Bio-Inspired Algorithms for Decentralized Round-Robin and Proportional Fair Scheduling

Roberto Pagliari, Y.-W. Peter Hong and Anna Scaglione

***Abstract*—In recent years, several models introduced in mathematical biology and natural science have been used as the foundation of networking algorithms. These bio-inspired algorithms often solve complex problems by means of simple and local interactions of individuals. In this work, we consider the development of decentralized scheduling in a small network of self-organizing devices that are modeled as pulse-coupled oscillators (PCOs). By appropriately designing the dynamics of the PCO, the network of devices can converge to a desynchronous state where the nodes naturally separate their transmissions in time. Specifically, by following Peskin's PCO model with inhibitory coupling, we first show that round-robin scheduling can be achieved with weak convergence, where the nodes' transmissions are separated by a constant duration, but the differences of their local clocks continue to shift over time. Then, by having each node accept coupling only from the pulses emitted by a subset of neighboring nodes, we show that it is possible to achieve strict desynchronization, where the difference between local clocks remain fixed over time. More interestingly, by having each node maintain two local clocks, we show that it is possible to further achieve proportional fair scheduling, where the time alloted to each node is proportional to their demands. The convergence of these algorithms is studied both analytically and numerically.**

***Index Terms*—Wireless Sensor Networks, Scheduling, PCO, Consensus.**

I. Introduction

LONG before man made communication network existed, natural phenomena had ways of creating what we can call *order*. One of the first scientific observers of these phenomena was the dutch physicist Christian Huygens, who in the 17th century studied the synchronization of two penduli mounted on the same beam, which he attributed to the "imperceptible motion of the air" [1]. Centuries later, in 1975, Charles Peskin [2] modeled the sinoatrial node cells that are responsible for the pumping of the heart as leaky integrators (a common resistor-capacitor series) followed by a threshold element; as the voltage across the capacitor reached a threshold, each cell would broadcast to its neighbors a current, perturbing the voltage at the other capacitors. The Pulse Coupled Oscillator (PCO) model was born. Since then, the model has become extremely popular in basic science and has offered insights on a number of phenomena, as is beautifully narrated in the book by Steven Strogatz [3]. Less obvious to the engineering arena was the idea that the PCO model offered a remarkably simple architecture suitable for self-organization in small networks of simple low-cost devices, as demonstrated in the works on network synchronization [4]–[7] and distributed estimation [8], [9]. It is not only the physical simplicity of the mechanisms that are striking, but also how interactive PCOs can achieve communication and computation in an intertwined and inseparable fashion. While there are compelling engineering reasons to employ the layered architecture in most modern communication systems, such as the ease of interoperability among different functionalities and the flexibility in system design, these examples from mother nature show that specific applications can be achieved in a simple and efficient manner by combining different functionalities and treating them jointly as a single module. This paper shows how the functionalities of the PCO can be expanded to provide not only common network synchronization, but also decentralized multiple access scheduling. This approach is particularly suitable for wireless sensor networks (WSNs) due to the simplicity of the algorithms and the robustness to dynamic changes of the environment.

WSNs have been exploited today by literally hundreds of applications, ranging from habitat, infrastructure, and environmental monitoring, to localization and tracking, to cooperation and coordination of mobile agents. To fulfill the needs of different sensor network applications, many sensor MAC protocols have been developed in the past (see, e.g., [10]–[12]). These schemes focus predominantly on two extremes of the spectrum, namely, random access (such as variants of slotted ALOHA or CSMA) on one side and centralized scheduling (such as TDMA or CDMA) on the other. Random access schemes are decentralized in nature, making them suitable for environments where the size of the network is unknown, the topology is dynamically changing, and the traffic demands are bursty. On the other hand, centralized scheduling provides a reliable means of transmitting data at a high and constant rate. Such requirements arise in a subclass of WSN applications, referred to as body area networks (BANs) [13], where small embedded devices constantly sample biological signals from the human body. However, centralized scheduling schemes require a master node to take charge of the process and may not be robust to dynamic changes of the environment or to failure of the master node. A decentralized approach to scheduling is thus desirable. Inspired by the PCO model, we propose in this paper a method to achieve a TDMA scheme in a decentralized manner. In addition to BANs, the algorithms we present are

Manuscript received 16 March 2009; revised 18 October 2009.

Roberto Pagliari is with the Department of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA (e-mail: rp294@cornell.edu).

Yao-Win Peter Hong is with the Institute of Communications Engineering, National Tsing Hua University, Hsinchu, Taiwan (e-mail: ywhong@ee.nthu.edu.tw).

Anna Scaglione is with the Department of Electrical and Computer Engineering, University of California, Davis, CA, USA(e-mail: ascaglione@ucdavis.edu).

Digital Object Identifier 10.1109/JSAC.2010.100506.

generally applicable to small networks composed of low-complexity devices that cannot support an explicit signaling channel for intra-node coordination.

In the literature, a number of papers has already been devoted to promoting the use of PCO as a synchronization primitive [4]–[7]. By inverting the polarity of the coupling signal, it has been observed in [3] that the PCO network can be led to a pulsing pattern that yields constant spacing between neighboring pulses. This emergent behavior, called *desynchronization*, has been utilized in [14], [15] to achieve round-robin scheduling, where each node in the network is allotted an equal share of the total transmission time. Specifically, in [14], [15], it has been shown numerically by the authors that a PCO scheme with negative coupling can produce a pattern where the firing of the nodes tend to be separated in time by an amount equal to T_f/n, where T_f is the period of each node's local clock and n is the number of nodes in the network. An alternative method to achieve this task has also been proposed in [16] based on the concept of average consensus. These methods fall under the general class of pulse coupling schemes, where communication is encoded in the nodes' pulsing times and computation is performed through local phase updates.

In this work, we first show that the original PCO model with inhibitory coupling yields only weak desynchronization, which refers to the case where a constant spacing between consecutive nodes' pulsing times is reached but the differences between their local phase variables shift constantly over time. Although round-robin scheduling is achieved in this case, the convergence is not robust to non-ideal effects of the environment since it relies on the constant interaction between PCOs. Then, by restricting each pulse coupling to only a subset of nodes, we show that it is possible to achieve strict desynchronization, where the phase differences among nodes also remain constant over time. Given knowledge of the number of nodes n, the primitive we describe is shown to achieve desynchronization with a faster rate than the scheme proposed in [16]. More interestingly, we also show that, by having each node maintain two local clocks instead of one, we can further achieve proportional fair scheduling where the time alloted to different nodes are proportional to their demands.

The remainder of this paper is organized as follows. In Section II, we review the basic mechanisms of the PCO. In Section III, we introduce the notion of strict and weak desynchronization and show how they can be achieved by appropriate designs of the PCO dynamics. In IV we show that proportional fair scheduling can be achieved by having each node maintain two local clocks instead of one. The proposed scheduling schemes and the theoretical results are verified through numerical simulations in Section V. Finally, we conclude in Section VI.

II. Background on Synchronization of PCOs

Pulse coupled oscillators (PCOs) are elements that pulse individually in a periodic manner if separated, but may alter their pulsing patterns in response to the signals heard from other elements if interconnected (hence the adjective "coupled"). In particular, the recipient of a pulse will move earlier or later its own firing by altering a local clock that regulates the pulse emission of the node. By following simple local coupling rules, networks of PCOs are known to produce a variety of pulsing patterns, which have been used to model the spiking of neurons in the brain [17] or the flashing of fireflies [3]. We consider a network of n nodes inter-connected through direct transmission links. When in isolation, each node emits a pulse periodically every T_f, which is referred to as the *firing cycle*. The evolution of the local time at node i is characterized by a *phase variable*

$$\Phi_i(t) = \frac{t}{T_f} + \phi_i \bmod 1, \tag{1}$$

which increases linearly from 0 to 1 in each cycle. When the phase $\Phi_i(t)$ reaches 1, node i will emit a pulse and reset its phase back to 0. Using an analogy with an alarm clock, $\Phi_i(t)$ is basically the local time at node i normalized by the firing cycle T_f, and ϕ_i is an initial offset at absolute time $t = 0$. The local phase variable at each node, say node i, is mapped to the local *state variable* $X_i(t)$ by the function f so that $X_i(t) = f(\Phi_i(t))$, where f is called *the PCO dynamics*[1].

When the nodes are interconnected, the reception of a pulse at any given node adds ε (called *coupling*) to the local state variable $X_i(t)$, altering the phase and, thus, the next pulsing time of the nodes receiving the coupling. Specifically, if node i receives a pulse at time t, it will immediately update its local phase variable at time $t + \eta$ for $\eta \to 0$ such that

$$\begin{aligned}\Phi_i(t^+) &= \lim_{\eta \downarrow 0} \Phi_i(t+\eta)\\ &= f^{-1}(X_i(t)+\varepsilon) = f^{-1}(f(\Phi_i(t))+\varepsilon).\end{aligned} \tag{2}$$

In practice, the pulse coupling may be affected by propagation or processing delays as well as false alarm or miss detection of pulses at the receiver. However, to gain theoretical insight on the convergence behavior, we consider, as in many works on PCOs, the idealized scenario where (a) the transceiver processing and propagation delays are negligible and (b) the pulses are always detected without error by other nodes. Practical issues that may arise by loosening these assumptions will be studied via numerical simulations in Section V. Following these idealized assumptions, it has been shown, in the seminal paper [3] by Mirollo and Strogatz, that convergence to the synchronous state can be achieved given conditions on the coupling strength ε and the dynamics f as stated below.

Theorem 1 (Synchronization [3]): *For any positive coupling, i.e. $\varepsilon > 0$, the set of initial states $\{\phi_1, \phi_2, \ldots, \phi_n\}$ that never result in synchrony has measure zero if the function f is smooth, monotonically increasing, and concave down.*

A well known example of the dynamics mentioned above is the one given by Peskin in [2] where $f(\Phi_i(t)) = (1 - e^{-\gamma \Phi_i(t)})/(1 - e^{-\gamma})$. More interestingly, it has also been shown in [3] that, with the dynamics f mentioned above and a negative coupling (i.e., $\varepsilon < 0$), the nodes will separate their firing times asymptotically by a constant amount. The same result occurs if the function f is concave-up and the coupling strength is positive $\varepsilon > 0$ (see Figure 1). Thus, it is evident that

[1] In the past, the PCO dynamics have often been taken as a solution of some partial differential equation that describes the uncoupled PCO (see e.g. [17] for more details).

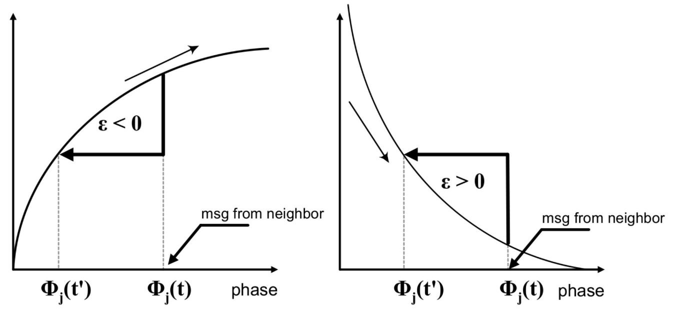


Fig. 1. Left: Strogatz model with concave-down f and $\varepsilon < 0$. A message received from the neighborhood causes the node to step back in phase (its next firing time increases). Right: PCO-based model with concave-up f and $\varepsilon > 0$. This is a similar model to the one shown on the left. After detecting a message, the node add a positive quantity to its state, but since f is concave-up, its phase decreases.

by tuning local parameters, such as the updating function or the coupling strength, the PCO dynamics can produce different global behaviors that can be used as the basis of decentralized scheduling algorithms, as shown in the following sections.

III. Round-Robin Scheduling with PCO Desynchronization

The problem of desynchronization can be considered the dual of the synchronization problem, where the states of the nodes coalesce under the action of the network dynamics. Instead, nodes that are desynchronized will alternate their pulses in order with constant spacing between each other. The constant spacing produced by the PCO dynamics can be utilized as the basis of round-robin scheduling [18]. That is, each node can utilize the time period between its own firing and the firing of the next node to transmit its data. Specifically, suppose that the nodes are indexed in an increasing order of their initial phases such that $\phi_1 < \phi_2 < \cdots < \phi_n$ (i.e., the nodes pulse in the inverse order of their indices). As we show later on, this firing order will be preserved by all the schemes proposed in this work. When each node fires exactly once, from node n to node 1, we say that we have completed a *round* of firing. In between two consecutive firings, the local phase variable of each node, say node i, will evolve according to eq. (1) and will be updated at every firing event based on eq. (2). We can imagine the set of n nodes as balls placed on a circle of circumference equal to 1 (the normalized period) moving clockwise at the same speed in decreasing order of their indices, as shown in Fig. 2. When a node, say node i, crosses the finish line (i.e., $\Phi_i(t) = 1$), it will emit a pulse and trigger the other nodes to update their local phase variables according to eq. (2).

To study the evolution of the system it is sufficient to track how the phase differences between two consecutively firing nodes change over time. Let

$$\Delta_i(t) = \Phi_i(t) - \Phi_{i-1}(t) \pmod 1, \tag{3}$$

for $i = 2, \ldots, n$, be the phase difference between nodes i and $i-1$ at time t and let

$$\Delta_1(t) = \Phi_1(t) - \Phi_n(t) \pmod 1. \tag{4}$$

Following our previous analogy, where we view the nodes as balls placed on a circle of circumference equal to 1, the parameter $\Delta_i(t)$ refers to the distance around the circumference between neighboring balls i and $i-1$. Notice that modulo 1 is taken in the definition of $\Delta_i(t)$ since node i resets its phase to 0 after its own firing, causing the difference $\Phi_i(t) - \Phi_{i-1}(t)$ to change signs when this occurs.

A. Notation and Definitions

Each firing modifies the phase differences between nodes and, thus, the state of the system can be described by the vector

$$\mathbf{\Delta}(t) = (\Delta_n(t), \ldots, \Delta_1(t))^T$$

which follows a trajectory over the n-dimensional plane defined by $\sum_{i=1}^{n} \Delta_i(t) = 1$, subject to the constraints that $\Delta_i(t) \in (0, 1)$, $\forall i$. We can achieve two types of convergence, namely, strict and weak desynchronization, as described below.

Definition 1: *The network is strictly desynchronized if the phase differences converge to a constant value as $t \to \infty$. This is equivalent to saying that, for any $\epsilon > 0$, there exists t^* such that*

$$|\Delta_i(t) - c_{\text{strict}}| < \epsilon \tag{5}$$

for all $t > t^*$ and $i = 1, \ldots, n$.

When strict desynchronization is reached, the phase difference between any two consecutive nodes converges to the same fixed value and, thus, the spacing between two consecutive firings also remains constant. However, to achieve equal spacing between nodes' firing times, it is actually not necessary to have the phase differences remain constant over time. In fact, the phase differences can shift frequently over time as long as the time between consecutive firings remain constant. This leads to the notion of weak desynchronization as described in the following.

Recall that the sequence of n consecutive firings starting at node n and ending at node 1 constitutes a *round* of firing. Let us denote by $t_i^{(k)}$ the firing time of node i in the k-th round. By definition, a round terminates after the firing of node 1, which occurs at time $t_1^{(k)}$ for round k. Therefore, we define the phase differences at the end of round k as

$$\Delta_i[k] \triangleq \lim_{\eta \downarrow 0} \Delta(t_1^{(k)} + \eta), \quad \text{for } i = 1, \ldots, n \tag{6}$$

and the state of the system in round k as

$$\mathbf{\Delta}[k] = (\Delta_n[k], \ldots, \Delta_1[k])^{\mathrm{T}}. \tag{7}$$

It is worthwhile to note that $\Delta_1[k]$ is the difference in firing time between node 1 and node n, which is the next one to fire. Since any node can be labeled as node 1 by a shift of the time of origin, the convergence of $\Delta_1[k]$ to a fixed value can be viewed as the convergence of firing time between any two consecutive nodes.

Definition 2: *The network is weakly desynchronized if the time elapsed between consecutive firings converges to a constant value, i.e., for any $\epsilon > 0$, there exists k^* such that*

$$|\Delta_1[k] - c_{\text{weak}}| < \epsilon \tag{8}$$

for all $k > k^*$ and $i = 1, \ldots, n$.

When strict desynchronization is reached, the nodes will fire at fixed time instants in each period. When weak desynchronization is reached, instead, the time between two consecutive firings converges to a constant, but the phase differences are not fixed for all t. Clearly, strict desynchronization implies weak desynchronization, but not the other way around.

Definition 3: *The convergence time R^* is the minimum number of rounds needed to achieve ϵ-desynchronization, i.e., convergence with ϵ accuracy.*

The analytical results reported in the following will make use of the concept of round, which is the sequence of n consecutive firings described previously. The strategy we use to prove convergence results can be briefly described as follows. Instead of considering the convergence of each phase difference separately, we look specifically at the convergence of the entire system state $\mathbf{\Delta}[k]$. By defining the fixed point of the system state as

$$\mathbf{\Delta}^* = \lim_{k\to\infty} \mathbf{\Delta}[k] = [\Delta_n^*, \ldots, \Delta_1^*]^T, \tag{9}$$

our goal is to show that, $\forall\epsilon > 0$, $\exists k^*$ such that

$$||\mathbf{\Delta}[k] - \mathbf{\Delta}^*||_1 < \epsilon \tag{10}$$

for all $k > k^*$, where $\|\cdot\|_1$ is the 1-norm. By considering the 1-norm, eq. (10) can be viewed as a bound for the sum of distances with respect to the fixed point, which also implies ϵ-convergence of $\Delta_1[k]$. In particular, if $\mathbf{\Delta}^* = c_{\text{strict}}\mathbf{1}_n$ (all elements are equal) we have strict desynchronization, in which case $c_{\text{strict}} = \frac{1}{n}$, since $||\mathbf{\Delta}^*||_1 = 1$. If $\mathbf{\Delta}^* = \mathbf{v}$ with $\frac{\mathbf{v}}{||\mathbf{v}||_1} \neq \mathbf{1}_n$, instead, we have weak desynchronization with $c_{\text{weak}} = \Delta_1^*$. Equation (10) will be used in the following to prove convergence and rate of convergence.

B. Weak Desynchronization

The protocol we describe first is inspired by the PCO model, introduced by Peskin [2], with inhibitory coupling. Rather than changing the polarity of the coupling, we maintain $\varepsilon > 0$ and introduce a convex function f (see Figure 1). This system produces the same global effects as described in the following. Specifically, by taking the dynamics as

$$f(\Phi_j(t)) = -\log(\Phi_j(t)) \tag{11}$$

and positive coupling $\varepsilon = -\log(1-\alpha)$, where $\alpha \in (0,1)$, the firing of node i at time t_i will trigger node j to update its phase as

$$\begin{aligned}\Phi_j(t_i^+) &= f^{-1}(f(\Phi_j(t_i)) - \log(1-\alpha)) \\ &= (1-\alpha)\Phi_j(t_i),\end{aligned} \tag{12}$$

which is consistent with that described in eq. (2). We can now state the following result (see Appendix A for the proof).

Theorem 2 (Weak Desynchronization): *For any $\alpha \in (0,1)$, the dynamics in* (12) *will reach weak desynchronization with $c_{\text{weak}} = \frac{\alpha}{1-(1-\alpha)^n}$, except over a set of initial conditions of measure zero. The convergence occurs in $R^* = \left\lceil -\frac{1}{n}\frac{\log\left(\frac{2}{\epsilon}\right)}{\log(1-\alpha)}\right\rceil$ rounds.*

Although the time required to achieve weak desynchronization scales favorably as $O(\frac{1}{n})$ with respect to the network size, the algorithm itself is not robust to detection errors as to be shown in Section V. This is due to the fact that the phase differences between nodes do not remain constant over time and, thus, achieving equal spacing between consecutive firing times must rely on constant interaction among the nodes. This motivates the study on strict desynchronization.

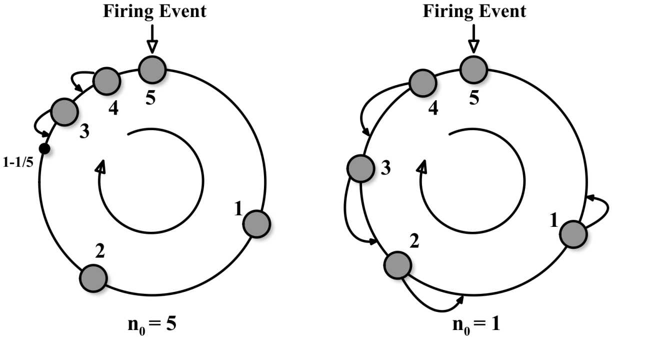


Fig. 2. Update in PCO-Based Round-Robin, with $n = 5$ nodes. Left: $n_0 = n$; in this case only the nodes whose phase is bigger than $1 - \frac{1}{5}$ shift back in phase. Right: $n_0 = 1$; a firing causes all the others to change their phase.

C. Strict Desynchronization

One of the reasons that the previous scheme can only achieve weak desynchronization is because the firing of each node will always impose a coupling on all the other nodes. Thus, the nodes phases will be constantly updated throughout each round. To achieve strict desynchronization, our intuition tells us that one should restrict the coupling caused by each firing event to only a subset of neighboring nodes. That is, the firing of a node should only push away the nodes whose phase is too close. Based on this intuition, we introduce the following updating rule (assuming node i fires at time t_i)

$$\Phi_j(t_i^+) = \begin{cases} f^{-1}(f(\Phi_j(t_i)) + \varepsilon), & \text{if } \Phi_j(t_i) \in \left(1 - \frac{1}{n_0}, 1\right) \\ \Phi_j(t), & \text{otherwise.}\end{cases}$$

$\forall j \neq i$, where the parameter $n_0 \geq 1$ controls the spacing between nodes, and the dynamics

$$f(\Phi_j(t)) = -\log\left(1 - n_0(1 - \Phi_j(t))\right) \tag{13}$$

is a concave-up function similar to that in (11) and in Fig. 1 but defined only over the range $\left(1 - \frac{1}{n_0}, 1\right)$. We can see that the phase of node j is pushed toward the point $1 - 1/n_0$ through a convex combination with parameter $\alpha = 1 - e^{-\varepsilon} \in (0,1)$. In fact,

$$\begin{aligned}\Phi_j(t^+) &= f^{-1}(f(\Phi_j(t)) + \varepsilon) \\ &= (1-\alpha)\,\Phi_j(t) + \alpha\left(1 - \frac{1}{n_0}\right),\end{aligned} \tag{14}$$

for all $j \neq i$ such that $\Phi_j(t_i) \in (1 - \frac{1}{n_0}, 1)$. It is interesting to note that (13) reduces to (12) when $n_0 = 1$. We can now state the the following results (see Appendix B for proof):

Theorem 3 (Strict Desynchronization): *For any $\alpha \in (0,1)$, the dynamics in* (13) *with $n_0 = n$ will reach strict desynchronization for all initial conditions, except over a set of measure zero.*

By assuming that the firing of each node imposes a coupling only on the node immediately firing after it (which is typically the case when the nodes are close to the desynchronous state), we show in Appendix C for a special case that strict desynchronization can be achieved in $R^* = O(\frac{n}{\log n})$ number of rounds. The algorithm we described reaches strict desynchronization in a way that depends on the number of active nodes. Therefore, extra complexity needs to be introduced to set the parameter n_0 appropriately.

On the other hand, in the DESYNCH method proposed in [16], each node accepts coupling only from one node (i.e., the node firing immediately after it), regardless of the number of nodes in the network. Therefore, knowledge of n is not required. Specifically, upon hearing the pulse emitted by node $i-1$, node i will update its local phase $\Phi_i(t)$ by deliberately moving it towards the middle-point between the phases of nodes $i+1$ and $i-1$, which is the time that node i should be firing if desynchronization is reached (i.e., the target phase). That is, by assuming that node $i-1$ fires at time t_{i-1}, node i will update its phase to

$$\Phi_i(t_{i-1}^+) = (1-\alpha)\Phi_i(t_{i-1}) + \alpha\Phi_i^{\text{target}}(t_{i-1}^+), \qquad (15)$$

where $\alpha \in (0,1)$ is the coupling parameter and $\Phi_i^{\text{target}}(t_{i-1}^+) = \frac{\Phi_{i+1}(t_{i-1}^+)+\Phi_{i-1}(t_{i-1}^+)}{2}$ is the target phase. Recall that $\Phi_{i-1}(t_{i-1}^+) = 0$ since a node resets its phase after emitting a pulse. However, it is necessary to note that node i can only attain knowledge of the phases of nodes $i-1$ and $i+1$ by observing their pulsing times. By assuming that t_{i+1} is the most recent firing time of node $i+1$, the estimate that node i has on the phase of node $i+1$ is

$$\hat{\Phi}_{i+1}(t) = \Phi_i(t) + [1 - \Phi_i(t_{i+1})],$$

where $1-\Phi_i(t_{i+1})$ is the phase difference between nodes $i+1$ and i at the time node $i+1$ was firing. However, the phase of node $i+1$ would have shifted by the time node $i-1$ is firing (due to the firing of node i) and, thus, would no longer be accurate when node i updates its local phase variable. A conjecture on the convergence of the DESYNCH protocol was given in [18] and is restated below.

Conjecture 1 ([18]): *n nodes governed by* DESYNC *will always be driven to strict desynchronization (for $n < 500$) in $R^* = \frac{1}{\alpha}n^2\log\left(\frac{1}{\epsilon}\right)$ rounds.*

D. Discussion

The different protocols we have discussed so far for round-robin scheduling belong to the same class of algorithms that use time of transmission to encode the state of each node and use pulse coupling as the basis of nodes' interaction. The basic PCO model with negative coupling provides convergence in a *weak* sense, which is still useful to schedule the nodes' duty-cycles, but is not robust to pulse transmission errors since the convergence relies on constant shifting of phases triggered by the pulse transmissions. When coupling of each pulse is limited to only a selected group of nodes, as in the case of the PCO-based scheme with thresholding or DESYNCH, strict desynchronization can be achieved.

Although the PCO-based scheme and DESYNCH both belong to the class of pulse-coupling algorithms, they have inherent differences. In DESYNCH, each node must identify the nodes firing immediately before and after it and deliberately record an estimate of their phases. In the PCO-based schemes, each node can accept a coupling whenever the relative time difference between the reception of a pulse and the local firing fall within a certain interval set by the threshold. There is no need for the PCO-based schemes to keep track of the adjacent nodes and their phases. However, the disadvantage of the PCO scheme lies in the fact that knowledge of the network size n is required in order to guarantee equal-share scheduling. Nevertheless, convergence to a fixed schedule can always be achieved even when n_0 is not equal to n, but the schedule may not result in equal share and the type of convergence will depend on the value of n_0 (e.g., weak for $n_0 < n$ and strict for $n_0 \geq n$ as we show numerically in Section V). More importantly, the convergence time of DESYNCH is $O(n^2)$ while PCO with $n_0 = n$ requires only $O(\frac{n}{\log n})$.

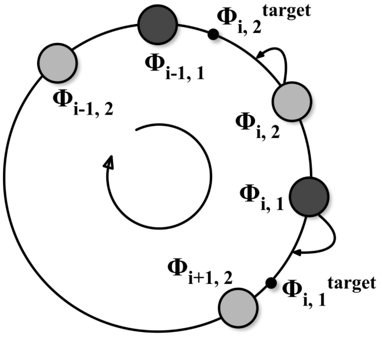


Fig. 3. Proportional Fairness: update of node i, caused by the firing of node $i-1$. Node i either expands or stretches its phase variables, depending on its neighbors' states. The objective of node i is given by $\Phi_{i,1}^{\text{target}}$ and $\Phi_{i,2}^{\text{target}}$.

In this section, we have shown how equal-share scheduling can be achieved by having nodes follow a simple pulse-coupling mechanism. In the following section, we shall extend the ideas of pulse coupling to the case of proportional fair scheduling by using more than one clocks at each node.

IV. PCO PRIMITIVES FOR PROPORTIONAL FAIR SCHEDULING

In many networking applications, different nodes may have different bandwidth requirements and, thus, should be allocated unequal shares of the transmission time that are proportional to their demands[2]. In this context, we assume that each node has a specific request that is quantified by the value K_i, where i is the node index. The value K_i lies between 1 and a maximum value $K_{\max}$, which is fixed *a priori* and equal for all the nodes, that is, $K_i \in [1, K_{\max}]$ with $K_{\max} < \infty$. Basically, K_i stands for the amount of resources node i is hoping to obtain. Since our objective is to derive a negotiation scheme not based on explicit exchange of information, the nodes may be requesting more than the actual available resource. Hence, our goal is to have each node obtain, by negotiation, a portion of time equal to $\frac{K_i}{K}T_f$, where $K = \sum_{i=1}^{n} K_i$ is the total demand of all nodes.

[2] An example of proportionally fair schedule is in the protocol IS-895, aimed at unleashing multi-user diversity taking into consideration fairness issues in the channel assignment.

Following the formulation in the previous section, an intuitive approach is to have each node, say node i, maintain K_i virtual nodes that employ the PCO-based scheme with the threshold mechanism or DESYNCH. Since each virtual node will obtain one out of K shares of the transmission time T_f, node i will obtain a total K_i/K portion of the transmission time. However, this approach is obviously inefficient since each node needs to emit K_i pulses in each cycle and the convergence time also increases proportionally. In the following, we show that proportional fair scheduling can actually be achieved by having each node maintain only *two* clocks and by embedding the demand K_i in the update rule. The basic idea is to treat each node as a spring of force with constant K_i. If we take many springs, each of which obeys the Hooke's law $F_i = -K_i x_i$ with x_i being the difference between the actual and natural lengths of spring i, and connect them together in a circle, then each spring will occupy a portion of the circle that is proportional to K_i/K. We can then imagine n users having different requests K_i as n springs connected together in a ring. The two pulses emitted by each node mark the boundaries of the spring on the circle and the update rule incorporates the strength of the spring that is parameterized by K_i. Each user needs then to communicate only with its two neighbors, implying that only two state variables are truly needed.

Each node has two phase variables, indicated by $\Phi_{i,1}(t)$ and $\Phi_{i,2}(t)$, where i is the index of the node. Let $\phi_i \in [0,1)$ be the initial phase of both phase variables of node i. When nobody is firing, the two phases will evolve as

$$\Phi_{i,1}(t) = (\phi_i + t/T_f) \bmod 1, \tag{16}$$

$$\Phi_{i,2}(t) = (\phi_i + t/T_f) \bmod 1, \tag{17}$$

and fire when either $\Phi_{i,1}(t)$ or $\Phi_{i,2}(t)$ reaches the value 1. Our goal is to expand the time between the firings of the two clocks at each node (with $\Phi_{i,1}(t)$ firing before $\Phi_{i,2}(t)$ as illustrated in Fig. 3) and use it as the transmission period scheduled for that node. By assuming that an extra δ portion of the unit share (defined as $1/K$ of the total time scheduled for the nodes' transmission) is used as guard band between the transmission periods of adjacently transmitting users, the network will require a total of $K + n\delta$ shares and node i should ideally obtain $\frac{K_i}{K+n\delta}$ portion of the transmission time when proportional fair scheduling is achieved. In that case, the target phases of node i should be

$$\begin{cases} \Phi_{i,1}^{\text{target}}(t) = \frac{\delta}{K_i+2\delta}\Phi_{i-1,1}(t) + \frac{K_i+\delta}{K_i+2\delta}\Phi_{i+1,2}(t) \\ \Phi_{i,2}^{\text{target}}(t) = \frac{K_i+\delta}{K_i+2\delta}\Phi_{i-1,1}(t) + \frac{\delta}{K_i+2\delta}\Phi_{i+1,2}(t). \end{cases} \tag{18}$$

In the proposed scheme, we allow node i to update the phases of its two local clocks immediately upon the firing of the first clock of node $i-1$. The update is ideally performed by moving the local phases towards the target such that

$$\begin{cases} \Phi_{i,1}(t_{i-1,1}^{+}) = (1-\alpha)\Phi_{i,1}(t_{i-1,1}) + \alpha\Phi_{i,1}^{\text{target}}(t_{i-1,1}^{+}) \\ \Phi_{i,2}(t_{i-1,1}^{+}) = (1-\alpha)\Phi_{i,2}(t_{i-1,1}) + \alpha\Phi_{i,2}^{\text{target}}(t_{i-1,1}^{+}) \end{cases} \tag{19}$$

where $\alpha \in (0,1)$ and $t_{i-1,1}$ is the firing time of the first clock of node $i-1$. The update is illustrated in Fig. 3. Recall that, at time $t_{i-1,1}^{+}$, we have $\Phi_{i-1,1}(t_{i-1,1}^{+}) = 0$ and, thus, the target phases reduce to $\Phi_{i,1}^{\text{target}}(t_{i-1,1}^{+}) = \frac{K_i+\delta}{K_i+2\delta}\Phi_{i+1,2}(t_{i-1,1}^{+})$ and $\Phi_{i,1}^{\text{target}}(t_{i-1,1}^{+}) = \frac{\delta}{K_i+2\delta}\Phi_{i+1,2}(t_{i-1,1}^{+})$.

The following result establishes the convergence of this idealized version of the updating procedure (see Appendix D for proof).

Theorem 4: [PROPORTIONAL FAIRNESS] *The algorithm in (19) converges for all initial conditions* $(\phi_1, \phi_2, \ldots, \phi_n)$, *except over a set of measure zero. Furthermore, by letting* $\Gamma_i(t) = \Phi_{i,1}(t) - \Phi_{i,2}(t) \pmod 1$, *and* $\Theta_i(t) = \Phi_{i,2}(t) - \Phi_{i-1,1}(t) \pmod 1$, *we have*

$$\lim_{t\to\infty} \Gamma_i(t) = \beta\frac{K_i}{K} \quad \textit{and} \quad \lim_{t\to\infty} \Theta_i(t) = \beta\frac{\delta}{K},$$

for all i, where $\beta = \frac{1}{1+\frac{n\delta}{K}}$ and $K = \sum_{i=1}^{n} K_i$.

The theorem indicates that the portion of time node i obtains converges to βK_i, where β is the unit share. When the guard interval is made arbitrarily small, that is, as δ goes to 0, node i will obtain exactly K_i/K portion of the firing cycle T_f for transmission. However, it is necessary in practice to have $\delta > 0$ not only to accommodate the synchronization errors and non-negligible pulse widths, but also to leave a vacant time window for external nodes to join the network. In any case, proportional fair scheduling is achieved since the time alloted to the users are indeed proportional to their demands.

This approach is similar in nature to the DESYNCH and thus has the same problem regarding the estimate of the phase of the node firing before it. Therefore, we can similarly replace the phase $\Phi_{i+1,2}(t_{i-1,1}^{+})$ in the update rule with the phase estimate

$$\hat{\Phi}_{i+1,2}(t) = \Phi_{i,1}(t) + [1 - \Phi_{i,1}(t_{i+1,2})],$$

where $t_{i+1,2}$ is the most recent firing time of the second clock of node $i+1$. This estimate is obtained by having node i observe the firing time of node $i+1$ and compute the relative phase difference between itself and node $i+1$ at the time of firing. However, since the estimate of the phase of node $i+1$ is no longer accurate by the time node $i-1$ is firing, it is possible that pulsing order between clocks $\Phi_{i,1}(t)$ and $\Phi_{i+1,2}(t)$ may exchange positions after the update. That is, the period in between the two pulses of nodes i and $i+1$ may overlap. To avoid this problem, we modify the target values defined in (18) as follows

$$\begin{cases} \tilde{\Phi}_{i,1}^{\text{target}}(t) = \min\left(\Phi_{i,1}^{\text{target}}(t), \frac{\Phi_{i,1}(t)+\Psi_{i+1,2}(t)}{2}\right), \\ \tilde{\Phi}_{i,2}^{\text{target}}(t) = \max\left(\Phi_{i,2}^{\text{target}}(t), \frac{\Phi_{i-1,1}(t)+\Phi_{i,2}(t)}{2})\right). \end{cases} \tag{20}$$

This choice avoids the possibility that the clocks interleave their phases. As discussed in [19], the fixed point is preserved and no other fixed points exist. The dynamics in (19) with target values defined by (20) is similar to the ideal case, with the only difference that the update is limited by max and min to avoid overlapping. Numerical simulations show that the convergence properties are not affected.

V. COMPUTER SIMULATIONS

In this section we provide and discuss the performances of the algorithm we discussed for achieving round-robin scheduling and proportional fairness based on PCO-like primitives. All simulations have been performed using MATLAB.

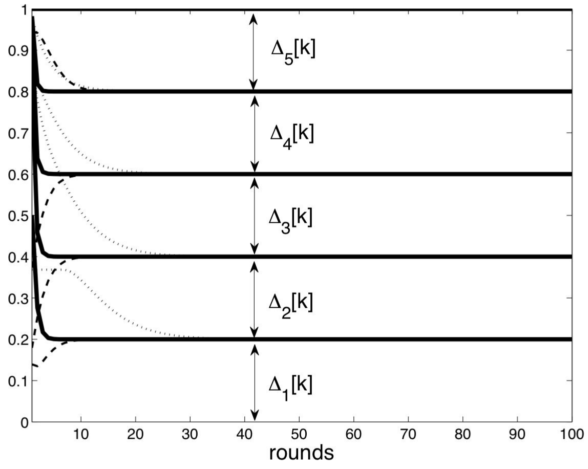


Fig. 4. Evolution of phase distances with $n = 5$ nodes, $n_0 = n$ and different values of α ($\alpha = 0.25$ [dotted], $\alpha = 0.5$ [dashed], $\alpha = 0.9$ [solid]).

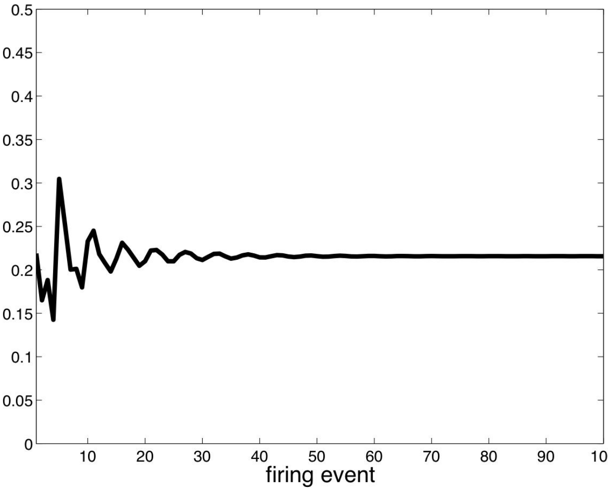


Fig. 6. Time between consecutive firings for the PCO-based algorithm, with $n = 5$, $n_0 = 3$ and $\alpha = 0.4$. Although the phase differences do not converge to the same value, the time elapsed between consecutive firings converges.

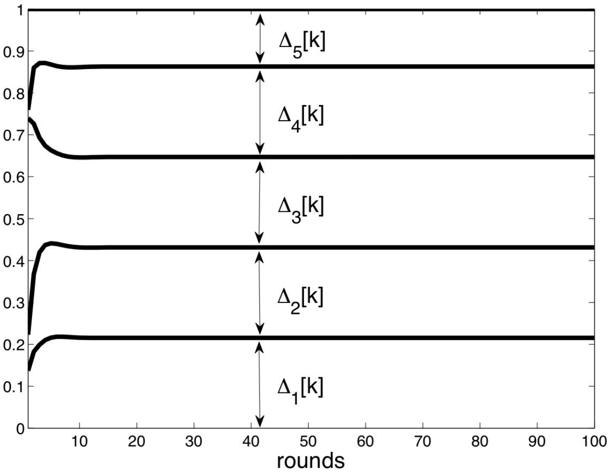


Fig. 5. Evolution of a network with $n = 5$ nodes, $n_0 = 3$ and $\alpha = 0.4$. We can see that the phase differences do not converge to the same value.

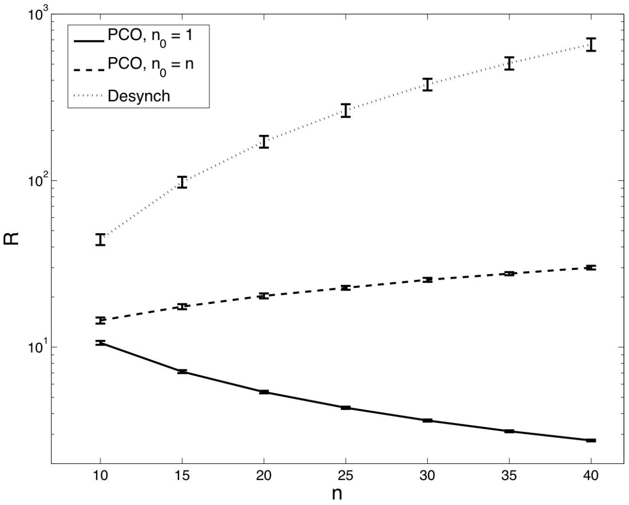


Fig. 7. Number of rounds R required to achieve an accuracy $\epsilon = 10^{-4}$ for (i) DESYNCH with $\alpha = 0.9$, (ii) PCO with $n = n_0$ and $\alpha = 0.75$, and (iii) PCO-based with $n_0 = 1$ and $\alpha = 8 \cdot 10^{-2}$. Each point has been obtained by averaging over a set of 1000 experiments.

Round-Robin Scheduling

In Figure 4, we show the evolution of the phase distances $\Delta_i[k]$ according to the dynamics specified in (13) with $n = 5$ and coupling parameter $\alpha = 0.25,\ 0.5,\ 0.9$. The difference between the curves from top to bottom are $\Delta_5[k]$, $\Delta_4[k]$, …, $\Delta_1[k]$, as indicated in the figure. Since all phase differences converge to $1/n = 0.2$, we can infer that strict desynchronization is achieved (see Section III-A). Also, as expected, increasing the coupling strength results in a faster convergence to the fixed point $\frac{1}{n}\mathbf{1}$.

In Figure 5, we show the evolution of the system state for a network with $n = 5$ nodes and parameters $\alpha = 0.4$ and $n_0 = 3 < n$. This is an illustration of weak desynchronization which occurs whenever $n_0 < n$. Recall that the dynamics described in (11) is a special case of (13) with $n_0 = 1 < n$. We observe that the phase differences do not converge to the same value, but the difference in consecutive firing times remains fixed as indicated in Figure 6, which shows the difference in firing time with respect to the firing events.

In Figure 7, we reported the number of rounds R required to achieve desynchronization, with $\epsilon = 10^{-4}$. We can see that the DESYNCH protocol exhibits a complexity that scales as $O(n^2)$, as conjectured in [18]. The PCO-based dynamics in (13) exhibits sublinear complexity, while in the case of weak desynchronization it is of the order of $\frac{1}{n}$.

Now, let us consider the non-ideal effects of the transmission channel. We consider the case where each node experiences miss detection with probability p each time a pulse is emitted. Assume that the miss detection probability p is equal for all nodes and that signal-to-noise ratio is sufficiently high such that false alarm is negligible[3]. In Fig. 8, we reported an example of weak desynchronization with $n = 10$ nodes and $\alpha = 0.1$ in two different cases: the ideal case and the case

[3] The issues regarding processing and propagation delays are not considered explicitly in this work since we are dealing with a network with closely located nodes, such as BAN. However, these issues can be treated by employing refractory periods as done in [4], but may affect the precision of the schedule.

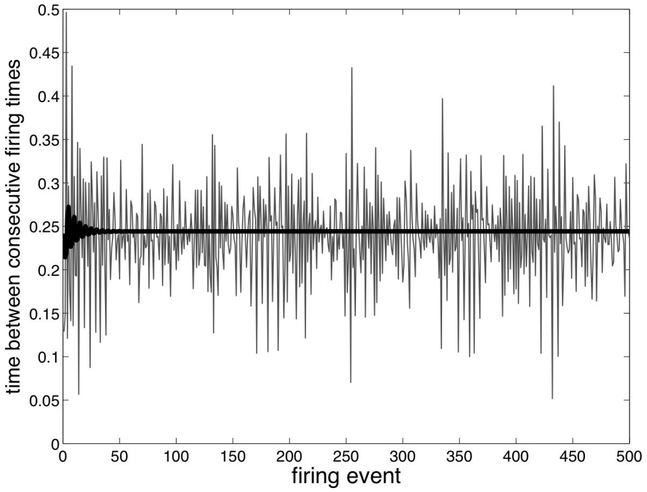


Fig. 8. Weak desynchronization with packet losses. In black is reported the time elapsed between consecutive firings in the ideal case ($p = 0$), while in grey is reported the same quantity with $p = 0.1$. In this case $n = 10$ and $\alpha = 0.1$. We can see that in the latter case the amount of time reserved by a node is not constant.

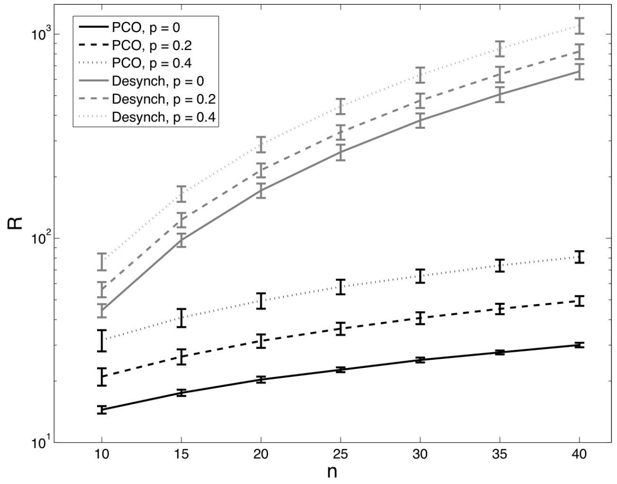


Fig. 9. Strict desynchronization achieved by PCO and DESYNCH in the ideal case and with noisy channel. The plot shows the number of rounds needed to achieve convergence, with $\epsilon = 10^{-4}$, $\alpha = 0.75$ for PCO and $\alpha = 0.9$ for DESYNCH. Data have been averaged over 1000 experiments for each point.

with miss detection probability $p = 0.1$. We can see that the desynchronous state is not steadily maintained in the case of weak desynchronization since it relies on constant interaction between PCOs, which can be disturbed by detection errors. In the case of strict desynchronization, as shown in Fig. 9, the desynchronous state can be better maintained since the interaction between PCOs are not required once the this state is reached. However, the time needed to reach the convergent state is now increased.

Proportional Fairness

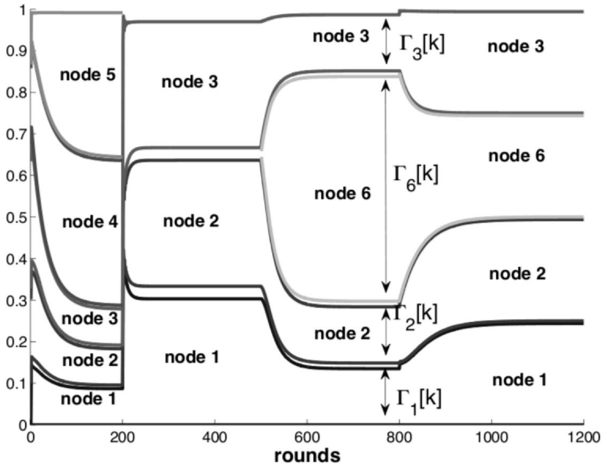


Fig. 10. Evolution of $\Phi_{i,1}(t)$ and $\Phi_{i,2}(t)$ with respect to $\Phi_{1,2}(t)$ when nodes leave or join the network, and the requests change over the time.

In Figure 10, we show the performance of the proportional fair scheduling scheme based on two local clocks, as described in Section IV. In this simulations, we allow the nodes to join or leave the system or adjust their requests at different time instants. We can see that the proposed scheme can dynamically adapt to these changes. Specifically, suppose that the network is initially composed of 5 nodes, whose requests are $\mathbf{K} = [5, 5, 5, 20, 20]$. At round 200, nodes 4 and 5 leave the network, while nodes 1, 2 and 3 remain in the network. We can see that the period T_f is then shared equally among the three nodes, since $K_1 = K_2 = K_3 = 5$. At round 500, node 6 with demand $K_6 = 20$ joins the network. The convergent state then allots to node 6 the biggest amount of time, while the rest is allotted an equal share of the remaining time. Finally, at round 800 the requests become all equal, i.e., $K_1 = K_2 = K_3 = K_6 = 20$ and the nodes converge toward a state where they all get the same portion of time.

Proportional fairness can also be achieved using the DESYNCH protocol by having each node generate a number of virtual nodes that is equal to its demand, say K_i. Hence, the total number of virtual nodes in the network would be $\sum_{i=1}^{n} K_i$. In Fig. 11, we compared the convergence time between DESYNCH with virtual nodes and the proposed proportional fair scheduling scheme. The parameter $K_{\max}$ has been set to 5 and 10, respectively. The request of each node has been generated uniformly at random between 1 and $K_{\max}$. We can see that the PCO-based scheme is faster than DESYNCH in both cases, even though the complexity is of the same order (which corresponds to the slope of the curve in the plot). Our scheme has the advantage that only two clocks are needed and, thus, it is easier to implement than the DESYNCH, which would require K_i clocks for each node.

In Fig. 12, we reported the evolution of the system with the effect of detection errors. The solid line refers to the case with $p = 0$, while the dotted one shows the case with $p = 0.1$. The network is composed of 5 nodes, with $\alpha = 0.03$ and the demands are given by $\mathbf{K} = [5, 2, 3, 4, 2]$. We can see that the proposed scheme is relatively robust to detection errors.

VI. CONCLUSIONS

In this work, we have proposed a class of PCO-based bio-inspired algorithms for decentralized scheduling in small networks of low-cost, low-complexity devices. The emergent properties of these protocols allow the network to self-organize

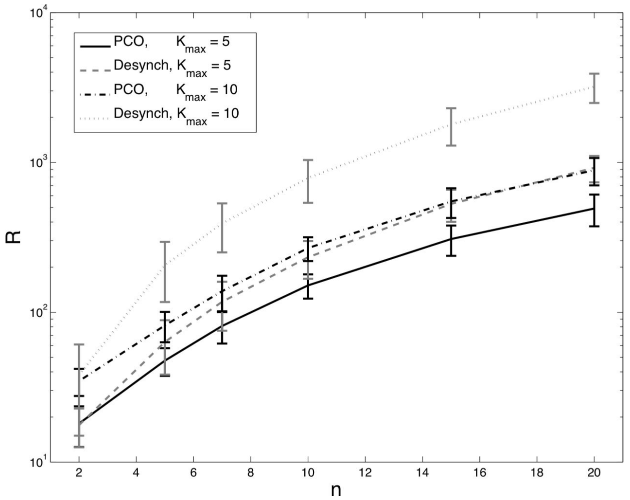


Fig. 11. Number of rounds required to reach desynchronization for DESYNCH and our PCO-based protocol, with $\epsilon = 10^{-2}$ and $\alpha = 0.9$ in both cases. Data have been average over a set of 1000 experiments for each point.

the activities of the nodes without the need of a master node. Specifically, by employing Peskin's model with inhibitory coupling, we have shown that convergence to weak desynchronization can be achieved. Then, by carefully restricting the coupling to a subset of neighboring nodes, we have shown that one can further achieve strict desynchronization, where the phase differences between nodes remain constant over time and, thus, is more robust to detection errors. Moreover, we have also found that, by utilizing two local clocks instead of one, the nodes can also achieve proportional fair scheduling, where the time alloted to the nodes are proportional to their demands. Future work will focus on the implementation of such primitives on practical sensor platforms.

ACKNOWLEDGMENT

Roberto Pagliari and Anna Scaglione were supported by the National Science Foundation, USA, under grants NSF-0834582 and Welch Allyn. Yao-Win Peter Hong was supported by the National Science Council, Taiwan, under grants NSC-95-2221-E-007-043-MY3, NSC-96-2628-E-007-012-MY2, and NSC-97-3114-E-007-002.

APPENDIX A
PROOF OF THEOREM 2

In this case, the firing of a node causes all the others to perform the update. We shown next that the evolution of the state vector, from round to round, can be described by an affine transformation. Consider for example, the case with three nodes, and assume that the initial offsets, at round 0, are $\phi_3(0) > \phi_2(0) > \phi_1(0)$. Note that, $\Phi_3(t_3) = 1$, $\Delta_3(t_3) = 1 - \Phi_2(t_3)$ while $\Delta_2(t_3) = \Phi_2(t_3) - \Phi_1(t_3)$ and $\Delta_1(t_3) = \Phi_1(t_3) - \Phi_3(t_3) \mod 1 \ = \Phi_1(t_3)$. Considering equation (12), we obtain that $\Delta_3(t_3^+) = 1 - (1-\alpha)\Phi_2(t_3) = 1 - (1-\alpha)(1 - \Delta_3(t_3))$ and $\Delta_2(t_3^+) = (1-\alpha)\Phi_2(t_3) - (1-\alpha)\Phi_1(t_3)$. Hence, when node 3 fires at time t_3, the state of

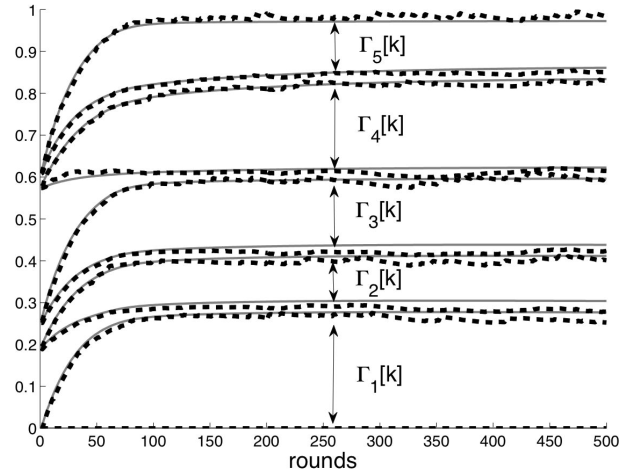


Fig. 12. Evolution of a system composed of $n = 5$ nodes, with requests vector $\mathbf{K} = [5, 2, 3, 4, 2]$, $\alpha = 0.03$ and $p = 0.1$. In the figure are reported both the ideal case (solid line) and the case with noisy channel (dotted line).

the system becomes linear:

$$\begin{aligned}\Delta_3(t_3^+) &= (1-\alpha)\Delta_3(t_3) + \alpha \\ \Delta_2(t_3^+) &= (1-\alpha)\Delta_2(t_3) \\ \Delta_1(t_3^+) &= (1-\alpha)\Delta_1(t_3)\end{aligned}$$

At time $t_2 > t_3$ node 2 fires, and, thus, the intervals between the nodes change as $\Delta_3(t_2^+) = (1-\alpha)\Delta_3(t_2)$, $\Delta_2(t_2^+) = (1-\alpha)\Delta_2(t_2) + \alpha$ and $\Delta_1(t_2^+) = (1-\alpha)\Delta_1(t_2)$. At time $t_1 > t_2$ node 1 fires, and the state of the system becomes

$$\begin{aligned}\Delta_3(t_1^+) &= (1-\alpha)\Delta_3(t_1) = (1-\alpha)^3\Delta_3(t_3) + (1-\alpha)^2\alpha \\ \Delta_2(t_1^+) &= (1-\alpha)\Delta_2(t_1) = (1-\alpha)^3\Delta_2(t_3) + (1-\alpha)\alpha \\ \Delta_1(t_1^+) &= (1-\alpha)\Delta_1(t_1) + \alpha = (1-\alpha)^3\Delta_1(t_3) + \alpha.\end{aligned}$$

The evolution of the system from round R to round $R+1$ is then $\mathbf{\Delta}[R+1] = \mathbf{M}_3\mathbf{\Delta}[R] + \mathbf{v}_3$, where $\mathbf{M}_3 = (1-\alpha)^3\mathbf{I}$ and $\mathbf{v}_3 = [(1-\alpha)^2\alpha, (1-\alpha)\alpha, \alpha]^T$. More generally, if we have n nodes the system evolves as $\mathbf{\Delta}[R+1] = \mathbf{M}\mathbf{\Delta}[R] + \mathbf{v}$ where $\mathbf{M} = (1-\alpha)^n\mathbf{I}$ and $\mathbf{v} = \alpha[(1-\alpha)^{n-1}, (1-\alpha)^{n-2}, \ldots, (1-\alpha), 1]^T$. Therefore, at round $R+1$

$$\begin{aligned}\mathbf{\Delta}[R+1] &= \mathbf{M}^R\mathbf{\Delta}[0] + \sum_{i=0}^{R-1}\mathbf{M}^i\mathbf{v} \\ &= \mathbf{M}^R\mathbf{\Delta}[0] + (\mathbf{I} - \mathbf{M})^{-1}\left(\mathbf{I} + \mathbf{M}^R\right)\mathbf{v}. \quad (21)\end{aligned}$$

$\mathbf{M}$ is a contraction, and, thus, from Eq. (21) we have $\mathbf{\Delta}^* = \lim_{R\to\infty}\mathbf{\Delta}[R] = (\mathbf{I} - \mathbf{M})^{-1}\mathbf{v}$. By expanding (21) we obtain

$$\begin{aligned}||\mathbf{\Delta}[R] - \mathbf{\Delta}^*|| &= ||\mathbf{M}^R\mathbf{\Delta}[0] + (1-(1-\alpha)^n)^{-1}\mathbf{M}^R\mathbf{v}|| \\ &= (1-\alpha)^{nR}\ ||\mathbf{\Delta}[0] + (1-(1-\alpha)^n)^{-1}\mathbf{v}|| \\ &\leq 2(1-\alpha)^{nR} < \epsilon. \quad (22)\end{aligned}$$

Condition (22) implies $R^* = -\frac{1}{n}\frac{\log\left(\frac{2}{\epsilon}\right)}{\log(1-\alpha)}$; whenever this quantity becomes less than 1 (number of firings less than n), we set $R^* = 1$ and the result of the theorem follows. The spacing between two consecutive firings is given by $\Delta_1^* T_f = ((\mathbf{I} - \mathbf{M})^{-1}\mathbf{v})_1 T_f = \frac{\alpha}{1-(1-\alpha)^n}T_f$.

The spacing between two consecutive rings converges to a constant with value $\Delta_1 T_f = \frac{\alpha}{1-(1-\alpha)^n}T_f$. This shows that,

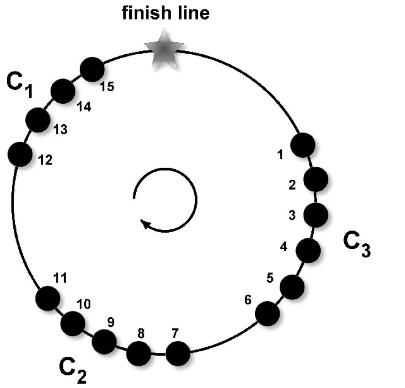


Fig. 13. Example of chains of nodes at time $t > 0$ in a network composed of $n = 15$ nodes. C_1, C_2 and C_3 are the three chains of the system at time t.

although the phase difference between any two nodes do not converge to a fixed value throughout the process, this scheme is sufficient to achieve round-robin scheduling where each node gets equal share of the channel. Interestingly, as $\alpha \to 0$, the spacing between two rings is converges to $1/n$, which is the value achieved in the case of strict desynchronization. However, as $\alpha \to 0$ the time to convergence increases as $1/\alpha$.

APPENDIX B
PROOF OF THEOREM 3

Without loss of generality, we assume that $\Delta_i(0) > 0, \forall i = 1, 2, \ldots, n$.[4] We first show that if two consecutive nodes are separated by a gap strictly bigger than zero, then, those nodes can never collapse into the same point. If this statement does not hold, then it implies that $\lim_{t\to\infty} \Delta_j(t) = 0$ for some j. This implies that $\lim_{t\to\infty} \Phi_{j-1}(t) = \Phi_j(t)$. But from (13), when $\Phi_{j-1}(t) \to \Phi_j(t)$ the firing of node j causes node $j-1$ to update as $\Phi_{j-1}(t^+) \to 1 - \frac{\alpha}{n} \neq 0$. Therefore, $\Delta_j(t_j^+) = \alpha/n \neq 0$, leading to a contradiction.

Prior to proceeding, we define a *chain* of nodes at time t as a set of consecutive nodes whose inter-distances are less or equal to $1/n$. An example is shown in Fig. 13 where, at time t, three chains are present in the system. We indicate with $\mathcal{C}(t)$ the set of chains in the system at time t, and its cardinality (the number of chains) with $|\mathcal{C}|$. In Fig. 13, for instance, $\mathcal{C}(t) = \{C_1, C_2, C_3\}$ and $|\mathcal{C}(t)| = 3$. Moreover, we call *size* of chain C_l the distance between the two extreme nodes in the chain. In Fig. 13, for example, the size of chain C_1 is equal to $\Delta_{13}(t) + \Delta_{14}(t) + \Delta_{15}(t) = \Phi_{15}(t) - \Phi_{12}(t)$. Notice that, by definition, these chains are separated by gaps bigger than $1/n$.

Lemma 1: *The cardinality of the set $\mathcal{C}(t)$ is strictly positive and non increasing.*

Proof: The first result is trivial, since there always exists a pair of adjacent nodes whose distance is less than, or equal to $1/n$ and, thus, $|\mathcal{C}(t)| \geq 1$. Now, suppose that $|\mathcal{C}(t)|$ increases. This implies that a node (or a group of nodes) leaves its chain and form a new one, due to the firing of another node in the same chain. But a node never pushes apart another one more than $1/n$, leading to a contradiction. ■

On the other hand, the cardinality of $\mathcal{C}(t)$ can decrease, and this happens if the phase-jump made by a node is such that its distance to the next one becomes less than $1/n$. Since the cardinality of $\mathcal{C}(t)$ is non increasing, and at least equal to one, the sequence of integer variables $\{|\mathcal{C}[k]|\}$ has a minimum, and this minimum is reached in finite time. Suppose k^* is the first iteration at which the minimum is reached. Then, $\forall k > k^*$ the set of chains (separated by gaps bigger than $1/n$) does not change. Because of (13) the size of each chain C_l increases monotonically to $\frac{C_l - 1}{n}$, while each gap decreases to $1/n$, and the result of Theorem 3 follows.

APPENDIX C
CONVERGENCE RATE OF STRICT DESYNCHRONIZATION

Recall that $n_0 = n$. We consider the special case where the firing of a node only affect its neighbor (i.e., the node firing immediately after it) and where initially $\Delta_i[0] < 1/n, \forall i = 2, \ldots, n$ and $\Delta_1[0] > 1/n$. In this case, $\Delta_1[k]$ will decrease monotonically towards $1/n$ in each round, as described in Appendix B, and, therefore, $\sum_{i=2}^{n} \Delta_i[k]$ will increase towards $1 - 1/n$. Since it must hold that $\sum_{i=1}^{n} \Delta_i[k] = 1$ (which implies that $\Delta_1[k] = 1 - \sum_{i=2}^{n} \Delta_i[k]$), the system state of round k is uniquely specified by the $n-1$ dimensional vector $\hat{\mathbf{\Delta}}[k] = (\Delta_n[k], \Delta_{n-1}[k], \ldots, \Delta_2[k])^T$. The distance to the fixed point at round k is then given by

$$\begin{aligned}\left\|\mathbf{\Delta}[k] - \frac{1}{n}\mathbf{1}_n\right\|_1 &= \left\|\hat{\mathbf{\Delta}}[k] - \frac{1}{n}\mathbf{1}_{n-1}\right\|_1 + \left(\Delta_1[k] - \frac{1}{n}\right)\\ &= \left\|\hat{\mathbf{\Delta}}[k] - \frac{1}{n}\mathbf{1}_{n-1}\right\|_1 + \left(1 - \sum_{i=2}^{n} \Delta_i[k] - \frac{1}{n}\right)\\ &= 2\left\|\hat{\mathbf{\Delta}}[k] - \frac{1}{n}\mathbf{1}_{n-1}\right\|_1 \qquad (23)\end{aligned}$$

where $\mathbf{1}_{n-1}$ is the $n-1$ dimensional vector of ones. The convergence time is the minimum number of rounds required to satisfy the condition $2\left\|\hat{\mathbf{\Delta}} - \frac{1}{n}\mathbf{1}_{n-1}\right\| < \epsilon$.

To study the convergence of $\hat{\mathbf{\Delta}}[k]$, we apply the methodology used in Appendix A to find the recursive update of the phase difference vector, which we can show is also affine. Let us consider the updates in the k-th round and recall that the firing of each round starts with node n. When node n fires, the phase of node $n-1$ is updated as (14) and, thus, the variables Δ_n and Δ_{n-1} become

$$\begin{aligned}\Delta_n(t_n^+) &= (1-\alpha)\Delta_n(t_n) + \frac{\alpha}{n} &(24)\\ \Delta_{n-1}(t_n^+) &= \Delta_n(t_n) + \Delta_{n-1}(t_n) - \Delta_n(t_n^+)\\ &= \alpha\Delta_n(t_n) + \Delta_{n-1}(t_n) - \frac{\alpha}{n}, &(25)\end{aligned}$$

where t_i is the firing time of node i in the k-th round. Then, as node $n-1$ fires, only node $n-2$ will update its phase and, thus, the variables $\Delta_{n-1}(t_{n-1}^+)$ and $\Delta_{n-2}(t_{n-1}^+)$ can be computed as

$$\Delta_{n-1}(t_{n-1}^+) = (1-\alpha)\alpha\Delta_n(t_n) + (1-\alpha)\Delta_{n-1}(t_n) + \frac{\alpha^2}{n}$$

[4] In fact, if the nodes choose their initial phases at random and independently, the probability that any two nodes pick exactly the same value is zero.

$$\begin{aligned}\Delta_{n-2}(t_{n-1}^{+}) &= \Delta_{n-2}(t_n)+\alpha\Delta_{n-1}(t_n)\\ &\qquad +\alpha^2\Delta_n(t_n)-\frac{\alpha}{n}-\frac{\alpha^2}{n}\end{aligned}$$

Next, the firing of node $n-2$ will then modify $\Delta_{n-2}(t_{n-1}^{+})$ and $\Delta_{n-3}(t_{n-1}^{+})=\Delta_{n-3}(t_n)$ and so on and so forth, until node 1, whose firing will have no effect on the others, since $\Delta_1 > 1/n$. By some algebraic manipulation, the state of the network at round k can be expressed as

$$\begin{aligned}\hat{\boldsymbol{\Delta}}[k] &= \mathbf{M}^k\hat{\boldsymbol{\Delta}}[0]+\sum_{i=0}^{k-1}\mathbf{M}^i\mathbf{v}\\ &= \mathbf{M}^k\hat{\boldsymbol{\Delta}}[0]+(\mathbf{I}-\mathbf{M})^{-1}(\mathbf{I}+\mathbf{M}^k)\mathbf{v} \qquad (26)\end{aligned}$$

where

$$\{\mathbf{M}\}_{ij}=\begin{cases}(1-\alpha)\alpha^{i-j}, & \text{for } i\ge j\\ 0, & \text{otherwise,}\end{cases}$$

and $\mathbf{v}=\frac{1}{n}(\alpha,\alpha^2,\ldots,\alpha^{n-1})^T$. By induction, we can show that

$$\{(\mathbf{I}-\mathbf{M})^{-1}\}_{ij}=\begin{cases}\frac{1}{\alpha}, & \text{for } i=j\\ \frac{1-\alpha}{\alpha}, & \text{for } i>j\\ 0, & \text{otherwise,}\end{cases}$$

and, thus, $(\mathbf{I}-\mathbf{M})^{-1}\mathbf{v}=\frac{1}{n}\mathbf{1}_{n-1}$. Therefore, by (23), the distance to the fixed point at round k can be bounded by

$$\begin{aligned}&\left|\left|\boldsymbol{\Delta}[k]-\frac{1}{n}\mathbf{1}_n\right|\right|_1\\ &= 2\left|\left|\mathbf{M}^k\hat{\boldsymbol{\Delta}}[0]+(\mathbf{I}-\mathbf{M})^{-1}(\mathbf{I}+\mathbf{M}^k)\mathbf{v}-\frac{1}{n}\mathbf{1}_{n-1}\right|\right|_1\\ &= 2\left|\left|\mathbf{M}^k\hat{\boldsymbol{\Delta}}[0]+(\mathbf{I}-\mathbf{M})^{-1}\mathbf{M}^k\mathbf{v}\right|\right|_1\\ &= 2\left|\left|\mathbf{M}^k\hat{\boldsymbol{\Delta}}[0]\right|\right|_1+2\left|\left|(\mathbf{I}-\mathbf{M})^{-1}\mathbf{M}^k\mathbf{v}\right|\right|_1, \qquad (27)\end{aligned}$$

where we used the fact that the norm of a product is less than the product of the norms. The terms in (27) can be evaluated as follows.

First of all, by the definition of $\mathbf{M}$ and $\mathbf{v}$, we have that

$$\mathbf{M}\mathbf{v}=\frac{(1-\alpha)}{n}(\alpha,2\alpha^2,\ldots,(n-1)\alpha^{n-1})^T \prec (1-\alpha)n\mathbf{v}, \qquad (28)$$

where $\prec$ is the element-wise inequality. By applying the bound k times, we have $\mathbf{M}^k\mathbf{v}\prec(1-\alpha)^k n^k\mathbf{v}$. Hence,

$$\begin{aligned}(\mathbf{I}-\mathbf{M})^{-1}\mathbf{M}^k\mathbf{v} &\prec (\mathbf{I}-\mathbf{M})^{-1}(1-\alpha)^k n^k\mathbf{v}\\ &= (1-\alpha)^k n^k\frac{1}{n}\mathbf{1}=(1-\alpha)^k n^{k-1}\mathbf{1}\end{aligned}$$

and the second term in (27) becomes

$$\left|\left|(\mathbf{I}-\mathbf{M})^{-1}\mathbf{M}^k\mathbf{v}\right|\right|_1<(1-\alpha)^k n^k. \qquad (29)$$

Now, consider the product $\mathbf{M}\hat{\boldsymbol{\Delta}}[0]$. Since $\mathbf{M}$ can be viewed as a convolution matrix, the i^{th} element of the vector $\mathbf{M}\hat{\boldsymbol{\Delta}}[0]$ is given by

$$\begin{aligned}\{\mathbf{M}\hat{\boldsymbol{\Delta}}[0]\}_i &= (1-\alpha)\sum_{j=1}^{i}\alpha^{i-j}\hat{\Delta}_j[0]<\frac{1-\alpha}{n}\alpha^i\sum_{j=1}^{i}\alpha^{-j}\\ &= \frac{1-\alpha}{n}\alpha^i\frac{\alpha^{-(i+1)}-\alpha^{-1}}{\alpha^{-1}-1}<\frac{1}{n}\alpha^i\alpha^{-n} \qquad (30)\end{aligned}$$

where the first inequality follows from the assumption that $\hat{\Delta}_j[0]<1/n,\ \forall j$. Therefore, $\mathbf{M}\hat{\boldsymbol{\Delta}}[0]\prec\alpha^{-n}\mathbf{v}$ and

$$\mathbf{M}^k\hat{\boldsymbol{\Delta}}[0]=\mathbf{M}^{k-1}\mathbf{M}\hat{\boldsymbol{\Delta}}[0]\prec(1-\alpha)^{k-1}n^{k-1}\alpha^{-n}\mathbf{v}.$$

Therefore, we have

$$\left|\left|\mathbf{M}^k\hat{\boldsymbol{\Delta}}[0]\right|\right|_1<(1-\alpha)^{k-1}n^{k-1}\alpha^{-n}. \qquad (31)$$

Hence, by (27), it follows that

$$\begin{aligned}\left|\left|\boldsymbol{\Delta}[k]-\frac{1}{n}\mathbf{1}_n\right|\right|_1 &< 2(1-\alpha)^k n^{k-1}\alpha^{-n}+(1-\alpha)^k n^k\\ &< 2[(1-\alpha)n]^k\left(1+\frac{\alpha^{-n}}{1-\alpha}\right).\end{aligned}$$

Finally, by setting $2[(1-\alpha)n]^k\left(1+\frac{\alpha^{-n}}{1-\alpha}\right)<\epsilon$, we can see that, to guarantee that the distance to the fixed point is less than ϵ, the number of rounds k must be at least equal to R^*, where

$$R^*=\left\lceil\frac{\log\frac{\epsilon}{2\left(1+\frac{\alpha-n}{1-\alpha}\right)}}{\log(1-\alpha)+\log(n)}\right\rceil=O\left(\frac{n}{\log n}\right), \qquad (32)$$

given that $1-\alpha>1/n$.

Appendix D
Proof of Theorem 4

By using variables Γ_i and Θ_i, the state of the network characterized by the $2n$-dimensional vector $\boldsymbol{\Delta}(t)=(\Theta_1(t),\Gamma_1(t),\ldots,\Theta_n(t),\Gamma_n(t))^{\mathrm{T}}$. From (19), node $i-1$'s firing causes node i to update its two clocks which, in turn, modify variables Θ_i, Δ_i and Θ_{i+1} as follows:[5]

$$\begin{aligned}\Theta_i' &= \Psi_i'=\alpha\frac{\delta}{K_i+2\delta}(\Theta_i+\Delta_i+\Theta_{i+1})+(1-\alpha)\Theta_i \qquad (33)\\ &= \left[1-\alpha\frac{K_i+\delta}{K_i+2\delta}\right]\Theta_i+\frac{\alpha\delta}{K_i+2\delta}\Delta_i+\frac{\alpha\delta}{K_i+2\delta}\Theta_{i+1}\end{aligned}$$

$$\begin{aligned}\Delta_i' &= \Phi_i'-\Psi_i' \qquad (34)\\ &= \frac{\alpha K_i}{K_i+2\delta}\Theta_i+\left[1-\alpha\frac{2\delta}{K_i+2\delta}\right]\Delta_i+\frac{\alpha K_i}{K_i+2\delta}\Theta_{i+1}\end{aligned}$$

$$\begin{aligned}\Theta_{i+1}' &= \Psi_{i+1}-\Phi_i'=(\Theta_i+\Delta_i+\Theta_{i+1})-(\Theta_i'+\Delta_i') \qquad (35)\\ &= \frac{\alpha\delta}{K_i+2\delta}\Theta_i+\frac{\alpha\delta}{K_i+2\delta}\Delta_i+\left[1-\alpha\frac{K_i+\delta}{K_i+2\delta}\right]\Theta_{i+1}\end{aligned}$$

while all the other variables remain the same. As done in the previous sections, we consider the state at discrete times and study its evolution. From one firing to another, the new state is obtained simply by multiplying the previous one by a matrix $\mathbf{M}_i$, equal to the identity matrix except for a 3-by-3 (positive) block centered at position $2i-1$, induced by the coefficients in (33), (34) and (35). The evolution of the system can then be described as:

$$\begin{aligned}\boldsymbol{\Delta}^* &= \cdots\mathbf{M}_{n-1}\mathbf{M}_n\mathbf{M}_1\mathbf{M}_2\ldots\mathbf{M}_{n-1}\mathbf{M}_n\boldsymbol{\Delta}[0]\\ &= \lim_{R\to\infty}\mathbf{M}^R\boldsymbol{\Delta}[0]=\mathbf{M}_\infty\boldsymbol{\Delta}[0] \qquad (36)\end{aligned}$$

[5] For clarity, we omit the time index.

where $\mathbf{M} = \prod_{i=1}^{n} \mathbf{M}_i$ and $\mathbf{M}_\infty = \lim_{k\to\infty} \mathbf{M}^k$. Each $\mathbf{M}_i$ is a full-rank matrix and, thus, $\mathbf{M}$ is full-rank as well. Moreover, it is not hard to see that $\mathbf{M}_i$ is left-stochastic for all i, which implies that $\mathbf{M}$ is left-stochastic. Since $\mathbf{M} = \prod_{i=1}^{n} \mathbf{M}_i$ is a primitive matrix[6], we can apply the Perron-Frobenius theorem and conclude that there is a positive eigenvalue $\lambda_{\max}$ strictly greater, in magnitude, than all the other eigenvalues. Since matrix $\mathbf{M}$ is left-stochastic all eigenvalue lie within the unit disc and, thus, the fixed point is unique, if there is one. Such vector must validate (33), (34) and (35) for all i. It can be verified that the vector

$$\mathbf{\Delta}^* = \frac{\beta}{K}(K_1, \delta, K_2, \delta, \ldots, K_n, \delta)^T \tag{37}$$

satisfies (33), (34) and (35) for all $i \in \{1, 2, \ldots, n\}$. Since $||\mathbf{\Delta}||_1 = 1$, we have that $\sum_{i=1}^{n} \Delta_i^* = \frac{\beta}{K}\sum_{i=1}^{n} K_i + \frac{\beta}{K} = 1$. Therefore, the vector defined in (37) with $\beta = (1 + \frac{n\delta}{K})^{-1}$ is the (unique) fixed point of the system.

[6] Matrix $\mathbf{M}$ is primitive if $\exists k : \forall k' > k,\ \mathbf{M}^{k'} > 0$. Since matrices $\mathbf{M_i}$ have the 3-by-3 block shifted over the diagonal, the product $\mathbf{M} = \prod_{i=n}^{1} \mathbf{M}_i$ has positive elements on the upper-triangular part and on the last row and, thus, $\mathbf{M}^2 > 0$.

REFERENCES

[1] Bennett, Schatz, Rockwood, and Wiesenfeld, “Huygens clocks,” *Proc. R. Soc. Lond. A.*, vol. 458, pp. 563 – 579, 2002.

[2] C. Peskin, “Mathematical aspects of heart physiology,” *Courant Institute of Mathematical Sciences*, 1975.

[3] R. Mirollo and S. Strogatz, “Synchronization off pulse-coupled biological oscilators,” *SIAM J. Appl. Math.*, 1990.

[4] Y.-W. Hong and A. Scaglione, “A scalable synchronization protocol for large scale sensor networks and its applications,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 5, pp. 1085–1099, 2005.

[5] G. Werner-Allen, G. Tewari, A. Patel, R. Nagpal, and M. Welsh, “Firefly-inspired sensor network synchronicity with realistic radio effects,” in *SensysÕ05*, 2005.

[6] D. Lucarelli and I.-J. Wang, “Decentralized synchronization protocols with nearest neighbor communication,” in *Sensys*, 2004.

[7] A. Tyrrell, G. Auer, and C. Bettstetter, “Biologically inspired synchronization for wireless networks,” *Series: Studies in Computational Intelligence, Springer*, 2007.

[8] S. Barbarossa and G. Scutari, “Decentralized maximum likelihood estimation for sensor networks composed of nonlinearly coupled dynamical systems,” *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3456–3470, jul 2007.

[9] ——, “Bio-inspired sensor network design: Distributed decision through self-synchronization,” *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 26–35, may 2007.

[10] Y.-W. Hong and P. Varshney, *Advanced Signal Processing for Sensor MAC Protocols.* John Wiley & Sons, 2007.

[11] C. Raghavendra, K. Sivalingam, and T. Znati, *A Survey of MAC Protocols for Sensor Networks.* Springer US, 2004, ch. 2.

[12] I. Demirkol, C. Ersoy, and F. Alagoz, “Mac protocols for wireless sensor networks: A survey,” *IEEE Commun. Mag.*, 2006.

[13] L. Schwiebert, S. Gupta, J. Weinmann, A. Salhieh, V. Shankar, V. Annamalai, M. Kochhal, and G. Auner, “Research challenges in wireless networks of biomedical sensors,” in *MOBICOM*, 2001.

[14] R. Pagliari, Y.-W. P. Hong, and A. Scaglione, “Pulse coupled oscillators’ primitives for collision-free multiple access in body area networks,” in *ISABEL*, 2008.

[15] Y.-W. P. Hong, A. Scaglione, and R. Pagliari, “Pulse coupled oscillators’ primitives for low complexity scheduling,” in *ICASSP*, 2009.

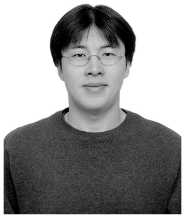
[16] A. Patel, J. Degesys, and R. Nagpal, “Desynchronization: The theory of self-organizing algorithms for round-robin scheduling,” in *SASO*, 2007.

[17] F. Hoppensteadt and E. Izhikevich, *Weakly connected neural networks.* Springer, 1997.

[18] J. Degesys, I. Rose, A. Patel, and R. Nagpal, “Desync: Self-organizing desynchronization and tdma on wireless sensor networks,” in *International Conference on Information Processing in Sensor Networks (IPSN)*, 2007.

[19] R. Pagliari, Y.-W. P. Hong, and A. Scaglione, “A low-complexity scheduling algorithm for proportional fairness in body area networks,” in *BodyNets*, 2009.

Roberto Pagliari received his M.Sc. degree in Electrical Engineering from the University of Parma, Italy, in 2005. Since 2007 he has been Ph.D. student in Electrical Engineering at Cornell University. His research focuses on theory and design of algorithms, with applications to distributed systems.



Yao Win Peter Hong received his B.S. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan, in 1999, and his Ph.D. degree in Electrical Engineering from Cornell University, Ithaca, NY, in 2005. He joined the Institute of Communications Engineering and the Department of Electrical Engineering at National Tsing Hua University, Hsinchu, Taiwan, in Fall 2005, where he is now an Associate Professor. He was also a visiting scholar at the University of Southern California during June-August of 2008. His research interests include cooperative communications, distributed signal processing for sensor networks, and PHY-MAC cross-layer designs for next generation wireless networks.

Dr. Hong received the best paper award for young authors from the IEEE IT/COM Society Taipei/Tainan chapter in 2005, the best paper award among unclassified papers in MILCOM 2005, and also the Junior Faculty Research Award from the College of EECS at National Tsing Hua University in 2009.

Anna Scaglione received the M.Sc. and the Ph.D. degrees in Electrical Engineering from the University of Rome La Sapienza, Rome, Italy, in 1995 and 1999, respectively. She is currently Associate Professor in Electrical Engineering at the University of California at Davis, CA. Prior to this she was postdoctoral researcher at the University of Minnesota, MN, in 1999-2000, Assistant Professor at the University of New Mexico, Albuquerque, NM, in 2000-2001, Assistant and Associate Professor at Cornell University, Ithaca, NY, in 2001-2006 and 2006-2008, respectively. She received the 2000 IEEE Signal Processing Transactions Best Paper Award, the NSF Career Award in 2002, the Fred Ellersick Award for the best unclassified paper in MILCOM 2005 and the 2005 Best paper for Young Authors of the Taiwan IEEE Comsoc/Information theory section. Her research is in the broad area of signal processing for communication systems. Her current research focuses on cooperative communication systems, decentralized processing for sensor networks, compressed sensing and body area networks.