



**HAL**  
open science

# Decentralized Time-Synchronized Channel Swapping for Ad Hoc Wireless Networks

George Smart, Nikos Deligiannis, Rosario Surace, Valeria Loscrì, Giancarlo Fortino, Yiannis Andreopoulos

► **To cite this version:**

George Smart, Nikos Deligiannis, Rosario Surace, Valeria Loscrì, Giancarlo Fortino, et al.. Decentralized Time-Synchronized Channel Swapping for Ad Hoc Wireless Networks. *IEEE Transactions on Vehicular Technology*, 2016, 65 (10), pp.8538 - 8553. 10.1109/TVT.2015.2509861 . hal-01241833

**HAL Id: hal-01241833**

**<https://hal.science/hal-01241833>**

Submitted on 9 Feb 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Decentralized Time-Synchronized Channel Swapping for Ad Hoc Wireless Networks

George Smart, Nikos Deligiannis, Rosario Surace, Valeria Loscri, Giancarlo Fortino, and Yiannis Andreopoulos

**Abstract**—Time-synchronized channel hopping (TSCH) is currently the most efficient solution for collision-free, interference-avoiding communications in ad hoc wireless networks, such as wireless sensor networks, vehicular networks, and networks of robots or drones. However, all variants of TSCH require some form of centralized coordination to maintain the time-frequency slotting mechanism. This leads to slow convergence to steady state and moderate time-frequency slot utilization, especially under node churn or mobility. We propose *decentralized time-synchronized channel swapping* (DT-SCS), a novel protocol for medium access control (MAC) in ad hoc wireless networks. Under the proposed protocol, nodes first converge to synchronous beacon packet transmissions across all available channels at the physical layer, with balanced number of nodes in each channel. This is done by the novel *coupling of distributed synchronization and desynchronization mechanisms*—which are based on the concept of pulse-coupled oscillators—at the MAC layer. *Decentralized channel swapping* can then take place via peer-to-peer swap requests/acknowledgments made between concurrent transmitters in neighboring channels. We benchmark the convergence and network throughput of DT-SCS, TSCH and the Efficient Multichannel MAC (EM-MAC) protocol (seen as the state-of-the-art in decentralized, interference-avoiding, multichannel MAC protocols) under simulated packet losses at the MAC layer. Moreover, performance results via a Contiki-based deployment on TelosB motes reveal that DT-SCS comprises an excellent candidate for decentralized multichannel MAC layer coordination by providing for: quick convergence to steady state, high bandwidth utilization under interference and hidden nodes, and high connectivity.

**Index Terms**—Channel hopping, pulse-coupled oscillators, decentralized medium access control (MAC), ad hoc networks.

## I. INTRODUCTION

**D**ATA-INTENSIVE ad hoc wireless networks, such as visual sensor networks [1]–[5], networks of mobile robots,

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org). G. Smart and Y. Andreopoulos were supported by the UK EPSRC: EP/I500331/1 and EP/K033166/1. N. Deligiannis was supported by the VUB strategic research programme M3D2.

G. Smart and Y. Andreopoulos are with the Electronic and Electrical Engineering Department, University College London, Roberts Building, Torrington Place, London, WC1E 7JE, UK. (e-mail: {george.smart, i.andreopoulos}@ucl.ac.uk).

N. Deligiannis is with the Department of Electronics and Informatics, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium, and also with iMinds, Ghent 9050, Belgium (email: [ndeligia@etro.vub.ac.be](mailto:ndeligia@etro.vub.ac.be)).

R. Surace and G. Fortino are with the Department of Electronics, Informatics and Systems, University of Calabria, Via P. Bucci, 87036 Rende (CS), Italy (e-mail: [rsurace@deis.unical.it](mailto:rsurace@deis.unical.it), [giancarlo.fortino@unical.it](mailto:giancarlo.fortino@unical.it)).

V. Loscri is with the FUN Team of Inria Lille-Nord Europe, 40 Avenue Halley, 59650, Villeneuve d’Ascq, France (e-mail: [valeria.loscri@inria.fr](mailto:valeria.loscri@inria.fr)).

vehicles and drones [6]–[10], and wireless capsule endoscopy [11], require:

- high bandwidth to transmit large amounts of sensory data (images, video, acceleration and position data, etc.) with low latency and the smallest possible impact on each sensor’s battery resources [12]–[14];
- spontaneous and quick convergence to a network steady-state when multiple sensors are suddenly activated to monitor an event [15]–[18], e.g., in vehicular networks [8], [9];
- robustness to interference in the unlicensed 2.4 GHz or the 5.9 GHz dedicated short range communication (DSRC) bands [9], [19], [20], used by ad hoc wireless network deployments.

Beyond these requirements, the recent thrust towards machine-to-machine (M2M) communications [21], [22] and the integration of wireless sensor networks with the generic internet infrastructure via 6LoWPAN support at the Network layer [4], [23], [24] call for the development of ad hoc communication protocols at the Medium Access Control (MAC) layer, i.e., protocols that do not depend on any pre-existing infrastructure, such as access points, interference-free control channels, and universal coordinated time (UTC) mechanisms.

The concept of channel hopping has gained acceptance as a good solution for wireless MAC layer coordination, with time-synchronized channel hopping (TSCH) [25] and multichannel DSRC [26] now comprising essential elements of the IEEE 802.15.4e-2012 [27], [28] and IEEE 802.11p [29] standards, respectively. Channel hopping enables nodes to hop between the available channels of the physical layer (e.g., the 16 channels of the 2.400-2.484 GHz band [28] or the 7 channels of the 5.850-5.925 GHz band [26]). This is performed such that transmitters and receivers are evenly spread across channels, so nodes are not constantly in a channel with excessive interference.

Fig. 1 depicts an example of the TSCH protocol [25], where an arbitrary topology is formed between 14 nodes (depicted at the bottom) [30]. Each node reserves timeslots within the slotframe interval (horizontal axis of the top part) and within the 16 channels of IEEE 802.15.4 (vertical axis of the top part). Unoccupied slots appear in white. As the slotframe interval of Fig. 1 repeats periodically, all nodes transmit and listen in different channels, thus avoiding concentrated interference. A similar structure applies for the IEEE 1609.4 multichannel DSRC extension [26] of the IEEE 802.11p [29]. Such slotframes have a rigid (pre-defined) structure and filling up the available slots follows a rather complex

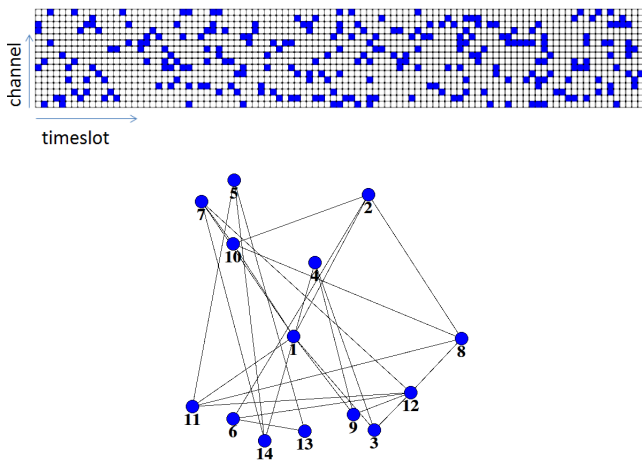


Fig. 1. TSCH example of 14 nodes derived by the 6tisch simulator: (top) slotframe structure with 101 timeslots and 16 channels of IEEE 802.15.4 (blue indicates used slots, white indicates unused slots); (bottom) corresponding connectivity mesh.

advertising request and acknowledgment (RQ/ACK) process on a “coordination” (or “control”) channel [25], [26], [28]. This channel is prone to interference and occasional self-inflicted collisions when the nodes are set to advertise slot reservations very aggressively. Conversely, if slot advertising is not aggressive and nodes leave the network (e.g., in vehicular networks [31] or networks of mobile robot and drones), their slots may remain unoccupied for long periods until another advertisement RQ/ACK process reassigns them to other nodes. This limits the bandwidth usage per channel (see, for instance, the large number of unoccupied slots in Fig. 1) and does not allow for quick convergence to the steady network state. However, both high bandwidth and quick convergence are important for ad hoc networks that must quickly converge to a steady operational state and transmit high data volumes under a periodic or event-driven schedule [5], [8], [9], [31]. Finally, TSCH [25] and the IEEE 1609.4 multichannel DSRC extension [26] of IEEE 802.11p [29] cannot be considered as infrastructure-less protocols, since (i) a coordinator node is required in order to maintain global time synchronization via beacon message broadcasts at slotframe boundaries [20], [25], [26], [28], [30]; and (ii) a dedicated coordination channel must be available for the advertisement RQ/ACK process [25] or the node rendezvous process [26].

#### A. Novelty and Contributions

We propose a novel decentralized multichannel MAC coordination framework, called *decentralized time-synchronized channel swapping* (DT-SCS). The proposed protocol leverages the concept of *pulse-coupled oscillators* (PCOs) [32], [33] at the MAC layer. In DT-SCS, nodes randomly join a channel and are automatically spread across the available channels. The nodes then achieve PCO-based coordination via the periodic transmission of *beacon* packets at the MAC layer. As such, for channels with an equal number of nodes, DT-SCS converges to synchronized beacon packet transmission at the MAC layer in a completely uncoordinated manner.

It is worth emphasizing that, while the use of PCOs as a means for decentralized synchronization or desynchronization is well-established for *single-channel* coordination [32], [33], this is the first approach to combine *both concepts* for *decentralized* time-synchronized transmissions in a *multichannel* setup. In addition, the proposed DT-SCS allows for arbitrary pairwise swaps between nodes in neighboring channels with minimal effort and without disrupting the established network operation. Finally, due to the inherent elastic adaptation of PCO mechanisms, our proposal is robust to interference, hidden nodes and node churn, which are frequent phenomena in ad hoc wireless—in particular, mobile—networks.

Our contributions are summarized below:

- We design DT-SCS, a novel protocol that simultaneously marries the key benefits of: (i) a *decentralized time-division multiple access (TDMA) schedule*, achieved with negatively-coupled PCOs (e.g., DESYNC [32]) within each channel; (ii) *spontaneous alignment of nodes’ timeslots across channels* by using positively-coupled PCOs (e.g., SYNC [33]), which allows for node pairwise channel swapping without jeopardizing the network stability or bandwidth efficiency; (iii) elastic (rather than rigid) time synchronization and *spontaneous adaptation of the available transmission slots across channels* via the SYNC/DESYNC coupling coefficients.
- We prove that DT-SCS converges to a balanced steady state and estimate the degree of node connectivity and the expected convergence time.
- We present detailed simulation and experimental results demonstrating the efficacy of the proposed protocol for distributed multichannel coordination in ad hoc wireless networks.
- Finally, we carry out detailed comparisons between DT-SCS, TSCH, and the decentralized Efficient Multichannel MAC (EM-MAC) protocol [34] in terms of convergence delay, bandwidth efficiency, and robustness to interference and hidden nodes.

#### B. Paper Organization

Regarding the remainder of the paper: Section II reviews related work on multichannel MAC coordination. Section III describes the proposed DT-SCS protocol. Section IV analyzes DT-SCS in terms of stability, connectivity, and convergence time. Section V presents simulation results, whereas Section VI presents experiments with an ad hoc wireless network deployment. Finally, Section VII concludes the paper.

## II. RELATED WORK ON MULTICHANNEL MAC

Multichannel MAC layer coordination has been addressed in numerous ways. The first category comprises schemes that assign channels to nodes in a *static* manner in order to balance them across the available channels of the used physical layer and maximize bandwidth utilization [27], [35]–[37]. Such solutions, however, achieve reduced node connectivity and are prone to persistent interference in any of the utilized channels. The most representative approach in this category is the IEEE

802.15.4e-2012 CSMA/CA [27] with the coordinator-based reserved slot allocation mode operating in multiple channels.

The second category comprises protocols based on *dynamic* coordination of node channel hopping. Hwang *et al.* [38] proposed a low-energy, receiver-driven, channel hopping scheme that does not require global time synchronization. Instead, each sender predicts the wake-up time of each receiver encountered, which is shown to minimize idle listening at the cost of significantly reduced bandwidth efficiency. Tang *et al.* [34] proposed EM-MAC, a multichannel protocol based on receiver initiated predictive wake-up. In EM-MAC, nodes select the channel for communication by following receiver-driven pseudo-random scheduling and predictive wake-up. EM-MAC is shown to be highly resilient to interference and jamming, albeit also at the cost of reduced duty cycling and bandwidth utilization.

The third category of approaches for multichannel coordination and hopping utilize a control (or “coordination”) channel, where nodes negotiate the channel to use for data transmission. Representative examples include, Y-MAC [39], A-MAC [40], MMAC [41], CAM-MAC [42], the IEEE 1609.4 multichannel DSRC [26] extension of IEEE 802.11p [29], and the TSCH option of IEEE 802.15.4e-2012 [27] that can be seen as the most representative approach from this category. However, the bandwidth and reliability of the control node or channel can become significant obstacles to the efficacy of multichannel MAC protocols in many mobile wireless infrastructures [1], [2], [6]–[9], especially under strong interference conditions. These issues are expected to become more pronounced within ad hoc deployments for M2M and 6LoWPAN networks [4], [21]–[24]. For these reasons, on-going efforts towards a *decentralized* TSCH mechanism [25] are based on distributed Aloha-based scheduling for the advertisement channel and a gossip mechanism for the propagation and response to advertisements. However, such mechanisms: (i) are still based on the time and energy-consuming RQ/ACK mechanism; (ii) have a rigid slotframe structure (see Fig. 1); and (iii) require a UTC mechanism (e.g., via a GPS unit [25]).

Table I summarizes the features of the most representative protocol from each of the three aforementioned categories of multichannel MAC coordination in conjunction with the proposed DT-SCS protocol. Overall, with the aim to improve channel hopping for ad hoc wireless networks, three key issues can be identified: (i) avoiding the dependence on a coordination channel and/or node and achieving *completely decentralized* time-frequency coordination with high connectivity and high bandwidth utilization; (ii) providing a decentralized approach for time synchronization in the network; (iii) making node synchronization and timeslot assignment dynamic under varying interference conditions and densities of nodes per channel.

### III. THE PROPOSED DT-SCS PROTOCOL

We first present an overview of the overall operation of the proposed DT-SCS protocol. The detailed operation of the SYNC and DESYNC mechanisms is given in the Section III-B.

TABLE I  
COMPARISON OF KEY FEATURES OF EXISTING MAC PROTOCOLS  
VERSUS THE PROPOSED DT-SCS.

Protocol	IEEE 802.15.4e 2012 [27]	TSCH [25]	EM-MAC [34]	Proposed DT-SCS
Coordination	Centralized	Centralized	Distributed	Distributed
Multichannel Operation	Yes	Yes	Yes	Yes
Channel Hopping	No	Yes	Yes	Yes
Convergence Time (s)	~ 1	~ 14	~ 4	~ 1.5
Connectivity	Low	Medium	Low	High
Network Throughput (kbps)	8 – 100	10 – 55	6 – 18	30 – 85
Resilience to Interference	Low	High	High	High

#### A. Introduction to the Basic Concept

Consider an ad hoc network comprising  $W$  wireless nodes randomly distributed in  $C$  channels [see left part of Fig. 2(a)], with each node broadcasting short *beacon* packets periodically every  $T$  seconds. Within each channel, nodes are assumed to be fully-connected (all listen to all) or densely-connected (only a small subset of nodes cannot be reached by all nodes). The proposed DT-SCS balances the number of nodes per channel and adjusts the transmission time of each node’s beacon packets to reach an evenly-distributed timeslot allocation within each channel. Specifically, the nodes in each channel perform PCO-based desynchronization (i.e., they are “DESYNC” nodes) and elect a single “SYNC” node to provide for cross-channel synchronization. Within each period, the SYNC node of each channel listens for the SYNC beacon message in the next channel<sup>1</sup> and adjusts the transmission time of its own beacon packet in its own channel using PCO-based synchronization [33]. SYNC nodes will also move to the next channel if they detect that less nodes are present there. In this way, the network can converge to the steady state with  $W_c = \frac{W}{C}$  nodes per channel<sup>2</sup>. The beacon packet transmission flow between DT-SCS nodes is illustrated in the right part of Fig. 2(a).

Once the system reaches the steady state, SYNC or DESYNC nodes in adjacent channels can swap channels and timeslots in pairs using a simple RQ/ACK scheme. Fig. 2(b) highlights the short interval between two consecutive beacon packet transmissions (stemming from two different nodes in a channel), during which RQ/ACK packet transmissions for channel swaps take place. If nodes join or leave the network, all remaining nodes adjust their beacon packet timings spontaneously, in order to converge to a new steady state. As shown in Fig. 2(a), the key aspect of DT-SCS is the spontaneous convergence of the ad hoc wireless network from a random state to a multichannel time synchronized beaconing, without the need

<sup>1</sup>We consider cyclic behavior between the last and the first channel. For instance, in IEEE 802.15.4 the SYNC node of Channel 16 listens for the SYNC beacon message of Channel 1.

<sup>2</sup>For simplicity, we assume that  $W$  is divisible by  $C$ . However, when this is not the case the scheme balances the number of nodes to  $W_c \in \left\{ \left\lfloor \frac{W}{C} \right\rfloor, \left\lceil \frac{W}{C} \right\rceil \right\}$  nodes per channel.

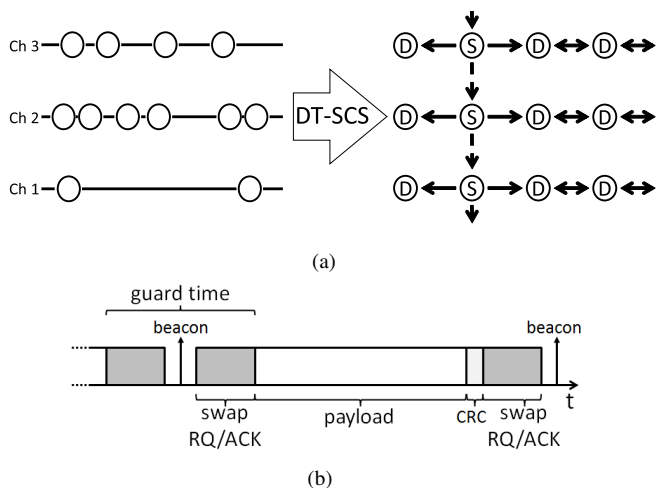


Fig. 2. (a, left) Initial random state of  $W = 12$  node in  $C = 3$  channels; (a, right) DT-SCS converged state with  $W_c = 4$  nodes per channel, showing the intra-channel desynchronization (solid lines) and inter-channel synchronization (dashed lines) between DESYNC (D) and SYNC (S) nodes, respectively. Arrows indicate the intended recipient of each beacon packet transmission. (b) The grey slots indicate the short transmitting/listening intervals where nodes can request and acknowledge swaps.

for a coordinator node or a coordinating channel.

Once convergence to steady state is achieved, the only overhead in the proposed DT-SCS protocol stems from handling swap requests as well as beacon packet broadcasts. Both, however, are very short packets (less than ten bytes). Beyond this, the nodes can also be set to “sparse listening” mode, as it will be detailed in the experimental section. Therefore the protocol overhead is minimal compared to data packet transmission and reception in data-intensive wireless networks.

Losses of beacon packets and timing errors due to interference cause node beacon times to *waver*, that is, nodes send beacon messages at incorrect times. As such, all nodes receiving these messages are similarly affected. If left untreated, this wavering may propagate through the network until all nodes are affected and the network is no longer considered converged. To combat this, we consider the notion of *coupling* between nodes, introduced by PCOs [33], [43]: instead of a DESYNC node jumping directly to the midpoint of its beacon neighbors, the node *slides towards the midpoint* with coupling factor  $\alpha$  ( $0 < \alpha < 1$ ); this is also known as *negative coupling* in the PCO literature [33], [43]–[45]. Similarly, a SYNC node gradually adjusts its beaconing time by coupling factor  $\beta$  ( $0 < \beta < 1$ ) to *align* with the beacon of the SYNC node in the next channel; this beacon alignment is also known as *positive coupling* [33], [46]. This work is the first to propose the usage of positive coupling for inter-channel synchronization in conjunction with concurrent intra-channel coordination achieved via negative coupling. As verified via simulations (Section V) and experiments (Section VI), appropriate selection of coupling factors ensures that any noise and instability in beacon timings is attenuated and does not propagate uncontrollably throughout all nodes and channels of DT-SCS.

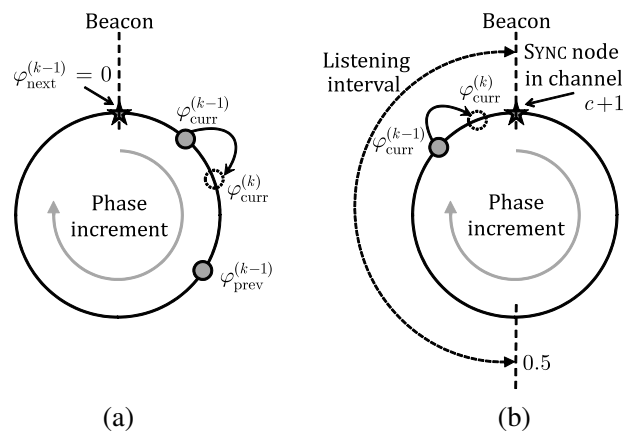


Fig. 3. (a) A DESYNC node performs its  $k$ th phase update when the next DESYNC or SYNC beacon packet is received in channel  $c$ . (b) A SYNC node performs its  $k$ th phase update when a SYNC beacon packet is received in channel  $c + 1$  while the phase of the current beacon broadcast is within its listening interval.

### B. New Multichannel Coupling via Joint SYNC-DESYNC

Synchronization and desynchronization primitives are algorithms for revising the beacon packet broadcast time of a node in a wireless network based on the broadcast times of beacon packets from other nodes within a certain time interval. The proposed DT-SCS protocol is the first to combine both SYNC and DESYNC algorithms in a joint framework for decentralized collision-free multichannel MAC. Consider  $W_c$  nodes being present in channel  $c$ , with  $c \in \{1, \dots, C\}$ , and the total nodes given by  $W = \sum_{c=1}^C W_c$ . Each node joins the network by broadcasting an initial beacon packet randomly in channel  $c$  at a time between  $[0, T)$  seconds. Each node repeats the transmission of its beacon packet upon the completion of its cycle, namely, every  $T$  seconds. For each node, the fraction of the way through its cycle at a given time  $t \in [0, T)$  is denoted as the node's *phase* [33], [43],  $\varphi \in [0, 1)$ .

As shown in Fig. 3, we can imagine the beacon packet transmission times as beads moving clockwise on a ring with period  $T = 1$  s [32]. When the phase of a node becomes one (i.e., the bead reaches the top of the ring in Fig. 3), a beacon packet is broadcast and its phase is reset to zero. Each node keeps the phase of received beacon packets and updates its own beacon phase  $\varphi_{curr}^{(k-1)}$  to  $\varphi_{curr}^{(k)}$  based on the utilized reactive listening primitive. Thus, superscript  $(k)$  indicates the  $k$ th *phase-update iteration*.

For the SYNC and DESYNC algorithms, it is immaterial which physical sensor node is linked to which beacon broadcast, as the phase update process is solely dependent on the received beacon packet times [32], [33], [43]–[45]. For this reason, we explicitly refer to *beacon packet transmission events* and not the physical nodes that broadcast them.

1) *DESYNC Phase Update via Negative Coupling*: During desynchronization in channel  $c$ , each node's beacon phase is updated once within each period  $T$ . As shown in Fig. 3(a), the phase of a node “curr” is updated based on the phases of received “prev” and “next” beacon messages, originating from nodes that transmitted their beacon before and after

node “curr”, respectively. Specifically, upon receiving the next beacon packet, the phase of node “curr” moves towards the middle of the interval between the phases of “prev” and “next” beacon messages, i.e., the phase values of the nodes become decoupled. The  $k$ th phase update of DESYNC with such negative coupling is expressed by<sup>3</sup> [32], [44]

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

with  $\alpha \in (0, 1)$  the DESYNC phase-coupling constant controlling the speed of the phase adaptation and  $\text{mod } 1$  denotes the modulo operation with respect to unity. Previous work [32], [44] showed that the reactive listening primitive of (1) disperses all beacon packet broadcasts in each channel  $c \in \{1, \dots, C\}$  at intervals of  $\frac{T}{W_c}$ . This leads to fair TDMA scheduling in channel  $c$  in steady state (SS). After  $k_{\text{ss}}$  iterations of (1), all beacon packets in channel  $c$  are periodic and the phase updates lead to convergence to SS, expressed by

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

with  $b_{\text{thres}}$  the preset convergence threshold, typically  $b_{\text{thres}} \in [0.001, 0.100]$ . Hence, in the steady state of the DT-SCS protocol, each node in channel  $c$  transmits data packets for  $T \left( \frac{1}{W_c} - b_{\text{thres}} \right) - t_{\text{swap}}$  seconds in the centre of its timeslot, where  $t_{\text{swap}}$  denotes the duration of the guard time per node. Therefore, the maximum number of nodes supported under collision-free TDMA per channel  $c$  is less than  $\left\lfloor \frac{1}{b_{\text{thres}}} \right\rfloor$ .

2) *SYNC Phase Update via Positive Coupling*: PCO-based synchronization with positive coupling [33] updates each SYNC node’s beacon phase according to a received beacon packet (from another SYNC node) that is within the *listening interval*  $\left[ \frac{T}{2}, T \right)$  (second half of the beaconing cycle) [see Fig. 3(b)]. Under the proposed DT-SCS protocol, the phase of each SYNC beacon in channel  $c$  changes after a SYNC beacon packet is received in channel  $c+1$  within the listening interval. Specifically, it moves closer to the phase of the node that sent the beacon packet in channel  $c+1$ . Hence, the  $k$ th phase update of PCO-based synchronization [33] is performed at  $\varphi_{\text{curr}}^{(k-1)} T$  s after the node’s last beacon packet transmission,  $0.5 < \varphi_{\text{curr}}^{(k-1)} < 1$ , via the positive coupling:

$$\varphi_{\text{curr}}^{(k)} = (1 + \beta) \varphi_{\text{curr}}^{(k-1)} \text{ (mod } 1), \quad (3)$$

with  $\beta \in (0, 1)$  the phase-coupling constant controlling the speed of the phase adaptation. Any beacon packets transmitted outside the listening interval  $(0.5, 1)$  are ignored with respect to the SYNC phase update. However, in the proposed DT-SCS, these packets are still processed to extract useful information, such as the total number of nodes in the current channel (see Section III-C). After  $\tilde{k}_{\text{ss}}$  phase updates, (3) converges to coordinated SYNC beacon packet broadcasts at intervals of

<sup>3</sup>Since (1) is applied when the next beacon packet is received, we have that  $\varphi_{\text{next}}^{(k-1)} = 0$  [see Fig. 3(a)]. However, we include  $\varphi_{\text{next}}^{(k-1)}$  in (1) to clarify that the operation of DESYNC depends on both the previous and next beacon packet phase.

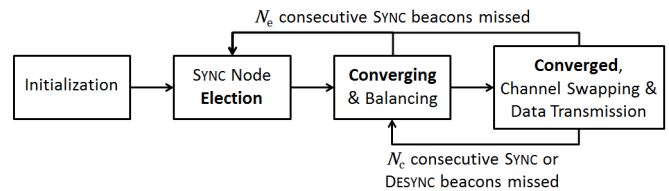


Fig. 4. Block diagram of the operational modes of DT-SCS. The values of  $N_e$  and  $N_c$  are set via experimentation with varying packet loss. Data transmission and channel swapping takes place only during the Converged mode.

$(1 \pm \tilde{b}_{\text{thres}}) \times T$  seconds [33]. Similar to the DESYNC case,  $\tilde{b}_{\text{thres}}$  is used<sup>4</sup> to detect convergence to SS under (2).

### C. Proposed DT-SCS Protocol Description

In an ad hoc wireless network comprising  $W$  nodes that apply the SYNC and DESYNC algorithms within  $C$  channels ( $C > 1$ ), throughput is equally balanced across all nodes when the number of nodes is balanced across all channels, that is, when  $W_c = \left\{ \lfloor \frac{W}{C} \rfloor, \lceil \frac{W}{C} \rceil \right\}$  nodes are present within each channel  $c \in \{1, \dots, C\}$ . Fig. 4 presents the basic stages of the proposed DT-SCS protocol, which are explained in the following.

1) *Node Initialization and Beacon Packet Contents*: When initialized, each node joins a channel  $c \in \{1, \dots, C\}$  randomly as a DESYNC node. Initially, nodes have their receivers constantly enabled and send their beacon messages according to the DESYNC rules.

Each beacon packet transmitted by each node in channel  $c$  contains:

- 1) the type of beacon packet (SYNC or DESYNC);
- 2) the node unique identity number (node id);
- 3) the node id of the SYNC node in channel  $c$  (NULL if none);
- 4) the number of unique nodes heard in channel  $c$ ,  $W_c$ ;
- 5) the number of unique nodes heard (directly or indirectly) in channel  $c+1$ ,  $W_{c+1}$ ;
- 6) the current mode that the node perceives channel  $c$  to be in, namely, Election mode, Converging mode, or Converged mode (see Fig. 4).

Each node can independently establish the information of parts 3 and 4 by listening on channel  $c$ . The information for part 5 is obtained when the SYNC node in channel  $c$  listens to the beacon packet from the SYNC node in channel  $c+1$ . Alternatively, this information can also be obtained when DESYNC nodes in channel  $c$  listen for an acknowledgment of a swap request and overhear a DESYNC beacon in channel  $c+1$ . Finally, the information in part 6 is acquired as described in the following two subsections.

2) *Election Mode*: Election of a SYNC node is initiated in each channel  $c$  when  $N_e$  consecutive periods have passed without receiving a SYNC beacon packet, or when nodes observe that all other nodes report the SYNC node id as NULL.

<sup>4</sup>The thresholds  $b_{\text{thres}}$  and  $\tilde{b}_{\text{thres}}$  for the respective cases of DESYNC and SYNC can have different values. For simplicity, in our implementation, we consider  $b_{\text{thres}} = \tilde{b}_{\text{thres}}$ .

The value of  $N_e$  can be set high enough to avoid reelecting a SYNC node just because SYNC beacon packets were lost due to interference. In our experiments, we found that  $N_e = 10$  provided for virtually no reelections when a SYNC node is already present in each channel  $c$ , while allowing for fast network response when a SYNC node actually leaves the channel.

Once the nodes in channel  $c \in \{1, \dots, C\}$  go to `Election` mode, they report this in part 6 of their beacon packets. Each node then randomly generates an 8-bit number,  $r \in [0, 255]$ , and transmits it in part 3 of its beacon packet. After one complete period, the node with the highest number is elected to become the SYNC node for this channel. In the unlikely case where the highest number is sent by more than one node, the node with the highest node id (part 2) is elected. All nodes confirm the selection in the subsequent periods by setting their SYNC node id (i.e., part 3) to the node id they have elected. Because beacon packets may occasionally be lost, there may be some sporadic cases where nodes may not unanimously agree to the same elected SYNC node. In such cases, nodes rectify their election according to the majority decision. Once all nodes set the SYNC id field to the same value, the `Election` mode (i.e., part 6 in the beacon message) changes to either `Converging` or `Converged` mode. This process ensures that (up to) one SYNC node is present per channel.

3) *Converging Mode via Node Balancing across all  $C$  Channels:* When nodes are in the `Converging` mode, no channel swapping takes place. However, in order to balance nodes within the available  $C$  channels, SYNC nodes can decide to switch to the next channel if less nodes are present therein, as described next.

During the `Converging` mode, all nodes apply the `DESYNC` and `SYNC` processes of Section III-B. The SYNC node in channel  $c$  listens to the next channel for SYNC and `DESYNC` beacons. By listening to the former, the SYNC node applies phase updates to converge to the synchronous state. By listening to the latter, it establishes the number of nodes present in the next channel, i.e.,  $W_{c+1}$  (part 5 of beacon packet contents). If

$$W_c - W_{c+1} - 1 \geq 0 \quad (4)$$

and  $c < C$ , then the SYNC node of channel  $c$  switches to channel  $c+1$  and joins as `DESYNC` node, thereby triggering a new SYNC node election in channel  $c$  (after  $N_e$  periods). Importantly, the SYNC node in the highest channel,  $C$ , can switch to channel 1, i.e., perform “cyclic” switching from highest to lowest channel, if

$$W_C - W_1 - 2 \geq 0. \quad (5)$$

This difference in the switching control for channel  $C$  prevents a race condition where nodes would be constantly switching between channels.

Via the new SYNC node of channel  $c$ , all nodes remaining in channel  $c$  will observe that  $W_{c+1}$  increased by one. Furthermore, after  $N_e$  consecutive misses of the beacon of the node id that switched,  $W_c$  is decreased by one, i.e., the node is confirmed as having departed channel  $c$ . The requirement of  $N_e$  consecutive misses before assuming that the node has left

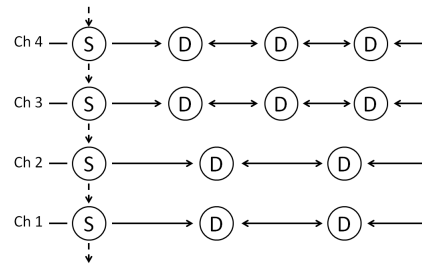


Fig. 5. Example of balancing under DT-SCS for a network of  $W = 14$  nodes in  $C = 4$  channels.

channel  $c$  avoids erroneously decreasing  $W_c$  due to a burst of interference in channel  $c$ .

The above process will lead to nodes moving from lower to higher channels, thereby enabling the network to converge to a balanced number of nodes across all  $C$  channels. That is, after balancing, there going to be  $W_c = \{\lfloor \frac{W}{C} \rfloor, \lceil \frac{W}{C} \rceil\}$  nodes in each channel  $c \in \{1, \dots, C\}$ . Examples of balancing are illustrated in Fig. 2(a) and Fig. 5. The example of Fig. 5 demonstrates that: (i) without the special condition for channel  $C$ , SYNC nodes would be cyclically switching in perpetuity; (ii) because switching occurs between successive channels during the `Converging` mode, channels with equal numbers of nodes are clustered together, with an ascending number of nodes per channel. This is an important feature of the algorithm, as it does not permit channels with unequal node counts to be interspersed. As nodes can only swap with their counterpart in neighboring channels<sup>5</sup> (a node in channel  $c$  may only swap with the concurrently-beaconing node in channel  $c+1$ ), ensuring that channels with equal numbers of nodes are grouped together greatly improves connectivity in the network.

4) *Converged Mode, Channel Swapping and Data Transmission:* Once nodes are in `Converging` mode and their SYNC or `DESYNC` beacon packets fall within the convergence threshold—i.e., (2) holds—they switch to the `Converged` mode. Nodes can thus begin data transmission following a short guard time interval after their beacon packet broadcast. The duration of their transmission lasts until another short guard time interval prior to the subsequent node beacon packet broadcast, as shown in Fig. 2(b). In `Converged` mode, nodes transmit data, send or acknowledge swap requests and swap channels.

*Limited listening:* Beyond the time required for receiving (or transmitting) data packets, all nodes switch on their transceiver only during beacon guard times [Fig. 2(b)]. This *limits the required listening to short time intervals* within each period of  $T$  sec. The guard time is used to allow for beacon variability due to SYNC or `DESYNC` beacon time adaptation via phase updates. This adaptation may cause beacon time fluctuations, the range of which is controlled via the coupling parameters  $\alpha$  and  $\beta$ . To reduce the listening time further, all nodes can switch to *sparse listening*, that is, they can opt to listen for beacons only once every several periods.

<sup>5</sup>In the network configuration of Fig. 5, `DESYNC` nodes can apply channel swapping only between channel 1 and channel 2, and between channel 3 and channel 4. However, SYNC nodes can still swap places between all four channels, as their beacon packet transmissions remain synchronous.

*Channel Swapping:* In Converged mode all DESYNC nodes of each channel  $c$  can opt to transmit swap requests in the next channel (or acknowledge swap requests from a node of channel  $c-1$ ) if, and only if,  $W_{c+1} = W_c$  (or  $W_c = W_{c-1}$ ), with  $W_{C+1} \equiv W_1$  and  $W_0 \equiv W_C$ . If a swap is acknowledged, the corresponding SYNC or DESYNC nodes swap channels in their subsequent beaconing cycle and remain in the new channel until another swap RQ/ACK event. Because the swap acknowledgment may not always be received by the requesting node, sporadic cases may occur where the node requesting the swap does not actually swap channels. To overcome this, every node that received a swap request transmits its first beacon packet towards the end of the guard time after performing the channel swap. This enables the node to detect that its swap-requesting “partner” is not sending its beacon in its old channel and has indeed carried out the swap. If, however, the swap partner did not carry out the swap, then the node returns to its original channel, resumes beaconing therein and requests a new swap.

By using the channel swapping mechanism, DT-SCS ensures each node can attempt to swap channels whenever: (i) the application requires; (ii) a node must reach nodes not present in its current channel; (iii) excessive interference is observed in a channel. Channel swapping should not be confused with channel switching: the former is done in a peer-to-peer RQ/ACK manner in Converged mode, while the latter is performed in order to balance the total nodes in  $C$  channels during the Converging mode, and it does not use an RQ/ACK mechanism. Finally, as depicted in Fig. 4, nodes in the Converged mode may move back to Converging mode if  $N_c$  consecutive beacon packets are not received from any SYNC or DESYNC node. Nodes move to Election mode if  $N_e$  consecutive SYNC beacon packets are not received. In both of these modes, no data transmission or channel swapping takes place and nodes listen constantly.

*Coupling Adaptation:* As mentioned in Section III-A, losing beacon packets due to interference may lead to beacon packet transmissions at incorrect times. To absorb transient oscillations of beacon times, while at the same time maintain fast convergence, the values of  $\alpha$ ,  $\beta$ ,  $N_e$ ,  $N_c$  can be adjusted per node. Setting  $\alpha, \beta \rightarrow 1$  and  $N_e, N_c \leq 3$  allows for very quick convergence and better suits channels experiencing low interference. Conversely, setting  $\alpha, \beta \rightarrow 0$  and  $N_e, N_c \geq 8$  provides for more stable operation under interference, albeit at the cost of slower convergence and reaction time. While the joint optimization of these parameters with DT-SCS deployments remains a future research topic, we refer to existing work on the impact of  $\alpha$  and  $\beta$  in single-channel DESYNC and SYNC [32], [33].

#### IV. PROTOCOL ANALYSIS

##### A. Balancing and Stability

As described in Section III-C3, during the Converging mode of the proposed DT-SCS protocol, SYNC nodes can decide to switch to the next channel if they detect less nodes present therein. We prove below that this mechanism leads to a balanced number of nodes per channel as illustrated in Figs. 2(a) and 5.

**Proposition 1.** *The proposed node balancing mechanism converges to  $W_c \in \{\lceil \frac{W}{C} \rceil, \lfloor \frac{W}{C} \rfloor\}$  nodes within each channel  $c \in \{1, \dots, C\}$ .*

*Proof:* See Appendix A. ■

Once  $C$  channels have balanced numbers of nodes, the DT-SCS protocol performs repeated PCO-based synchronization [33] across channels and desynchronization [32] within each channel. The former technique leads to synchronized beacon transmissions of SYNC nodes across channels, while the latter ensures fair TDMA scheduling between the nodes in a channel. The following proves the efficacy of the algorithm.

**Proposition 2.** *For each channel  $c$ , the proposed DT-SCS protocol converges to equidistant beacon packet transmissions at intervals of  $T\left(\frac{1}{W_c} \pm b_{thres}\right)$  seconds, with  $W_c \in \{\lceil \frac{W}{C} \rceil, \lfloor \frac{W}{C} \rfloor\}$  and the SYNC beacons are broadcast concurrently in all channels.*

*Proof:* See Appendix B. ■

##### B. Connectivity

Via channel swapping, the SYNC node in each channel  $c$  can eventually reach any node in the remaining channels  $\bar{c} \in \{1, \dots, C\}$ ,  $\bar{c} \neq c$ , except for other SYNC nodes, since they are concurrently transmitting. Hence, the degree of connectivity of a SYNC node is

$$D_{\text{SYNC}} = W - C. \quad (6)$$

Similarly, for all channels with  $\lfloor \frac{W}{C} \rfloor$  or  $\lceil \frac{W}{C} \rceil$  nodes, all DESYNC nodes can swap channels in order to reach any other SYNC or DESYNC node, except for the DESYNC nodes that are synchronous to them. In the Converged mode, the

$$C_{\text{high}} = W - \left\lfloor \frac{W}{C} \right\rfloor C \quad (7)$$

highest channels will have

$$W_{\text{DESYNC,high}} = \left\lfloor \frac{W}{C} \right\rfloor - 1 \quad (8)$$

DESYNC nodes (and one SYNC node), while the

$$C_{\text{low}} = C - \left( W - \left\lfloor \frac{W}{C} \right\rfloor C \right) \quad (9)$$

lowest channels will have

$$W_{\text{DESYNC,low}} = \left\lceil \frac{W}{C} \right\rceil - 1 \quad (10)$$

DESYNC nodes (and one SYNC node).

**Proposition 3.** *The average degree of connectivity of a DESYNC node is*

$$D_{\text{DESYNC}} = \frac{1}{W - C} \left[ (C_{\text{high}} W_{\text{DESYNC,high}})^2 + (C_{\text{low}} W_{\text{DESYNC,low}})^2 + C_{\text{low}} C_{\text{high}} \times (W_{\text{DESYNC,high}} + W_{\text{DESYNC,low}}) \right]. \quad (11)$$

*Proof:* See Appendix C. ■



In the example of Fig. 2 we get  $D_{\text{SYNC}} = 9$  and  $D_{\text{DESYNC}} = 9$ , while in the example of Fig. 5 (and following the node placement of Fig. 1), we get  $D_{\text{SYNC}} = 9$  and  $D_{\text{DESYNC}} = 7.2$ . For the same wireless node placement, TSCH achieves average connectivity of 3.5 under its default configuration (see Section VI for details), which is illustrated at the bottom of Fig. 1.

### C. Estimation of Convergence Time

The protocol initiates with  $W$  nodes randomly joining  $C$  channels. To estimate the expected time for DT-SCS to converge to the steady-state, we must estimate the probability that the ensemble of  $W$  nodes will reach combination  $i$  out of  $\mathcal{C}_{W,C}$  possible combinations, with each combination comprising  $[W_1(i) \dots W_C(i)]$  nodes within  $C$  channels. Our convergence time estimate is summarized in the following proposition.

**Proposition 4.** *Under no packet losses and no hidden nodes in the network, the expected delay until convergence in DT-SCS is*

$$d_{W,C} = TN_e \sum_{i=1}^{\mathcal{C}_{W,C}} \left( \Pr(i) \max_{\forall c} \left| W_c(i) - \left\lfloor \frac{W}{C} \right\rfloor \right| \right). \quad (12)$$

with

$$\mathcal{C}_{W,C} = \frac{(W+C-1)!}{(C-1)!W!}. \quad (13)$$

and

$$\Pr(i) = \prod_{c=1}^{C-1} \left[ \binom{W_{\text{res},c}(i)}{W_c(i)} \frac{(c-1)^{W_{\text{res},c}(i)-W_c(i)}}{c^{W_{\text{res},c}(i)}} \right]. \quad (14)$$

with  $\forall i : W_{\text{res},c}(i) = W - \sum_{m=1}^{c-1} W_m(i)$ .

*Proof:* See Appendix D. ■

## V. SIMULATION RESULTS

All simulations for DT-SCS were performed in Matlab, by extending the event-driven simulator for the DESYNC protocol by [32]. Since DT-SCS is a MAC-layer protocol, the utilized simulation is reporting results in function of packet loss experienced within each of the 16 channels of the IEEE802.15.4. The results are compared against TSCH simulation results produced via the 6tisch simulator [30], which is the most accurate TSCH simulator available in public. Since the 6tisch simulator allows for link and timeslot establishment between nodes, we have modified it to also simulate the operation of the EM-MAC protocol [34], which is one of the most-prominent decentralized protocols in the literature. Specifically, we disabled the use of the control channel for slot RQ/ACK and enabled each node to: (i) join the network by selecting channels and wake-up times pseudorandomly; (ii) send beacon packets notifying senders about their listening slots and wake-up times; (iii) predict the wake-up times and channels of receiver(s) and join them to send packets as per the established configuration; (iv) blacklist channels if packet loss above 15% is observed, as per the original EM-MAC proposal [34]. Overall, our comparison is indicative for a broad range of wireless networks encountered in vehicular or mobile node

environments because other solutions, like the IEEE 1609.4 extension [26] of the IEEE 802.11p [29] standard, also use slotframe and reservation mechanisms similar to TSCH.

We use data payload size of 60 bytes. Packet loss is simulated by randomly dropping packets to mimic interference conditions experienced within the 2.4 GHz unlicensed band. Simulations were repeated 100 times and average results are reported. In the vast majority of the reported results, the span of 95% confidence intervals was found to be only  $\pm 15\%$  from the average values.

Unless otherwise stated, all simulations assume an ad hoc network consisting of  $W = 64$  nodes in the  $C = 16$  channels of IEEE 802.15.4. Concerning the configuration of the proposed DT-SCS, we used  $\alpha = 0.6$ ,  $\beta = 0.6$  for the DESYNC and SYNC parameters of (1) and (3). We set  $T = 100$  ms,  $t_{\text{swap}} = 12$  ms<sup>6</sup>,  $b_{\text{thres}} = 0.01$  and  $N_e = N_c = 10$ . Under the specified settings and excluding the guard time periods (13 ms), one data packet of 60 bytes can be sent between two consecutive beacon packets within the same channel.

Regarding TSCH, we use the default 6tisch settings for timeslots (101 slots) per slotframe and channels (16 channels). Each node has, on average, two outgoing (data sending) links and one incoming (data receiving) link. In addition, the `--traffic` parameter of 6tisch is set to 0.75, which, under the established setup, corresponds to two timeslots per node link within each slotframe. Convergence is assumed for TSCH when ten consecutive slotframes are observed with less than 5% change in timeslot allocations amongst nodes.

Finally, concerning EM-MAC, following the low duty cycle of the original paper [34], we used one outgoing and one incoming slot per node within the 16 channels available, with maximum sleep time interval per node equivalent to 100 slots. The wakeup slot duration was set to be equal to the slot duration of TSCH. The use of these settings ensured minimal clock drift between transmitters and receivers in EM-MAC, and therefore the exponential chase algorithm proposed in the original implementation of the protocol was found to be unnecessary in our implementation. Convergence is assumed for EM-MAC when at least 70% of the nodes have established the wake up time pattern (and channel) to send to their receiver node. The use of 70% was found to provide for the best compromise between convergence and robustness to packet loss and clock drift between sender and receiver nodes.

### A. Node Balancing and Connectivity

Firstly, we show that the proposed node balancing mechanism within DT-SCS converges to  $\lfloor \frac{W}{C} \rfloor$  or  $\lceil \frac{W}{C} \rceil$  nodes per channel. Figs. 6(a)-(b) show the initial and final node beacon-packet phases versus the channel number for  $W = 14$  nodes in  $C = 4$  channels. In the initial state [see Fig. 6(a)], a random number of nodes, each with a random phase, enter each channel. In this example, we have  $W_{\{1,2,3,4\}} = [5, 3, 2, 4]$ . In the converged state [see Fig. 6(b)], the nodes have been

<sup>6</sup>We opted for the reported values of  $T$  and  $t_{\text{swap}}$  such that, under the expected number of nodes per channel in steady state (i.e., 4 nodes), the duration of the data payload interval in-between the guard times becomes 13 ms, which is similar to the data payload interval of TSCH.

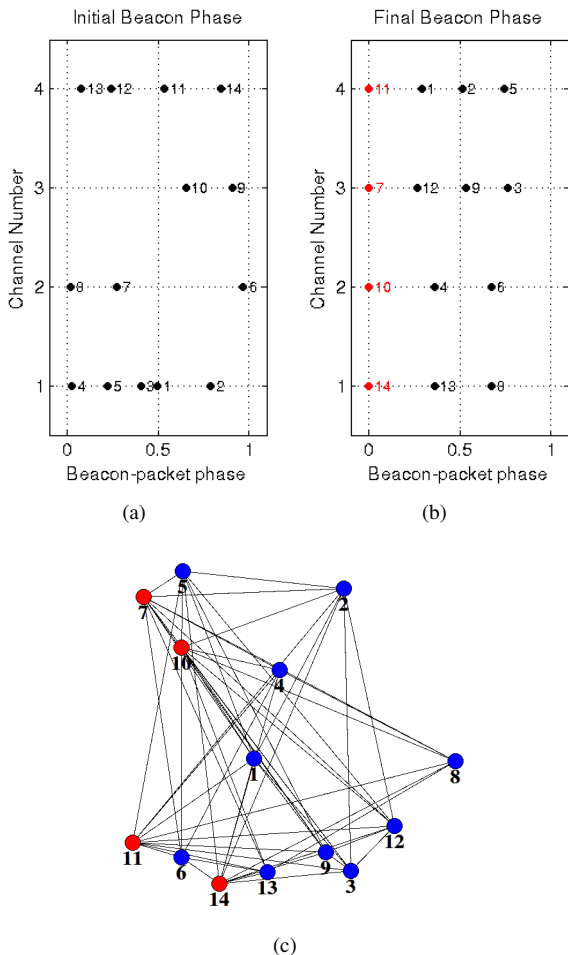


Fig. 6. Initial (a) and final (b) node beacon-packet phase locations versus channel number. Each node has a unique id, with SYNC nodes indicated in red. (c) Corresponding connectivity between DT-SCS nodes in the Converged mode, with node swapping enabled.

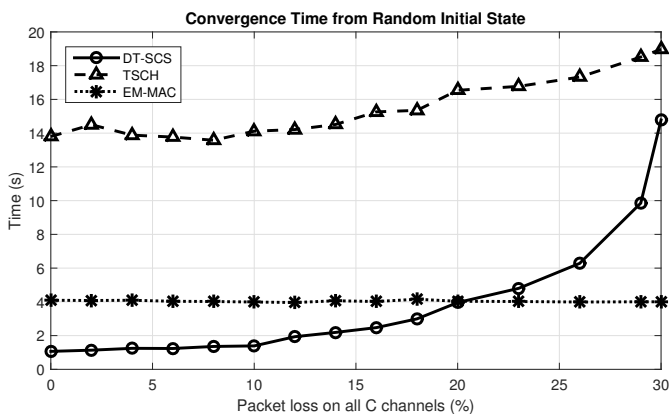


Fig. 7. Average time required for convergence in DT-SCS, TSCH and EM-MAC when 64 nodes join 16 channels randomly during initialization.

balanced within the channels (with the elected SYNC nodes indicated in red), where the two highest channels have four nodes and the two lowest channels have three nodes.

TABLE II  
THEORETICAL (PROPOSITION 4) VS. SIMULATION CONVERGENCE TIME OF DT-SCS UNDER VARIOUS SETTINGS AND NO PACKET LOSSES.

$\{W, C\}$	Simulation	Proposition 4	% Error
$\{64, 16\}$	1.296	1.123	13.4
$\{48, 12\}$	1.138	1.064	6.5
$\{32, 4\}$	1.3850	1.410	1.8
$\{25, 3\}$	0.958	1.013	5.7
$\{12, 3\}$	0.509	0.662	30.1
$\{8, 2\}$	0.308	0.419	35.1

### B. Convergence Time

An important aspect of the proposed protocol is the time required to reach the Converged mode from a random initial state.

Table II presents the average convergence time of DT-SCS versus the corresponding theoretical result of Proposition 4 under a variety of settings for  $W$  and  $C$  (all other settings are left as described previously) and no packet loss. Evidently, for the majority of cases, Proposition 4 predicts the simulation convergence time with less than 15% error, and the maximum prediction error is below 36%. Importantly, under no packet loss, the DT-SCS convergence time was always found to be below 1.5 s for all settings.

We can now investigate convergence of DT-SCS under the occurrence of packet losses, and also in comparison to the time required by TSCH and EM-MAC to achieve a stable contention-free slot allocation via their centralized and distributed advertising mechanisms. Fig. 7 presents the related results for  $W = 64$  nodes under varying packet loss percentage imposed on each of the 16 channels of IEEE 802.15.4. Even though these simulations do not incorporate all the aspects of propagation and interference experienced in a real testbed, the results in Fig. 7 demonstrate that DT-SCS reduces the required convergence time by 22.04–91.61% in comparison to TSCH. Such quick convergence occurs because, contrary to TSCH, the proposed DT-SCS protocol does not require nodes to advertise and acknowledge free slots, which is a process that is detrimental to the convergence time. In addition, DT-SCS converges to the steady-state faster than EM-MAC for packet loss rates below 20%. However, EM-MAC exhibits a very stable convergence behaviour as the low duty cycle and low connectivity of the protocol ensure that, even under high packet loss rate, the majority of nodes establish the wake-up pattern to rendezvous with their receiver within 4 sec.

Subsequently, we study the time required for the network to return to the steady state under the effect of churn (which is typically encountered in mobile and vehicular networks). In this case, the network was initially in steady state, but the arrival or departure of nodes (i.e., the effect of churn) caused the network to return to Converging mode. Fig. 8 depicts a comparison of the re-convergence speed of the proposed DT-SCS against that of TSCH for different churn conditions, namely, low, medium and high churn. These conditions correspond to 5%, 25%, and 50% of nodes arriving or leaving the network, respectively. The results show that, under medium and high churn and packet loss rates up to 22–25%,

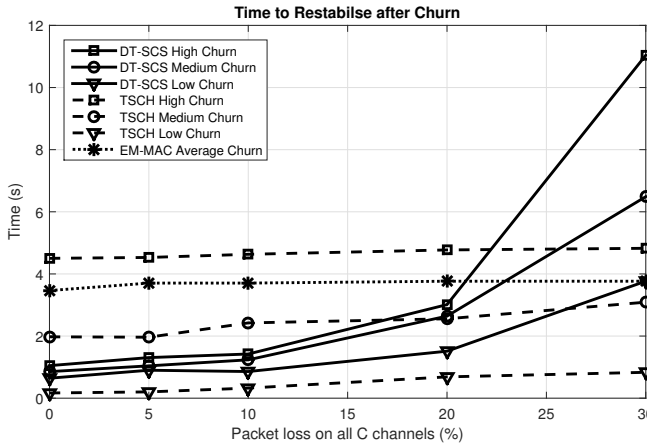


Fig. 8. Average time required for DT-SCS, TSCH and EM-MAC to return to steady state under varying degrees of node churn.

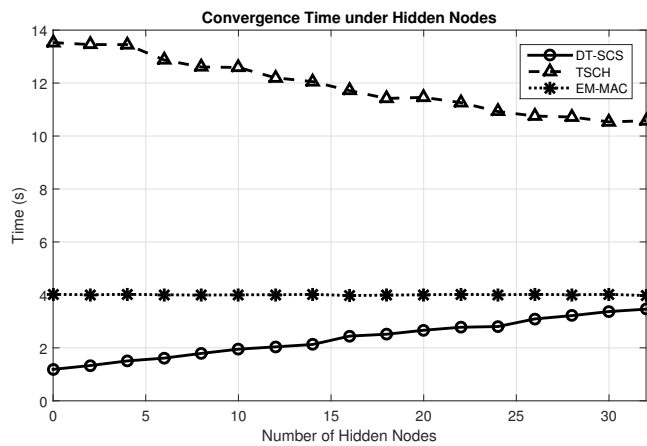


Fig. 9. Average convergence time under increasing number of hidden nodes.

DT-SCS reduces the time that the network requires to return to steady state in comparison to TSCH and EM-MAC (where we only show the average results as all churn cases exhibited very similar behaviour). This is because under medium and high churn all protocols will require extensive reconfiguration to return to steady state. Similar to the convergence from a random initial state, the proposed DT-SCS achieves quicker convergence in comparison to TSCH. On the contrary, TSCH and EM-MAC offer faster convergence when the packet loss rate is high, or when low churn is experienced. This is to be expected since, under low churn, only the TSCH or EM-MAC nodes that have lost communication links will engage in re-advertising actions. Conversely, the proposed protocol will force all nodes to re-converge. Moreover, under high packet loss, few nodes receive advertising RQ/ACK, and so the schedule remains largely stable. In both cases, however, the disadvantage is that not all abandoned TSCH slots (or EM-MAC sender-receiver pairings) are reoccupied, thereby leading to lower bandwidth utilization.

We now investigate the convergence speed of DT-SCS when some of the nodes in the network are hidden from other nodes. In particular, we measure the time to achieve convergence

TABLE III  
AVERAGE CONVERGENCE TIME (IN SECONDS) UNDER TARGETED INTERFERENCE.

	DT-SCS	TSCH	EM-MAC
On a random channel ( $c \neq 1$ )	1.2496	14.2186	4.095
On TSCH control channel	1.2496	73.9126	4.095

to steady state when a number of randomly chosen nodes<sup>7</sup> cannot communicate with a random subset of twelve other nodes in the considered setup. In order to make an extensive investigation of the effect of hidden nodes, we vary their number from 0 to 32 and repeat the DT-SCS convergence process multiple times in order to measure the average convergence time. The results in Fig. 9 show that, irrespective of the presence of hidden nodes, the convergence speed of DT-SCS is significantly higher than that of TSCH and EM-MAC. When hidden nodes are present, the required convergence time of DT-SCS increases by up to 3 seconds, while that of TSCH decreases by up to 3 seconds (albeit still remaining almost three times higher than that of DT-SCS). This is to be expected since TSCH nodes simply miss RQ packets from hidden nodes. On the other hand, the Converging mode of DT-SCS will perform channel switching until all nodes join channels with non-hidden terminals.

Next, we study the convergence time of the proposed protocol against TSCH under the effect of targeted interference, namely, high packet losses on a given channel. In this regard, we devise the following experiment: We apply packet loss of 30% on channel  $\hat{c}$  of DT-SCS, TSCH and EM-MAC, while all other channels  $c \neq \hat{c}$  suffer from packet loss of 2%. We explore two cases, that is, (i) when  $\hat{c}$  is a random channel ( $\hat{c} \in \{1, \dots, 16\}$ ), or (ii)  $\hat{c}$  is the control channel of TSCH and a specific channel (e.g.,  $c = 1$ ) of DT-SCS or EM-MAC. Table III shows that the convergence time of all protocols is increased with targeted interference. However, contrary to the proposed DT-SCS and EM-MAC, TSCH is particularly vulnerable to packet losses on the control channel, whereby the convergence time is increased by 444%. This underlines the importance of the decentralised, infrastructure-less, nature of the proposed protocol and EM-MAC and highlights potential problems with centralised protocols that rely on control nodes or coordination channels. Particularly, under high control channel interference, a network deployment using TSCH will struggle to maintain time synchronisation across all channels, thereby suffering from a loss of performance.

### C. Bandwidth Efficiency

To assess the steady state performance of the proposed DT-SCS against TSCH and EM-MAC, we compare the total payload bits successfully received by all DT-SCS nodes per second versus the equivalent results obtained via the 6tisch simulator for TSCH (and its modification for the EM-MAC simulation). Fig. 10 shows that our approach achieves a substantially-higher slot and channel utilization than TSCH, leading to bandwidth gains of 27.12–40.63%. At the same

<sup>7</sup>Nodes can be SYNC or DESYNC.

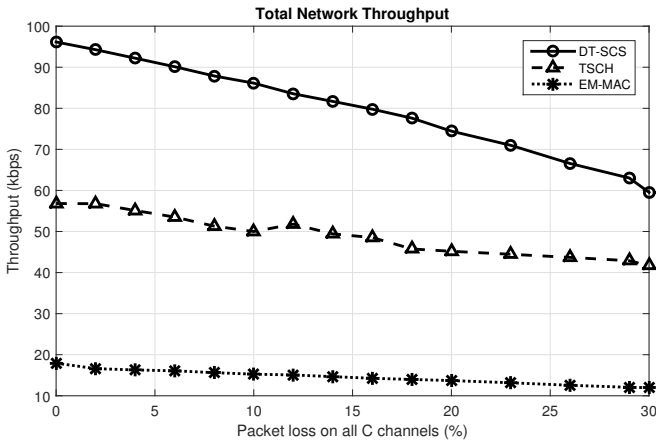


Fig. 10. Comparison of bandwidth utilization (total payload transmitted by all nodes per second) between the proposed DT-SCS, TSCH and EM-MAC.

time, it offers more than five times the network throughput of EM-MAC. This is because DT-SCS allows for all nodes to use all the available time in-between their own beacon and the next beacon (barring the guard time intervals) for contention-free transmission. On the contrary, TSCH requires advertisement and confirmation actions and imposes a rigid slot allocation. Such a rigid slotframe allocation imposes strict limitations on the available bandwidth per node, restrictions that are not applied by the proposed DT-SCS protocol. Moreover, EM-MAC imposes a low duty cycle due to the receiver-driven rendezvous policy applied in the protocol, thereby leading to substantially lower network throughput.

## VI. EXPERIMENTS WITH TELOS B MOTES

We implemented both DT-SCS and TSCH as applications in the Contiki 2.7 operating system running on low-power TelosB motes<sup>8</sup>. By utilizing the NullMAC and NullRDC network stack options in Contiki, we control all node interactions at the MAC layer via our code. Our DT-SCS implementation follows the protocol description of Section III-C with  $T = 100$  ms,  $\alpha = \beta = 0.6$ ,  $N_c = N_c = 10$  and guard time of 12 ms [Fig. 2(b)] for increased robustness to interference. During the Converged mode all nodes switch to “sparse listening”, i.e., they listen for beacons only once every twenty periods, unless high interference noise is detected<sup>9</sup>. Concerning TSCH, our implementation follows the advertisement RQ/ACK and slotframe structure of the 6tisch simulator and TSCH standard [27], [30], [47], namely: channel 11 of IEEE 802.15.4 for advertisements, RQ/ACK ratio of 1/9, slotframe comprising 101 slots of 15ms each, and one node (at the center of our deployment) was set to broadcast the slotframe beacon for global time synchronization. For both frameworks, we utilized

<sup>8</sup>Due to the fact that the simulation experiments of the previous section showed that EM-MAC achieves substantially lower network throughput and lower connectivity than the proposed DT-SCS, we do not utilize EM-MAC in our experiments and instead compare only against TSCH.

<sup>9</sup>In the Converged mode, we determine the interference noise floor in-between transmissions by reading the CC2420 RSSI register and switch to regular listening of all anticipated neighboring beacons per period if high interference noise levels are detected. Therefore, the option of sparse listening does not affect the stability of DT-SCS.

the TelosB high-resolution timer (`rtimer` library) for setting transmission and listening events. We remark that the slotframe period and guard time settings are similar to those of the IEEE 1609.4 [26] extension of IEEE 802.11p [29].

Similar to our simulations, all experiments are based on the deployment of  $W = 64$  nodes in the  $C = 16$  channels of IEEE 802.15.4. For DT-SCS, this leads to  $W_c = 4$  nodes per channel in the steady state. The 64 TelosB motes were placed in four neighboring rooms on the same floor of an office building, with each room containing 16 nodes. Each DT-SCS node (either SYNC or DESYNC) could reach up to 48 other nodes via channel swapping. This agrees with the values for  $D_{\text{SYNC}}$  and  $D_{\text{DESYNC}}$ . On the other hand, each TSCH node could reach only up to 4 other nodes under the used configuration. Overall, our setup corresponds to scenarios involving dense network topologies and data-intensive communications once the ad hoc wireless nodes are activated from a *suspended* state.

### A. Results under Interference

We investigate the convergence time of DT-SCS and TSCH under varying interference levels. Rapid convergence to the steady network state is extremely important when the entire set of nodes is initiated from a suspended state, or when sudden changes happen in the network (e.g., multiple nodes join or leave). We carried out 100 independent tests, with each room containing an interference generator for 25 tests. To generate interference, an RF signal generator was used to create an unmodulated carrier in the center of each channel. The carrier amplitude was adjusted to alter the signal-to-noise-ratio (SNR) at each receiver [48]. The nodes were set to maximum transmit power (+0 dBm) in order to operate under the best SNR possible. As an indication, in the utilized experimental environment, jamming signal powers of 6.00, 9.00, 10.00 and 10.25 dBm correspond to average packet losses of 0.3%, 1.7%, 11.4%, 29.4%, respectively.

Fig 11(a) shows the time required for DT-SCS and TSCH to converge under varying interfering signal power levels. The results obtained with our Contiki implementation corroborate that the proposed DT-SCS reduces the convergence time by an order of magnitude in comparison to TSCH. Moreover, the difference in convergence time between the proposed protocol and TSCH is increasing with the interference level. This result demonstrates the key advantages of the DT-SCS protocol with respect to TSCH, namely that: (i) it is fully decentralized and (ii) it does not depend on an advertisement and acknowledgement scheme.

Next, we investigate the convergence time of the proposed DT-SCS protocol and TSCH under the effect of targeted interference on a given channel. Concerning the former, given that there is no coordination channel, we explore how the interference on channel  $c+1$  effects the convergence in channel  $c$ . A moderate level of interference (that is, 5 dBm) in channel  $c+1$  causes fluctuations in the SYNC node beacon of channel  $c \in \{1, \dots, 16\}$ , which in turn causes the average convergence time to increase from 1.223 to 1.518 seconds. When the same level of interference is also applied on channel  $c$ , the convergence time is further increased to 2.738 seconds. Regarding

TSCH, we observed that interference in the advertisement channel led to unstable behavior and, for the cases where convergence was eventually achieved, more than 30 seconds were required. This demonstrates the detriment of depending on a coordination channel for advertisements.

### B. Bandwidth Results

We measured the total network bandwidth (that is, total payload bits per second) achieved under DT-SCS and TSCH. Interference was applied as described in Section VI-A. The results, depicted in Fig 11(b), show that DT-SCS systematically achieves more than 40% increase in the total network throughput, irrespective of the interference level. Both protocols experience a significant loss of throughput under high interference levels (that is, above 10 dBm), which is, however, substantially more severe for TSCH. In effect, when interference is above 12 dBm, the bandwidth obtained with TSCH drops to zero because of the inability of TSCH to recover lost slots through advertising. On the contrary, even at high interference levels, DT-SCS recuperates bandwidth utilization due to the elasticity of SYNC and DESYNC mechanisms and the high values used for  $N_e, N_c$ .

## VII. CONCLUSION

We propose a novel protocol for ad hoc wireless networks that performs decentralized time-synchronized channel swapping (DT-SCS) and circumvents certain convergence and network utilization problems of existing designs, such as the state-of-the-art time-synchronized channel hopping (TSCH) protocol. The unique aspect of our approach is the use of pulsed-coupled oscillators that *concurrently* perform synchronization and desynchronization across *multiple channels*. This allows for rapid convergence to the steady state in a completely decentralized manner, that is, without requiring a node or channel coordinator, or time synchronization via a universal coordinated time mechanism. DT-SCS spontaneously adapts to node churn and varying packet losses, while offering high degree of connectivity through channel swapping. Experimentation via simulations and a real Contiki-based implementation on TelosB motes shows that, in comparison to TSCH and the Efficient Multichannel MAC (EM-MAC) protocol, the proposed DT-SCS leads to a significant reduction of the convergence time and substantially higher network throughput utilization. These traits render the proposed DT-SCS an excellent candidate for vehicular or mobile deployments that collect and communicate large quantities of information in a decentralized manner.

### APPENDIX A PROOF OF PROPOSITION 1

*Proof:* During the Converging mode, a SYNC node may switch from channel  $c$  to  $c+1$ , or from channel  $c-1$  to  $c$ . A SYNC node switch occurring simultaneously between channels  $c-1 \rightarrow c$  and  $c \rightarrow c+1$  at the  $k$ th period can be

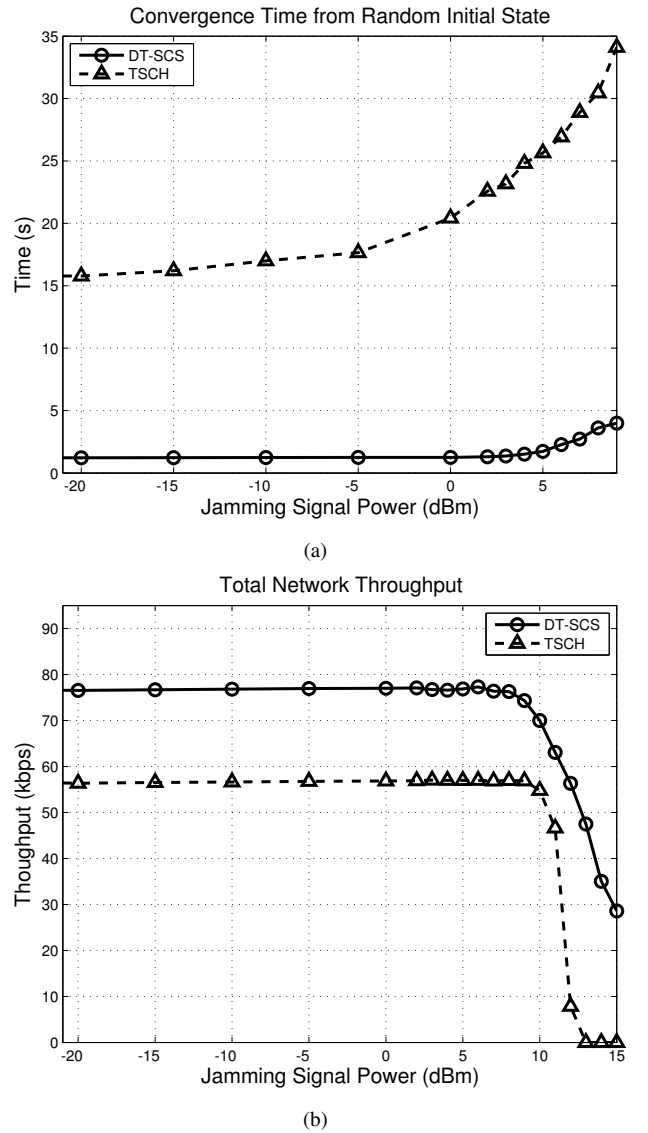


Fig. 11. (a) Average time required for DT-SCS to reach Converged mode and for TSCH to reach a stable slotframe allocation under varying interfering signal power levels. (b) Total network bandwidth (total payload bits transmitted by all nodes per second) between the proposed DT-SCS and TSCH under varying signal power levels.

expressed stochastically for the number of nodes in channel  $c \in \{1, \dots, C\}$  by

$$\begin{aligned} \overline{W}_c^{(k+1)} = & \overline{W}_c^{(k)} - u \left[ \overline{W}_c^{(k)} - \overline{W}_{c+1}^{(k)} - 1 \right] p_{c+1}^{(k)} \overline{W}_c^{(k)} \\ & + u \left[ \overline{W}_{c-1}^{(k)} - \overline{W}_c^{(k)} - 1 \right] p_c^{(k)} \overline{W}_{c-1}^{(k)}, \end{aligned} \quad (15)$$

while for channel  $C$ ,

$$\begin{aligned} \overline{W}_C^{(k+1)} = & \overline{W}_C^{(k)} - u \left[ \overline{W}_C^{(k)} - \overline{W}_1^{(k)} - 2 \right] p_1^{(k)} \overline{W}_C^{(k)} \\ & + u \left[ \overline{W}_{C-1}^{(k)} - \overline{W}_C^{(k)} - 1 \right] p_C^{(k)} \overline{W}_{C-1}^{(k)} \end{aligned} \quad (16)$$

and for channel 1,

$$\begin{aligned} \overline{W}_1^{(k+1)} = & \overline{W}_1^{(k)} - u \left[ \overline{W}_1^{(k)} - \overline{W}_2^{(k)} - 1 \right] p_2^{(k)} \overline{W}_1^{(k)} \\ & + u \left[ \overline{W}_C^{(k)} - \overline{W}_1^{(k)} - 2 \right] p_1^{(k)} \overline{W}_C^{(k)}, \end{aligned} \quad (17)$$

where  $u[\cdot]$  is the unit-step function,  $\overline{W}_{c-1}^{(k)}$ ,  $\overline{W}_c^{(k)}$  and  $\overline{W}_{c+1}^{(k)}$  are the expected numbers of nodes at channels  $c-1$ ,  $c$  and  $c+1$  during the  $k$ th period, and  $p_c^{(k)}$  is the probability the SYNC node will successfully switch to channel  $c$  during the  $k$  period. We remark that  $p_c^{(k)}$  is smaller than unity since (typically) only a single (SYNC) node will switch channels or—in the case of interference—no node will manage to switch.

For every channel  $c \in \{1, \dots, C\}$  the transition system formed by (15) is written in matrix form as

$$\overline{\mathbf{w}}^{(k+1)} = \mathbf{G}^{(k)} \overline{\mathbf{w}}^{(k)} \quad (18)$$

with

$$\overline{\mathbf{w}}^{(k+1)} = \left[ \overline{W}_1^{(k+1)} \overline{W}_2^{(k+1)} \dots \overline{W}_{C-1}^{(k+1)} \overline{W}_C^{(k+1)} \right]^T, \quad (19)$$

$$\overline{\mathbf{w}}^{(k)} = \left[ \overline{W}_1^{(k)} \overline{W}_2^{(k)} \dots \overline{W}_{C-1}^{(k)} \overline{W}_C^{(k)} \right]^T \quad (20)$$

and

$$\mathbf{G}^{(k)} = \begin{bmatrix} 1 - g_1^{(k)} & 0 & 0 & \dots & g_C^{(k)} \\ g_1^{(k)} & 1 - g_2^{(k)} & 0 & \dots & 0 \\ 0 & g_2^{(k)} & \ddots & \dots & 0 \\ \vdots & \vdots & \ddots & 1 - g_{C-1}^{(k)} & 0 \\ 0 & 0 & 0 & g_{C-1}^{(k)} & 1 - g_C^{(k)} \end{bmatrix} \quad (21)$$

where  $\forall c < C : g_c^{(k)} = u \left[ \overline{W}_c^{(k)} - \overline{W}_{c+1}^{(k)} - 1 \right] p_{c+1}^{(k)}$  and  $g_C^{(k)} = u \left[ \overline{W}_C^{(k)} - \overline{W}_1^{(k)} - 2 \right] p_1^{(k)}$ .

The eigenvectors of the system in (18) are given by

$$\mathbf{w}^{(SS)} = \left[ \left[ \frac{W}{C} \right] \dots \left[ \frac{W}{C} \right] \right]^T. \quad (22)$$

This is because  $\mathbf{w}^{(SS)}$  vectors in the form of (22) lead to

$$\forall c : \begin{cases} u \left[ \overline{W}_c^{(SS)} - \overline{W}_{c+1}^{(SS)} - 1 \right] = 0 \\ u \left[ \overline{W}_C^{(SS)} - \overline{W}_1^{(SS)} - 2 \right] = 0 \end{cases} \\ \Rightarrow \forall c : g_c^{(SS)} = 0.$$

Thus, we have

$$\forall c : \lim_{k \rightarrow \infty} W_c^{(k)} \in \left\{ \left[ \frac{W}{C} \right], \left[ \frac{W}{C} \right] \right\}. \quad (23)$$

Note that the transition matrix  $\mathbf{G}^{(k)}$  in (21) has all its columns summing to unity, while its entries are non-negative and smaller than unity. As such, via the Perron–Frobenius theorem [49], we find that the maximum magnitude of all eigenvalues of  $\mathbf{G}^{(k)}$  is unity, that is, all eigenvalues of any instantiation of  $\mathbf{G}$  are within (or on) the unit circle. Therefore, the system in (18) will reach a vector of the form of (22), or will oscillate between multiple of these vectors (fixed points). However, no oscillations can occur, since, due to the switching rules of (4) and (5), all higher-numbered channels must contain  $\left[ \frac{W}{C} \right]$  nodes and all lower-numbered channels must contain  $\left[ \frac{W}{C} \right]$  nodes. Thus, the proposed balancing mechanism converges to a single fixed point in (22). ■

## APPENDIX B PROOF OF PROPOSITION 2

*Proof:* PCO-based synchronization is well-known to achieve convergence [33]. Hence, during the `Converging` mode, all SYNC nodes will converge to synchronous beacons across all  $C$  channels, given that their beacon packet broadcasts are only affected by other SYNC node broadcasts. PCO-based desynchronization within each channel is then equivalent to anchored desynchronization [45]. The latter is proven to converge to a steady state wherein the packet broadcasts are equidistant within the beacon period, i.e., at intervals of  $T \left( \frac{1}{W_c} \pm b_{\text{thres}} \right)$  seconds. Once this is achieved and all nodes are balanced across all channels (the latter is ensured via Proposition 1), the system moves to `Converged` mode.

Channel swapping events do not affect the converged beacon packet transmissions within each channel since: (i) nodes between unbalanced channels cannot perform swaps; (ii) swapping requests and acknowledgments are done in the guard time periods; (iii) once swapping is acknowledged, nodes broadcast their first beacon packet in their new channel at the end of the guard period. In this way, they can confirm that the node they are swapping with has left the channel. Selecting the post-beacon guard period to be smaller than  $b_{\text{thres}} T$  seconds ensures that the convergence is not disturbed by channel swapping. ■

## APPENDIX C PROOF OF PROPOSITION 3

*Proof:* The average degree of connectivity of a DESYNC node is given by the total number of connections established by DESYNC nodes divided by the total number of DESYNC nodes (i.e.,  $W - C$ ). The total number of connections is found by multiplying the number of DESYNC nodes with the number of connections established by each of them. Particularly, each DESYNC node in a channel can connect to (i) all the SYNC nodes, (ii) the remaining DESYNC nodes in the same channel and (iii) the DESYNC nodes in other balanced channels (i.e., channels with the same number of nodes) that do not have the same phase. Hence, the number of connections established by DESYNC nodes in the highest and lowest channels is

$$C_{\text{high}} W_{\text{DESYNC,high}} \times (C_{\text{high}} W_{\text{DESYNC,high}} + C_{\text{low}})$$

and

$$C_{\text{low}} W_{\text{DESYNC,low}} \times (C_{\text{low}} W_{\text{DESYNC,low}} + C_{\text{high}}),$$

respectively. Summing the above expressions and dividing by  $W - C$  leads to (11).

## APPENDIX D PROOF OF PROPOSITION 4

*Proof:* To derive the possible combinations of  $W$  nodes in  $C$  channels, we begin by assuming zero nodes in channels  $1, 2, \dots, C-1$ ; this means that all  $W$  nodes must be in channel  $C$ . If zero nodes exist in channels  $1, 2, \dots, C-2$  and one node exists in channel  $C-1$ , this means that  $W-1$  nodes must be in channel  $C$ . Continuing this expansion, we can cover all

possible cases (two nodes in channel  $C - 1$  and  $W - 2$  nodes in channel  $C$  and so on). For the non-trivial case of  $C \geq 2$  and  $W \geq 2C$ , this leads to the following summation:

$$\mathcal{C}_{W,C} = \sum_{i_1=0}^W \sum_{i_2=0}^{W-i_1} \cdots \sum_{i_{C-2}=0}^{W-\sum_{j=1}^{C-3} i_j} \binom{C-2}{\sum_{j=1}^{C-2} i_j + 1}. \quad (24)$$

By calculating the result of the series of (24), we reach (13).

Since nodes join a channel randomly, once each node makes a decision, it is a “success” or “fail” process for each channel: “success” if the node joins it, “fail” otherwise. The probability of “success” is  $\frac{1}{C}$ , while the probability of “fail” is  $\frac{C-1}{C}$ . Hence, for the first channel, the probability of combination  $i$  (out of  $\mathcal{C}_{W,C}$ ) having  $W_1(i)$  nodes (“successes”) out of  $W$  (based on the binomial distribution) is:

$$\Pr(i, \text{Ch1}) = \binom{W}{W_1(i)} \frac{(C-1)^{W-W_1(i)}}{C^W}. \quad (25)$$

For the second channel, the probability of combination  $i$  having  $W_2(i)$  nodes out of  $W - W_1(i)$  possible nodes [since we assumed that  $W_1(i)$  nodes have chosen to join the first channel] is:

$$\Pr(i, \text{Ch2}) = \binom{W - W_1(i)}{W_2(i)} \frac{(C-1)^{W-W_1(i)-W_2(i)}}{C^{W-W_1(i)}}. \quad (26)$$

Iterating this for all channels, we can derive in a similar fashion  $\Pr(i, \text{Ch3}), \dots, \Pr(i, \text{Ch}\{C-1\})$ . The remaining number of nodes, i.e.,  $\left[ W - \sum_{c=1}^{C-1} W_c(i) \right]$  nodes, will be joining channel  $C$  with probability  $\Pr(i, \text{Ch}\{C\}) = 1$ . Since these probabilities are independent, the probability of combination  $i$  having the node distribution:  $[W_1(i) \dots W_C(i)]$  in channels  $1, \dots, C$  is given by (14). Notice that the assumption of nodes deciding first on whether to join channel 1, then whether to join channel 2, etc., is not restrictive. In fact, the above analysis can be expressed with any order of channels without affecting the result. In other words, the numbering of channels stated above has no effect on  $\Pr(i)$ .

We can then estimate the expected delay until convergence via (12), with the expression in the maximization of (12) establishing the largest imbalance of the node distribution of combination  $i$  from the average number of nodes per channel,  $\left\lfloor \frac{W}{C} \right\rfloor$ . This expresses the channel that will experience the highest number of channel switches until convergence (each requiring  $N_e$  periods for SYNC node election and  $N_c$ ). ■

## REFERENCES

- [1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A survey on wireless multimedia sensor networks,” *Computer networks*, vol. 51, no. 4, pp. 921–960, 2007.
- [2] Y.-C. Tseng, Y.-C. Wang, K.-Y. Cheng, and Y.-Y. Hsieh, “iMouse: an integrated mobile surveillance and wireless sensor system,” *IEEE Computer*, vol. 40, no. 6, pp. 60–66, 2007.
- [3] N. Deligiannis, F. Verbist, J. Slowack, R. v. d. Walle, P. Schelkens, and A. Munteanu, “Progressively refined wyner-ziv video coding for visual sensors,” *ACM Transactions on Sensor Networks*, vol. 10, no. 2, p. 21, 2014.
- [4] J. Xu, Y. Andreopoulos, Y. Xiao, and M. van der Schaar, “Non-stationary resource allocation policies for delay-constrained video streaming: Application to video over internet-of-things-enabled networks,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 782–794, 2014.
- [5] A. Vinel, E. Belyaev, K. Egiastian, and Y. Koucheryav, “An overtaking assistance system based on joint beaconing and real-time video transmission,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 5, pp. 2319–2329, 2012.
- [6] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, “Distributed approach for coverage and patrolling missions with a team of heterogeneous aerial robots under communication constraints,” *International Journal on Advanced Robotic Systems*, vol. 10, no. 28, 2013.
- [7] L. Merino, A. Gilbert, J. Capitán, R. Bowden, J. Illingworth, and A. Ollero, “Data fusion in ubiquitous networked robot systems for urban services,” *Annales des Télécommunications*, vol. 67, no. 7-8, pp. 355–375, 2012.
- [8] M. Gerla and L. Kleinrock, “Vehicular networks and the future of the mobile internet,” *Computer Networks*, vol. 55, no. 2, pp. 457–469, 2011.
- [9] N. Alam, A. T. Balaie, and A. G. Dempster, “A dsrc-based traffic flow monitoring and lane detection system,” in *Proc. 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*. IEEE, 2011, pp. 1–5.
- [10] Y.-Y. Shih, W.-H. Chung, P.-C. Hsiu, and A.-C. Pang, “A mobility-aware node deployment and tree construction framework for zigbee wireless networks,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2763–2779, 2013.
- [11] N. Deligiannis, F. Verbist, J. Barbarien, J. Slowack, R. Van de Walle, P. Schelkens, and A. Munteanu, “Distributed coding of endoscopic video,” in *IEEE International Conference on Image Processing (ICIP’11)*. IEEE, 2011, pp. 1813–1816.
- [12] Y. Andreopoulos and M. Van der Schaar, “Adaptive linear prediction for resource estimation of video decoding,” *IEEE Trans. on Circ. and Syst. for Video Technol.*, vol. 17, no. 6, pp. 751–764, 2007.
- [13] N. Kontorinis, Y. Andreopoulos, and M. Van Der Schaar, “Statistical framework for video decoding complexity modeling and prediction,” *IEEE Trans. on Circ. and Syst. for Video Technol.*, vol. 19, no. 7, pp. 1000–1013, 2009.
- [14] H. Besbes, G. Smart, D. Buranapanichkit, C. Kloukinas, and Y. Andreopoulos, “Analytic conditions for energy neutrality in uniformly-formed wireless sensor networks,” *IEEE Trans. Wireless Comm.*, vol. 12, no. 10, pp. 4916–4931, 2013.
- [15] G. Smart, N. Deligiannis, Y. Andreopoulos, R. Surace, V. Loscri, and G. Fortino, “Decentralized time-synchronized channel swapping for wireless sensor networks,” in *11th European Conference on Wireless Sensor Networks (EWSN’14), poster presentation*, 2014.
- [16] D. Buranapanichkit, N. Deligiannis, and Y. Andreopoulos, “Convergence of desynchronization primitives in wireless sensor networks: A stochastic modeling approach,” *IEEE Trans. on Signal Process.*, vol. 63, no. 1, pp. 221–233, 2015.
- [17] N. Deligiannis, J. F. Mota, G. Smart, and Y. Andreopoulos, “Decentralized multichannel medium access control: Viewing desynchronization as a convex optimization method,” in *Proc. 14th Int. Conf. on Inf. Process. in Sensor Net. (IPSN 2015)*. ACM, 2015, pp. 13–24.
- [18] —, “Fast desynchronization for decentralized multichannel medium access control,” *IEEE Trans. on Commun.*, vol. 63, no. 9, pp. 3336–3349, 2015.
- [19] S. Pollin, M. Ergen, M. Timmers, A. Dejonghe, L. Van der Perre, F. Cathoor, I. Moerman, and A. Bahai, “Distributed cognitive co-existence of 802.15. 4 with 802.11,” in *International Conference on Cognitive Radio Oriented Wireless Networks and Communications*. IEEE, 2006, pp. 1–5.
- [20] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, and K. Pister, “Openwsn: a standards-based low-power wireless development environment,” *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 5, pp. 480–493, 2012.
- [21] G. Wu, S. Talwar, K. Johansson, N. Himayat, and K. D. Johnson, “M2m: From mobile to embedded internet,” *IEEE Communications Magazine*, vol. 49, no. 4, pp. 36–43, 2011.
- [22] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, “Home m2m networks: architectures, standards, and qos improvement,” *IEEE Communications Magazine*, vol. 49, no. 4, pp. 44–52, 2011.
- [23] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [24] G. Mulligan, “The 6lowpan architecture,” in *Proc. 4th Workshop on Embedded Networked Sensors*. ACM, 2007, pp. 78–82.
- [25] A. Tinka, T. Watteyne, and K. Pister, “A decentralized scheduling algorithm for time synchronized channel hopping,” in *Ad Hoc Networks*. Springer, 2010, pp. 201–216.
- [26] J. B. Kenney, “Dedicated short-range communications (dsrc) standards in the united states,” *Proc. of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

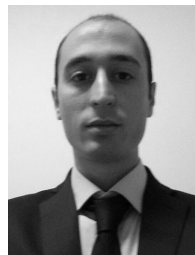
- [27] IEEE 802.15.4e-2012, "IEEE Standard for Local and Metropolitan Area Networks. Part 15.4: Low-Rate Wireless Personal Area Networks (LRWPANs) Amendment 1: MAC Sublayer," *IEEE Std. 802.15.4e*, April 2012.
- [28] P. Zand, S. Chatterjea, K. Das, and P. Havinga, "Wireless industrial monitoring and control networks: The journey so far and the road ahead," *Journal of Sensor and Actuator Networks*, vol. 1, no. 2, pp. 123–152, 2012.
- [29] IEEE 802.11p, "IEEE Standard for Information Technology Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks Specific Requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications; Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std. 802.11p*, July 2010.
- [30] P. Thubert, T. Watteyne, M. R. Palattella, X. Vilajosana, and Q. Wang, "IETF 6TSCH: Combining IPv6 connectivity with industrial performance," in *Proc. IEEE 2013 7th Int. Conf. on Innov. Mob. and Internet Serv. in Ubiqu. Comput. (IMIS)*. IEEE, 2013, pp. 541–546.
- [31] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi, "Dissemination and harvesting of urban data using vehicular sensing platforms," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 882–901, 2009.
- [32] J. Degeys, I. Rose, A. Patel, and R. Nagpal, "Desync: self-organizing desynchronization and tdma on wireless sensor networks," in *Proc. 6th International Conference on Information Processing in Sensor Networks (IPSN'07)*. ACM, 2007, pp. 11–20.
- [33] R. Pagliari, Y.-W. P. Hong, and A. Scaglione, "Bio-inspired algorithms for decentralized round-robin and proportional fair scheduling," *IEEE J. on Select. Areas in Commun.*, vol. 28, no. 4, pp. 564–575, May 2010.
- [34] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "EM-MAC: a dynamic multichannel energy-efficient MAC protocol for wireless sensor networks," in *Proc. 12th ACM Int. Symp. on Mob. Ad Hoc Netw. and Comput.*, ser. MobiHoc '11. New York, NY, USA: ACM, 2011, pp. 23:1–23:11.
- [35] H. K. Le, D. Henriksson, and T. Abdelzaher, "A practical multi-channel media access control protocol for wireless sensor networks," in *Proc. of the 7th international conference on Information processing in sensor networks*, 2008, pp. 70–81.
- [36] S. M. George, W. Zhou, H. Chenji, M. Won, Y. O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah, "Distressnet: a wireless ad hoc and sensor network architecture for situation management in disaster response," *IEEE Communications Magazine*, vol. 48, no. 3, pp. 128–136, 2010.
- [37] D. Buranapanichkit and Y. Andreopoulos, "Distributed time-frequency division multiple access protocol for wireless sensor networks," *IEEE Wirel. Comm. Lett.*, vol. 1, no. 5, pp. 440–443, Oct. 2012.
- [38] J. Hwang, T. Kim, J. So, and H. Lim, "A receiver-centric multi-channel mac protocol for wireless networks," *Computer Communications*, 2012.
- [39] Y. Kim, H. Shin, and H. Cha, "Y-MAC: An energy-efficient multi-channel mac protocol for dense wireless sensor networks," in *Proc. of the 7th international conference on Information processing in sensor networks*, 2008, pp. 53–63.
- [40] M. J. Miller and N. H. Vaidya, "A mac protocol to reduce sensor network energy consumption using a wakeup radio," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 228–242, 2005.
- [41] M. Ali, T. Suleman, and Z. A. Uzmi, "Mmac: A mobility-adaptive, collision-free mac protocol for wireless sensor networks," in *IEEE International Performance, Computing, and Communications Conference (IPCCC'05)*. IEEE, 2005, pp. 401–407.
- [42] T. Luo, M. Motani, and V. Srinivasan, "Cam-mac: A cooperative asynchronous multi-channel mac protocol for ad hoc networks," in *International Conference on Broadband Communications, Networks and Systems (BROADNETS'06)*. IEEE, 2006, pp. 1–10.
- [43] J. Degeys and R. Nagpal, "Towards desynchronization of multi-hop topologies," in *Proc. IEEE Internat. Conf. on Self-Adaptive and Self-Organizing Syst. (SASO'08)*, 2008, pp. 129–138.
- [44] A. Patel, J. Degeys, and R. Nagpal, "Desynchronization: The theory of self-organizing algorithms for round-robin scheduling," *Proc. IEEE Internat. Conf. on Self-Adaptive and Self-Organizing Syst. (SASO'07)*, July 2007.
- [45] C.-M. Lien, S.-H. Chang, C.-S. Chang, and D.-S. Lee, "Anchored desynchronization," in *Proc. IEEE INFOCOM'12*, 2012, pp. 2966–2970.
- [46] J. Klinglmayr and C. Bettstetter, "Self-organizing synchronization with inhibitory-couples oscillators: convergence and robustness," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 7, no. 3, Sep. 2012.
- [47] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. Pister, "A realistic energy consumption model for tsch networks," *IEEE Sensor Journal*, 2013.
- [48] C. A. Boano, T. Voigt, C. Noda, K. Romer, and M. Zúñiga, "Jamlab: Augmenting sensor network testbeds with realistic and controlled interference generation," in *2011 10th International Conference on Information Processing in Sensor Networks (IPSN'11)*. IEEE, 2011, pp. 175–186.
- [49] V. G. Kulkarni, *Modeling and analysis of stochastic systems*. CRC Press, 1995, vol. 36.



**George Smart** (S'13) is a PhD student in the Department of Electronic and Electrical Engineering at University College London (UK). His research interests are in wireless sensor networks, embedded systems software, digital signal processing and hardware design.



**Nikos Deligiannis** (S'08, M'10) is assistant professor with the Electronics and Informatics (ETRO) department, Vrije Universiteit Brussel (VUB). He received the Diploma in electrical and computer engineering from University of Patras, Greece, in 2006, and the PhD in applied sciences (awarded with Highest Honors and congratulations from the Jury) from Vrije Universiteit Brussel, Belgium, in 2012. From Jun. 2012 to Sep. 2013, he was a postdoctoral researcher with the Department of Electronics and Informatics, Vrije Universiteit Brussel. From Oct. 2013 to Feb. 2015, he was working as postdoctoral research associate at the Department of Electronic and Electrical Engineering at University College London, UK. During that period, he was also acting as technical consultant on big visual data technologies at the British Academy of Film and Television Arts (BAFTA), UK. His current research interests include big data processing, compressed sensing, ad hoc networking for the internet-of-things, distributed computing, and visual search. Prof. Deligiannis has authored over 60 journal and conference publications, book chapters, and holds two patent applications (one owned by iMinds, Belgium and the other by BAFTA, UK). He was a recipient of the 2011 ACM/IEEE International Conference on Distributed Smart Cameras Best Paper Award and the 2013 Scientific Prize FWO-IBM Belgium.



**Rosario Surace** obtained his Ph.D. in Systems and Computer Science Engineering at the Department of Computer Science, Modeling, Electronics and Systems Engineering (DIMES), University of Calabria (Italy) in 2015, with a thesis on "Design and Performance Evaluation of Algorithm for Wireless Self-Organizing Systems".

From October 2012 to March 2013 he was a visiting researcher at Electronic and Electrical Engineering Department, University College London (UCL).

Since September 2013, he works as a Technical Services Engineer at RAI - Radiotelevisione Italiana.





**Valeria Loscri** is a permanent researcher of the FUN Team in Inria Lille Nord Europe since October 2013. She got her Masters degree in Computer Science and PhD in Systems Engineering and Computer Science in 2003 and 2007 respectively, both at University of Calabria (Italy). In 2006 she spent 6 months as visiting researcher at Rice University under the supervision of Prof. Knightly, where she worked on the MAC layer of wireless mesh networks. She authored more than 60 publications in journals, conferences, workshops and book chapters. She is involved in

several programs and organization committees such as SWANSITY, PASC workshop, AdHocNow. Her research interests focus on performance evaluation, self-organizing systems, robotics networks, nanocommunications, visual communication paradigm.



**Giancarlo Fortino** (SM12) received the Laurea (B.S and M.S) and Ph.D in computer engineering from the University of Calabria (Unical), Italy, in 1995 and 2000, respectively. He is currently an Associate Professor of Computer Engineering (since 2006) with the Department of Informatics, Modeling, Electronics and Systems (DIMES) at Unical. He holds the scientific Italian national habilitation for Full Professorship (since 2013) and, since 2015, is Adjunct Professor of Computer Engineering at the Wuhan University of Technology (China) and

Adjunct Senior Research Fellow at the Italian National Research Council. He authored more than 250 publications in journals, conferences, and books. His research interests include distributed computing, wireless sensor networks, software agents, cloud computing, and Internet of Things. He is currently an associate editor of IEEE Trans. on Affective Computing, IEEE Trans. on Human-Machine Systems, Information Fusion, Engineering Application of Artificial Intelligence, Journal of Network and Computer Applications. He is co-founder and CEO of Sensyscal S.r.l., a Unical spin-off, engaged in advanced applied research and development of IoT-based systems.



**Yiannis Andreopoulos** (M'00, SM'14) is Reader (Assoc. Professor) in Data and Signal Processing Systems in the Department of Electronic and Electrical Engineering of University College London (U.K.). His research interests are in wireless sensor networks, error-tolerant computing and multimedia systems.

He received the 2007 Most-Cited Paper award from the Elsevier EURASIP Signal Processing: Image Communication journal and a best-paper award from the 2009 IEEE Workshop on Signal Processing

Systems. Dr. Andreopoulos was Special Sessions Co-chair of the 10th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2009) and Programme Co-chair of the 18th International Conference on Multimedia Modeling (MMM 2012) and the 9th International Conference on Body Area Networks (BODYNETS 2014). He is (or has been) Associate editor of the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE SIGNAL PROCESSING LETTERS and the Elsevier Image and Vision Computing journal.