

A Low-Complexity Scheduling Algorithm for Proportional Fairness in Body Area Networks

Roberto Pagliari
Dept. of Electrical and
Computer Engineering
Cornell University
rp294@cornell.edu

Yao-Win Peter Hong
Department of Electrical
Engineering
National Tsing Hua University
ywhong@ee.nthu.edu.tw

Anna Scaglione
Dept. of Electrical and
Computer Engineering
UC Davis
anna@ucdavis.edu

ABSTRACT

Pulse-coupled oscillator networks are a system of pulsing devices that alter their pulsing patterns in response to the pulsing events at other nodes. It is well known that these networks can produce a number of different pulsing patterns, among which the synchrony of pulsing. The primitive we study in this paper belongs to the class of desynchronization algorithms whose objective is collision resolution in networks with extremely low power devices, which also are bound to have very low complexity. In this paper, we introduce a new decentralized scheduling algorithm based on the coupled oscillator model to achieve a schedule that allows to multiplex in time a small network of very simple transmission devices, based on the requests negotiated by the nodes.

Keywords

pulse-coupled oscillators, TDMA, scheduling

1. INTRODUCTION

The design of multiple access for Wireless Body Area Networks (WBAN), should account for two basic application requirements: i) the sampling of biological signals occurs regularly, making rather inefficient the use of asynchronous Carrier Sensing Multiple Access (CSMA) protocols that are adopted in most commercial interfaces, such as for examples, 802.11, Bluetooth, and 802.15.4 [12]; ii) the signals are samples at heterogeneous rates that depend on the nature of the phenomenon under observation [8] (for example, motion sensors, ECG sensors, breathing sensors, blood glucose sensors etc.). The number of sensors used in different applications vary and may range from a handful of sensors to a few dozens.

Thus, in this context one would want a multiple access control (MAC) primitive capable of generating a regular Time Division Multiple Access (TDMA) schedule with a possibly variable number of nodes and satisfying heterogeneous bandwidth demands. Even if WBANs have naturally

a clustered structure, decentralized MAC protocols are often lightweight and adaptive. The advantage of CSMA is precisely its decentralized nature, that makes it suitable in applications where nodes are intermittent, the size of the network is unknown and the traffic demands are unequal. However, for periodically sampled data, TDMA is more efficient and there is no reason to employ an asynchronous medium access policy, if TDMA can be achieved in a simple decentralized fashion.

As observed by several Authors (see e.g. [12]), the specifications of Zigbee [3], Bluetooth [1], Mica Motes [2] and other commercial platforms are orders of magnitude above the limits imposed by many WBAN applications. In general, as we argued in [10], engineering WBANs can considerably benefit from employing an integrated system point of view in designing network primitives, based on novel and simple physical models for the communications interactions and for the computation of the network state.

The objective of this paper is the design and the analysis of a scheduling protocol aimed at achieving proportional fairness in dividing the bandwidth among the nodes of a WBAN. In contrast to classical scheduling protocols, our decentralized strategy is based on the emergent behavior of Pulse-Coupled Oscillators (PCO) networks driven by appropriate state dynamics, leading to a proportionally fair TDMA schedule among the active nodes. Pulse-coupled oscillators (PCO) denote pulsing nodes that alter their emission patterns in response to the pulsing events heard from other elements (hence the adjective coupled). The emission of a pulse is usually referred to as firing event, whose effect is to move earlier or later the recipients firing of the next pulse, by altering a local state that is regulating the next pulse emission of the node. PCO has been proposed first in [5] as a powerful primitive for wireless networks synchronization. By changing the update rule, PCOs can produce a variety of pulsing patterns in addition to synchrony [9]. The simplicity of the hardware interface required, the scalability and the low computational complexity of our protocol make it particularly suitable for biomedical sensor networks and brain-computer interfaces, which place extremely tight constraints on the size and power of the sensors employed.

Background

Describing the node transmissions as those of PCO elements, in [4], the authors found that it was possible to enforce a TDMA schedule in a decentralized fashion, assigning to each node $1/n$ -th of the frame duration, where n denotes the number of nodes in a fully connected network. More specifi-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. BodyNets '09 April 1-3, 2009 Los Angeles, CA, USA.
Copyright 2009 ICST 978-963-9799-41-7 ...5.00 .

cally, [4, 11] introduced two PCO protocols: 1) a scheduling method called *Desync* where the nodes adjust their firing time in each period based on two firing events of other sensors, i.e. the firing that occurred right before its own and the one that occurred right after it; 2) a protocol called *Inverse-MS* which is essentially the same as the PCO technique proposed for achieving synchrony in [5], except with a negative coupling. Building on these results, in [10, 6] we proposed two new PCO schemes to achieve TDMA distributedly with faster convergence compared to [4, 11].

The basic hypothesis in [4, 11, 10, 6] is that the pulses sent by the PCO elements have a very short duration compared to the transmission period and, therefore, do not collide. At the end of the protocol, the pulse emissions are equally spaced in the time period of the frame. While the protocol is running, the nodes do not know how much time is available for transmission before the next pulsing event and, unfortunately, none of the papers [4, 11, 10] explains how this self-organizing primitive should be used for multiplexing data transmissions in the same bandwidth without collisions. To overcome this limitation, in [6] we proposed an implementation of the PCO-MAC as a reservation-based MAC protocol, where each transmission frame is divided into a reservation (or scheduling) phase and a data transmission phase. However, this method requires frame synchronization, thereby increasing the complexity of the system.

Another drawback of the PCO state dynamics proposed in [4, 11, 10, 6] is that they lead to uniform TDMA scheduling only, where the same amount of time is allotted to each sensor's transmission.

Our main contribution in this work is the design of a bio-inspired decentralized medium access control (MAC) protocol based on the PCO primitive that achieves a proportional-fair schedule, as opposed to the uniform TDMA schedule, while enabling transmission of sensor data in the same bandwidth/time of the PCO protocol. The key difference of our protocol compared to the ones we cited above is that each PCO element emits two pulses, negotiating adaptively with the other nodes a suitable collision free interval between its own two pulses. By tracking two state variables per PCO node, we naturally increase the flexibility of the method and overcome the two limitations of the previous work in [4, 11, 10, 6] discussed above, with a modest increase in complexity.

In addition, our method readily defines the data transmission period as well as the guard time between sensors' transmissions representing the window of opportunity for new sensors to join the network without disrupting the collision-free transmission.

Clearly, our approach is a departure from classical MAC protocols, because: i) the desired TDMA scheduling can be achieved without a master-slave architecture, nor an absolute reference time; ii) regular data is exchanged during the agreement phase and it is not necessary to split the network activities into reservation and data transmission subframes. The class of algorithms we investigate provides powerful, yet very simple mechanisms to share resources, whose design can be easily implemented in energy efficient hardware. For energy scavenging sensors, the state variables can be chosen so as to ensure appropriate duty-cycles patterns, to save energy.

2. PULSE-COUPPLING BASED NETWORK SCHEDULING

Consider a network composed by n sensors, denoted by s_0, s_1, \dots, s_{n-1} , fully-connected through direct transmission links. We answer the following question: *Can TDMA be achieved without a master-node architecture, nor a common reference time?*

In the simplest model, each node has a local clock of period T_f , which we refer to as the firing cycle. The firing of the local clock causes the node to transmit a pulse before resetting the clock cycle and starting to count again until T_f . When in isolation this results in the periodic emission of pulses at the rate of $1/T_f$. However, in an interconnected network, the pulses are the tool of interaction among nodes. The firing time of a node, in fact, embeds the information of its local state. Therefore, the generic node i receiving a pulse from node j at time t , can use this information to change its local state accordingly. With coupling rules properly chosen, this process drives the network to attain desirable emergent properties. In mathematical terms, the generic i th node is characterized by a local variable $\Phi_i(t)$, whose time-evolution can be modeled as follows:

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

where $\Phi_i(t)$ is the phase of node i at time t (local time normalized w.r.t. T_f), and $\phi_i \in [0, 1)$ is the initial phase offset at time $t = 0$. When the clock reaches 1, node i fires a pulse $p(t)$ and then $\Phi_i(t)$ returns to 0. At the same time, the generic node j receiving the pulse, updates its phase variable $\Phi_j(t)$ according to

$$\Phi_j(t^+) = f\left(\tilde{\Delta}_{j1}(t), \tilde{\Delta}_{j2}(t), \dots, \tilde{\Delta}_{jN}(t)\right) \quad (1)$$

where $\tilde{\Delta}_{jk}$ indicates the phase difference between nodes j and k , estimated by node j . Different algorithms have been proposed, and they differ because of: i) the updating rule (represented by the function $f(\cdot)$ in (1)), ii) the updating time (when to update), and iii) the number of nodes performing the update, when a given node fires (the network topology).

In the Inverse-MS [4], for example, when a node emits a message, the others update their phases according to the following model:

$$\Phi_j(t^+) = f(\Delta_{ji}(t)) = (1 - \alpha)\Phi_j(t), \forall j \neq i$$

assuming that i is firing at time t . This mechanism drives all the nodes to *weak convergence* [4], in the sense that the phase differences converge to T_f/n although the firing times continuously change drifting forward. In contrast to Inverse-MS, in the protocol proposed in [6] when node i fires, only the nodes that are set to fire within the interval $(T_f - T_f/n, T_f)$ are allowed to perform the update. In this case, the updating function is non-linear, but it preserves the ordering of the nodes and can achieve strong convergence, i.e. the firing times settle to a periodic pattern, where nodes space uniformly their firing epochs. In the *Desync* protocol in [4] and in the method introduced in [10], when node i fires, only the node with the smallest phase updates its state.

The important point to emphasize is that by properly choosing the parameters, the algorithms proposed in [4, 6, 10] achieve *strong* synchronization, i.e., the phase difference between any two consecutive nodes converges to T_f/n ,

which is the sought TDMA schedule, and the firing times converge to a constant value modulo T_f . Therefore, PCO or consensus-based primitives can be employed to perform round-robin activities in a network according to a uniform TDMA schedule in a distributed, self-organizing, adaptive (adds or removals of nodes are easily handled), and simple (low-complexity) fashion, without requiring an absolute time reference.

The question we want to ask is whether it is possible to assign different portions of the period T_f to different nodes non uniformly. This is equivalent to say that the nodes have different requests, and they are willing to agree, by interacting, on the portions of time assigned to them in a "fair" manner, as we will explain in the following.

3. A PULSE-COUPLED PRIMITIVE FOR PROPORTIONAL FAIRNESS

We assume that each of the sensors s_i has a specific request, K_i , which is the portion of the period T_f the node is willing to obtain. K_i is an integer, for simplicity, within a specified interval $[1, K_{\max}]$. For example, if $K_{\max} = 10$, $K_i = 5$ means that node i is requesting the 50% of the period T_f .

In general, the total amount of the requests $K = \sum_{i=0}^{n-1} K_i$ may exceed the available resource T_f , and, therefore, the fair solution of the problem is, for each node, to obtain a portion of the period equal to:

$$\frac{K_i}{\sum_{i=1}^N K_i} T_f = \frac{K_i}{K} T_f. \quad (2)$$

Achieving *proportional fairness* for us means to allow the nodes to hold on to a portion of the frame of size equal to (2) that preserves the ratio of the requests. For example, if all the nodes request the 50% of the period, then, it makes sense that they will obtain a portion of time equal to T_f/n . Instead, if a node requests the 80% of T_f and all the remaining nodes request the 20% of T_f , then, we would like the former to get a portion of time which is four times bigger than the others. Our main objective is to obtain proportional fairness as an emergent PCO behavior, without explicit knowledge neither of the total number of active nodes nor of their requests K_i .

In order to obtain proportional fairness, we endow each node with two state variables, namely, $\Psi_i(t)$ and $\Phi_i(t)$. Each PCO in our model fires twice, when each of the state variables reaches 1. In a nutshell, the node tries progressively to separate the firing times that correspond to these two state variables so that the time between them is equal to K_i/K , where $K = \sum K_i$. This can be obtained as follows.

Initially, each node has $\Psi_i(0) = \Phi_i(0) = \Omega_i$, with Ω_i a random number in $[0, 1)$. While nobody is firing, the two phases evolve as:

$$\begin{cases} \Psi_i(t) &= (\Omega_i + t/T) \bmod 1 \\ \Phi_i(t) &= (\Omega_i + t/T) \bmod 1. \end{cases} \quad (3)$$

If no firings occur, the node is free to utilize the entire frame for its own transmission, leaving a small gap of time in the order of the pulse duration to allow other nodes to join the network. Updates occur whenever there is a firing event caused by one node having its $\Phi_j(t) = 1$, which in turn causes the two phase variables of other nodes to spread

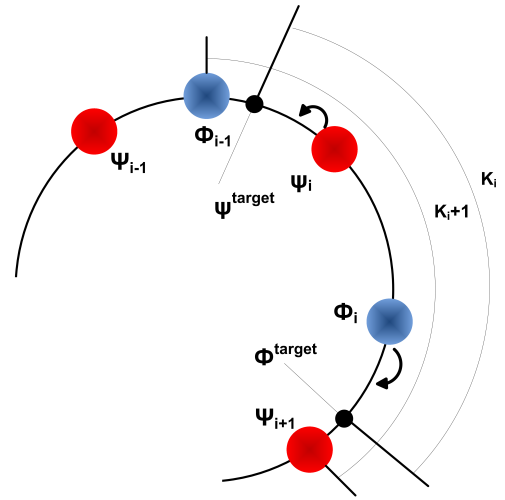


Figure 1: Update of node i due to the firing of the phase variable Φ_{i-1} of node $i-1$ at time t_2 .

apart. The nodes will use the gap created between its two firing times as its own collision free interval for transmission.

In the following, the indices are going to be ordered based on the order of the nodes' firing times; the indices are in descending order, so that node $i+1$ fires before node i , which fires before node $i-1$ and so on. Upon hearing the signals emitted by nodes $i+1$, node i records the firing time t_1 at which $\Psi_{i+1}(t_1) = 1$. Then, node i fires its two pulses. After that, upon hearing the signals emitted by node $i-1$, node i records the firing time t_2 at which $\Phi_{i-1}(t_2) = 1$ and updates its state. Specifically, the update at time t_2 of the two state variables of node i are aimed at making the difference $\Phi_i^{\text{target}}(t_2) - \Psi_i^{\text{target}}(t_2)$ as close as possible to K_i/K_{sum} . We will show that this can be obtained through the following update (see Figure 1):

$$\begin{cases} \Psi_i(t_2^+) &= \alpha \Psi_i^{\text{target}}(t_2) + (1 - \alpha) \Psi_i(t_2) \\ \Phi_i(t_2^+) &= \alpha \Phi_i^{\text{target}}(t_2) + (1 - \alpha) \Phi_i(t_2) \end{cases} \quad (4)$$

where $\alpha \in (0, 1)$ and

$$\begin{aligned} \Psi_i^{\text{target}}(t_2) &= \Phi_{i-1}(t_2) + \frac{1}{2} \frac{\Psi_{i+1}(t_2) - \Phi_{i-1}(t_2)}{K_i + 1} \\ &= \frac{1/2}{K_i + 1} \Psi_{i+1}(t_2) \\ \Phi_i^{\text{target}}(t_2) &= \Phi_{i-1}(t_2) + \frac{K_i + 1/2}{K_i + 1} (\Psi_{i+1}(t_2) - \Phi_{i-1}(t_2)) \\ &= \frac{K_i + 1/2}{K_i + 1} \Psi_{i+1}(t_2). \end{aligned}$$

In the update we assume that node i has access to the state of node $i+1$ at time t_2 .

Remark 1: Note that node i has only access to the state of node $i+1$ at time t_1 . Therefore, it only has an outdated estimate of $\tilde{\Psi}_{i+1}$ of $\Psi_{i+1}(t_2)$. This is similar to what was assumed in analyzing the Desynch protocol in [4]. Since the nodes transmit information during the interval between their firing events, which are marked by their two state variables, Φ_i and Ψ_i , to guarantee the absence of collisions in our setup, it is necessary to preserve the alternating order

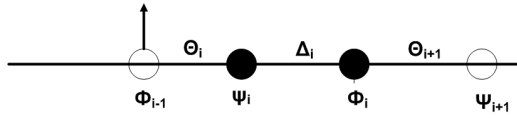


Figure 2: The state variable during the update of node i .

of the state variables, $\dots > \Phi_{i+1}(t) > \Psi_{i+1}(t) > \Phi_i(t) > \Psi_i(t) > \Phi_{i-1}(t) \dots$. As a result, these intervals cannot overlap. The main problem with using outdated estimates is that in a non-steady state condition it is possible to have $\Psi_{i+1}(t_2^+) < \Phi_i(t_2^+)$ and, therefore, having overlapping intervals which in turn cause collisions. However, in Section 4 we will show how to modify the dynamics in (4) so as to avoid clocks overlapping. We also discuss why using an estimate $\tilde{\Psi}_{i+1}$ in lieu of $\Psi_{i+1}(t_2)$ does not affect the convergence. The next theorem proves that the dynamics in (4) converge to the desired proportional fair schedule. More specifically, we prove that the time gap $\Delta_i(t) = \Phi_i(t) - \Psi_i(t)$ between the two state variables of each node converges to $\beta \frac{K_i}{K} T_f$, where $\beta = 1/(1 + n/(2K))$.

THEOREM: 1. *For any initial condition $(\Omega_1, \Omega_2, \dots, \Omega_n)$, algorithm (4) converges. Furthermore, let $\Delta_i(t) = \Phi_i(t) - \Psi_i(t)$, $\Theta_i(t) = \Psi_i(t) - \Phi_{i-1}(t)$, and $\beta = 1/(1 + n/(2K))$. Then, $\lim_{t \rightarrow \infty} \Delta_i(t) = \beta K_i/K$ and $\lim_{t \rightarrow \infty} \Theta_i(t) = \beta/(2K)$, for all i .*

PROOF. To aid our proof, in Figure 2 we show the state variables as points over the circle with perimeter T_f . The firing events occur in counter-clockwise order. The situation described is the update of node i , according to (4). In the following we omit the time dependency.

The update made by node i when node $i-1$ fires, modifies variables Θ_i , Δ_i and Θ_{i+1} as follows:

$$\begin{aligned} \Theta'_i &= \Psi'_i = \frac{1}{2} \frac{\alpha}{K_i + 1} (\Theta_i + \Delta_i + \Theta_{i+1}) + (1 - \alpha)\Theta_i \quad (5) \\ &= \left[1 - \alpha \frac{K_i + 1/2}{K_i + 1} \right] \Theta_i + \frac{1}{2} \frac{\alpha}{K_i + 1} \Delta_i + \frac{1}{2} \frac{\alpha}{K_i + 1} \Theta_{i+1} \end{aligned}$$

$$\begin{aligned} \Delta'_i &= \Phi'_i - \Psi'_i \quad (6) \\ &= \alpha \frac{K_i}{K_i + 1} \Theta_i + \left[1 - \frac{\alpha}{K_i + 1} \right] \Delta_i + \alpha \frac{K_i}{K_i + 1} \Theta_{i+1} \end{aligned}$$

$$\begin{aligned} \Theta'_{i+1} &= \Psi_{i+1} - \Phi'_i = \Psi_{i+1} - (\Theta'_i + \Delta'_i) \quad (7) \\ &= \frac{1}{2} \frac{\alpha}{K_i + 1} \Theta_i + \frac{1}{2} \frac{\alpha}{K_i + 1} \Delta_i + \left[1 - \alpha \frac{K_i + 1/2}{K_i + 1} \right] \Theta_{i+1} \end{aligned}$$

while all the other variables remain the same.

Based on our indices order, the first who performs the update is node n , followed by $n-1$ and so forth. Given the initial condition provided by the vector

$$\begin{aligned} \mathbf{x}[0] &= (\Delta_1[0], \Theta_2[0], \Delta_2[0], \Theta_3[0], \dots, \Delta_n[0], \Theta_1[0]) \\ &= (0, \Omega_2 - \Omega_1, \dots, \Omega_n - \Omega_1) \end{aligned}$$

the evolution of the system can be described as:

$$\begin{aligned} \mathbf{x}_{\text{inf}} &= \dots \mathbf{M}_{n-1} \mathbf{M}_n \mathbf{M}_1 \mathbf{M}_2 \dots \mathbf{M}_{n-1} \mathbf{M}_n \mathbf{x}[0] \\ &= \lim_{k \rightarrow \infty} \mathbf{M}^k \mathbf{x}[0] = \mathbf{M}_{\infty} \mathbf{x}[0] \quad (8) \end{aligned}$$

where $\mathbf{M} = \prod_{i=1}^n \mathbf{M}_i$ and $\mathbf{M}_{\infty} = \lim_{k \rightarrow \infty} \mathbf{M}^k$. The state of the network, at iteration k , can be characterized by a vector, namely, \mathbf{x} , whose components are given by the variables Δ_i and Θ_i . Thus, we define the state vector at iteration k as $\mathbf{x}[k] = (\Delta_1[k], \Theta_2[k], \Delta_2[k], \Theta_3[k], \dots, \Delta_n[k], \Theta_1[k])$. In order to achieve proportional fairness, in the limit for $k \rightarrow \infty$ $\mathbf{x}[k]$ to converge to

$$\mathbf{x}_{\text{inf}} = (K_1/K, 0, \dots, K_n/K, 0)$$

that corresponds to the case where the generic node i uses a portion of time equal to K_i/K of the frame to transmit data. Since $\sum K_i/K = 1$, with $\Theta_i = 0, \forall i$ there would not be any remaining empty interval to accommodate external nodes willing to join the network. Thus, a preferable fixed point for (8) is

$$\mathbf{x}_{\text{inf}} = \beta(K_1/K, \Theta_{2,\text{inf}}, K_2/K, \dots, K_n/K, \Theta_{1,\text{inf}})$$

where $\Theta_{i,\text{inf}} = \lim_{t \rightarrow \infty} \Theta_i(t)$, and $\sum x_{i,\text{inf}} = 1$ (assuming $T_f = 1$). If $\Theta_{i,\text{inf}} > 0, \forall i$, external nodes will have room to join the network.

Consider equation (8). Each \mathbf{M}_i is a full-rank matrix composed of a 3-by-3 positive block at position $2i-1$, with coefficients given by (5)-(6)-(7), while the rest is 1 on the diagonal and 0 elsewhere. A finite product of full-rank matrices is full-rank, thus, \mathbf{M} is full-rank. In particular, it is not hard to see that \mathbf{M}_i is left-stochastic for all i , and, therefore, \mathbf{M} is left-stochastic. $\mathbf{M} = \prod_{i=1}^n \mathbf{M}_i$ in Eq.(8) is, in fact, a non-negative primitive matrix (see Appendix). Therefore, we can conclude that there is a positive eigenvalue $\lambda_{\text{max}} = 1$ strictly greater, in magnitude, than all the other eigenvalues, with eigenvector \mathbf{w} . This eigenvalue is also eigenvalue of \mathbf{M}_{∞} , corresponding to the eigenvector \mathbf{w} (unique), because $\mathbf{M}\mathbf{w} = \mathbf{w}$ implies $\mathbf{M}^k \mathbf{w} = \mathbf{w}$, for any $k > 0$. Therefore, $\lim_{k \rightarrow \infty} \mathbf{M}^k \mathbf{w} = \mathbf{M}_{\infty} \mathbf{w} = \mathbf{w}$. Hence, \mathbf{w} is fixed point of \mathbf{M} , as well as of \mathbf{M}_{∞} and what is left for us to show is that \mathbf{w} has the desired structure \mathbf{x}_{inf} . Consider the vector:

$$\mathbf{w}_i = [w_1, w_2, \dots, 1/(2K), K_i/K, 1/(2K), \dots, w_{2n-1}, w_{2n}]^T. \quad (9)$$

We have that $\mathbf{M}_i \mathbf{w}_i = \mathbf{w}_i, \forall i$. In fact, \mathbf{M}_i has a positive 3×3 block centered at $2i-1, 2i-1$, which modifies entries $(2i-2, 2i-1, 2i)$ of \mathbf{w}_i , while the remaining entries remain constant, because the rest of the matrix is equal to one on the diagonal and zero elsewhere. By equation (5), we have

$$\left(1 - \alpha \frac{K_i + 1/2}{K_i + 1} \right) \frac{1}{2K} + \frac{1}{2} \frac{\alpha}{K_i + 1} \frac{K_i}{K} + \frac{1}{2} \frac{\alpha}{K_i + 1} \frac{1}{2K} = \frac{1}{2K}$$

and, by Equation (6) (similarly for (7))

$$\alpha \frac{K_i}{K_i + 1} \frac{1}{2K} + \left(1 - \frac{\alpha}{K_i + 1} \right) \frac{K_i}{K} + \alpha \frac{K_i}{K_i + 1} \frac{1}{2K} = \frac{K_i}{K}.$$

If \mathbf{w} is eigenvector of $\mathbf{M} = \prod \mathbf{M}_i$, then \mathbf{w} is eigenvector for any \mathbf{M}_i , or, in other words, \mathbf{w} is the intersection of the eigenvectors of matrices \mathbf{M}_i , of the form given in (9). Thus,

$$\mathbf{w} = [K_1/K, 1/(2K), \dots, K_n/K, 1/(2K)]^T \quad (10)$$

where $K = \sum K_i$. Because \mathbf{M} is nonnegative primitive and stochastic, from the Frobenius-Theorem [7] we know that $\mathbf{M}_{\infty} = \lim_{k \rightarrow \infty} \mathbf{M}^k$ has eigenvalue 1, while the others are strictly less than 1 in absolute value. Moreover, from the same theorem, the (positive) eigenvector corresponding to eigenvalue 1 is unique, up to an amplification factor (if \mathbf{w} is eigenvector, so is $\beta \mathbf{w}$). Since we have shown that \mathbf{w} is the

eigenvector of \mathbf{M}_∞ with eigenvalue 1, then \mathbf{w} is the (unique) fixed point of the system.

We know from the theorem that row i of \mathbf{M}_∞ is composed of elements equal to the i -th entry of \mathbf{w}_{norm} . For any initial condition $\mathbf{x}[0]$ such that $\sum x_i[0] = 1$, with $x_i[0] > 0, \forall i$, and $\mathbf{M}_\infty = \beta[w_1 \mathbf{1}^T; w_2 \mathbf{1}^T; \dots; w_{2n} \mathbf{1}^T]$, we have

$$\begin{aligned} \lim_{k \rightarrow \infty} \mathbf{M}^k \mathbf{x}[0] &= \mathbf{M}_\infty \mathbf{x}[0] \\ &= \beta[w_1 \mathbf{1}^T; w_2 \mathbf{1}^T; \dots; w_{2n} \mathbf{1}^T] \mathbf{x}[0] \\ &= \beta[K_1/K, 1/(2K), \dots, K_n/K, 1/(2K)]^T \end{aligned}$$

because $\sum x_i[0] = 1$. Since $\sum x_i[0] = 1$, then $\beta \sum w_i = 1$ and,

$$\beta \sum w_i = \beta \left(\frac{n}{2K} + 1 \right) = 1 \Rightarrow \beta = \frac{1}{1 + \frac{n}{2K}}. \quad (11)$$

Therefore, Algorithm (4) converges to the (unique) fixed point given by $\mathbf{w}_{\text{norm}} = \beta[K_1/K, 1/(2K), \dots, K_n/K, 1/(2K)]$, which corresponds to the desired proportional fairness scheduling policy. \square

As we have shown in Equation (11), the portion of time node i obtains by iteration, goes to $\beta K_i/K$, as the time evolves. Obviously we would like to have $\beta = 1$, which corresponds to having no empty spaces between different nodes, and this happens when $n/(2K) \rightarrow 0$. However, as we will see in the next section, in order to allow external nodes to join the network, β must be less than 1 and there is a trade-off between the number of nodes and K . In any case, the ratios between the requests are preserved, i.e., the ratio between $\beta K_i/K$ and $\beta K_j/K$ is equal to $K_i/K_j, \forall i, j, i \neq j$. Finally, notice that $\beta = \frac{1}{1 + \frac{n}{2K}} > \frac{1}{1 + 1/2} = 2/3$.

4. CROSS-LAYER SCHEDULING AND DATA TRANSMISSION

As we mentioned before, the order of nodes' clocks needs to be preserved so that the nodes can transmit without collisions in the interval gap between their own firing events. This allows to use the bandwidth as the schedule is computed and it has also the benefit of making unnecessary to identify explicitly the node that it is firing. Thus, we need to avoid the overlapping of intervals that are between the two firing events of different nodes. In Remark 1 in Section 3 we noted that, even if node i can only obtain an estimate of Ψ_{i+1} , the value $(\tilde{\Psi}_{i+1} + \Phi_i)/2$ is the true value, since it was obtained when node $i+1$ fired, hence, before the update of both node i and $i+1$.

To ensure that this constraint is met at all times, one can modify the update equations as follows:

$$\begin{cases} \Psi_i(t_2^+) &= \alpha \max(\Psi_i^{\text{target}}(t_2), \Psi_i/2) + (1 - \alpha) \Psi_i(t_2) \\ \Phi_i(t_2^+) &= \alpha \min(\Phi_i^{\text{target}}(t_2), (\Psi_{i+1} + \Phi_i)/2) \\ &+ (1 - \alpha) \Phi_i(t_2). \end{cases} \quad (12)$$

To see that the model in (12), intuitively, converges, let us make some observations. For the dynamics in (4) Theorem 1 showed that the fixed point is given by \mathbf{w}_{norm} . Therefore, when node $i-1$ fires, as shown in Figure 2, we have that $\Phi_{i-1} = 0$, $\Phi_i - \Psi_i = \beta K_i/K$, and $\Psi_i - \Phi_{i-1} = \Psi_{i+1} - \Phi_i = \beta/(2K)$. Now, if we analyze the system in (12), we see that $\max(\Psi_i^{\text{target}}, \Psi_i/2) = \Psi_i^{\text{target}}$ and $\min(\Phi_i^{\text{target}}, (\Psi_{i+1} +$

$\Phi_i)/2) = \Phi_i^{\text{target}}$ (where we omitted the time indices for brevity). In fact,

$$\begin{aligned} \Psi_i^{\text{target}} &= \beta \frac{1/2}{K_i + 1} \left(\frac{\beta}{K} + \beta \frac{K_i}{K} \right) = \frac{\beta}{2K} \frac{K_i + 1}{K_i + 1} = \frac{\beta}{2K} \\ &> \Psi_i/2 = \frac{\beta}{4K} \end{aligned}$$

and

$$\begin{aligned} \Phi_i^{\text{target}} &= \frac{K_i + 1/2}{K_i + 1} \left(\frac{\beta}{K} + \beta \frac{K_i}{K} \right) = \frac{\beta}{K} (K_i + 1/2) \\ &< (\Psi_{i+1} + \Phi_i)/2 = \frac{\beta}{K} (K_i + 3/4). \end{aligned}$$

Therefore we have

$$\begin{aligned} \Psi(t_2^+) &= \alpha \Psi(t_2) + (1 - \alpha) \Psi(t_2) = \Psi(t_2) \\ \Phi(t_2^+) - \Psi(t_2^+) &= (1 - \alpha) \beta \frac{K_i}{K} + \alpha \left(\frac{\beta}{K} (K_i + 1/2) - \frac{\beta}{2K} \right) \\ &= \beta \frac{K_i}{K}. \end{aligned}$$

Thus, \mathbf{w}_{norm} is also a fixed point of (12). Could there be other fixed points? The answer is no. In fact, consider the case where $\Phi_i - \Psi_i = \gamma \neq \beta K_i/K$, and $\Psi_i - \Phi_{i-1} = \Psi_{i+1} - \Phi_i = \gamma_0$. We could have $\max(\Psi_i^{\text{target}}, \Psi_i/2) \neq \Psi_i^{\text{target}}$ or $\min(\Phi_i^{\text{target}}, (\Psi_{i+1} + \Phi_i)/2) \neq \Phi_i^{\text{target}}$ or both.

Suppose, for instance, that $\max(\Psi_i^{\text{target}}, \Psi_i/2) = \Psi_i/2$ and $\min(\Phi_i^{\text{target}}, (\Psi_{i+1} + \Phi_i)/2) = (\Psi_{i+1} + \Phi_i)/2$. We would have

$$\begin{aligned} \Phi_i(t_2^+) - \Psi_i(t_2^+) &= (1 - \alpha) \gamma + \alpha \left(\frac{\Psi_{i+1} + \Phi_i}{2} - \frac{\Psi_i}{2} \right) \\ &= (1 - \alpha) \gamma + \alpha \frac{\Psi_{i+1} + \gamma}{2} = \gamma \quad (13) \end{aligned}$$

which is not true for any i and Ψ_{i+1} . Moreover, we have $\Psi_i(t_2^+) = \alpha \frac{\Psi_i(t_2)}{2} + (1 - \alpha) \Psi_i(t_2) = (1 - \alpha/2) \Psi_i(t_2) \neq \Psi_i(t_2)$. We can make similar observations for the other two cases. In conclusion, model (12) has a unique fixed point, which is given by \mathbf{w}_{norm} . If we look at (12), we see that the dynamics the system follows, is equivalent to (4). The only difference is that, in this case, the clocks of node i are bounded by $\Psi_i/2$ and $(\Phi_i + \Psi_{i+1})/2$, but the behavior is the same, as well as the fixed point. Therefore, we provided an explanation on why model (12) converges, while preserving the interval gap between the two firings of each node as a collision free slot for the node transmission. This is further detailed in the next section.

4.1 The Proposed Scheduling Algorithm

Based on the two state PCO model described above we are now ready to introduce an algorithm that combines both the scheduling of the network activities with the transmissions of data in the same bandwidth adaptively. The algorithm is able to accommodate new nodes willing to join the network, and re-arrange the system when nodes already in the network leave and the adaptation occurs as the nodes continue to transmit their data. Furthermore nodes do not use a common time reference as in reservation protocols, which increases the system complexity. In our proposed MAC protocol each node, say node i , follows the dynamics in (4). Suppose that, at iteration k , the firing times of node i , namely Φ_i and Ψ_i , are $t_\Phi[i, k]$ and $t_\Psi[i, k]$ respectively. At time $t_\Phi[i, k]$ node i fires, i.e., he emits a pulse. Within the

interval $(t_\Phi[i, k], t_\Psi[i, k])$, node i is allowed to transmit regular data. At time $t_\Psi[i, k]$, node i stops transmitting data and delivers his second firing. The problem of this scheme is that it is not easy for the nodes to distinguish between short pulses and regular data traffic, and this introduces additional complexity on the implementation of physical devices. To overcome this difficulty, we stipulate that node i transmits data between its own firing times. If so, the channel transitions IDLE->BUSY, and BUSY->IDLE, identify the firing times $\Phi_i(t) = 1$ and $\Psi_i(t) = 1$, respectively, of node i . Thus, in our protocol, node i transmits data between the firings of $\Phi_i(t)$ and $\Psi_i(t)$. While not transmitting, node i can interpret as firing events the transitions of the channel from IDLE to BUSY or viceversa, and identify the $\Phi_j(t)$ and $\Psi_j(t)$ firing events of the other nodes. Alternatively, a unique signature can be assigned to be the PCO pulse.

An external node willing to join the network can identify an interval with no transmissions (whose beginning is identified by a BUSY->IDLE channel state transition) to begin negotiating with other nodes some space. Hence, the incoming node only needs to realize when transmissions stop, at which point it can perform its first firing according to (3). On the other hand, if a node leaves the network, the others will start eroding the interval of time left empty, and they will redistribute the resource according to their requests.

As we have seen so far, the fixed point of the system (10) is such that the time elapsed between Φ_i and Ψ_i converges to $\beta(K_i/K)T$. On the other hand, the time difference between two consecutive firings of different nodes, $\Psi_i - \Phi_{i-1}$ goes to $\beta(1/(2K))T$. Therefore, in order for the network to host new nodes we need the quantity $\beta(1/(2K))T$ to be greater than the duration of the message an external node delivers to join the network. Hence, if δ is the duration of this message, we must satisfy the condition $\beta \frac{T}{2K} > \delta$, or $K < \frac{\beta T/\delta}{2}$. We have then $K = \sum K_i \leq nK_{\max} < \beta \frac{T_f/\delta}{2}$ and, thus,

$$n \leq \beta \frac{T_f/\delta}{2K_{\max}} \leq \frac{3}{4} \frac{T_f/\delta}{K_{\max}} = n_{\max}.$$

n_{\max} is the maximum number of nodes a network can handle, in the sense that if $n = n_{\max}$ no external nodes can join the network, unless some of them leave the system. If, for example, $T = 1s$, $\delta = 1ms$, and $K_{\max} = 10$, then for $n > n_{\max} = 70$, no external nodes can join the network. *We cannot have the cake and eat it too*: as long as we wish to allow new nodes to join the network, we need to have periods of time with no transmission, and therefore, there is a trade-off between the total bandwidth that can be used and the number of nodes a network may accommodate. n_{\max} is a rough estimate, since variables Θ_i may be less $\beta T_f/(2K)$ if the network has not reached the steady-state yet. However, n_{\max} gives an idea on the maximum number of nodes the network can handle.

A simple event-driven algorithms that implements our protocol is given in Al. 1. The SFD signal event is the start of frame delimiter, which indicates the beginning of the stream from another node. When SFD is set to low-high it indicates the transition of the channel from busy to idle; when set to high-low it indicates a transition of the channel from busy to idle. Those events identify variables $\Phi_j(t)$ and $\Psi_j(t)$, for some external node j . Alternatively, the nodes can explicitly indicate their firing times with a dedicated signature waveform that delimits their slot. Between the two local firings of node i , the node transmits data, while

the "enable" flag is used to determine when to perform the update.

Algorithm 1 Event-driven description of our proportional fairness scheme.

1. *SFD Interrupt*
 2. **if**(SFD_Interrupt == high-low)
 3. next = 1 - Φ_i ; SFD_Interrupt == low-high;
 4. **else**
 5. SFD_Interrupt == high-low;
 6. **if**(~enabled)
 7. **return**;
 8. (Ψ_i, Φ_i) = update($\Psi_i, \Phi_i + next$)
 9. set_timer_Ψ; set_timer_Φ; enabled = 0;
 10. *Timer_Φ_tick*
 11. start_stream; set_Φ(t + T);
 12. *Timer_Ψ_tick*
 13. stop_stream; set_Ψ(t + T); enabled = 1;
-

In algorithm Al.2, is shown the procedure for joining the network. The incoming node waits until it senses that there is no signal in the channel, which means the last firing event $\Psi_i = 1$ happened, for some i . Then, the node emits a pulse, and begins performing algorithm Al.1.

Algorithm 2 Join procedure for an external node.

1. set SFD_Interrupt = high-low;
 2. *SFD Interrupt*
 3. start_stream(δ); start_algorithm 1;
-

5. NUMERICAL RESULTS

In Figure 3 is reported the evolution of variables Φ_i and Ψ_i for each node, with respect to Ψ_1 , when $\Psi_1 = 0$. In this case, the requests of the nodes are given by the vector $\mathbf{K} = [10, 10, 4, 4, 2]$. As we can see, nodes 1 and 2, requesting $K_1 = K_2 = 10$, obtain the biggest portion of the period, and these portions are equal. Notice that the ratio of the intervals are the same w.r.t. the ratios of the requests K_i s.

In Figure 4 is reported the evolution of variables Φ_i and Ψ_i for each node when the requests are $\mathbf{K} = [10, 10, 10, 10, 10]$. As expected, the nodes obtain an equal amount of the total period T_f . We would like remark that the intervals (Φ_i, Ψ_{i+1}) are different in Figures 3 and 4. This is because

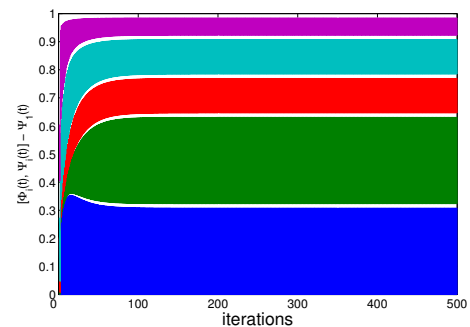


Figure 3: The colored area is the elapsed time between Φ_i and Ψ_i . $\mathbf{K} = [10, 10, 4, 4, 2]$.

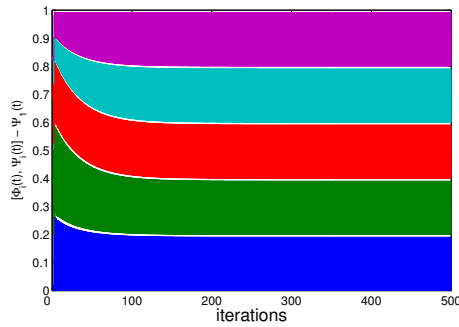


Figure 4: The colored area is the elapsed time between Φ_i and Ψ_i . $\mathbf{K} = [10, 10, 10, 10, 10]$.

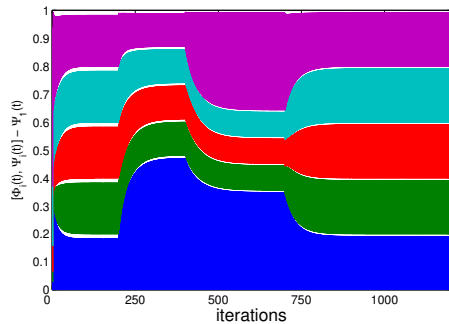


Figure 5: Evolution of Φ_i and Ψ_i , with respect to Ψ_1 , when the objective vector \mathbf{K} changes over the time.

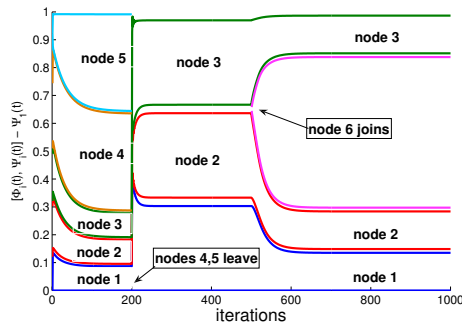


Figure 6: Evolution of Φ_i and Ψ_i , with respect to Ψ_1 , when nodes join and leave the network.

variables Θ_i converge to $1/(2K)$, which is different in the two cases.

In Figure 5 the nodes change their request as the time evolves. At the beginning the requests of the nodes are given by the vector $\mathbf{K} = [5, 5, 5, 5, 5]$. At iteration 200 node 1 changes its demand to $K_1 = 20$. At time 400 node 5 changes its demand to $K_5 = 20$. Finally, at iteration 700, the requests vector becomes $\mathbf{K} = [10, 10, 10, 10, 10]$. We can observe that the network adaptively changes the portion of time for each sensor, according to the actual requests.

In Figure 6 we allow nodes to leave or enter the network. The initial configuration of the network is characterized by 5 nodes, whose requests are $\mathbf{K} = [5, 5, 5, 20, 20]$. At iteration

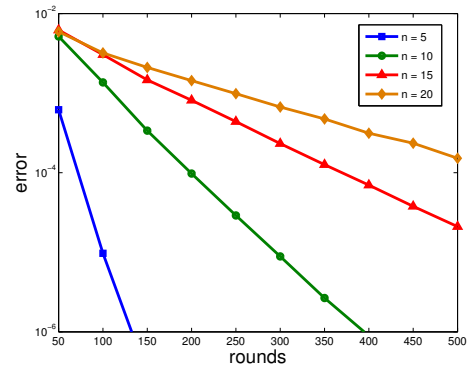


Figure 7: Error as a function of the number of iterations, for different values of N .

200, nodes 4 and 5 leave the network. Therefore, nodes 1, 2 and 3 remain, and we can see that they share the period T_f equally, since $K_1 = K_2 = K_3 = 5$. At iteration 500 a new node, node 6, joins the network, and its demand is $K_6 = 20$. The final state of the network is then given by 4 nodes: node 6 obtains the biggest amount of time, while the rest is equally divided by the remaining nodes, node 1, 2 and 3.

In order to evaluate the performance of the proposed protocol, we consider the following metric

$$\text{error} = E \{ \|\mathbf{x}[L] - \mathbf{w}_{\text{norm}}\|^2 \}$$

where $\mathbf{x}[L] = (\Delta_1, \Theta_1, \dots, \Delta_n, \Theta_n)[L]$ is the state of the network after a fixed number of updates equal to L .

In Figure 7 is reported the error, defined above, as a function of the number of iterations, for different values of the number of nodes N and random initial conditions. As we can see, the error asymptotically converges to zero, and the speed of convergence decreases with the number of nodes, as expected.

6. CONCLUSIONS

In this paper we introduced a bio-inspired algorithm to schedule the access to the wireless medium. This scheme is adaptive, serves heterogeneous traffic, does not require an absolute time-reference, and is fully-decentralized. We provided a proof of convergence of the proposed algorithm, and derived an implementation for event-driven embedded software architectures, such as TinyOS. Finally, we provided performance results, in terms of the achievable accuracy, as a function of the number of iterations and the number of nodes involved.

The results we provided indicate that there are promising alternatives to research in the class of PCO MAC protocols that could lead to self-organizing schemes for collision-free multiple access. Future work will include the practical implementation of the PCO-MAC protocol we proposed.

7. REFERENCES

- [1] Bluetooth specification. In *SIG Bluetooth* - <http://www.bluetooth.com>.
- [2] www.xbow.com.
- [3] Zigbee specification. In *ZB Alliance - ZigBee Document 053474r06*, 2004.

- [4] J. Degeysys, 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)*, April 2007.
- [5] Y.-W. Hong and A. Scaglione. A scalable synchronization protocol for large scale sensor networks and its applications. *IEEE Journal on Selected Areas in Communications*, 23(5), 2005.
- [6] Y.-W. Hong, A. Scaglione, and R. Pagliari. Pulse coupled oscillators primitive for low complexity scheduling. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (Submitted)*, April 2009.
- [7] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge Univ Pr, 1990.
- [8] E. Jovanov. A survey of power efficient technologies for wireless body area networks. In *Proc. of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Vancouver, Canada, 2008.
- [9] R. Mirollo and S. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM J. Appl. Math.*, 50:1645–1662, 1998.
- [10] R. Pagliari, Y.-W. P. Hong, and A. Scaglione. Pulse coupled oscillators' primitives for collision-free multiple access in body area networks. In *First International Symposium on Applied Sciences in Bio-Medical and Communication Technologies, ISABEL 2008*, October 2008.
- [11] A. Patel, J. Degeysys, and R. Nagpal. Desynchronization: The theory of self-organizing algorithms for round-robin scheduling. In *IEEE Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, July 2007.
- [12] L. Schwiebert, S. K. Gupta, and J. Weinmann. Research challenges in wireless networks of biomedical sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, Rome, Italy, 2001. ACM.

APPENDIX

LEMMA: 1. $\mathbf{M} = \prod_{i=1}^n \mathbf{M}_i$ is a nonnegative primitive matrix. In particular, $\mathbf{M}^2 > 0$.

PROOF. The matrix \mathbf{M}_i has a 3×3 positive block centered at position $(2i - 1, 2i - 1)$ for all i . The rest of the matrix is equal to one on the main diagonal and zero elsewhere. When $i = 1$, the 3×3 block is divided into one 2×2 block on the top-left of the matrix, a 2×1 on the top-right, a 1×2 block at the bottom-left, and a 1×1 block at the bottom-right. This is because the update of node i modifies variables Δ_1 , Θ_2 and Θ_{n+1} , and the latter is in common with node n . We will refer to the nonzero block of matrix \mathbf{M}_i to as \mathbf{S}_i .

$\mathbf{M} = \prod_{i=1}^n \mathbf{M}_i$ is a nonnegative matrix, since it is a product of nonnegative matrices. We prove now that \mathbf{M} is also primitive, by induction. The first product $\mathbf{M}_{n-1}\mathbf{M}_n$ is a product of two nonnegative matrices with 3×3 positive blocks centered at $(2n - 3) \times (2n - 3)$ and $(2n - 1) \times (2n - 1)$, respectively. The submatrix $\mathbf{M}_{n-1}\mathbf{M}_n(1 : 2n - 5, 1 : 2n - 5)$ is equal to the identity matrix, since both \mathbf{M}_{n-1} and \mathbf{M}_n are equal to the identity matrix in that block.

The elements corresponding to the positive elements of \mathbf{S}_{n-1} and \mathbf{S}_n are positive. This is because \mathbf{M}_{n-1} and \mathbf{M}_n are both positive on the main diagonal. Consider, now, the elements on the right side of block \mathbf{S}_{n-1} in $\mathbf{M}_{n-1}\mathbf{M}_n$. The elements on row $2n - 2$ are positive, because they were in \mathbf{S}_n , and, as we said, they remain in $\mathbf{M}_{n-1}\mathbf{M}_n$. The elements at rows $2n - 3$ and $2n - 2$ are given by the product of a row in \mathbf{M}_{n-1} whose $2n - 2$ th element is positive, and a column in \mathbf{M}_n , whose $2n - 2$ th element is positive too. Therefore, the elements on the right side of block \mathbf{S}_{n-1} in $\mathbf{M}_{n-1}\mathbf{M}_n$ are positive.

Suppose at step i , with $i = n, n - 1, \dots, 3$, matrix \mathbf{M}_i is positive on the diagonal and on the upper-triangular part $\mathbf{M}_i(2i - 2 : n, 2i - 2 : n)$. Because \mathbf{M}_{i-1} is positive on the diagonal, then $\mathbf{M}_{i-1}\mathbf{M}_i$ is positive on the diagonal and on the triangle $\mathbf{M}(2i - 1 : n, 2i - 1 : n)$. Moreover, each element at rows $2i - 3$ and $2i - 4$ of $\mathbf{M}_{i-1}\mathbf{M}_i$, is a product of a row with a positive entry at $2i - 1$, and a column which is positive at position $2i - 1$. Therefore, $\mathbf{M}_{i-1}\mathbf{M}_i$ is positive on the triangle $\mathbf{M}(2i - 4 : n, 2i - 4 : n)$, while the rest of the matrix is one on the diagonal and zero elsewhere.

At the last step we multiply \mathbf{M}_1 by $\prod_{i=2}^n \mathbf{M}_i$. Each element of the top row of $\prod_{i=1}^n \mathbf{M}_i$ is a product of the top row of \mathbf{M}_1 and the corresponding column of $\prod_{i=2}^n \mathbf{M}_i$. They both have a positive entry at position 2, therefore, the top row of the resulting matrix is positive. Similarly, the bottom row of $\prod_{i=1}^n \mathbf{M}_i$ is positive.

Thus, matrix $\mathbf{M} = \prod_{i=1}^n \mathbf{M}_i$ is positive on both the right triangular part and the bottom row. Element (i, j) of \mathbf{M}^2 is the product of row i and column j of \mathbf{M} . Row i has, at least, one positive element (the last element). Column j has the last element which is positive too, therefore the product is positive. Since this is true for every (i, j) , then $\mathbf{M}^2 > 0$. In Figure 8 an example, with $n = 6$, is depicted. \square

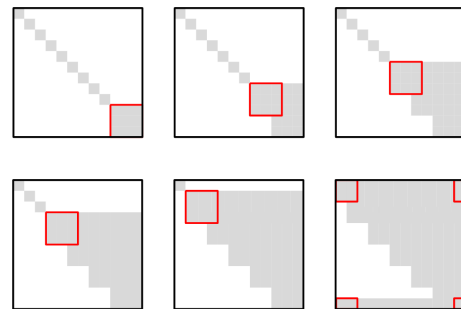


Figure 8: Spreading of non-zero elements of the product $\prod_{i=1}^n \mathbf{M}_i$. From the top-left to the bottom-right it is shown the evolution of the product, with $n = 6$ (non-zero elements are in gray).