise, i.e., $\Delta \sim \mathcal{U}(0, \sigma_{\Delta})$. ■

Due to the measurement noise and the interaction between fire message broadcasts, for each phase update k of each node, the PDF of its phase, $P_{\Phi_{\text{curr}}^{(k)}}$, changes after applying (1). This changes the probability of convergence to SState, as follows:

$$\begin{aligned} \Pr\left[\left|\Phi_{\text{curr}}^{(k)} - \mu_{\Phi_{\text{curr}}^{(k)}}\right| \leq b_{\text{thres}}\right] &= \int_{-b_{\text{thres}}}^{b_{\text{thres}}} P_{\Phi_{\text{curr}}^{(k)}}(u - \mu_{\Phi_{\text{curr}}^{(k)}}) du \\ &= \text{erf}\left(\frac{b_{\text{thres}}}{\sqrt{2}\sigma_{\Phi_{\text{curr}}^{(k)}}}\right), \end{aligned} \quad (5)$$

where $\text{erf}(u)$ is the error function [18]. Notice that (5) holds under the assumption that $P_{\Phi_{\text{curr}}^{(k)}}$ converges to a normal distribution, which will be shown to be the case. We therefore use a stochastic criterion for convergence based on the confidence intervals of the normal distribution [18]. By defining the confidence coefficient

$$c_{\text{conf}} = \Pr\left[\left|\Phi_{\text{curr}}^{(k)} - \mu_{\Phi_{\text{curr}}^{(k)}}\right| \leq b_{\text{thres}}\right], \quad 0 < c_{\text{conf}} < 1, \quad (6)$$

²The use of the modulo operator in (4) is imposed because $\varphi_{\text{fire}} \in [0, 1)$.

and replacing in (5), we reach that, for convergence under (5):

$$\sigma_{\Phi_{\text{curr}}^{(k)}} = \frac{b_{\text{thres}}}{\sqrt{2} \times \text{erf}^{-1}(c_{\text{conf}})}. \quad (7)$$

Thus, we determine the phase-update iteration k_{SSState} for which $\sigma_{\Phi_{\text{curr}}^{(k_{\text{SSState}})}}$ is closest to the right-hand side of (7). That is, we determine the phase update iteration leading to convergence to SState with probability that closely matches c_{conf} .

Definition 3 (Steady State): We define desynchronization being in *steady state* with $c_{\text{conf}} \times 100\%$ confidence, at the k_{SSState} th phase-update iteration, where

$$k_{\text{SSState}} = \arg \min_{\forall k \in \mathbb{N}} \left| \sigma_{\Phi_{\text{curr}}^{(k)}} - \frac{b_{\text{thres}}}{\sqrt{2} \times \text{erf}^{-1}(c_{\text{conf}})} \right|, \quad (8)$$

with $\sigma_{\Phi_{\text{curr}}^{(k)}}$ the standard deviation at the k th iteration of (1). ■

Proposition 1. *Under Definitions 1-3, the number of firing cycles for the DESYNC phase update to converge is*

$$k_{\text{desync}} = \arg \min_{\forall k \in \mathbb{N}} \left| \sigma_{\text{desync}}^{(k)} - \frac{b_{\text{thres}}}{\sqrt{2} \times \text{erf}^{-1}(c_{\text{conf}})} \right|, \quad (9)$$

with

$$\sigma_{\text{desync}}^{(k)} = \sqrt{\left\| \mathbf{v}_W^{(k)} \right\|^2 \sigma_{\Phi^{(0)}}^2 + \sum_{j=1}^k \left\| \mathbf{v}_W^{(j)} \right\|^2 \sigma_{\Delta}^2}, \quad (10)$$

$$\mathbf{v} = \left[\frac{\alpha}{2} \quad 1 - \alpha \quad \frac{\alpha}{2} \right], \quad (11)$$

and

$$\mathbf{v}_W^{(j)} = \underbrace{(\mathbf{v} * \dots * \mathbf{v})}_j \Big|_W,$$

produced by j consecutive circular convolutions of period W .

Proof: For the first firing cycle, we denote the initial phase random variables by the $1 \times W$ vector

$$\Phi^{(0)} = \left[\dots \quad \left(\Phi_{\text{prev}}^{(0)} \right)_{\text{prev}} \quad \Phi_{\text{prev}}^{(0)} \quad \Phi_{\text{curr}}^{(0)} \quad \Phi_{\text{next}}^{(0)} \quad \dots \right] \quad (12)$$

and the corresponding additive measurement noise sources [independent identically distributed (i.i.d.) random variables] from Definition 2 by the $1 \times W$ vector

$$\Delta = \left[\dots \quad \left(\Delta_{\text{prev}} \right)_{\text{prev}} \quad \Delta_{\text{prev}} \quad \Delta_{\text{curr}} \quad \Delta_{\text{next}} \quad \dots \right]. \quad (13)$$

The number of elements before and after $\Phi_{\text{curr}}^{(0)}$ and Δ_{curr} in (12) and (13) depends on how many fire message broadcasts precede or follow the broadcast under consideration within the firing cycle. The first iteration of the phase update of (1) is

$$\Phi^{(1)} = \left[\mathbf{v} * \left(\Phi^{(0)} + \Delta \right) \right]_W \pmod{1}. \quad (14)$$

The circular convolution performs periodic extension of the phase and noise vectors of (12) and (13), which corresponds

to the circular dependency between consecutive firing cycles. Due to this dependency, the exact number of elements before and after $\Phi_{\text{curr}}^{(0)}$ and Δ_{curr} does not affect $\Phi^{(1)}$. Notice that (14) holds under the assumption that the statistics of all phase variables of all firings are “stale” [4, 15], i.e., they correspond to the initial state (Definition 1). This is because we do not assume any knowledge of *the order of firings* (which may not be fixed [12]). Thus, Proposition 1 makes use of the statistics of the *previous* phase update iteration. From (14), we get

$$\begin{aligned} \Phi_{\text{curr}}^{(1)} &= (1 - \alpha) \left(\Phi_{\text{curr}}^{(0)} + \Delta_{\text{curr}} \right) \\ &+ \frac{\alpha}{2} \left(\Phi_{\text{prev}}^{(0)} + \Delta_{\text{prev}} + \Phi_{\text{next}}^{(0)} + \Delta_{\text{next}} \right) \pmod{1}. \end{aligned} \quad (15)$$

Thus, $\mu_{\Phi_{\text{curr}}^{(1)}} = \mu_{\Phi_{\text{curr}}^{(0)}}$ and $\sigma_{\Phi^{(1)}} = \|\mathbf{v}\| \sqrt{(\sigma_{\Phi^{(0)}}^2 + \sigma_{\Delta}^2)}$. Generalizing (14) to k iterations, we reach

$$\Phi^{(k)} = \left(\mathbf{v}_W^{(k)} * \Phi^{(0)} \right)_W + \sum_{j=1}^k \left(\mathbf{v}_W^{(j)} * \Delta \right)_W \pmod{1} \quad (16)$$

where Δ is the i.i.d. measurement noise vector per iteration. Therefore, we obtain $\mu_{\Phi_{\text{curr}}^{(k)}} = \mu_{\Phi_{\text{curr}}^{(0)}}$ and $\sigma_{\Phi^{(k)}} = \sigma_{\text{desync}}^{(k)}$, shown in (10). It can now be shown that the statistics of $\Phi_{\text{curr}}^{(k)}$ satisfy the three conditions for the generalized form of the central limit theorem to be applicable [18, pp. 219-220], and thus $\Phi_{\text{curr}}^{(k)}$ becomes a normally-distributed random variable after a few phase updates, i.e.,

$$\Phi_{\text{curr}}^{(k)} \sim \mathcal{N} \left(\mu_{\Phi_{\text{curr}}^{(0)}}, \sigma_{\text{desync}}^{(k)} \right) \pmod{1} \quad (17)$$

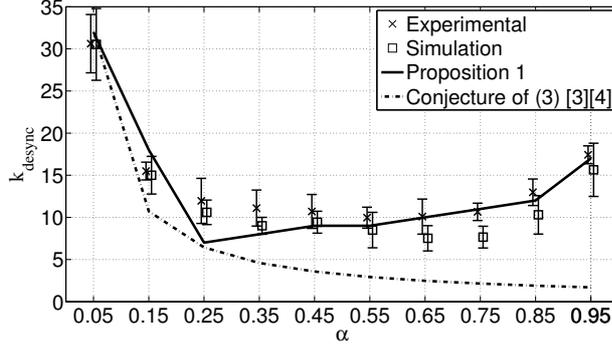
Hence, we reach (9) for convergence under Definition 3. ■

Proposition 1 shows how k_{desync} is affected by α as well as by the noise assumptions expressed by $\sigma_{\Phi^{(0)}}$ and σ_{Δ} in Definitions 1 and 2. Interestingly, according to (9), the number of nodes, W , does not influence the convergence to SState. This contradicts the conjecture of Degeys *et al.* [4] given by (3).

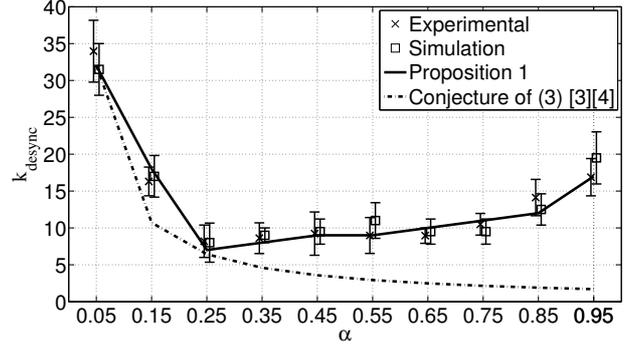
In Definition 1, every node’s initial phase random variable, i.e., $\Phi^{(0)}$, was assumed to be i.i.d. uniform. However, Proposition 1 holds for any i.i.d. random variable $\Phi^{(0)}$ that satisfies the three conditions for the generalized form of the central limit theorem to be applicable [18, pp. 219-220].

4. EXPERIMENTAL VALIDATION

We conduct experiments using imote2 Crossbow sensors with TinyOS1.x. The nodes use the IEEE 802.15.4 standard with the default 2.4GHz Chipcon CC2420 transceiver. By measuring the oscillatory behavior of each node’s phase in the SState, the standard deviation of the oscillating phase around the SState value of each node’s phase was $\sigma_{\Delta} = 0.34\text{ms}$. Also, the phase statistics over all nodes were confirmed as marginally white. We present results under $b_{\text{thres}} = 0.001$, with coupling constants $\alpha \in \{0.05, \dots, 0.95\}$ and number of nodes $W \in \{4, 16\}$, where we use $c_{\text{conf}} = 1 - 10^{-4}$ to detect



(a) $W = 4, b_{\text{thres}} = 0.001$



(b) $W = 16, b_{\text{thres}} = 0.001$

Fig. 2. Required firing cycles for convergence to SState for DESYNC for various values of α . Vertical error bars correspond to one standard deviation from the experimental (or simulation) mean values.

convergence. Each node reported the number of firing cycles until convergence was detected via a special report message to a base station listening passively to all messages. This enabled the automated collection of 50 such results per number of nodes and coupling constant. To cross-validate our results with simulations, we used the Matlab code of Degeysys *et al.* [4]. To simulate the noise conditions, we apply zero-mean additive noise in the phase update with $\sigma_{\Delta} = 0.34\text{ms}$ and set each node to misfire with probability 0.4% (300 Matlab runs per pair $\{W, \alpha\}$).

4.1. DESYNC Results

The results of the proposed model in relation to the experimental measurements are shown in Fig. 2. We also include the order-of-convergence conjecture of (3) [4], by scaling the order estimate to fit within the range of the obtained experiments and simulations. The results show that the WSN tends to converge to steady state faster when α decreases (until $\alpha = 0.25$), since the presence of measurement noise causes higher-amplitude oscillations for strong coupling. However, for very small values of α , the convergence iterations increase dramatically due to weakened coupling between phase-neighboring nodes. The proposed model is within the standard deviation of the experimental results for all cases. As shown in Fig. 2, previous estimates do not accurately match the behavior observed in the convergence iterations. It is important to note that contrary to the proposed model, previous estimates [4] do not take into account the measurement noise conditions. Moreover, the results of Fig. 2 show no statistical dependence on W , which agrees with our analysis.

4.2. Indicative Application

Consider a WSN that initially comprises W nodes, where nodes are expected to join or leave the network every T_{swap} seconds. In fair TDMA scheduling, the bandwidth per node

is $\frac{B_{\text{WSN}}}{W}$ bps, where B_{WSN} is the maximum application-layer bandwidth in IEEE 802.15.4 [4]. The fluctuating number of nodes in the WSN will result in bandwidth loss as, each time nodes join or leave, the system needs to converge to SState before transmission resumes with equal slot size [4, 14, 19]. Using the proposed model, we can derive an estimate of the expected bandwidth per node under such conditions. Specifically, if the DESYNC firing-cycle period is T seconds and the node joining or exiting occurs (on average) every T_{swap} seconds, the expected bandwidth per node can be estimated as $B_{\text{swap}} = \left(1 - \frac{k_{\text{desync}}T}{T_{\text{swap}}}\right) \frac{B_{\text{WSN}}}{W}$. The factor $\frac{k_{\text{desync}}T}{T_{\text{swap}}}$ expresses the normalized loss of bandwidth per node due to convergence to TDMA.

Let a WSN comprising $W = 10$ nodes with $T = 1\text{s}$ and $1 \sim 3$ nodes entering or exiting the network every T_{swap} s, with $T_{\text{swap}} \in [70, 130]\text{s}$. The proposed model provides the setting for α that minimizes the convergence iterations (i.e., k_{desync}) and leads to the maximum achievable bandwidth per node. In particular, previous work used $\alpha = 0.95$ instead of the best option, which according to our analysis is $\alpha = 0.25$. Experimental results show that selecting α through the proposed model brings gains of up to 13% in the bandwidth per node compared with the standard settings used in existing works [3, 4].

5. CONCLUSIONS

A novel stochastic estimation framework for the convergence iterations to fair TDMA scheduling is proposed for the DESYNC primitive. Our analytic result is validated based on simulations and experiments with a fully-meshed network of wireless sensors. Our model can be used to estimate the best operational parameters (and the associated delay) to establish fair TDMA scheduling under several desynchronization-based WSN protocols. In this way, it facilitates the analytic understanding of the temporal evolution of the desynchronization process in WSNs.

6. REFERENCES

- [1] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. Strogatz, "Distributed synchronization in wireless networks," *IEEE Signal Process. Mag.*, vol. 25, no. 5, pp. 81–97, Sep. 2008.
- [2] R. Leidenfrost and W. Elmenreich, "Firefly clock synchronization in an 802.15.4 wireless network," *Eurasip J. Embed. Syst.*, vol. 2009, Article id. 18585, 17 pages, 2009.
- [3] R. Pagliari, Y.-W. Hong, and A. Scaglione, "Bio-inspired algorithms for decentralized round-robin and proportional fair scheduling," *IEEE J. on Select. Areas in Commun.*, vol. 28, no. 4, pp. 564–575, May 2010.
- [4] J. Degeysys, I. Rose, A. Patel, and R. Nagpal, "Desync: Self-organizing desynchronization and TDMA on wireless sensor networks," *Proc. 6th Int. Symp. on IPSN*, pp. 11–20, Apr. 2007.
- [5] A. Mutazono, M. Sugano, and M. Murata, "Energy efficient self-organizing control for wireless sensor networks inspired by calling behavior of frogs," *Computer Communications*, vol. 35, no. 6, pp. 661–669, 2012.
- [6] H. Yamamoto, N. Wakamiya, and M. Murata, "An inter-networking mechanism using stepwise synchronization for wireless sensor networks," in *Bio-Inspired Models of Network, Information, and Computing Systems*. Springer, 2012, pp. 276–287.
- [7] T. Nakano, "Biologically inspired network systems: a review and future prospects," *IEEE Trans. on Syst., Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 5, pp. 630–643, Sep. 2011.
- [8] J. Klinglmayr and C. Bettstetter, "Self-organizing synchronization with inhibitory-couples oscillators: convergence and robustness," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 7, no. 3, Sep. 2012.
- [9] A. Patel, J. Degeysys, and R. Nagpal, "Desynchronization: the theory of self-organizing algorithms for round-robin scheduling," *Proc. IEEE Internat. Conf. on Self-Adaptive and Self-Organizing Syst. (SASO'07)*, pp. 87–96, Jul. 2007.
- [10] S. Ashkiani and A. Scaglione, "Discrete dithered desynchronization," *arXiv preprint arXiv:1210.2122*, 2012.
- [11] S. Choochaisri, K. Apicharttrisorn, K. Korprasertthaworn, P. Taechalertpaisarn, and C. Intanagonwiwat, "Desynchronization with an artificial force field for wireless networks," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 2, pp. 7–15, 2012.
- [12] C. Lien, S. Chang, C. Chang, and D. Lee, "Anchored desynchronization," in *Proc. IEEE INFOCOM'12*, 2012, pp. 2966–2970.
- [13] I. Bojic, V. Podobnik, I. Ljubi, G. Jezic, and M. Kusek, "A self-optimizing mobile network: auto-tuning the network with firefly-synchronized agents," *Information Sciences, Elsevier*, vol. 182, no. 1, pp. 77–92, 2012.
- [14] D. Buranapanichkit and Y. Andreopoulos, "Distributed time-frequency division multiple access protocol for wireless sensor networks," *IEEE Wireless Comm. Letters*, vol. 1, no. 5, pp. 440–443, Oct. 2012.
- [15] J. Degeysys and R. Nagpal, "Towards desynchronization of multi-hop topologies," in *Proc. IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'08)*, 2008, pp. 129–138.
- [16] R. Pagliari and A. Scaglione, "Scalable network synchronization with pulse-coupled oscillators," *IEEE Trans. on Mobile Computing*, vol. 10, no. 3, pp. 392–405, Mar. 2011.
- [17] Y.-W. Hong, A. Scaglione, and R. Pagliari, "Pulse coupled oscillators' primitive for low complexity scheduling," in *IEEE International Conference Acoustics, Speech and Signal Processing (ICASSP'09)*, 2009, pp. 2753–2756.
- [18] A. Papoulis, *Probability and Statistics*. Prentice Hall, 1989.
- [19] H. Besbes, G. Smart, D. Buranapanichkit, C. Kloukinas, and Y. Andreopoulos, "Analytic conditions for energy neutrality in uniformly-formed wireless sensor networks," *IEEE Trans. on Wireless Communications*, pp. 1–16, "to appear".