POLITECNICO DI TORINO
Repository ISTITUZIONALE

Assessing the Effectiveness of Channel Hopping in IEEE 802.15.4 TSCH Networks

Original
Assessing the Effectiveness of Channel Hopping in IEEE 802.15.4 TSCH Networks / Cena, G., Scanzio, S., Ghazivakili, M., Demartini, C.G., Valenzano, A.. - In: IEEE OPEN JOURNAL OF THE INDUSTRIAL ELECTRONICS SOCIETY. - ISSN 2644-1284. - 4:(2023), pp. 214-229. [10.1109/OJIES.2023.3287943]

Availability:
This version is available at: 11583/2992004 since: 2024-08-28T08:53:26Z

Publisher:
IEEE

Published
DOI:10.1109/OJIES.2023.3287943

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

18 November 2026

IEEE Open Journal of the
Industrial Electronics Society

Received 7 May 2023; accepted 17 June 2023. Date of publication 20 June 2023;
date of current version 3 July 2023. The review of this article was arranged by Associate Editor Gerhard Hancke.

Digital Object Identifier 10.1109/OJIES.2023.3287943

Assessing the Effectiveness of Channel Hopping in IEEE 802.15.4 TSCH Networks

GIANLUCA CENA [1] (Senior Member, IEEE), STEFANO SCANZIO [1] (Senior Member, IEEE), MOHAMMAD GHAZI VAKILI[2,3], CLAUDIO GIOVANNI DEMARTINI [4] (Senior Member, IEEE), AND ADRIANO VALENZANO [1] (Senior Member, IEEE)

[1]National Research Council of Italy (CNR-IEIIT), I-10129 Torino, Italy
[2]Department of Computer Science, University of Toronto, Toronto, ON M5S 3G4, Canada
[3]Department of Chemistry, University of Toronto, Toronto, ON M5S 3H6, Canada
[4]Dipartimento di Automatica e Informatica, Politecnico di Torino, I-10129 Torino, Italy

CORRESPONDING AUTHOR: STEFANO SCANZIO (e-mail: stefano.scanzio@cnr.it)

ABSTRACT Time slotted channel hopping (TSCH) is an enhanced access mechanism for IEEE 802.15.4 that improves many aspects of the original protocol, like determinism and power consumption. Besides preventing collisions, time slotting allows motes to stay asleep for most of the time. In addition, channel hopping achieves more stable communication, in terms of packet losses and transmission latency, in spite of wireless spectrum variations. In this article, the specific effects of the latter mechanism are investigated from both qualitative and quantitative points of view. In particular, a thorough experimental campaign was carried out on real devices, deployed in a real environment and performing realistic data exchanges, when they are exposed to different interference conditions, to assess the benefits it brings over the case when transmissions are performed on a fixed-channel (FC). Experimental results confirm that, when the transmission frequency is kept repeatedly changing, communication quality is sensibly less affected by colocated Wi-Fi infrastructures, which makes the network behavior more predictable, and hence, intrinsically more dependable.

INDEX TERMS Channel hopping, IEEE 802.15.4, Industrial Internet of Things, time slotted channel hopping (TSCH), wireless sensor networks (WSNs).

I. INTRODUCTION

Wireless sensor networks (WSN) are mainly characterized by low power consumption, which makes them suitable for battery-powered devices. Among the many technologies for which products are currently available off-the-shelf, IEEE 802.15.4 [1] certainly plays a relevant role, and can be used as the basis for implementing real solutions (e.g., ZigBee [2]) that can be readily embedded in both new designs and legacy systems. Other relevant technologies include, for instance, Bluetooth low energy (BLE) [3], LoRaWAN [4], IEEE 802.11ah (Wi-Fi "HaLow") [5], and near field communication (NFC). Although they are often seen as direct competitors, every technology shows peculiar features, which make it optimal for specific application scenarios. A distinctive feature of IEEE 802.15.4 is that, probably, it is the only widespread solution for which both hardware (chipsets, evaluation boards, etc.) and software (protocol stacks, often released as open source) are currently available that support meshing out-of-the-box. By exploiting forwarding techniques, wide areas can be covered without the need to explicitly deploy a network infrastructure.

The basic, non-beaconed IEEE 802.15.4 medium access control (MAC) mechanism operates asynchronously: nodes access the shared communication support autonomously (without the need to ask for prior permission) when they need to do so. This leads to two main drawbacks. First, since nodes can start a transmission on air at any time, receivers must be always on, which fairly increases power consumption. Second, relying on random access unavoidably leads to collisions, which worsen communication reliability. The latter issue can be tackled with automatic repeat request (ARQ) mechanisms [6], but they waste time, spectrum, and energy. At the limit, in large networks with many connected devices, network congestion may sometimes occur, where

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

communication is (almost) prevented for a while. In turn, this may cause misbehavior of applications, hence leading to system malfunction. A beacon-enabled operating mode is defined by legacy IEEE 802.15.4 that relies on superframes and guaranteed time slots, but it offers limited features and is not always supported by commercial solutions. Moreover, the lack of frequency diversity may occasionally lead to intolerably large latency and losses, as a consequence of narrowband disturbance or when entering a deep fade condition.

To cope with aforementioned issues, time slotted channel hopping (TSCH) was first defined in the IEEE 802.15.4e amendment in 2012 [7], along with deterministic and synchronous multichannel extension (DSME) and low-latency deterministic network (LLDN), and in 2015, it was rolled into the main specification [1]. TSCH belongs to the data-link layer, and only deals with single-hop frame transmissions between neighbor nodes that can communicate directly. Multihop transmission between any pair of nodes in a mesh network is usually accomplished by routing protocols located at the network layer. This is the case of the IPv6 routing protocol for low-power and lossy networks (RPL), which relies on the prior activities of the IETF 6LoWPAN working group whose outcomes helped laying the ground for the Internet of Things (IoT). Protocol suites have been recently defined that achieve ubiquitous connectivity and predictable behavior, like IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) [8]. 6TiSCH is a strong candidate for connecting devices at the perception layer in Industrial IoT (IIoT) architectures, where information are collected from sensors and transferred for further processing.

This article elaborates on the benefits brought by channel hopping (CH) in TSCH, focusing on the improvements it brings over pure time slotted solutions. As will be shown, CH is not meant to increase, in absolute terms, the quality of communication perceived by motes. Instead, it makes them see equivalent links whose quality roughly coincides with the average of what they would observe if their radio module were separately tuned on every one of the channels managed by TSCH. Experimental results confirm the more predictable nature of TSCH, which can be profitably exploited to tangibly improve determinism of distributed sensing/control systems in typical environmental conditions.

The rest of this article is organized as follows. Section II reports on the recent advances that concern TSCH, whereas Section III summarizes TSCH basics and provides some preliminary discussions on how CH affects performance. The experimental testbed and the performance indicators used to evaluate communication quality are presented in Section IV, while experimental results for three measurement campaigns, along with the final discussion, are provided in Section V. Finally, Section VI concludes this article.

II. STATE OF THE ART

Many aspects about TSCH behavior have been studied in depth in the recent scientific literature. For example, in [9], black and white listing were considered to increase reliability. Similarly, in [10] and [11], probabilistic black listing was taken into account. Several articles take advantage of the division of time into slots performed by TSCH to analyze traffic scheduling strategies [12], [13]. Others are more focused on power consumption [14], [15]. In particular, in [16], proactive reduction of idle listening is presented and its behavior assessed. In [17], the effect of TSCH configuration parameters was analyzed with respect to reliability, communication latency, and power consumption. Other mathematical models, specific for shared cells and based on Markov chains, were presented in [18] and [19]. In [20], the interference between colocated TSCH networks was studied, whereas techniques to improve coexistence of TSCH and Wi-Fi were proposed and experimentally evaluated in [21]. In [22], the effects of Wi-Fi on TSCH were measured in a controlled environment, which did not require interleaving. Finally, in [23], redundancy was envisaged to improve both reliability and latency.

Among the works that aim to experimentally assess the benefits CH brings to IEEE 802.15.4, the most relevant is probably [24]. In that article, which precedes the definition of TSCH, the interleaving technique was used to collect information about the quality of communication of the links between 46 motes deployed in a real environment, on every channel in the 2.4-GHz industrial, scientific, and medical (ISM) band. In particular, the packet delivery ratio was evaluated. Starting from such a thorough spectrum characterization, the effects of CH were analyzed a posteriori, including multihop communication based on the gradient routing. A similar technique was employed in [25] and [26], which focus on single hops and also consider the IEEE 802.15.4g OFDM-based PHY in the sub-GHz band. The related datasets include more than one hundred million samples, which enable effective postanalysis for a variety of purposes.

The main differences between the aforementioned approaches and what we did in our work are as follows:

1) We evaluated the link behavior in typical WSN operating conditions, when packets are spaced by minutes (and not tens of milliseconds), to avoid overloading the wireless spectrum;
2) We relied on the standard MAC retransmission mechanism, thus leveraging time-frequency diversity as implemented in real TSCH devices;
3) We measured the transmission latency, and from it, we inferred the link failure rate (and not vice versa);
4) We analyzed CH effectiveness versus interfering Wi-Fi traffic (and not against multipath fading).

Since the same communication protocol has been studied using two completely different methodologies, obtaining similar results should not be taken for granted.

The testbed we employed resembles the one in [27]. However, that work was aimed at defining a mathematical model for the statistical distribution of latency on a TSCH link, whereas this one explicitly focuses on the comparison between fixed channel (FC) transmission and CH when the amount of Wi-Fi interference varies. To reliably assess the improvements brought by the latter mechanism in spite of

the intrinsic variability of the wireless spectrum, the use of interleaving is demanded.

III. TIME SLOTTED CHANNEL HOPPING (TSCH)

TSCH is an enhanced MAC mechanism for IEEE 802.15.4 that combines two distinct but interrelated MAC mechanisms: time slotting and CH.

A. TIME SLOTTING

Time slotting belongs to the class of time division multiple access (TDMA) [28] approaches, where the transmission medium is shared among the nodes belonging to the same network on the basis of time. Nodes in TDMA are assigned specific and nonoverlapping time windows in which they can freely and exclusively access the medium, as opposed to carrier sense multiple access (CSMA) mechanisms where no preliminary agreement exists. The easiest way to do so is to split time into periodically recurring statically defined structures, customarily denoted superframes, which are the basis for bandwidth allocation to nodes.

Time slotting is a specific flavor of TDMA that foresees that time is split into elementary intervals, denoted *timeslots* (or just slots), which have exactly the same duration. Every slot accommodates one frame exchange, either acknowledged or not. Therefore, its duration must account for one data frame with maximal payload plus the related acknowledgment frame (as well as interframe spaces). While this approach is in general not very efficient from the point of view of bandwidth allocation, the limited maximum payload foreseen by IEEE 802.15.4 and the fact that the main focus is on low power consumption make it perfectly acceptable.

According to time-slotting rules, frame transmissions take place on *links*, each of which is characterized by the addresses of the involved source and destination nodes and is assigned a slot identified by its relative position within the slotframe (SlotOffset). In stable operating conditions, i.e., when all nodes belonging to the network are time synchronized, this mechanism is able to prevent collisions among them. To this purpose, a suitable network-wide schedule must be defined in advance for exchanges so that every slot in the slotframe is allocated for transmission to one node at most (non-shared links). In practice, every node has to maintain only its view of the schedule, restricted to those slots it is interested to either read or write. The properties of solutions based on TDMA and time slotting are well known, and have been largely analyzed and debated in the scientific literature [29], [30]. For this reason, they will not be considered further in the following.

B. CHANNEL HOPPING (CH)

A second technique is additionally foreseen by TSCH, namely CH. It is worth remarking that setting up a WSN that only exploits time slotting can be easily achieved on real TSCH equipment by enforcing it to operate on an FC (e.g., by redefining the hopping sequence so that all channels coincide, or by setting the length of such a sequence to one). Conversely, CH can be hardly implemented without a shared reference grid for time (like the one provided by time slotting). In fact, having the nodes in the network jointly and orderly wandering among channels in a seemingly random way (which in reality takes place according to a predefined pattern) requires not only cooperation, but also strict time synchronization among them.

Timings for CH in TSCH come for free from the time slotting mechanism. In fact, the transition between any slot and the next one also drives frequency switching. Moreover, the *absolute slot number* (ASN) globally maintained by TSCH for the whole network is also used to decide on which physical channel every frame transmission actually takes place. ASN is a counter encoded on 5 bytes, that is initialized to 0 at network formation and is increased by one on every slot. Every node updates its own copy of ASN separately, but all copies remain coherent because they have the same synchronized view of time. In addition, the ASN value is included by nodes in enhanced beacons, which are control frames that are sent periodically in every slotframe, hence allowing new nodes to (re-)synchronize.

Concerning access rules, a channel offset parameter (ChOffset) is additionally defined on a per-link basis. It permits a number of frames, characterized by different transmitters and receivers, to be sent in the same slot (identified by SlotOffset) but using different frequencies. Overall, the TSCH schedule can be seen as a matrix where every cell corresponds to a link. The column and row of any cell represent the related slot and channel offsets, respectively. According to the IEEE 802.15.4 specification, the physical channel (PhyCh) on which frame transmission for any given link is performed is computed as

$$\text{PhyCh} = \text{HopSeqList}[(\text{ASN} + \text{ChOffset})\%\text{HopSeqLen}]$$

where symbol % denotes the modulo operator, HopSeqList is the pseudorandom sequence of channels used for hopping, and HopSeqLen is its length. As an example, Fig. 1 depicts a link described by the cell at coordinates SlotOffset $= 3$, ChOffset $= 2$ in the TSCH matrix, when the slotframe includes 101 slots (edges between adjacent slotframes have been emphasized using thicker lines) and the hop sequence list of the current OpenWSN 6TiSCH implementation is considered.

CH brings two main benefits to the basic time-slotting mechanism. First, it increases the network bandwidth available on the whole to applications, by enabling several nodes (up to the number of available physical channels) to perform transmissions concurrently (at any time, the bandwidth available to any single node remains clearly the same). In the case of the ISM band operating at 2.4 GHz, up to 16 channels are available, which increases the peak overall throughput from 250 kb/s to 4 Mb/s, not considering spatial channel reuse. This means that either larger networks (including a higher number of devices), or shorter sampling periods (enabling a finer monitoring of the physical system) are supported.

Second, it improves resilience against narrowband interference and disturbance, that is, radiation phenomena that affect a single WSN channel or, at worst, very few adjacent ones,

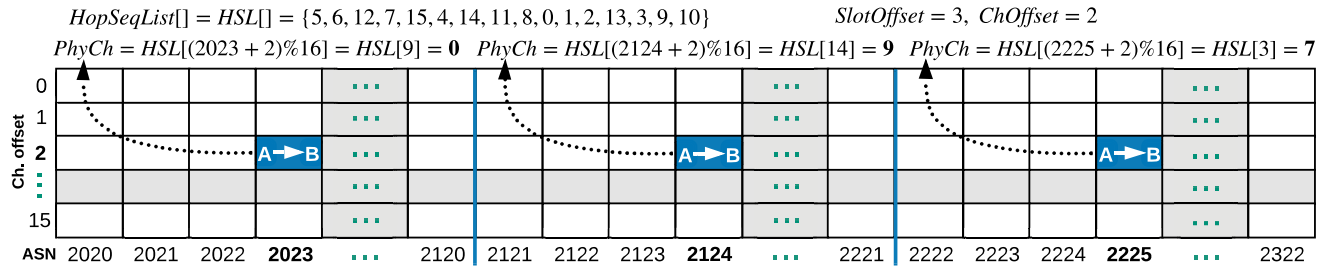


FIGURE 1. CH behavior: The physical channel on which subsequent transmissions of the link A → B are performed keeps changing.

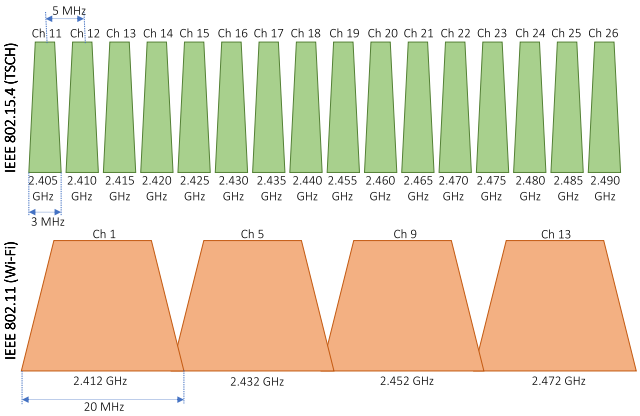


FIGURE 2. Interference between IEEE 802.15.4 (TSCH) and IEEE 802.11 (Wi-Fi) in the 2.4-GHz ISM band.

like the traffic on colocated Wi-Fi networks. In fact, having subsequent transmissions (and retransmissions) on the same link occurring on different physical channels decreases the likelihood that a specific frame is definitely lost (more on this later). Fig. 2 highlights how the 16 channels of IEEE 802.15.4 overlap with channels 1, 5, 9, and 13 of Wi-Fi in the ISM band. As can be seen, a single Wi-Fi channel interferes with four adjacent IEEE 802.15.4 channels.

Basically, the joint use of time slotting and CH in TSCH enables mixed time-frequency diversity [6] in IEEE 802.15.4 and makes transmission attempts for the same frame, as well as subsequent packets in the same stream, as independent as possible.

In this article, we are interested in the second aspect, as the former mainly concerns configuration of the TSCH matrix and consists in finding the optimal schedule for transmissions. In the following, we will consider how CH contributes to the behavior of real slotted networks. Nonslotted solutions, like the non-beaconed IEEE 802.15.4 mode, were not taken into account, as the behavior of CSMA/CA differs noticeably from TDMA [31].

C. EFFECTS OF CH IN TSCH

Quantitative assessment of CH effects can be performed in terms of a number of performance indicators about the *quality of communication* (QoC) experienced by a pair of nodes involved in the transmission of a data stream over a link. In this article, QoC is mainly defined in terms of reliability and timeliness. When dealing with distributed systems where devices are interconnected by a digital network, communication *reliability* is the probability for a packet exchange to succeed, which empirically coincides with the fraction of transmitted packets that are not lost, evaluated over a long enough time span. Concerning *timeliness*, we have to distinguish between one-way and round-trip latency. The former consists in the time elapsing between the transmission request for a packet in the originating node and its reception on the target node, while the latter measures the time between a request and the related response, both taken on the initiator. Because of the erraticness of the wireless spectrum, latency characterization needs to be performed on a statistical basis.

As we will see, CH does not necessarily improve QoC performance indicators for links. In fact, in the unusual (but not completely unrealistic) case where spectrum conditions do not vary appreciably over time, performing a preliminary channel scan and tuning transmissions on a fixed frequency, chosen among the less crowded and less disturbed ones, is likely to provide better performance than CH. Much more realistically, adaptive mechanisms can be exploited that from time to time evaluate the conditions of a group of channels and select the best one. Besides articles available in the scientific literature [32], [33], [34], the ZigBee specification [2] as well foresees a frequency agility feature. The main problem, in these cases, is the need to evaluate spectrum conditions repeatedly, either periodically (which implies increased power consumption for motes) or on demand, when the QoC perceived by the motes drops below an acceptable threshold, and to propagate this information to all network nodes. In the meanwhile, applications may suffer from degraded (and possibly inadequate) network performance, until a new channel is found [35]. Moreover, in large mesh networks, it is unlikely that a single frequency exists that ensures optimal performance over the whole area covered by the WSN.

A distinctive advantage of CH over adaptive approaches is its inherent simplicity. In fact, the transmission channel keeps changing continuously, irrespective of the actual spectrum conditions. Therefore, this mechanism has zero intervention time and does not drain additional energy for channel management. Generally speaking, what we expect from CH is not, in absolute terms, higher communication reliability or lower latency with respect to pure time slotting performed on an

FC. Instead, it ensures on average good performance out-of-the-box, which likely will remain stable over time without requiring any specific (re)configuration based on environmental conditions. Performance of CH can in theory be improved through black- and white-listing techniques [9], which dynamically change the hopping sequence to exclude badly behaving channels. In practice, these mechanisms are quite complex and not in widespread use in existing implementations (for instance, they have not been explicitly included in 6TiSCH).

In TSCH, what actually matters is the overall amount of interference and disturbance that impact on the network (all channels considered), and not the extent to which every single channel is affected. This makes it particularly resilient in those cases where, e.g., the operating frequencies of colocated Wi-Fi infrastructures may vary, either automatically (smart access points) or by hand (IT managers). Or, more simply, where (moving) users can access the Internet (or the local intranet) through different access points (AP) tuned on distinct channels so that the shape of the traffic (i.e., how it is distributed over the spectrum) may change over time.

IV. EXPERIMENTAL EVALUATION

A nonnegligible problem when measuring the performance of wireless communication technologies in the real world is that the conditions of the surrounding environment keep changing unpredictably, hence undermining severely not only reproducibility, but also repeatability of experiments. This means that performing several times the same experiment, in the same place and with the same devices deployed in the same positions, will generally lead to different results.

Performing experiments in a radio-frequency (RF) anechoic chamber by injecting selected traffic patterns solves the problem about repeatability. However, doing so hides the unpredictability of real environments, which is intrinsic of any wireless technologies and represents their most interesting aspect. While a methodology based on real traffic capture and replay could be exploited, reproducing the effects of a plurality of aspects (e.g., position and strength of interferers, including multipath propagation) would require an overly complex setup, well beyond our means and goals.

For this reason, we resorted to a simple yet effective method, already exploited in [36]. We set up multiple different experiments that run (almost) in parallel on similar networks, made up of identical motes deployed approximately in the same positions. Thus, the radio signals they send and receive propagate over very similar paths and suffer from about the same interference and disturbance coming from the surroundings. In our case, we analyzed a pair of TSCH links, each one between exactly two motes. Experiments are then interleaved according to an $ABAB\ldots AB$ scheme, in such a way that transmission on air is only enabled on one link at a time. Statistics about communication quality evaluated this way can be realistically thought of as if frame transmissions on the different links were performed in approximately the same, although (partially) unknown, spectrum conditions.

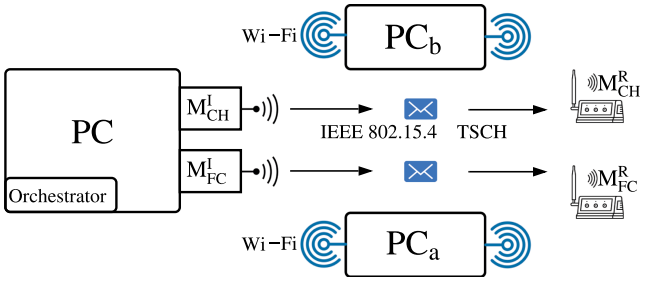


FIGURE 3. Testbed used for the experimental campaigns.

A. EXPERIMENTAL SETUP

The setup we used to assess the effects of CH is shown in Fig. 3, and consists of two pairs of motes: M^I_{CH} and M^I_{FC} are directly connected to the same Linux PC and act as initiators, while M^R_{CH} and M^R_{FC} are located about 2-m apart from initiators and act as responders. Subscript CH identifies the pair of motes where CH was enabled (all 16 channels in the 2.4-GHz band were used), whereas FC denotes the pair where it was disabled (so that all transmissions took place on an FC). The slotframe size was left to the default value (101 slots), and so its duration was $T_{sf} = 2.02$ s.

We relied on OpenMote B devices [37] running the OpenWSN [38] REL-1.24.0 operating system. Such motes employ the TI CC2538 System-On-Chip microcontroller. Computation is managed by an ARM Cortex M3 CPU embedded in the microcontroller, equipped with 512 kB of flash memory and 32 kB of SRAM memory. Communication is managed by two transceivers: the first is embedded in the TI CC2538 microcontroller and is used for transmission in the 2.4-GHz band, whereas the second is an Atmel AT86RF215 operating in the 868/915 MHz frequency range (not used in our experimental campaigns). OpenWSN includes the most recent version of the 6TiSCH protocol stack, which is based on TSCH. Its source code is freely available and can be modified at will. The code can be cross compiled on the PC by means of, e.g., the ARM `gcc` toolchain, and downloaded to the target mote via the USB interface using OpenVisualizer, a software tool provided with OpenWSN for controlling and monitoring network operation in real time, including the selection of root motes (in our case, M^I_{CH} and M^I_{FC}).

To allow FC and CH mote pairs to operate at the same time, some preliminary configurations are needed. In particular, two operations were performed. First, two different PAN identifiers were selected for FC and CH to make the two networks independent. This can be easily achieved by compiling the OpenWSN code with the option `panid=0x1234`, which specifies the selected PAN identifier. Second, CH was disabled on FC motes. To this purpose, the variable `ieee154e_vars.singleChannel` in the file `openstack/02a-MAClow/IEEE802154E.c` has to be set to the channel on which all transmissions must take place.

In addition to the PC running OpenVisualizer (to which motes M^I_{CH} and M^I_{FC} are connected) used to carry out measurements, two other PCs running Linux (i.e., PC_a and PC_b in Fig. 3) played an important role in experimental campaigns. These PCs are equipped with two Wi-Fi adapters each (dual-band TP-Link TL-WDN4800, managed through the `ath9k` driver) and are used to inject additional interfering traffic on air. Adapters were purposely configured to comply to the IEEE 802.11g standard, to disable frame aggregation and make the amount of traffic they generate more predictable. A program was created that generates a specific random traffic pattern on each adapter. The software, which is similar to the one used in [39], relies on a finite state machine that periodically evolves between two states, *inactive* and *active*. In the inactive state, the adapter remains idle for a time interval whose duration follows a truncated exponential distribution with mean 250 ms and maximum limited to 10 s. In the active state, a sequence of packets with size 1500 B are sent, spaced by 400 μs. The number of packets in every sequence is chosen randomly according to an exponential distribution whose average is 100, truncated to a maximum of 500 packets.

A second specific software, identified as *orchestrator* in Fig. 3, was specifically implemented to coordinate the operations of all the devices involved in the measurement procedure. More details are provided in the next section.

B. MEASUREMENT PROCEDURE

Each experiment consists in the periodic transmission of packets between initiators (M^I_{CH} and M^I_{FC}) and the corresponding responders (M^R_{CH} and M^R_{FC}), to evaluate the quality of communication on the related links. Two-way communications are considered so that latency can be easily computed on the PC that controls motes M^I_{CH} and M^I_{FC}, without the need for external instruments to measure absolute time differences.

An extremely simple and proven protocol based on the request–response paradigm is employed to this extent, i.e., ICMPv6 as implemented by the `ping` command. Thanks to its inherent simplicity, ICMPv6 introduces negligible delays, improving accuracy and precision of statistics on communication latency. From a functional point of view, it closely resembles application protocols used in real WSNs that suit the IIoT paradigm, like the constrained application protocol (CoAP). At the physical layer, internet control message protocol (ICMP) *echo request* and *echo reply* messages are encoded on 87 B and 90 B, respectively, while acknowledgement (ACK) frames are 33-B long. Since the network bit rate is 250 kb/s, the duration on air of the three aforementioned frames is 348, 360, and 132 μs, respectively.

Logs were obtained from the output of `ping` commands. In particular, the round-trip time was included for every successful request. Performing the experimental analysis on the logs obtained from an external tool like `ping`, instead of modifying the code of the motes, makes it possible applying it also to networks already deployed in their intended operating environments and conditions. This option is quite valuable for network administrators, because it permits to assess network performance in a noninvasive way. A sample script for the estimation of the packet loss probability was made freely available in [40].

In every experimental campaign a variable number N_{cond} of distinct *nominal conditions* was analyzed contextually. Every condition is characterized by specific settings, which refer to aspects we can control, like the number N_{int} of interfering Wi-Fi nodes actively transmitting on air. Both FC and CH transmissions were considered, exploiting the two pairs of motes. Other aspects, like the background traffic generated by Wi-Fi equipment in nearby premises (including APs), were mostly unknown and subject to unpredictable fluctuations. To make results about the different nominal conditions comparable, the related experiments were split into small *slices* and interleaved. This method is sketched in Fig. 4, together with the sequence of operations performed by the orchestrator.

Measurement in our setup takes place on a cyclic basis, and the related procedure consists of two nested loops. The inner loop iterates over all nominal conditions, starting from the initial one. First, ten `ping` requests are sent from M^I_{FC} to M^R_{FC} over an FC, spaced by 120-s, followed by a 120-s gap where both initiators cease generating new packets, to ensure that all internal queues are emptied before a new set of measurements is started. After that, other ten `ping` requests are sent from M^I_{CH} to M^R_{CH} using CH, followed, again, by a 120-s gap. Finally, the settings related to the next condition are loaded by the orchestrator, which takes at least 10 s, and the aforementioned steps are repeated. This generates $2 \cdot N_{cond}$ small *batches* of sample points (one batch per slice). Batches for FC and CH in the same nominal conditions were adjacent and captured within a limited time span, therefore, we can reasonably assume that phenomena not under our control (background traffic) affected them more or less in the same way. At this point, a new iteration of the inner loop can be performed, which resumes from the initial condition. The outer loop repeats all the aforementioned operations N_{loop} times, so as to acquire a large number of sample points, hence achieving a satisfactory statistical characterization of the link behavior.

The outcome of every experiment can be described by the sequence $(\mathcal{M}_1, \ldots, \mathcal{M}_{N_{loop}})$, where

$$\mathcal{M}_j = \left(\mathcal{B}_j^{1,FC}, \mathcal{B}_j^{1,CH}, \ldots, \mathcal{B}_j^{N_{cond},FC}, \mathcal{B}_j^{N_{cond},CH}\right) \quad (1)$$

is the subsequence of batches captured in the repetition $j \in [1, N_{loop}]$ of the measurement procedure (inner loop) and $\mathcal{B}_j^{l,m}$ denotes the specific batch acquired with nominal condition $l \in [1, N_{cond}]$ and transmission mode $m \in \{FC, CH\}$. For every pair $\langle l, m \rangle$, all batches acquired in the experiment were (orderly) merged together to yield an aggregate sequence of sample points

$$\mathcal{S}^{l,m} = \bigcup_{j \in [1, N_{loop}]} \mathcal{B}_j^{l,m}. \quad (2)$$

Statistical indices computed on $\mathcal{S}^{l,m}$ permit to characterize the QoC of transmission mode m under nominal condition l.

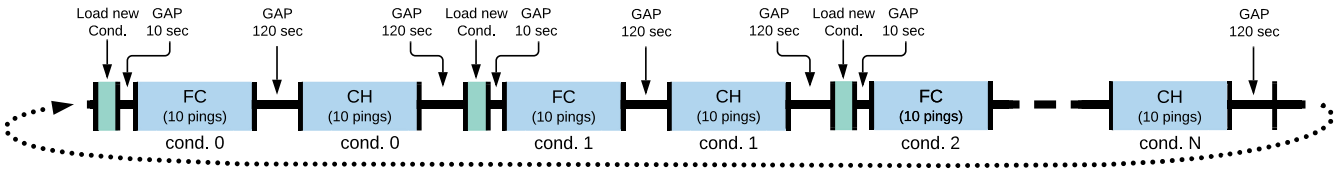


FIGURE 4. **Interleaving technique to allow fair comparison between FC and CH for different nominal experimental conditions.**

C. PERFORMANCE INDICATORS

From the logs obtained in the different experiments, and in particular for every sequence $\mathcal{S}^{l,m}$, a number of metrics, which are relevant for distributed applications that benefit from predictable behavior, can be evaluated.

1) COMMUNICATION RELIABILITY

One of the most important metrics is the *packet loss probability* P_{L}. It can be approximated by the packet loss ratio, defined as the fraction of transmission requests that failed. Let N and N_{L} denote the overall number of requests issued in every experiment and the number of them that failed, respectively. Then

$$P_{\mathrm{L}} \simeq f_{\mathrm{L}} = \frac{N_{\mathrm{L}}}{N}. \tag{3}$$

Since we considered a pair of motes interacting over a TSCH link through the request–response paradigm, the two-way path has to be taken into account. In this case, a failure occurs when either the request packet is lost in the forward direction or the response packet is lost in the return direction.

Communication reliability is often expressed as $1 - P_{\mathrm{L}}$. In the vast majority of the experiments, we carried out, no packets went lost, which implies that the measured f_{L} is 0. This does not mean that reliability was 100%. Instead, it depends on the fact that with the default retry limit (15) set by 6TiSCH on OpenMotes B and the noncritical interference level observed in our lab, packet losses (which, because of ARQ, must not be confused with the failure of a single attempt) are indeed rare events. However, they may occur over longer time spans (months or years) or in the presence of severe interference.

2) TRANSMISSION LATENCY

Although WSNs were not conceived to support real-time applications, a better determinism can be useful in many contexts. For instance, if time stamping is carried out by the PC that collects sampled values, accuracy and precision of timings depend on the delays introduced by the WSN. Taking timestamps directly on motes is a viable solution, but doing so requires that a suitable clock synchronization mechanism [41] is made available to the applications running on them, to ensure a coherent time base.

Let d_i denote the *transmission latency* of the ith request, which is defined only for successful requests and coincides with the round-trip time. Then, the minimum ($d_{\min}$), mean (d_{avg}), and maximum ($d_{\max}$) values, as well as the standard deviation (d_{std}) and relevant percentiles (e.g., d_{p95}), can be easily evaluated from logs. In a real TSCH network, latency is made up of three distinct contributions.

1) The *queuing time* in the initiator waiting for a suitable (negotiated) slot for the request packet. Since time slotting operates periodically and independently from the requests issued by applications, this time can be modeled as a random variable uniformly distributed between 0 and the slotframe duration T_{sf}. As such, its expected value, variance, and standard deviation are equal, in theory, to $T_{\mathrm{sf}}/2$, $T_{\mathrm{sf}}^2/12$, and $T_{\mathrm{sf}}/\sqrt{12}$, respectively. If situations occur where more than one packet is buffered for the link in the initiator, an additional delay has to be considered in previous equations.
2) The *basic communication time* effectively taken for a two-way packet exchange according to TSCH rules, also considering the (short) delays introduced by the operating system and the protocol stack of the involved motes. This time is fixed, and mostly depends on the position in the slotframe of the cells used for the links on the forward and return paths. Provided that enough sample points are available and the amount of interference and disturbance is not excessively high, it can be easily and reliably estimated as the minimum measured latency $d_{\min}$. In fact, the probability for $d_{\min}$ to exceed the true basic communication time by αT_{sf} (in the absence of transmission errors) is equal to $(1-\alpha)^{N-N_{\mathrm{L}}}$. For example, when $N - N_{\mathrm{L}} = 2500$ (a value comparable with our datasets), the probability to overestimate this time by more than 5 ms (that is, $\alpha = 0.0025$, a quarter of the slot time) is less than 0.2%.
3) The *retransmission time* taken by failed transmission attempts, denoted $\hat{d}$ for short. Since the default configuration was used for the TSCH matrix (which allocates exactly one cell per direction to the link between initiator and responder), this time corresponds to an integer number of slotframes, that is, $\hat{d}_i = r_i \cdot T_{\mathrm{sf}}$, where r_i is the overall number of retries performed for the packets involved in request i, both directions considered. For each sample in the log, r_i can be found as the unique nonnegative integer for which $d_{\min} + r_i T_{\mathrm{sf}} \leq d_i < d_{\min} + (r_i + 1)T_{\mathrm{sf}}$. The value of r_i is bounded by the retry limit R_{L}, and it is usually quite low. In the absence of failed attempts, $\hat{d}_i = 0$.

Queuing and basic communication times do not depend on environmental conditions (described by l), and not even on the transmission mode (either FC or CH, as specified by m). In particular, the basic communication time only depends on the TSCH matrix, whose schedule is decided automatically by the

6TiSCH Operation Sublayer (6top) [42] every time the network is formed (just before the experiment starts, in our case). Despite varying among experiments, it is the same for all the sample points related to the same network (i.e., transmission mode m) collected in every single experiment. Thus, samples d_i can be profitably offset by subtracting $d_{\min}$, which provides the *unbiased latency* $\breve{d}_i = d_i - d_{\min}$ (that does not depend on the TSCH matrix). Several statistical indices about $\breve{d}$, namely, mean value ($\breve{d}_{\text{avg}}$), maximum ($\breve{d}_{\max}$), and percentiles ($\breve{d}_{\text{p95}}$), can be found in the same way, e.g., $\breve{d}_{\text{avg}} = d_{\text{avg}} - d_{\min}$. Standard deviation, instead, is just the same, $\breve{d}_{\text{std}} = d_{\text{std}}$.

The only contribution to the latency that is influenced by environmental conditions and CH adoption is the retransmission time $\hat{d}$. Reliable *estimates* for statistical indices about it can be derived from r_i values, e.g., $\hat{d}_{\text{avg}} = \bar{r}T_{\text{sf}}$, $\hat{d}_{\text{std}} = \sigma_r T_{\text{sf}}$, $\hat{d}_{\max} = r_{\max}T_{\text{sf}}$, and $\hat{d}_{\text{p95}} = r_{\text{p95}}T_{\text{sf}}$, where $\bar{r}$, σ_r, $r_{\max}$, and r_{p95}, are the mean value, standard deviation, maximum, and 95-percentile of the set $\{r_i\}$, only considering successful requests.

Statistical indices for $\hat{d}$ can be also derived from the unbiased latency. They will be denoted with the prime symbol. The estimated average is obtained by subtracting the expected value of the queuing time from the mean unbiased latency, $\hat{d}'_{\text{avg}} = \breve{d}_{\text{avg}} - T_{\text{sf}}/2$. Since random phenomena affecting queuing, basic communication, and retransmission times are statistically uncorrelated, the variance of their sum equals the sum of their variances. Basic communication time variance can be neglected, as it is due to small jitters affecting execution times on motes. Therefore, the standard deviation can be also estimated by subtracting the theoretical queuing time, that is, $\hat{d}'_{\text{std}} = \sqrt{d^2_{\text{std}} - T^2_{\text{sf}}/12}$.

Discrepancies between estimates $\hat{d}$ and $\hat{d}'$ become negligible provided that the number $N - N_{\text{L}}$ of sample points is large enough, as can be seen by observing the mean value and standard deviation reported in the following tables.

3) EQUIVALENT FAILURE PROBABILITY

A very interesting quantity can be derived from logs, referred to as the *equivalent failure probability* for transmission attempts $\tilde{\epsilon}$, which describes in a synthetic way the amount of interference and disturbance affecting communication [27]. More specifically, it represents the failure probability ϵ that, under the assumption that attempts can be modeled as Bernoulli trials (i.e., that their outcomes are independent and time invariant), makes the theoretical value of some specific performance indicator equal to the corresponding statistics obtained from measured samples. In our case, we sought the value of ϵ that makes the theoretical probability $P_{r=0}$ that no retransmissions are performed for a two-way exchange (both directions considered) equal to the measured fraction $f_{r=0}$ of request–response pairs in the experimental logs for which no retries were carried out.

The value of $f_{r=0}$ can be found as follows: first, the value of $d_{\min}$ is evaluated from the logs; then, the number $N_{r=0}$ of request–response pairs that did not experience any retransmissions is evaluated by counting the number of sample points for which $d_i < d_{\min} + T_{\text{sf}}$, from which

$$f_{r=0} = \frac{N_{r=0}}{N}. \tag{4}$$

The value $P_{r=0}$ corresponds instead to the probability that the transmissions of both the request and the response packets succeed at the first attempt. Under the hypotheses that the failure probability on the forward and return paths is the same and that attempts are statistically independent, this implies

$$P_{r=0} = (1-\epsilon)^2. \tag{5}$$

By equating (4) and (5), we obtain

$$\tilde{\epsilon} = 1 - \sqrt{\frac{N_{r=0}}{N}}. \tag{6}$$

As shown in [27], modeling TSCH communications this way is a sensible choice and provides a very good approximation. For example, the cumulative distribution function (CDF) of the latency computed from $\tilde{\epsilon}$ for single-hop two-way communications was found to match very well experimental results. Besides the value provided by (6), which refers to the whole experiment, a moving average can be also applied to samples in order to determine how $\tilde{\epsilon}$ varies with time.

4) ENERGY CONSUMPTION

This is a very important metric for WSNs, as motes are often powered on batteries or exploit energy harvesting. CH indirectly affects this quantity because of ARQ. Every time a retransmission attempt takes place, the transmitting mote spends some additional energy E_{tx}, while on the receiving mote a frame reception occurs in the place of idle listening (receivers wake up and listen to the channel according to the TSCH schedule, irrespective of transmitters). Let E_{rx} and E_{listen} be the energies the recipient spends in the two cases, respectively (they are typically similar). In the absence of overprovisioning [43], network stability requires the slotframe repetition rate to be strictly faster than the mean packet transmission rate on every link. This means that idle listening is unavoidable, unless specific techniques are adopted to switch off the receiving interface when no packets are expected to arrive [16], [44]. In typical operating conditions, it may actually occur more often than frame reception.

Overall, the increase in the consumed energy for a single retry, both sides considered, is $\Delta E = E_{\text{tx}} + E_{\text{rx}} - E_{\text{listen}}$. Such energies basically depend on the payload size and the specific mote implementation (both hardware and software, including optimizations). For example, by using the energy model described in [15], $E_{\text{tx}} = 485.7\,\mu$J, $E_{\text{rx}} = 651.0\,\mu$J, and $E_{\text{listen}} = 303.3\,\mu$J, which means that $\Delta E = 833.4\,\mu$J. On average, the increase in energy consumption due to retries for any single successful request–response exchange over a link is $\Delta\bar{E}_{\text{s}} = \bar{r}\Delta E$, whereas for the (very few) failed exchanges, $\Delta\bar{E}_{\text{f}} \geq (R_{\text{L}} - 1)\Delta E$. Therefore, $\bar{r}$ also provides interesting hints on the effects of spectrum conditions and CH adoption on power consumption.

TABLE 1. Link QoC: Two 7-Day Interleaved Experiments (CH Vs. FC_{12} and CH Vs. FC_{20}), No Purposely Injected Interfering Traffic, Only Background Traffic Generated by Nearby Infrastructure Wi-Fi Networks (More Than 12 Visible APs)

	Link mode m	Measured round-trip time $d_{\min}$	d_{avg}	d_{std}	d_{p95}	$d_{\max}$ [s]	Losses P_{L}	$\tilde{\epsilon}$ [%]	Unbiased latency $\breve{d}_{\text{avg}}$	$\breve{d}_{\text{p95}}$	$\breve{d}_{\max}$ [s]	Retransmission time $\hat{d}_{\text{avg}}$	$\hat{d}'_{\text{avg}}$	$\hat{d}_{\text{std}}$	$\hat{d}'_{\text{std}}$	$\hat{d}_{\text{p95}}$	$\hat{d}_{\max}$ [s]	Energy $\Delta\bar{E}_{\text{s}}$ [μJ]
Exp. 1	FC_{12}	0.822	2.203	1.054	4.372	9.982	0.0	8.7	1.381	3.550	9.160	0.374	0.371	0.891	0.878	2.020	8.080	154.3
	CH	0.502	1.709	0.879	3.626	6.835	0.0	4.7	1.207	3.124	6.333	0.198	0.197	0.638	0.658	2.020	6.060	81.7
Exp. 2	FC_{20}	0.661	1.736	0.672	2.651	4.695	0.0	1.8	1.075	1.990	4.034	0.071	0.065	0.372	0.335	0.000	2.020	29.3
	CH	0.510	1.738	0.864	3.532	7.788	0.0	5.2	1.228	3.022	7.278	0.216	0.218	0.661	0.637	2.020	6.060	89.1

V. RESULTS

Three measurement campaigns were carried out to analyze the effect of CH on QoC performance indicators. Two distinct links were contextually evaluated in every experiment, one exploiting CH and the other tuned on an FC. In particular, every campaign consisted of two experiments performed in sequence, where the FC link was tuned on WSN channels 12 and 20, respectively. In the following, the related transmission modes will be denoted CH, FC_{12}, and FC_{20}.

A. COMMUNICATION QUALITY VERSUS BACKGROUND TRAFFIC

The first experimental campaign was aimed at analyzing network behavior in unmodified, typical environmental conditions. Our lab is characterized by a certain amount of Wi-Fi background traffic due to the activity of more than a dozen nearby APs, to which mobiles and notebooks can associate. This traffic is usually low and does not impair communication on the WSN. On the other hand, it is not completely negligible, and represents a realistic example of what can be found when WSNs are deployed in the real world. Two separate experiments were carried out back-to-back (besides the time to restart the testbed), each of which lasted one week. Results are reported in Table 1.

To avoid queuing phenomena, the period we chose for `ping` transmissions was set to 120 s. This time is not so short, and may cause fluctuations on $\tilde{\epsilon}$ that affect pairs $\langle \mathcal{B}_j^{l,\text{FC}}, \mathcal{B}_j^{l,\text{CH}} \rangle$ of interleaved batches. Although decreasing this period improves accuracy, it would also make the TSCH link work in nontypical operating conditions, voiding one of the main goals of this work. The inner measurement loop, which includes ten `ping` requests for FC and ten for CH, lasted about 40 min. If the spectrum does not vary excessively in the meanwhile, interleaving is quite effective and performance indicators measured in the same experiment for the two transmission modes can be compared quite reliably. This is not necessarily true for distinct experiments, since they were carried out at different times (spaced by many days), which means that the related spectrum conditions may have changed significantly. In all cases, interference generated by background Wi-Fi traffic was mild, this leading on average to few retransmissions. Given the high retry limit set by default by OpenWSN on OpenMotes B, losing a packet is indeed a very rare event, and never occurred in this experimental campaign.

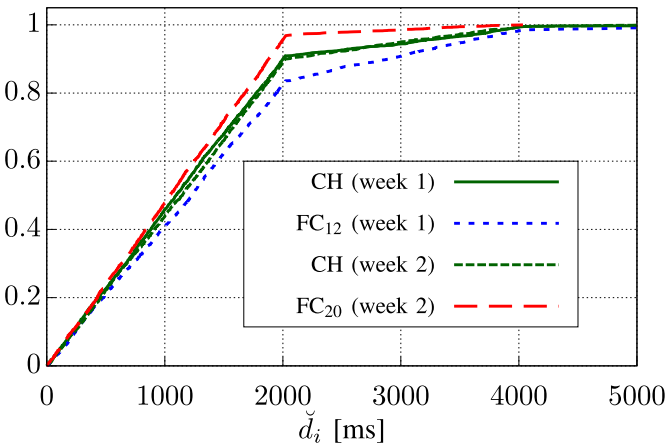


FIGURE 5. Cumulative distribution function of unbiased latency $\breve{d}_i$.

In the first experiment, CH managed to outdo FC_{12}. Unbiased mean value $\breve{d}_{\text{avg}}$, standard deviation d_{std}, and maximum $\breve{d}_{\max}$ were consistently lower, and the same held for the equivalent failure rate $\tilde{\epsilon}$ (4.7% versus 8.7%). Conversely, in the second experiment, CH performed worse than FC_{20} (measured $\tilde{\epsilon}$ values were 5.2% and 1.8%, respectively). This is because the amount of interference on the WSN channel 20 was mostly negligible, making CH seemingly not beneficial. Interestingly, statistics about latency for CH in the two experiments were quite similar, which means that overall spectrum conditions remained approximately the same over the two weeks. It is worth noting that operations like reconfiguring a nearby AP on a different Wi-Fi channel are not expected to impact on the behavior of TSCH.

In Fig. 5, the CDFs of the unbiased latency $\breve{d}_i$ are shown. A detailed explanation of their peculiar shape can be found in [27]. As can be seen, distributions of delays for physical channels differed sensibly (channel 20 behaved better than channel 12), whereas they were mostly identical when CH was enabled. The aforementioned results show that, if raw values are considered for communication performance indicators at any given time, it is untrue that CH always brings improvements over FC transmission for all the available frequencies.

To provide some hints about variations of the link QoC over time, two timing diagrams are included in Fig. 6, which show the equivalent failure rate $\tilde{\epsilon}_{\text{w}}$ calculated over a moving window that includes $n_{\text{w}} = 180$ sample points (about 12 h). As seen previously, CH behaved better than FC_{12} in the first

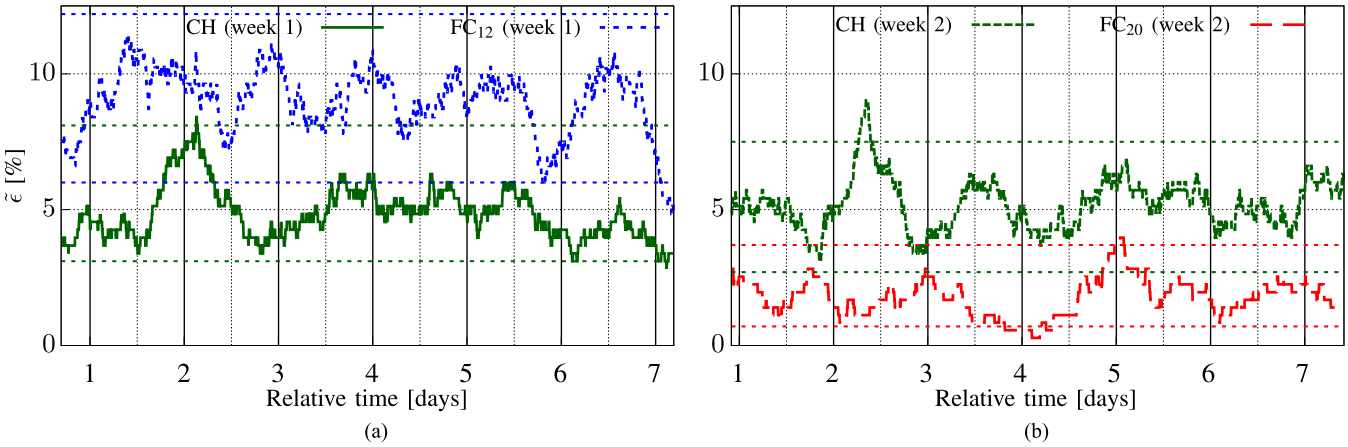


FIGURE 6. Equivalent failure rate $\tilde{\epsilon}_w$ (computed on a 180-sample moving window) versus time, two seven-day experiments (CH versus FC_{12} and CH versus FC_{20}) performed almost back-to-back (two weeks overall duration), horizontal dashed lines represent the 95% confidence intervals. (a) CH vs. FC_{12} (week 1). (b) CH vs. FC_{20} (week 2).

experiment (week 1) and worse than FC_{20} in the second (week 2). According to plots, the perceived link quality keeps varying. On the other hand, since experiments were performed almost back-to-back, variations of the spectrum conditions are expected to be limited, as highlighted by the seamless curve related to CH, which does not show abrupt changes between the two plots (first and second week). Increasing the width n_w of the moving window leads to lower variations of $\tilde{\epsilon}_w$ values, which remained, on average, more or less the same over the whole campaign. This suggests that the mean link quality was likely subject to slow drifts.

Assuming that transmission attempts can be modeled as Bernoulli trials (in [27], it was shown that this is an acceptable approximation in typical operating condition), the number of request–reply pairs for which no retransmissions were performed in a sequence including n_w requests can be described as a random variable K with binomial distribution $B(n_w, P_{r=0})$, whose probability mass function is

$$p_K(k) = \binom{n_w}{k} (P_{r=0})^k (1 - P_{r=0})^{n_w - k}. \tag{7}$$

This was checked against experimental results. For example, in the experiment for FC_{12} the measured $f_{r=0}$ was 0.833, which can be used as an estimate for $P_{r=0}$. By setting $n_w = 180$, according to (7), there is a 2.5% probability that K is below 139 and the same that it exceeds 159. From these bounds on K the 95% confidence interval for $\tilde{\epsilon}_w$ can be found through (6). Such confidence intervals can be easily computed for all cases described in Table 1 using the Clopper–Pearson method. We obtain [6.0%, 12.2%] for FC_{12} and [0.7%, 3.7%] for FC_{20}, while for CH it is [3.1%, 8.1%] in the first experiment and [2.7%, 7.5%] in the second. These intervals are shown as dashed horizontal lines in the plots of Fig. 6. As can be seen, variations of $\tilde{\epsilon}_w$ over time can be explained by the random Bernoulli process we used to model transmission, and not necessarily they are due to sudden and substantial variations of the mean interference level.

B. COMMUNICATION QUALITY VERSUS INJECTED INTERFERING TRAFFIC

From the aforementioned results, one may (incorrectly) infer that CH is useless. The point is that, this mechanism was mainly conceived to face phenomena like severe narrowband interference (besides multipath fading), which is not the case of our lab. For this reason, a second experimental campaign was performed where a nonnegligible amount of interfering Wi-Fi traffic was purposely injected on air. As in the previous campaign, two experiments were carried out where performance indicators for CH were evaluated together with FC_{12} and FC_{20}, respectively, by using the interleaving technique. However, a number N_{int} of interfering Wi-Fi stations were deployed this time close to our testbed, each one tuned on a distinct Wi-Fi channel (chosen among 1, 5, 9, and 13). Several interference patterns were selected for each experiment, where one, two, three, and four Wi-Fi stations were switched ON (the latter pattern causes interference on every WSN channel). The case with no interferers (in the presence of background traffic only) was also considered, which corresponds to the experiments in the previous campaign.

Patterns are identified with the notation "c_1-c_2-c_3-c_4," where c_k either represents the Wi-Fi channel on which the kth interferer injects traffic or is set to 0 if that interferer is inactive. In the FC case, a further distinction can be made among patterns, depending on whether any of the interferers overlaps with the frequency chosen for the TSCH link. This is expressed as a Boolean value termed *OVL*, depicted as a solid dot (•) in the tables and the figures. For instance, Wi-Fi channels 1 and 9 completely overlap with WSN channels 12 and 20, respectively. When the transmission frequencies of an interferer and the WSN link do not overlap, they can be either close to each other (e.g., Wi-Fi channel 5 is "adjacent" to WSN channels 12 and 20) or spaced wide apart (e.g., Wi-Fi channel 13 is located "far" away from the WSN channel 12).

The two experiments took place several weeks after the first campaign, and they were not performed back-to-back, to

TABLE 2. Link QoC: Two Nine-Day Interleaved Experiments (CH Versus FC_{12} and CH Versus FC_{20}) by Varying the Purposely Injected Interfering Traffic, Ten Different Patterns, at Most One Wi-Fi Station Per Channel, Wi-Fi Channel 1 (Resp., 9) Overlaps With WSN Channel 12 (Resp., 20)

	Link	Nominal conditions l		N_{int}	$d_{\min}$	d_{avg}	d_{std}	d_{p95}	$d_{\max}$	P_{L}	$\tilde{\epsilon}$	$\hat{d}_{\text{avg}}$	$\hat{d}'_{\text{avg}}$	$\hat{d}_{\text{std}}$	$\hat{d}'_{\text{std}}$	$\hat{d}_{\text{p95}}$	$\hat{d}_{\max}$	$\Delta\bar{E}_{\text{s}}$
	mode m	Wi-Fi traffic	Pattern	/ *OVL*	[s]					[%]		[s]						[μJ]
Exp. 3 (CH vs. FC_{12})	FC_{12}	none	0-0-0-0	0	1.964	3.217	0.927	5.424	6.871	0.0	6.4	0.263	0.243	0.719	0.720	2.020	4.040	108.5
		far	0-0-0-13	1	1.954	3.271	0.964	5.336	7.911	0.0	7.6	0.317	0.307	0.788	0.768	2.020	4.040	130.8
		adjacent	0-5-0-0	1	1.949	3.288	0.890	5.274	5.960	0.0	8.3	0.323	0.329	0.741	0.672	2.020	2.020	133.3
		overlapping	1-0-0-0	1 •	1.949	3.778	1.693	7.126	13.457	0.0	15.7	0.828	0.819	1.541	1.589	4.040	10.100	341.6
		two non-ovl.	0-0-9-13	2	1.954	3.211	0.946	5.266	6.939	0.0	6.9	0.283	0.247	0.739	0.744	2.020	4.040	116.8
		ovl.+far	1-0-0-13	2 •	1.942	3.852	1.684	7.477	13.372	0.0	17.5	0.909	0.900	1.574	1.580	4.040	10.100	375.0
		ovl.+adj.	1-5-0-0	2 •	1.957	4.064	1.831	7.787	12.614	0.0	21.5	1.117	1.097	1.739	1.736	4.040	10.100	460.8
		three non-ovl.	0-5-9-13	3	1.943	3.219	0.985	5.287	7.264	0.0	7.8	0.309	0.266	0.745	0.793	2.020	4.040	127.5
		ovl.+adj.+far	1-5-9-0	3 •	1.951	4.013	1.720	7.725	10.141	0.0	20.8	1.085	1.052	1.644	1.618	4.040	8.080	447.6
		all	1-5-9-13	4 •	1.946	4.222	1.982	8.224	13.786	0.0	24.7	1.307	1.266	1.860	1.894	6.060	10.100	539.2
	CH	none	0-0-0-0	0	1.463	2.731	0.971	4.912	7.193	0.0	6.9	0.283	0.258	0.739	0.776	2.020	4.040	116.8
		one chan.	0-0-0-13	1	1.474	2.801	0.958	4.976	6.096	0.0	8.0	0.323	0.317	0.776	0.760	2.020	4.040	133.3
		one chan.	0-5-0-0	1	1.502	2.826	1.051	5.147	8.451	0.0	7.6	0.337	0.314	0.885	0.875	2.020	6.060	139.0
		one chan.	1-0-0-0	1	1.489	2.778	0.990	4.940	7.399	0.0	6.9	0.289	0.279	0.764	0.800	2.020	4.040	119.2
		two chan.	0-0-9-13	2	1.464	2.881	1.172	5.275	7.722	0.0	8.5	0.404	0.407	0.990	1.017	2.020	6.060	166.7
		two chan.	1-0-0-13	2	1.471	2.922	1.057	5.117	7.498	0.0	10.2	0.438	0.441	0.939	0.881	2.020	4.040	180.7
		two chan.	1-5-0-0	2	1.466	2.912	1.103	5.277	6.835	0.0	10.6	0.444	0.436	0.929	0.936	2.020	4.040	183.2
		three chan.	0-5-9-13	3	1.466	3.021	1.243	5.451	7.464	0.0	12.3	0.560	0.545	1.095	1.098	2.020	4.040	231.0
		three chan.	1-5-9-0	3	1.467	3.113	1.352	5.687	9.810	0.0	14.0	0.646	0.636	1.220	1.220	4.040	8.080	266.5
		four chan.	1-5-9-13	4	1.475	3.083	1.356	5.555	9.880	0.0	12.4	0.600	0.598	1.226	1.224	4.040	8.080	247.5
Exp. 4 (CH vs. FC_{20})	FC_{20}	none	0-0-0-0	0	0.665	1.732	0.665	2.637	4.699	0.0	1.2	0.046	0.057	0.305	0.319	0.000	2.020	19.0
		far	1-0-0-0	1	0.666	1.745	0.659	2.647	4.446	0.0	1.7	0.067	0.069	0.364	0.306	0.000	2.020	27.6
		adjacent	0-5-0-0	1	0.671	1.712	0.700	2.656	4.247	0.0	1.9	0.075	0.031	0.380	0.387	0.000	2.020	30.9
		overlapping	0-0-9-0	1 •	0.681	2.088	1.104	4.417	5.958	0.0	9.3	0.398	0.397	0.899	0.937	2.020	4.040	164.2
		two non-ovl.	1-5-0-0	2	0.663	1.748	0.658	2.637	4.597	0.0	1.7	0.067	0.075	0.364	0.305	0.000	2.020	27.6
		ovl.+far	1-0-9-0	2 •	0.667	2.036	1.046	4.259	6.702	0.0	8.3	0.358	0.359	0.854	0.869	2.020	4.040	147.7
		ovl.+adj.	0-5-9-0	2 •	0.666	1.971	0.947	3.995	6.561	0.0	7.4	0.297	0.295	0.733	0.746	2.020	4.040	122.5
		three non-ovl.	1-5-0-13	3	0.672	1.762	0.642	2.651	3.886	0.0	1.7	0.067	0.080	0.364	0.269	0.000	2.020	27.6
		ovl.+adj.+far	1-5-9-0	3 •	0.674	2.065	1.053	4.385	6.964	0.0	8.2	0.349	0.381	0.865	0.877	2.020	6.060	144.0
		all	1-5-9-13	4 •	0.677	1.938	1.021	4.088	7.423	0.0	6.4	0.297	0.251	0.852	0.838	2.020	6.060	122.5
	CH	none	0-0-0-0	0	0.521	1.730	0.921	3.466	7.572	0.0	4.6	0.208	0.199	0.717	0.713	2.020	6.060	85.8
		one chan.	1-0-0-0	1	0.517	1.884	1.001	4.021	6.055	0.0	9.1	0.370	0.357	0.832	0.813	2.020	4.040	152.7
		one chan.	0-5-0-0	1	0.506	1.812	1.039	4.063	6.520	0.0	6.7	0.297	0.296	0.804	0.860	2.020	4.040	122.5
		one chan.	0-0-9-0	1	0.527	1.858	1.046	3.998	6.276	0.0	7.4	0.343	0.321	0.891	0.869	2.020	4.040	141.5
		two chan.	1-5-0-0	2	0.507	1.821	0.942	3.654	6.318	0.0	7.4	0.309	0.304	0.782	0.740	2.020	4.040	127.5
		two chan.	1-0-9-0	2	0.523	1.886	1.061	4.116	6.534	0.0	8.7	0.370	0.353	0.865	0.887	2.020	4.040	152.7
		two chan.	0-5-9-0	2	0.505	1.950	1.147	4.397	6.168	0.0	10.4	0.450	0.435	0.962	0.988	2.020	4.040	185.7
		three chan.	1-5-0-13	3	0.513	2.157	1.390	4.620	7.725	0.0	13.0	0.626	0.634	1.234	1.261	4.040	6.060	258.3
		three chan.	1-5-9-0	3	0.527	2.125	1.332	4.609	8.273	0.0	12.4	0.592	0.588	1.190	1.197	4.040	6.060	244.2
		four chan.	1-5-9-13	4	0.523	2.244	1.475	5.128	11.716	0.0	14.4	0.701	0.711	1.370	1.355	4.040	10.100	289.2

check what happens to the spectrum over longer time spans. As a consequence, the measured performance indicators for the cases related to the pattern "0-0-0-0" (with no interferers) do not match exactly those obtained previously. This confirms that spectrum conditions were actually ever changing, although slowly. Hence, the closer the experiments are in time, the more similar the interference due to the background traffic. Results for the experiments where CH was interleaved with FC_{12} and FC_{20} are reported in the upper and the lower parts of Table 2, respectively.

Because of the high number of considered patterns (ten, which enlarge the inner loop to about 400 min), the interleaving technique was not as effective as in the former campaign when different interference patterns are considered. As can be seen from Fig. 6, in such a large time interval fluctuations of the background traffic are typically more pronounced, and $\tilde{\epsilon}_{\text{w}}$ may vary, in absolute terms, up to $\pm 3\%$ (up to $\pm 4\%$ for channel 12). Therefore, we cannot assume that the background traffic (the contribution to disturbance not under our control) was mostly the same for the different conditions we analyzed (characterized by the injected traffic pattern). Nevertheless FC and CH can be still compared reliably for any given nominal interference condition l, and it is reasonable to assume that the background traffic did not vary drastically in the course of the same experiment, which means that differences observed among interleaved conditions depended for the most part on the injected traffic; hence, useful hints can be obtained by comparing them.

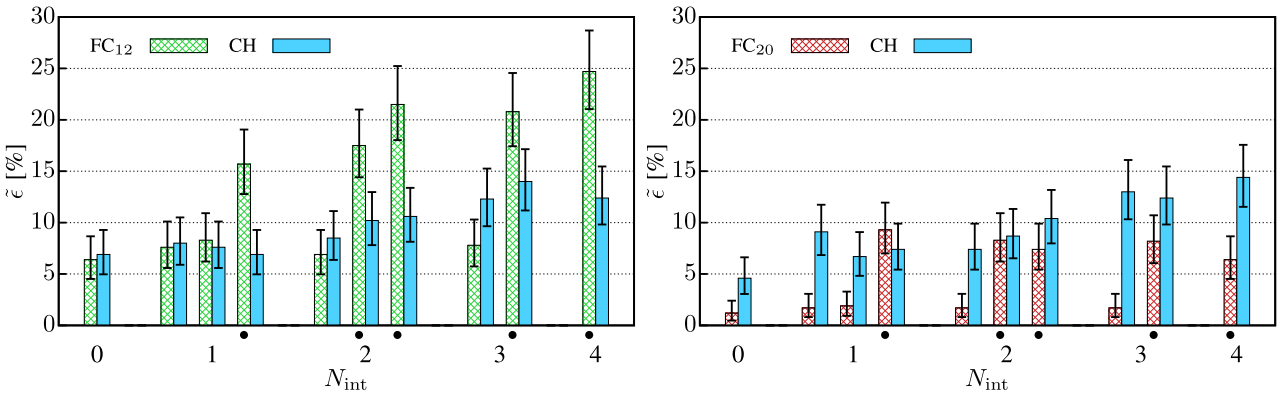


FIGURE 7. Equivalent failure probability $\tilde{\epsilon}$ versus number N_{int} of interferers for CH versus FC_{12} and CH versus FC_{20} (• denotes overlapping).

To better highlight differences between the CH and FC behavior, two bar charts are included in Fig. 7 that show the equivalent failure probability $\tilde{\epsilon}$, along with the 95% confidence interval, for the different interfering conditions (grouped according to the number N_{int} of interferers). As can be seen, in both the experiments $\tilde{\epsilon}$ for CH stayed in similar ranges ([6.9%, 14.0%] and [4.6%, 14.4%], respectively) and varied in the same way, gradually worsening as N_{int} increases. The Pearson correlation coefficient between $\tilde{\epsilon}$ and N_{int} is quite high (0.897 for the first experiment and 0.932 for the second). This is because the transmission frequency in CH keeps hopping, and hence, what matters is the mean interference on all the considered channels, that in our testbed mainly depends on N_{int}.

On the contrary, link quality for FC was either good or bad. The only aspect that really matters in this case is whether the specific channel in use overlaps with Wi-Fi interferers, as given by *OVL*. Unsurprisingly, interferers tuned on nonoverlapping channels were completely irrelevant, irrespective of how close frequencies were selected. This means that filters in the RF modules of the involved equipment operated as intended, and no adjacent-channel interference (ACI) phenomena took place [45]. From a practical viewpoint, $\tilde{\epsilon}$ for FC_{12} stayed in the range [6.4%, 8.3%] when *OVL* is false, which is somehow comparable to the value 8.7% reported in Table 1, and grown to [15.7%, 24.7%] otherwise. Similarly, $\tilde{\epsilon}$ for FC_{20} lain in the range [1.2%, 1.9%] when *OVL* is false, comparable to the value 1.8% in Table 1, and increased to [6.4%, 9.3%] otherwise. Pearson correlation coefficients between $\tilde{\epsilon}$ and *OVL* for FC_{12} and FC_{20} are 0.941 and 0.976, respectively, which implies that they are strongly correlated. The very same conclusions can be also drawn by considering statistics on latency, e.g., the retransmission time $\hat{d}$.

In this campaign, variability of results among different conditions was not negligible. For example, let us consider FC_{12} in those cases where *OVL* is true. In theory, the perceived link quality should be the same. By merging all the related sample points (there are 1500), the resulting value for $\tilde{\epsilon}$ is 20%. The 95% and 99% confidence intervals when 300 sample points are considered (i.e., for any specific condition among those taken into account previously), evaluated using Clopper–Pearson, are [16.7%, 23.7%] and [15.7%, 24.8%], respectively. We can observe that the latter interval comprises all the considered experiments (FC_{12} with *OVL* true), but the former only includes three out of five. This means that intrinsic randomness of Bernoulli processes is not enough to justify by itself variations. As said previously, because of the wider spacing between slices and the nonnegligible interference, modeling transmission attempts in TSCH as Bernoulli trials with fixed failure probability no longer provides a completely faithful approximation of reality.

C. COMMUNICATION QUALITY VERSUS NARROWBAND INTERFERING TRAFFIC

The last experimental campaign was aimed at evaluating the effect of narrowband interference on a TSCH link. As in the previous campaign, two experiments were carried out where CH was interleaved with FC_{12} and FC_{20} by varying the number of active interferers. However, this time all Wi-Fi stations were tuned on the same frequency, that is, either channel 1 or 9, which overlap with WSN channels 12 and 20, respectively. We did not consider the cases where interferers did not overlap with the FC link, as the previous analysis highlighted that they do not bring any noticeable effects. To maintain coherence among results throughout this article, the shape and amount of traffic generated by every single interferer remained the same as in the previous campaign. In order not to impair operations of nearby Wi-Fi networks too much, the duration of each experiment was purposely limited to about three days. In fact, having three interferers on the same channel generates a sizable traffic on air, and causes a nonnegligible amount of collisions. Preliminary experiments showed that, increasing interference further, occasionally caused TSCH instability (by impairing the transmission of enhanced beacons and RPL messages), which makes its behavior (and results) unreliable. Results are reported in Table 3. The notation for interfering patterns was augmented with superscripts that denote the number of interferers per channel, e.g., "0-0-9^{ℓ}-0" means that ℓ Wi-Fi stations were actively injecting interfering traffic at the same time on channel 9. As for the previous campaign, two bar charts have been included in Fig. 8 to enable CH and

TABLE 3. Link QoC: Two Three-Day Interleaved Experiments (CH Versus FC_{12} and CH Versus FC_{20}) by Varying the Purposely Injected Interfering Traffic, 0–3 Active Wi-Fi Interferers Transmitting on the Same Channel, Wi-Fi Channel 1 (Resp., 9) Overlaps With WSN Channel 12 (Resp., 20)

	Link	Nominal conditions l		N_{int}	$d_{\min}$	d_{avg}	d_{std}	d_{p95}	$d_{\max}$	P_{L}	$\tilde{\epsilon}$	$\hat{d}_{\text{avg}}$	$\hat{d}'_{\text{avg}}$	$\hat{d}_{\text{std}}$	$\hat{d}'_{\text{std}}$	$\hat{d}_{\text{p95}}$	$\hat{d}_{\max}$	$\Delta\bar{E}_{\text{s}}$
	mode m	Wi-Fi traffic	Pattern	/ *OVL*	[s]					[%]		[s]						[μJ]
Exp. 5 (CH vs. FC_{12})	FC_{12}	none	0-0-0-0	0	0.587	1.969	1.258	4.559	8.048	0.0	7.6	0.398	0.372	1.050	1.115	2.020	6.060	164.2
		one interf.	1-0-0-0	1 •	0.593	2.281	1.462	5.778	8.686	0.0	13.0	0.683	0.678	1.325	1.341	4.040	8.080	281.8
		two interf.	1^2-0-0-0	2 •	0.594	2.870	2.081	6.804	15.058	0.0	22.3	1.287	1.266	2.002	1.998	6.060	14.140	531.0
		three interf.	1^3-0-0-0	3 •	0.591	3.320	2.481	8.448	14.630	0.0	27.1	1.667	1.719	2.406	2.411	6.060	12.120	687.8
	CH	none	0-0-0-0	0	1.592	2.809	0.934	4.847	7.331	0.0	4.8	0.220	0.207	0.725	0.730	2.020	4.040	90.8
		one interf.	1-0-0-0	1	1.584	2.880	1.016	4.927	9.664	0.0	6.8	0.303	0.286	0.867	0.831	2.020	8.080	125.0
		two interf.	1^2-0-0-0	2	1.584	3.108	1.291	5.339	11.080	0.0	11.1	0.523	0.514	1.172	1.152	2.020	8.080	215.8
		three interf.	1^3-0-0-0	3	1.588	3.504	1.666	6.827	11.122	0.0	18.2	0.909	0.906	1.529	1.560	4.040	8.080	375.0
Exp. 6 (CH vs. FC_{20})	FC_{20}	none	0-0-0-0	0	0.330	1.426	0.667	2.323	4.163	0.0	1.4	0.057	0.086	0.333	0.325	0.000	2.020	23.5
		one interf.	0-0-9-0	1 •	0.339	1.859	1.302	4.225	10.014	0.0	10.6	0.505	0.510	1.141	1.164	2.020	8.080	208.4
		two interf.	0-0-9^2-0	2 •	0.343	2.279	1.736	5.922	10.927	0.0	18.4	0.947	0.926	1.620	1.635	4.040	10.100	390.7
		three interf.	0-0-9^3-0	3 •	0.338	3.157	2.887	8.136	21.462	0.6	29.5	1.848	1.809	2.822	2.827	6.060	20.200	762.4
	CH	none	0-0-0-0	0	0.126	1.400	0.936	3.391	5.823	0.0	5.5	0.240	0.264	0.727	0.732	2.020	4.040	99.0
		one interf.	0-0-9-0	1	0.122	1.482	1.101	3.680	7.894	0.0	7.5	0.335	0.350	0.891	0.934	2.020	6.060	138.2
		two interf.	0-0-9^2-0	2	0.131	1.563	1.042	3.633	5.829	0.0	9.9	0.416	0.422	0.907	0.863	2.020	4.040	171.6
		three interf.	0-0-9^3-0	3	0.135	1.641	1.249	4.104	8.040	0.0	10.2	0.479	0.496	1.071	1.105	2.020	6.060	197.6

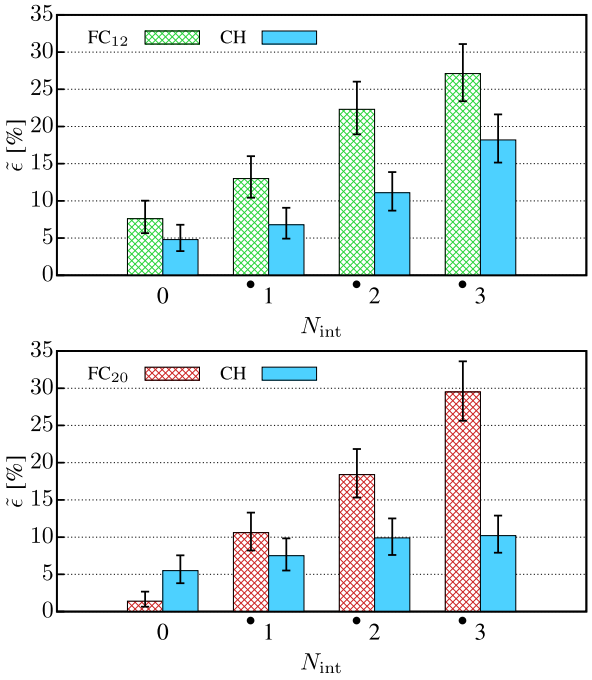


FIGURE 8. Equivalent failure probability $\tilde{\epsilon}$ versus N_{int} for CH versus FC_{12} and CH versus FC_{20} when interferers operate on the same channel (• denotes overlapping).

FC to be compared at a glance. As can be seen, in both cases, the quality of communication on the WSN link monotonically decreased as the injected traffic (described by the number of Wi-Fi interferers, which coincides with ℓ) grown higher. Correlation between $\tilde{\epsilon}$ and N_{int} was always greater than 0.96, and it exceeded 0.99 for FC. However, performance indicators for FC worsened much more quickly than for CH, and in the case of three active interferers, some packet went even lost in FC_{20}.

For example, in the experiment described in the upper part of Table 3, $\tilde{\epsilon}$ increased from 7.6% to 27.1% for FC_{12}, whereas it went from 4.8% to 18.2% for CH. This behavior is even more evident in the second experiment, carried out a few days later and reported in the lower part of the table. While the quality of communication offered by FC_{20} in the absence of injected traffic was indeed very good (and actually better than CH), it worsened quickly as more interferers were switched on: $\tilde{\epsilon}$ increased from 1.4% to 29.5% for FC_{20} and from 5.5% to 10.2% for CH. The reason is that, with FC all transmission attempts (and retries) suffered from the combined interference caused by the background and the purposely injected traffic, whereas with CH, thanks to hopping, most of the attempts (three out of four, on average) were performed on channels affected only by background traffic.

For any single experiment, accuracy of results was worse than in the previous campaigns. Besides the low number of available sample points (320 per interfering condition), this is due to the poor predictability of Wi-Fi behavior when high amounts of traffic are generated by several stations on the same channel. In fact, collisions make the relation between $\tilde{\epsilon}$ and ℓ not linear, therefore, it makes little sense comparing the results of the two experiments. In spite of this, some useful hints can be nevertheless obtained by looking at the different interfering conditions in the same experiment.

D. DISCUSSION

Modern industrial plants and automation systems are based on distributed architectures that either demand or, at least, take advantage of a deterministic network behavior. For closed-loop control, all process data exchanged among sensors, actuators, and controllers must take place within predefined deadlines. However, several other functions exist, many of which related to the Industry 4.0 paradigm, that do not require strictly deterministic communication. Remarkable examples are IIoT systems that rely on wireless sensor networks, e.g., condition-based monitoring, power and energy metering, reactive/proactive maintenance, online diagnostics, and

human-centered manufacturing. Although no strict deadlines are usually defined for such applications, the correct operation of algorithms that autonomously take decisions based on the perceived state of the underlying physical system benefits from higher communication reliability (less samples get lost) and lower latency (shorter reaction times improve both accuracy and effectiveness in providing the desired feedback).

Industrial environments are expected to suffer from non-negligible disturbance, including the electromagnetic noise due to high-power industrial equipment and interference generated by wireless networks like Wi-Fi, ZigBee, BLE, and Wireless IO-Link. It is worth noting that noise, e.g., from welding guns, usually does not affect the ISM band, and Wi-Fi is often deemed more aggressive than the other wireless communication technologies. Therefore, our findings mostly apply to shop floors as well. As can be seen from experimental results, not necessarily CH provides a better quality of communication than FC solutions in absolute terms. In fact, setting TSCH to operate on FC 20 is seemingly the best option (see rightmost plot in Fig. 7). What does change is stability over time: if transmissions take place on an FC and network performance is not periodically checked, sooner or later communication quality might drop consistently as a consequence of changes in the spectrum conditions. In this case, e.g., data collected from the WSN may possibly suffer from excessive latency, and even worse, severe losses.

The aforementioned statement becomes clear observing the results in the last campaign about narrowband interfering traffic. In the unfortunate (but not unlikely) event that interference on the channel chosen for FC transmission (either 12 or 20, in our case) grows out of control for a while, communication on links may be impaired seriously. On the contrary, CH manages to noticeably mitigate the effects of such a kind of events by reducing the variability of the observed behavior. This can be seen from the results in the lower part of Table 3, better visualized in the lower plot of Fig. 8. Exploiting an FC link tuned on channel 20 (for example, determined by a preliminary network scan) initially provides very good performance, but is unable to effectively cope with, e.g., a sudden increase of Wi-Fi traffic on channel 9.

Further benefits of CH can be appreciated by observing the rightmost column in all the aforementioned tables, which reports the average increase $\Delta\bar{E}_s$ of energy consumption due to retries for any two-way exchange (we can ignore failures, as they were either absent or very rare). As can be seen, variability in power consumption due to environmental conditions lowers, making duration of batteries more predictable.

E. PRACTICAL IMPLICATIONS

Unless the operating frequency is periodically adjusted based on the feedback about channel conditions (either by hand or using adaptive mechanisms), performance of FC solutions unavoidably drifts, and may sometimes fall below an acceptable threshold because of nonuncommon events (a newly installed high-power equipment, like a nonperfectly shielded industrial microwave oven, a wireless link used as a cable replacement to interconnect a subsystem over the air, etc.), some of which are hard to diagnose and fix (e.g., someone who occasionally transfers/streams bulk traffic from a mobile over Wi-Fi). True, sooner or later someone will notice that problems have arisen in data collection and will consequently undertake corrective actions, but this requires time and effort, and causes gaps in sample logs that may span over several days (and even weeks or months), making condition-based monitoring ineffective. As a real example, a number of LCD monitors communicating over Wi-Fi, deployed by the IT department in the shop floor of a large highly automated industrial plant to provide service information to the personnel, happened to cause troubles to wireless sensors and actuators located in the same area for several months before being diagnosed and fixed.

Instead, CH somehow "averages" the behavior of all the available channels, and prevents communication quality from degrading excessively, as long as not all frequencies are affected by interference. Since packet generation and slotframe repetition are asynchronous processes, we can in fact assume that the channel used for the initial transmission attempt of every frame is selected randomly according to a uniform distribution. In the case of transmission errors, retries are performed on pseudorandomly chosen channels. Again, we can intuitively assume that they are selected with the same likelihood, since every channel appears only once in the hopping list. Thus, the failure probability observed by the ARQ mechanism approximately corresponds to the arithmetic mean of what motes see on every single physical channel.

Very importantly, CH does not require any intervention (either periodic or on demand) from plant managers to reconfigure the transmission frequency of motes, hence decreasing maintenance cost and making it suitable for unattended (sub)systems.

VI. CONCLUSION

CH is a quite interesting approach that complements time slotting in TSCH. Despite its simplicity, it improves communication quality tangibly, which is beneficial for applications that rely on WSNs and require predictable behavior, e.g., monitoring systems in industrial plants for reactive and proactive maintenance. For this reason, it is very well suited to connect devices at the perception layer in IIoT architectures. The main goal of this article is to shed some light on the exact meaning of the word "improvement" when applied to CH. To this extent, a thorough measurement campaign was carried out on real devices deployed in a real environment. To enable reliable comparison of different solutions in spite of the unpredictability of the wireless spectrum, we exploited a well-established method that consists in interleaving experiments on a low-cost testbed. The very same approach can be used in other contexts and for a broader range of comparisons.

Results showed that CH is able to flatten the spectrum conditions, that is, every pair of communicating motes sees an equivalent transmission channel whose characteristics resemble, more or less, the average of the available physical channels. Therefore, while it is not true that a better quality

of communication is always achieved in absolute terms, more stable operations are, on average, guaranteed over time. In particular, CH tangibly reduces the likelihood that pathological conditions occur where communication worsens (due, e.g., to traffic on colocated Wi-Fi networks) and stays below an acceptable threshold for possibly long periods of time.

REFERENCES

[1] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Std. 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015), 2020.

[2] Zigbee Alliance, *ZigBee Specification (Rev. 21)*, 2015, pp. 1–565. [Online]. Available: https://zigbeealliance.org

[3] Core Specification Working Group, *Bluetooth Core Specification (ver 5.2)*, 2019, pp. 1–3256. [Online]. Available: https://www.bluetooth.com/

[4] LoRa Alliance Technical Committee, *LoRaWAN 1.1 Specification*, 2017, pp. 1–101. [Online]. Available: https://lora-alliance.org/

[5] *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 2: Sub 1 GHz License Exempt Operation*, IEEE Std. 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016), 2017.

[6] G. Cena, S. Scanzio, L. Seno, and A. Valenzano, "Comparison of mixed diversity schemes to enhance reliability of wireless networks," in *Ad-Hoc, Mobile, and Wireless Networks*, M. R. Palattella, S. Scanzio, and S. C. Ergen, Eds. Cham, Switzerland: Springer, 2019, pp. 118–135.

[7] *IEEE Standard for Local and Metropolitan Area Networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer*, IEEE Std 802.15.4e-2012, (Amendment to IEEE Std 802.15.4-2011), Apr. 2012.

[8] P. Thubert, "An architecture for IPv6 over the TSCH mode of IEEE 802.15.4," Tech. Rep. RFC 9030, May 2021. [Online]. Available: https://www.rfc-editor.org/rfc/rfc9030.txt

[9] S. Zoppi, H. M. Gürsu, M. Vilgelm, and W. Kellerer, "Reliable hopping sequence design for highly interfered wireless sensor networks," in *Proc. IEEE Int. Symp. Local Metrop. Area Netw.*, Jun. 2017, pp. 1–7.

[10] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre, "LABeL: Link-based adaptive BLacklisting technique for 6TiSCH wireless industrial networks," in *Proc. 20th ACM Int. Conf. Model., Anal. Simul. Wireless Mobile Syst.*, Nov. 2017, pp. 25–33, doi: 10.1145/3127540.3127541.

[11] G. Cena, S. Scanzio, and A. Valenzano, "Ultra-low power wireless sensor networks based on time slotted channel hopping with probabilistic blacklisting," *Electronics*, vol. 11, no. 3, p. 304, Jan. 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/3/304

[12] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On optimal scheduling in duty-cycled industrial IoT applications using IEEE802.15.4e TSCH," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3655–3666, Oct. 2013.

[13] A. Elsts, S. Kim, H. Kim, and C. Kim, "An empirical survey of autonomous scheduling methods for TSCH," *IEEE Access*, vol. 8, pp. 67147–67165, 2020.

[14] S. Jeong, J. Paek, H. Kim, and S. Bahk, "TESLA: Traffic-aware elastic slotframe adjustment in TSCH networks," *IEEE Access*, vol. 7, pp. 130468–130483, 2019.

[15] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. S. J. Pister, "A realistic energy consumption model for TSCH networks," *IEEE Sensors J.*, vol. 14, no. 2, pp. 482–489, Feb. 2014.

[16] S. Scanzio, G. Cena, and A. Valenzano, "Enhanced energy-saving mechanisms in TSCH networks for the IIoT: The PRIL approach," *IEEE Trans. Ind. Inform.*, vol. 19, no. 6, pp. 7445–7455, Jun. 2023.

[17] S. Scanzio et al., "Wireless sensor networks and TSCH: A compromise between reliability, power consumption, and latency," *IEEE Access*, vol. 8, pp. 167042–167058, 2020.

[18] D. De Guglielmo, B. Al Nahas, S. Duquennoy, T. Voigt, and G. Anastasi, "Analysis and experimental evaluation of IEEE 802.15.4e TSCH CSMA-CA algorithm," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1573–1588, Feb. 2017.

[19] C. Ouanteur, L. Bouallouche-Medjkoune, and D. Aïssani, "An enhanced analytical model and performance evaluation of the IEEE 802.15.4e TSCH CA," *Wireless Pers. Commun.*, vol. 96, no. 1, pp. 1355–1376, Sep. 2017.

[20] T. van der Lee, S. Raza, G. Exarchakos, and M. Gunes, "Towards co-located TSCH networks: An inter-network interference perspective," in *Proc. IEEE Glob. Commun. Conf.*, 2018, pp. 1–6.

[21] M. Chwalisz and A. Wolisz, "Towards efficient coexistence of IEEE 802.15.4e TSCH and IEEE 802.11," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp.*, 2018, pp. 1–7.

[22] M. Gürsu, M. Vilgelm, S. Zoppi, and W. Kellerer, "Reliable coexistence of 802.15.4e TSCH-based WSN and Wi-Fi in an aircraft cabin," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2016, pp. 663–668.

[23] R. Koutsiamanis, G. Z. Papadopoulos, X. Fafoutis, J. M. D. Fiore, P. Thubert, and N. Montavont, "From best effort to deterministic packet delivery for wireless industrial IoT networks," *IEEE Trans. Ind. Inform.*, vol. 14, no. 10, pp. 4468–4480, Oct. 2018.

[24] T. Watteyne, A. Mehta, and K. Pister, "Reliability through frequency diversity: Why channel hopping makes sense," in *Proc. 6th ACM Symp. Perform. Eval. Wireless Ad Hoc, Sensor, Ubiquitous Netw.*, 2009, p. 116–123.

[25] J. Muñoz, E. Riou, X. Vilajosana, P. Muhlethaler, and T. Watteyne, "Overview of IEEE802.15.4 g OFDM and its applicability to smart building applications," in *Proc. Wireless Days*, 2018, pp. 123–130.

[26] J. Muñoz, P. Muhlethaler, X. Vilajosana, and T. Watteyne, "Why channel hopping makes sense, even with IEEE802.15.4 OFDM at 2.4 GHz," in *Proc. Global. Internet Things Summit*, 2018, pp. 1–7.

[27] G. Cena, C. G. Demartini, M. Ghazi Vakili, S. Scanzio, A. Valenzano, and C. Zunino, "Evaluating and modeling IEEE 802.15.4 TSCH resilience against Wi-Fi interference in new-generation highly-dependable wireless sensor networks," *Ad Hoc Netw.*, vol. 106, 2020, Art. no. 102199.

[28] Q. Wang, K. Jaffrès-Runser, Y. Xu, J. Scharbarg, Z. An, and C. Fraboul, "TDMA versus CSMA/CA for wireless multihop communications: A. stochastic worst-case delay analysis," *IEEE Trans. Ind. Inform.*, vol. 13, no. 2, pp. 877–887, Apr. 2017.

[29] D. D. Falconer, F. Adachi, and B. Gudmundson, "Time division multiple access methods for wireless personal communications," *IEEE Commun. Mag.*, vol. 33, no. 1, pp. 50–57, Jan. 1995.

[30] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, "The evolution of MAC protocols in wireless sensor networks: A survey," *IEEE Commun. Surv. Tuts.*, vol. 15, no. 1, pp. 101–120, First Quarter 2013.

[31] Q. Wang, K. Jaffrès-Runser, Y. Xu, J. Scharbarg, Z. An, and C. Fraboul, "TDMA versus CSMA/CA for wireless multi-hop communications: A comparison for soft real-time networking," in *Proc. IEEE World Conf. Factory Commun. Syst.*, 2016, pp. 1–4.

[32] V. P. Rao and D. Marandin, "Adaptive channel access mechanism for zigbee (IEEE 802.15.4)," *J. Commun. Softw. Syst.*, vol. 2, no. 4, pp. 283–293, 2017.

[33] A. Lavric, V. Popa, and S. Sfichi, "Adaptive channel selection algorithm for a large scale street lighting control ZigBee network," *Elektronika ir Elektrotechnika*, vol. 19, no. 9, pp. 105–109, Nov. 2013.

[34] M. S. Kang, J. W. Chong, H. Hyun, S. M. Kim, B. H. Jung, and D. K. Sung, "Adaptive interference-aware multi-channel clustering algorithm in a ZigBee network in the presence of WLAN interference," in *Proc. 2nd Int. Symp. Wireless Pervasive Comput.*, 2007, pp. 200–205.

[35] M. A. Sarijari, M. S. Abdullah, A. Lo, and R. A. Rashid, "Experimental studies of the ZigBee frequency agility mechanism in home area networks," in *Proc. 39th Annu. IEEE Conf. Local Comput. Netw. Workshops*, 2014, pp. 711–717.

[36] G. Cena, S. Scanzio, and A. Valenzano, "A prototype implementation of Wi-Fi seamless redundancy with reactive duplication avoidance," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Automat.*, Sep. 2018, vol. 1, pp. 179–186.

[37] OpenMote, Feb. 21, 2021. [Online]. Available: https://www.industrialshields.com/open-mote-b-industrial-shields-open-source-device-ready-for-internet-of-things

[38] OpenWSN, Feb. 21, 2021. [Online]. Available: https://openwsn.atlassian.net/wiki/

[39] G. Cena, S. Scanzio, and A. Valenzano, "Experimental evaluation of techniques to lower spectrum consumption in Wi-Red," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 824–837, Feb. 2019.

[40] S. Scanzio et al., "Wireless sensor networks dataset (TSCH a compromise between reliability, power consumption, and latency)," *IEEE Dataport*, 2020, doi: 10.21227/fg62-bp39.
[41] M. Mongelli and S. Scanzio, "A neural approach to synchronization in wireless networks with heterogeneous sources of noise," *Ad Hoc Netw.*, vol. 49, pp. 1–16, 2016.
[42] Q. Wang, X. Vilajosana, and T. Watteyne, "6TiSCH operation sublayer (6top) protocol (6P)," Internet Engineering Task Force (IETF), IETF RFC 8480, Nov. 2018.
[43] G. Cena, S. Scanzio, L. Seno, A. Valenzano, and C. Zunino, "Energy-efficient link capacity overprovisioning in time slotted channel hopping networks," in *Proc. IEEE 16th Int. Conf. Factory Commun. Syst.*, 2020, pp. 1–8.
[44] S. Scanzio, G. Cena, A. Valenzano, and C. Zunino, "Energy saving in TSCH networks by means of proactive reduction of idle listening," in *Ad-Hoc, Mobile and Wireless Networks*, L. A. Grieco, G. Boggia, G. Piro, Y. Jararweh, and C. Campolo, Eds. Cham, Switzerland: Springer, 2020, pp. 131–144.
[45] G. Cena, S. Scanzio, and A. Valenzano, "Improving effectiveness of seamless redundancy in real industrial Wi-Fi networks," *IEEE Trans. Ind. Inform.*, vol. 14, no. 5, pp. 2095–2107, May 2018.



GIANLUCA CENA (Senior Member, IEEE) received the M.S. degree in electronic engineering and the Ph.D. degree in information and system engineering from the Politecnico di Torino, Torino, Italy, in 1991 and 1996, respectively.

Since 2005, he has been a Director of Research with the Institute of Electronics, Information Engineering and Telecommunications, National Research Council of Italy (CNR-IEIIT), Torino. He has co-authored about 170 papers and one patent in his research interests, which include wired and wireless industrial communication systems, real-time protocols, and automotive networks.

Dr. Cena was the recipient of the Best Paper Award of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS in 2017 and of the IEEE Workshop on Factory Communication Systems in 2004, 2010, 2017, 2019, and 2020. He has served as the Program Co-Chairman of the IEEE Workshop on Factory Communication Systems in 2006 and 2008, and as the Track Co-Chairman in six editions of the IEEE Conference on Emerging Technologies and Factory Automation. Since 2009, he has been serving as an Associate Editor for the IIEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.



STEFANO SCANZIO (Senior Member, IEEE) received Laurea and Ph.D. degrees in computer science from Politecnico di Torino, Torino, Italy, in 2004 and 2008, respectively.

From 2004 to 2009, he was with the Department of Computer Engineering, Politecnico di Torino, where he was involved in research on speech recognition and in classification methods and algorithms. Since 2009, he has been with the National Research Council of Italy, where he is currently a Senior Researcher with the institute CNR-IEIIT. He teaches several courses on computer science with Politecnico di Torino. He has authored and co-authored more than 80 papers in international journals and conferences, in the areas of industrial communication systems, real-time networks, wireless networks, and artificial intelligence.

Dr. Scanzio took part in the Program and Organizing Committees of many international conferences of primary importance in his research areas. He was the recipient of the 2017 Best Paper Award of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and the Best Paper Awards for three papers presented at the IEEE Workshops on Factory Communication Systems in 2010, 2017, and 2019, and for a paper presented at the IEEE International Conference on Factory Communication Systems in 2020. He is an Associate Editor for *Ad Hoc Networks* (Elsevier), IEEE ACCESS, and *Electronics* (MDPI) journals.



MOHAMMAD GHAZI VAKILI received the Ph.D. degree in computer engineering from the Politecnico di Torino University, Torino, Italy, in 2021.

He is currently a Postdoctoral Fellow with the Matter Lab, Department of Chemistry, University of Toronto, Toronto, Canada, jointly with Zapata Computing, where he works on material discovery with deep learning. He is currently working on Quantum and Quantum inspire machine learning algorithms to discover new material. His research interests include machine learning, industrial optimization, and quantum computing in Industry 5.Q (future of the factories).



CLAUDIO GIOVANNI DEMARTINI (Senior Member, IEEE) received the Ph.D. degree in information and systems engineering from the Politecnico di Torino, Torino, Italy, in 1987.

His teaching activities include graduate-level courses on "information systems" and "innovation management and product development." From 2003 to 2012, he was the Deputy Dean of the Industrial Engineering and Management Faculty and Deputy Chancellor of the Politecnico di Torino. He was in charge as a Member of the Academic Senate and Head of the Department of Computer Engineering until 2019. As a member of the Board of Directors of the Education Foundation—Compagnia di San Paolo, he addressed issues related to the Education Community. His main research interests include distributed systems, formal description techniques, software engineering, education, product life cycle, and innovation management.



ADRIANO VALENZANO (Senior Member, IEEE) received the Laurea degree magna cum laude in electronic engineering from Politecnico di Torino, Torino, Italy, in 1980.

He has been a Director of Research with the National Research Council of Italy (CNR) since 1991. He is currently with the Institute of Electronics, Computer and Telecommunication Engineering, Torino, where he is responsible for research concerning distributed computer systems, local area networks, and communication protocols. He has co-authored approximately 200 refereed journal and conference papers in the area of computer engineering.

Dr. Valenzano was the recipient of the 2013 IEEE IES and ABB Lifetime Contribution to Factory Automation Award. He was also the recipient of the Best Paper Award for the paper published in the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS during 2016, and the Best Paper Awards for the papers presented at the 5th, 8th, 13th, 15th and 16th IEEE Workshops on Factory Communication Systems, in 2004, 2010, 2017, 2019, and 2020, respectively. He has served as a Technical Referee for several international journals and conferences, also taking part in the program committees of international events of primary importance. Since 2007, he has been serving as an Associate Editor for the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.

Open Access funding provided by 'Consiglio Nazionale delle Ricerche-CARI-CARE-ITALY' within the CRUI CARE Agreement