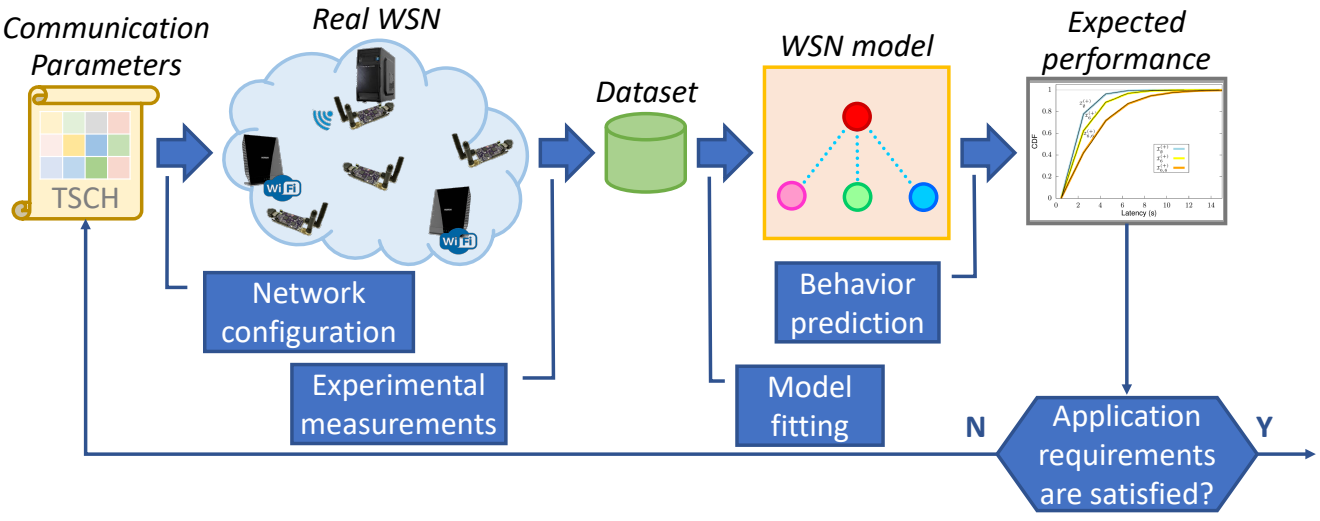
Evaluating and Modeling IEEE 802.15.4 TSCH Resilience against Wi-Fi Interference in New-Generation Highly-Dependable Wireless Sensor Networks

Graphical Abstract

Evaluating and Modeling IEEE 802.15.4 TSCH Resilience against Wi-Fi Interference in New-Generation Highly-Dependable Wireless Sensor Networks

Gianluca Cena, Claudio G. Demartini, Mohammad Ghazi Vakili, Stefano Scanzio, Adriano Valenzano, Claudio Zunino



Highlights

Evaluating and Modeling IEEE 802.15.4 TSCH Resilience against Wi-Fi Interference in New-Generation Highly-Dependable Wireless Sensor Networks

Gianluca Cena, Claudio G. Demartini, Mohammad Ghazi Vakili, Stefano Scanzio, Adriano Valenzano, Claudio Zunino

- The correct behavior of networked systems depends on the quality of communication
- Time slotted channel hopping effectively counteracts disturbance and interference
- Wi-Fi networks are a major source of interference for wireless sensor networks
- Wi-Fi interference on data exchanges between motes can be modeled satisfactorily
- Obtained model is a valuable tool to ease setting of communication parameters

Evaluating and Modeling IEEE 802.15.4 TSCH Resilience against Wi-Fi Interference in New-Generation Highly-Dependable Wireless Sensor Networks

Gianluca Cena[a], Claudio G. Demartini[b], Mohammad Ghazi Vakili[b], Stefano Scanzio[a,*], Adriano Valenzano[a], Claudio Zunino[a]

[a]*National Research Council of Italy (CNR-IEIIT), Corso Duca degli Abruzzi 24, I-10129 Torino, Italy*
[b]*Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, I-10129 Torino, Italy*

Abstract

Thanks to its ability to effectively counteract disturbance and interference, including the traffic generated by co-located Wi-Fi networks, Time Slotted Channel Hopping (TSCH) is currently gaining momentum in many application fields characterized by demanding reliability and determinism requirements. In particular, the ability of TSCH to change transmission frequency on every attempt sensibly mitigates packet losses and latencies, improving the overall behavior in a tangible way.

In this paper, the communication quality achieved by TSCH in a setup that includes real motes exposed to a realistic interfering traffic is evaluated experimentally. A theoretical model is also developed, based on quite simple assumptions about the effectiveness of time and frequency diversity, which satisfactorily matches the real behavior. The model permits to determine how much network parameters like, e.g., the retry limit, actually affect communication, and can be exploited to find proper settings for them. Finally, the ability of channel hopping to prevent narrowband interference from disrupting communication is assessed. As results show, this mechanism makes motes suffer from an equivalent interference that roughly corresponds to the mean interference evaluated over all physical channels.

Keywords: Time Slotted Channel Hopping (TSCH), IEEE 802.15.4, Communication performance modeling, Wireless Sensors Networks (WSN), Resilience to disturbance and interference, IEEE 802.11 (Wi-Fi)

1. Introduction

Wireless Sensor Networks (WSN) were introduced about two decades ago to provide low-cost connectivity over the air to simple and inexpensive devices [1]. One of the primary goals of WSNs was to collect data from a multitude of sensors, possibly deployed over wide areas, without the need to set up a wired network infrastructure. Among the application fields where these solutions are mostly employed there are, e.g., precision agriculture [2, 3], environmental monitoring [4, 5], natural disaster management [6, 7], and diagnostics in large industrial plants [8, 9].

One of the most popular transmission technologies adopted in WSNs is IEEE 802.15.4 [10]. It is characterized by very low power consumption, which makes it suitable for battery-powered devices. Moreover, since most of the related medium access control (MAC) mechanism is carried out in software, it enables simple and highly-flexible implementations. Although IEEE 802.15.4 defines a beacon-oriented transmission mode with Guaranteed Time Slots (GTS) to improve determinism, beaconless operations, which provide asynchronous network access, are customarily adopted in real-world applications. To increase reliability of legacy WSNs, the Time Slotted Channel Hopping (TSCH) mechanism was subsequently defined. TSCH is an enhanced MAC technique, which sensibly decreases the likelihood that packets sent by applications are dropped, due to either disturbance (e.g., electromagnetic noise) or interference from nearby wireless devices.

In this paper, the ability of TSCH to counteract such phenomena is analyzed. In particular, we considered the relevant case where the source of interference coincides with the traffic of co-located Wi-Fi networks. A basic model is first introduced, which relies on a simple yet relevant channel error model at the physical layer and describes higher-layer request-response interactions like those defined by the Constrained Application Protocol (CoAP) [11]. Then, this model has been validated by means of a thorough experimental campaign carried out on real devices. Results highlighted the advantages TSCH offers over legacy WSN technologies in the presence of heavily loaded Wi-Fi channels.

Here, we focused on star network topologies, where all motes are located one hop away from the root. This choice is not particularly limiting. In fact, in several application contexts a number of simple devices (sensors, actuators) have to be connected over the air to the wired infrastructure by using specific hubs, each of which acts as the gateway for the wireless devices in line of sight. This is the case of, e.g., the control system of robotized production cells in industrial plants, where the application master has to communicate with decentralized

*Corresponding author.
Email addresses: gianluca.cena@ieiit.cnr.it (Gianluca Cena), claudio.demartini@polito.it (Claudio G. Demartini), mohammad.ghazivakili@polito.it (Mohammad Ghazi Vakili), stefano.scanzio@ieiit.cnr.it (Stefano Scanzio), adriano.valenzano@ieiit.cnr.it (Adriano Valenzano), claudio.zunino@ieiit.cnr.it (Claudio Zunino)

peripherals fastened to moving parts of the equipment. Besides IEEE 802.15.4, two relevant examples are given by IO-Link Wireless [12] and, over large areas, LoRaWAN [13]. For star topologies, simple and meaningful closed-form expressions can be derived for performance indices.

This paper is structured as follows: in Section 2 the TSCH protocol is briefly recalled and the problem of Wi-Fi interference is described, while in Section 3 a simple theoretical model is introduced to characterize the communication quality experienced by request-response interactions. In Section 4 the hardware setup and the procedure employed to perform experimental measurements are illustrated. Results for two experimental campaigns, carried out with and without channel hopping, are presented and commented in Section 5. Finally, in Section 6 some conclusions are drawn.

2. TSCH

The now superseded IEEE 802.15.4e [14] specification defined three operating modes [15]: Time Slotted Channel Hopping (TSCH), Deterministic and Synchronous Multi-channel Extension (DSME), and Low Latency Deterministic Network (LLDN). Only TSCH and DSME were retained in the most recent version of IEEE 802.15.4. In particular, in the past few years, TSCH has been the most popular option for real-world devices, e.g., those based on WirelessHART [16].

In this paper we will explicitly consider the 6TiSCH protocol, which layers IPv6 over the TSCH mode of IEEE 802.15.4 [17]. The reason of this choice is twofold. First, TSCH provides a noticeably higher communication reliability than legacy IEEE 802.15.4 operating modes, retaining the same very-low power consumption. Even better results can be obtained, from these points of view, in dense environments, where the likelihood of intra-network collisions cannot be neglected. Second, IP adoption tangibly eases integration of sensor networks in the existing communication backbones, and simplifies application development by enabling asynchronous request-response interactions with motes.

2.1. TSCH Protocol Basics

TSCH is located in the data-link layer and takes care of frames exchanges between neighboring nodes. As the name suggests, it relies on two distinct but interrelated mechanisms: *time slotting* and *channel hopping*.

2.1.1. Time slotting

Time slotting, which resembles more generic time division multiple access (TDMA) approaches [18], splits time into fixed duration windows called *slotframes*, each of which consists in a fixed number N_{slot} of *slots*. All slots have the same duration T_{slot}, which is selected in such a way to accommodate one data frame and, for confirmed transmissions, the related acknowledgment frame. After network configuration has been carried out, some slots, uniquely identified by their position in the slotframe, are assigned to specific *links*. Every link is characterized by the source and destination nodes involved in the frame exchange (the broadcast address is also allowed for the destination). This means that the slot is exclusively devoted to the (unidirectional) communication between said nodes, thus preventing any collisions.

Each node, whose time is kept synchronized with the other nodes of the network [19], maintains a copy of the descriptors for the links that are relevant to itself, and wakes up only when, coherently to its schedule, it either has a pending packet ready to be sent or expects that a packet can be (possibly) received. On the one hand, this behavior is essential in order to reduce energy consumption [20]. On the other hand, the complexity of maintaining the nodes synchronized is one of the main drawbacks of TDMA-based techniques (and, consequently, of TSCH).

Slotframe transmission is repeated indefinitely, which means that, at the data-link layer, communications between adjacent nodes are scheduled cyclically with period $T_{\text{slfr}} = N_{\text{slot}} \cdot T_{\text{slot}}$. In reality, not necessarily a given slot is actually used in every slotframe. In fact, the 6TiSCH paradigm [21], which layers IPv6 over TSCH, provides users of the IP layer, e.g., applications communicating through CoAP, with true asynchronous network access. This implies that periodicity of data exchanges, as seen at the application layer, is very loosely tied to the slotframe duration and structure defined at the data-link layer, which permits to easily change the former at runtime without the need to reconfigure MAC parameters. Besides, on demand node access is fully supported.

Thanks to a suitable configuration of the transmission schedule (as per the links in the slotframe), collisions between nodes of the same network are prevented in typical operating conditions. Actually, shared cells can be defined, which are not reserved to a single source node. For them, collision may occur, and hence a suitable collision avoidance procedure is defined based on random exponential backoff. However, shared cells are rather peculiar, and usually they are not employed to support application data exchanges.

2.1.2. Channel Hopping

While intra-network interference can be suitably prevented by having the nodes cooperate with each other (by means, e.g., of time slotting), unpredictable phenomena like disturbance (for instance, noise generated by industrial equipment) and external interference (due to wireless devices that rely on different MAC protocols) can not. In this case, time diversity coupled with Automatic Retransmission reQuest (ARQ) techniques is customarily exploited to ensure adequate reliability in spite of possible transmission errors, which corrupt the frame while it is traveling on air.

A drawback of the above approach is that it is weak against heavy narrowband disturbance/interference affecting the physical transmission channel [22]. An effective solution consists in coupling time diversity with frequency diversity [23], for instance by re-tuning the radio block on every transmission attempt. In TSCH, such a technique is referred to as channel hopping. From a practical point of view, the resulting mixed diversity approach [24, 25] is noticeably more resilient to disturbance and external interference. As we will see, the equivalent

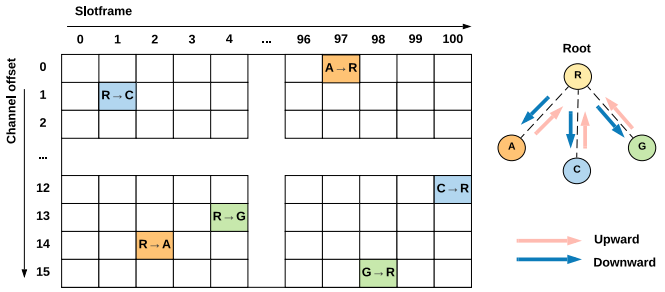


Figure 1: Sample TSCH Matrix (3 downward and 3 upward cells).

level of disturbance experienced by packet transmissions is sort of an average among the amounts of disturbance that separately affect every physical channel. This also implies that, as long as at least one of the channels is not severely disturbed, packet delivery will eventually succeed.

When channel hopping is enabled, the schedule of transmissions in the slotframe is actually defined through a TSCH matrix, an example of which is schematically depicted in Fig. 1. Every cell, found at the intersection between a specific slot (identified by the related column) and a specific channel offset (corresponding to the row), describes a link. Starting from the channel offset, an ever-increasing shared counter denoted Absolute Slot Number (ASN), and a globally defined hopping table (*macHoppingSequenceList*), both senders and receivers can determine the physical channel on which the frames (DATA and ACK) related to a given link will be actually transmitted at any time.

Besides channel hopping, two other mechanisms can be possibly adopted in WSNs based on TSCH, known as channel black-listing and white-listing. They operate by selectively disabling and enabling some of the physical channels in the hopping table, respectively, in an attempt to increase performance [26]. As a matter of fact, many real implementations, including those we used in this work, do not support these techniques.

2.2. *Wi-Fi Interference*

A non-negligible problem one has to face when deploying WSNs in residential or industrial areas is that, very likely, other wireless devices actively transmitting over the air are found nearby, which may interfere with communication. In particular, Wi-Fi network infrastructures based on the IEEE 802.11 technology [27], also known as Basic Service Sets (BSS), are today widely employed in home and office environments, but very often they are also found in factories, hospitals, shopping malls, and so on. Most of them still transmit in the 2.4 GHz industrial, scientific, and medical (ISM) band, which is the same that is used by many devices communicating via IEEE 802.15.4.

A careful planning of frequencies of co-located networks, so as to limit interference, is often unfeasible, as different kinds of networks are typically managed by different owners/administrators. This means that, unavoidably, packets exchanged over WSNs will suffer from delays due to interference and collisions. In the worst case, a substantial fraction of packets may be even dropped.

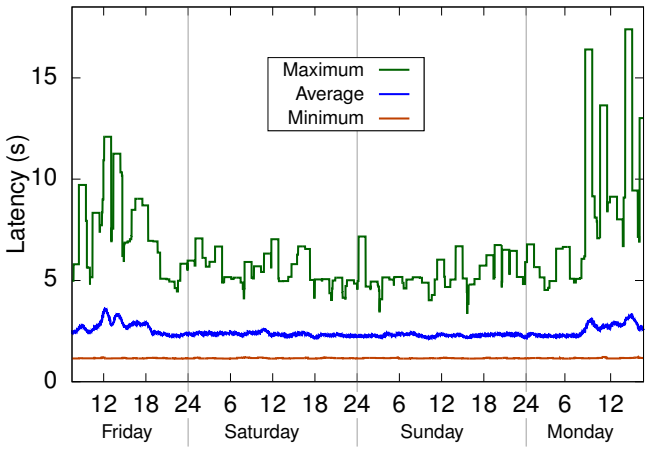


Figure 2: Measured `ping` round-trip time (Max./Avg./Min. values, Friday-to-Monday, 1-hour moving average).

As an example, Fig. 2 shows the round-trip time reported by the `ping` utility when it is run continuously in our experimental TSCH network setup over four days (Friday to Monday). To make plots more readable, statistics (maximum, average, and minimum) have been computed over a sliding window including 120 samples, which correspond to one hour. As can be seen, the latency is lower during weekend and overnight in weekdays. Conversely, in working hours the worst-case latency can grow up to three times as much. It is worth noting that no high-power electrical machines were present close to the testbed. This means that worsening of the communication quality was due for the most part to the contextual traffic generated by Wi-Fi devices (mobiles, PCs, access points) located in premises near our laboratory. As shown in Fig. 3, more than ten BSSs were visible from our setup. Above figures provide some clues on the practical extent to which Wi-Fi may affect the communication quality perceived by WSN motes.

One could argue that WSNs were not conceived to support deterministic traffic, and that delays and occasional packet losses are perfectly acceptable in many application contexts (e.g., precision agriculture). On the other hand, the higher reliability offered by TSCH makes this protocol suitable also

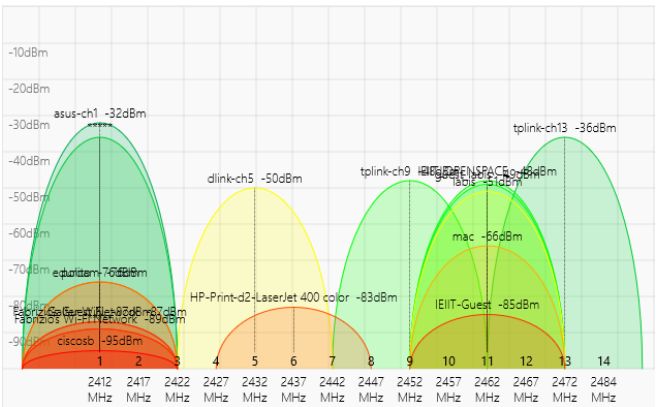


Figure 3: Wi-Fi spectrum usage during experimental campaigns.

for mid-criticality scenarios (e.g., industrial plants, mission-critical applications, disaster management), where some guarantees have to be provided for what concerns the time taken to deliver information [28, 29].

For the sake of truth, a number of solutions exist that are characterized by low power consumption and very low latency. This is the case of the Wireless Short-Packet (WSP) protocol optimized for energy harvesting defined by IEC 14543-3 [30], which suits the needs of lighting control applications in Home Electronic Systems (HES), or the Low Latency Deterministic Network (LLDN) protocol, defined as an amendment to IEEE 802.15.4 and targeted to industrial applications. Such solutions were not conceived for meshing, which in some cases (e.g., LLDN) is not even supported. Generally speaking, conserving energy implies small duty cycles on intermediate nodes, which unavoidably impair network responsiveness for asynchronous packets because of the tradeoff between energy and latency [31]. Other approaches are based on wake-up radios. In this case, an additional ultra-low-power receiver module is included, whose purpose is to wake up on demand the primary radio [32]. However, commercial products that comply with such technology are hardly available at present.

By carefully configuring the schedule of links in the slotframe, and in the absence of disturbance and interference, end-to-end communications (including two-way ones) are allowed to take place within the same slotframe [33, 34, 35]. This means that the round-trip delay for queries based on request-response paradigm can be, in theory, as low as 1–2 s. TSCH networks are quite reactive and, especially in the case where motes are not powered only on batteries (e.g., when energy harvesting [36, 37] is exploited), they can be even adopted to reliably close control loops with slow dynamics.

In this context, the ability to estimate the quality of communication TSCH achieves in the real world as a function of the amount of interfering Wi-Fi traffic is very important for deciding whether or not the requirements demanded by the applications can be meet. Moreover, assessing what improvements TSCH offers with respect to legacy WSN solutions from a quantitative point of view can be quite valuable in order to determine if benefits are worth the additional protocol complexity.

The effects of the mutual interference between different wireless network technologies has been widely studied in the recent past. In [38, 39] the cross-interference between Wi-Fi and ZigBee was analyzed. The effect of IEEE 802.15.4 traffic on a Wi-Fi network was evaluated in [40]. The other direction, i.e., the influence of Wi-Fi traffic on legacy (non-TSCH) IEEE 802.15.4 networks, has been evaluated in many research works [41, 42, 43, 44]. The specific interference caused by Wi-Fi on 6TiSCH networks (based on IEEE 802.15.4 with TSCH) was evaluated in [45] and [46], with particular attention to the benefits brought by channel hopping on performance. Finally, solutions that combine channel hopping with white- and blacklisting to improve reliability were evaluated in [47].

Unlike above works, this paper analyzes the effects of Wi-Fi interference on TSCH by means of a theoretical model. Many theoretical models were proposed in the scientific literature for IEEE 802.15.4 and IEEE 802.11. Most of them only consider a single technology, excluding any interactions with other wireless protocols. Earlier investigations, based on Markov chains, concerned Wi-Fi [48, 49, 50]. This approach was then extend to tackle TSCH as well. In particular, in [51, 52] some analytical derivations are proposed to describe transmission in shared cells. Papers dealing with the mutual interaction between IEEE 802.15.4 and IEEE 802.11 typically refer to older versions of the standards. In particular, [53, 54] report on some theoretical investigation about the effect of Wi-Fi traffic on a pre-TSCH version of IEEE 802.15.4. In our work, measurements obtained from a real setup are used to suitably set the parameters of a mathematical model, so that a number of performance indices related to the quality of communication perceived by motes can be obtained through simple computations.

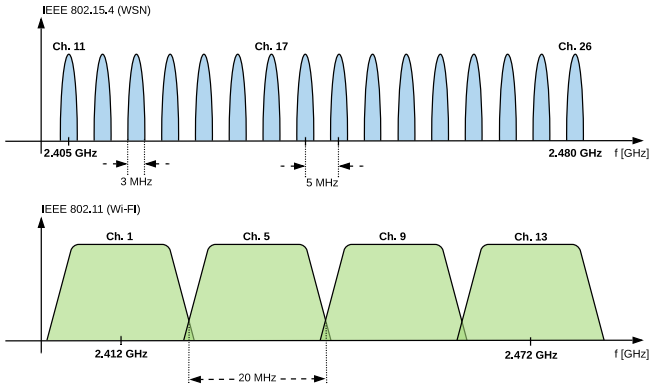


Figure 4: Overlapping channels in IEEE 802.11 and 802.15.4 (ISM band).

From a practical point of view, the physical layers of IEEE 802.15.4 and IEEE 802.11 have different characteristics. In particular, the width of a single OFDM channel in IEEE 802.11 is 20 MHz (22 MHz for DSSS). By contrast, up to 16 IEEE 802.15.4 channels are defined in the ISM band, whose frequencies fall in the range from 2.405 to 2.480 GHz, and the width of each one of such channels is 5 MHz. In the following, the type of channel (Wi-Fi or WSN) will be explicitly specified whenever it can not be clearly inferred from the context.

As can be seen in Fig. 4, each single Wi-Fi link overlaps with about 4 adjacent WSN channels (or 8, when channel bonding is exploited). This implies that when (at least) four Wi-Fi interfering nodes are actively transmitting on air, tuned on channels 1, 5, 9, and 13, then every single IEEE 802.15.4 channel in the ISM band unavoidably suffers from interference.

3. Two-way Communication Model

In the following, the behavior of a TSCH link between two adjacent motes in the presence of unpredictable phenomena, like disturbance and interference from devices not belonging to the network, will be analyzed from a theoretical viewpoint. Intra-network interference will not be taken into account since, once the network topology has settled and the transmission schedule (as per the slotframe) has been defined and configured on motes, collisions no longer occur in typical operating conditions. Single-hop communication (between adjacent nodes)

can be easily modeled starting from TSCH operations and the channel error model.

3.1. Packet Losses on a Single Hop

Retransmission techniques are customarily adopted in wireless networks to increase reliability. Therefore, more than one transmission attempt could be performed on air for the same packet: exactly one initial attempt plus, in the case of failures, a variable number of retries. Single attempts are considered as failed when the frame is sent on air but the related ACK is not received back. In such event, acknowledged transmissions are automatically repeated by the MAC up to r_{L} times, where the *retry limit* r_{L} coincides with the *macMaxFrameRetries* parameter of the TSCH protocol. This means that a packet is definitely lost only if all its $r_{\mathrm{L}} + 1$ attempts fail. Optionally, clear channel assessment (CCA) can be enabled, which defers a transmission attempt when the channel is sensed busy. From our point of view, this condition is almost the same as the lack of an ACK. In fact, in both cases the retry counter is increased by one and a retransmission of the frame is scheduled in the next suitable slot. It is worth pointing out that retransmission in TSCH does not take place immediately. Instead, the transmitter has to wait for a link (either dedicated or shared) targeted to the destination device.

In the following, we assume that single transmission attempts are modeled as Bernoulli trials with failure probability ϵ. Such hypothesis seemingly provides a rough approximation of the actual channel behavior. Nevertheless, subsequent attempts in our TSCH setup are spaced wide enough, so that time diversity sensibly lessen statistical dependence between them. If, as customarily done in real networks, a single dedicated link in the slotframe is allocated between any pair of motes, transmission attempts in the related slots are spaced by the slotframe duration (about 2 s), which is longer than the default MSDU lifetime in IEEE 802.11 (i.e., the maximum time after which the transmission process for a frame in Wi-Fi, including retries, is terminated). For shared links, a random exponential backoff mechanism is additionally employed to prevent collisions, which further enlarges the time between retries. Above behavior was verified for the real devices we used for experimentation. Additionally, because of channel hopping, unlikely subsequent attempts for the same packet are performed on the same physical channel, which means that frequency diversity is exploited as well. If, from a practical point of view, diversity techniques make retries (almost) statistically independent, then the *packet loss ratio* P_{L} on the link can be computed as

$$P_{\mathrm{L}} = \epsilon^{r_{\mathrm{L}}+1}. \tag{1}$$

For example, in the case $r_{\mathrm{L}} = 15$ and assuming that 20% of attempts fail, $P_{\mathrm{L}} = 6.55 \times 10^{-12}$, while when the failure probability grows up to 50% we have $P_{\mathrm{L}} = 1.526 \times 10^{-5}$.

3.2. Failure Rate for Two-way Communication

To easily match our theoretical analysis with experimental data captured from a real testbed, we will consider request-response transactions implemented by the `ping` utility, which relies on the Internet Control Message Protocol (ICMP) [55]. In fact, in this case the same mote (the root) is in charge of generating the test traffic and performing measurements on it. The behavior of `ping` closely resembles those applications where sensors are explicitly queried via CoAP. Each one of such transactions is made up of an *echo request* message, sent by the originating mote to the target mote, and an *echo reply* message, returned by the target mote to the originating mote. On single-hop paths, like the ones found in the star network topologies we are considering here, this means that one packet is sent by the root in the *downward* direction, followed by a subsequent packet, generated as a reply by the target mote in the *upward* direction.

Table 1: Glossary of Quantities.

Quantity	Description	Value
N_{ch}	Number of physical channels	16
N_{slot}	Number of slots in a slotframe	101
T_{slot}	Duration of a slot	20 ms
T_{slfr}	Period of the slotframe	2.02 s
r_{L}	Max. number of MAC retransmissions (retry limit)	15
ϵ	Failure probability for single attempts	-
P_{L}^{x}	Packet loss ratio in direction $x \in \{D, U\}$	-
P_{L}^{T}	Two-way loss ratio	-
P_{r}^{x}	Probability to perform r retries in direction $x \in \{D, U\}$	-
P_{r}^{T}	Probability to perform r retries in both directions	-
N_{sam}	Number of samples per experiment	2880
d_i	Two-way transmission latency of the i-th `ping` request	-
$d_{\mathrm{wait},i}$	Waiting time of the i-th `ping` request	$< T_{\mathrm{slfr}}$
d_{comm}	Two-way network communication time	-
$d_{\mathrm{retr},i}$	Retransmission time of the i-th `ping` request	-
r_i	Total number of retries for the i-th `ping` request	-
D	Two-way transmission latency (random variable)	-
D_{wait}	Waiting time (random variable)	$< T_{\mathrm{slfr}}$
D_{retr}^{x}	Retransmission time in direction $x \in \{D, U\}$ (rand. var.)	-
D_{retr}^{T}	Two-way retransmission time (random variable)	-
μ_d	Two-way mean transmission latency	-
$\hat{\mu}_r$	Estimated mean number of retries (two-way)	-
$\hat{P}_{r}^{T}$	Empirical probability to perform r retries (two-way)	-
$\hat{\epsilon}_P$	Estimated failure probability (from $\hat{P}_{0}^{T}$)	-
$\hat{\epsilon}_D$	Estimated failure probability (from μ_d)	-

The probability P_{L}^{T} that a request-response transaction fails, we denote for short *two-way loss ratio*, corresponds to

$$P_{\mathrm{L}}^{T} = 1 - \left(1 - P_{\mathrm{L}}^{D}\right)\left(1 - P_{\mathrm{L}}^{U}\right), \tag{2}$$

where P_{L}^{D} and P_{L}^{U} are the packet loss probabilities for the request and the response, respectively. By assuming that the downward and upward links suffer from the same failure probability P_{L}, which is very likely to happen if, as in our setup, motes are based on the same radio blocks, this can be rewritten as

$$P_{\mathrm{L}}^{T} = 1 - (1 - P_{\mathrm{L}})^2 = 2P_{\mathrm{L}} - P_{\mathrm{L}}^2, \tag{3}$$

and, under our simplified channel error model,

$$P_{\mathrm{L}}^{T} = 1 - \left(1 - \epsilon^{r_{\mathrm{L}}+1}\right)^2 = 2\epsilon^{r_{\mathrm{L}}+1} - \epsilon^{2(r_{\mathrm{L}}+1)}. \tag{4}$$

Using for ϵ the same values above (20% and 50%) clearly leads to very low values for the two-way loss ratio as well, which implies that finding a good match between the theoretical model and experimental samples starting from this quantity is hardly possible. For instance, even in the case when $\epsilon = 0.5$ (which denotes a quite severe interference), only one packet is lost, on average, every 11 days, if the `ping` period is set to 30 s. Setting smaller periods is not recommended, because we want to prevent packets from remaining queued in motes' transmission buffers.

3.3. Transmission Latency

Our simple model can be reliably checked against experimental data by considering latencies in the place of losses. In particular, let d_i be the round-trip delay related to the i-th `ping` request, $i \in [1...N_{\text{sam}}]$, where N_{sam} is the number of samples taken in the experiment. In the following, it is denoted for simplicity *two-way transmission latency*. For a single-hop link, d_i is made up of three contributions:

1. The *waiting time* $d_{\text{wait},i}$, elapsing from the instant the request packet is queued at the originating mote and the beginning of the downward slot to the target mote. In the absence of queuing phenomena, the packet is sent at the earliest opportunity, which implies that $0 \le d_{\text{wait},i} < T_{\text{slfr}}$. It is worth noting that there is no such contribution for the reply packet, since operations of the ICMP responder process in the target mote are directly triggered by the request conveyed in the downward slot. All it is needed is that the responder has enough time to serve the request and enqueue its reply before the beginning of the upward slot.
2. The two-way network *communication time* $d_{\text{comm},i}$ in the case there are no failed attempts (best case), measured from the beginning of the downward slot (to the recipient mote), where the request packet is sent, up to the end of the upward slot (back to the originator), where the reply packet is returned. This time depends on slotframe configuration, which is managed automatically by motes using the 6TiSCH Operation Sublayer Protocol (6P). Since the downward and upward slots are assigned fixed positions in the slotframe (until, following a topological change in the network, a new configuration is defined), the time between them is fixed as well, and will be denoted d_{comm}.
3. The *retransmission time* $d_{\text{retr},i}$ spent for backoffs (when an attempt is deferred because the CCA function has sensed the channel busy) or retries (when an attempt fails due to a collision or noise pulses corrupting the frame). Let r_i^D and r_i^U denote the numbers of retransmissions carried out in the downward and upward direction, respectively, while $r_i^T = r_i^D + r_i^U$ is the total number of retries, both directions considered. All these quantities depend on the amount of interference and disturbance on air. If dedicated slots are considered, as happens when network formation has settled, every time packet transmission is tried again, in either direction, the latency increases by exactly one slotframe. On the whole, the retransmission time only depends on r_i^T and is given by $d_{\text{retr},i} = r_i^T \cdot T_{\text{slfr}}$, where $r_i^T \in [0...2r_{\text{L}}]$.

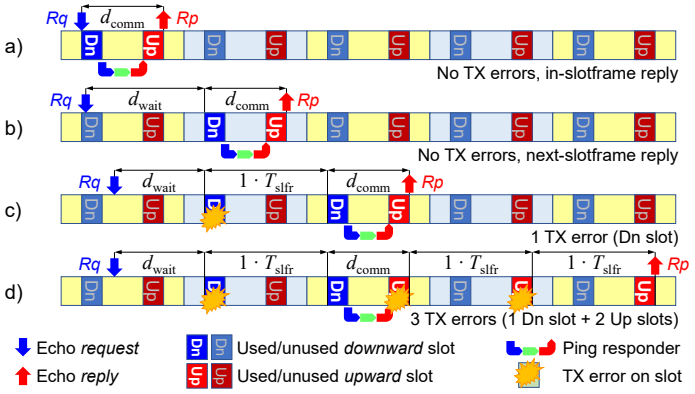


Figure 5: Single-hop request-response transaction in TSCH.

Overall, the two-way transmission latency of the i-th `ping` request can be expressed as

$$d_i = d_{\text{wait},i} + d_{\text{comm}} + r_i^T \cdot T_{\text{slfr}}. \tag{5}$$

A time diagram describing how the above contributions may affect request-response transactions is shown in Fig. 5. Diagrams a) and b) refer to the case where no transmission errors are experienced (and hence, no retries are performed). In particular, a) is related to the best case, where there is no waiting time, while in b) the request is carried out just after the relevant slot has begun, so that the originator has to wait for the next opportunity. Diagrams c) and d) refer instead to the case when some transmission errors occur that corrupt one or more frames, this meaning that some retries are performed. In c) only the downward link is affected (once), while in d) also the upward link suffers (twice) from errors.

3.4. Number of Retransmissions

The numbers of retries carried out for a packet on the downward and upward directions can be modeled as random variables, denoted R^D and R^U, respectively. As said above, it can be reasonably assumed that both directions are affected by the same amount and kind of interference. Therefore, R^D and R^U can be considered as independent and identically distributed (i.i.d.) random variables.

For each single direction $x \in \{D, U\}$ let $P_r^x \doteq \mathbb{P}(R^x = r)$ be the probability that a correctly delivered packet underwent exactly r retries (besides the initial attempt). Under our hypotheses about the channel error model, P_r^x can be evaluated as the probability to incur in r failures multiplied by the probability to succeed at the $r + 1$ attempt. Since latency is only defined for packets whose transmission was successful, the conditional probability given that the packet eventually arrived at destination has to be considered. This yields

$$P_r^x = \frac{1}{1 - P_{\text{L}}} \epsilon^r (1 - \epsilon) = \frac{1 - \epsilon}{1 - \epsilon^{r_{\text{L}}+1}} \epsilon^r. \tag{6}$$

Since P_r^x does not depend on the direction x, in the following it will be simply denoted P_r.

When both the downward and upward directions in a two-way request-response transaction are taken into account, the

probability $P_r^T \doteq \mathbb{P}(R^D + R^U = r)$ to incur, on the whole, in exactly r retries is, for $0 \le r \le r_\mathrm{L}$,

$$P_r^T = \sum_{k=0\ldots r} P_k \cdot P_{r-k} = \sum_{k=0\ldots r} \frac{(1-\epsilon)\epsilon^k}{1-\epsilon^{r_\mathrm{L}+1}} \cdot \frac{(1-\epsilon)\epsilon^{r-k}}{1-\epsilon^{r_\mathrm{L}+1}} = \\ = \left(\frac{1-\epsilon}{1-\epsilon^{r_\mathrm{L}+1}}\right)^2 (1+r)\,\epsilon^r, \tag{7}$$

where k and $r-k$ are the numbers of retries in the downward and upward directions, respectively, whereas for $r_\mathrm{L} < r \le 2r_\mathrm{L}$

$$P_r^T = \sum_{k=r-r_\mathrm{L}\ldots r_\mathrm{L}} P_k \cdot P_{r-k} = \sum_{k=r-r_\mathrm{L}\ldots r_\mathrm{L}} \frac{(1-\epsilon)\epsilon^k}{1-\epsilon^{r_\mathrm{L}+1}} \cdot \frac{(1-\epsilon)\epsilon^{r-k}}{1-\epsilon^{r_\mathrm{L}+1}} = \\ = \left(\frac{1-\epsilon}{1-\epsilon^{r_\mathrm{L}+1}}\right)^2 (1+2r_\mathrm{L}-r)\,\epsilon^r. \tag{8}$$

In fact, all cases have to be considered where the sum of the numbers of retries in both directions equals r, but no more than r_L retries are performed in any single direction. Overall

$$P_r^T = \left(\frac{1-\epsilon}{1-\epsilon^{r_\mathrm{L}+1}}\right)^2 (1+\min(r, 2r_\mathrm{L}-r))\,\epsilon^r. \tag{9}$$

3.5. Modeling the Transmission Latency

The two-way transmission latency of `ping` requests can be modeled as a random variable D, and the same holds for the waiting time D_wait and the time spent in the two directions for retransmissions, denoted D_retr^D and D_retr^U, respectively. It is worth remembering that d_comm is constant. As a consequence

$$D = D_\mathrm{wait} + D_\mathrm{retr}^D + d_\mathrm{comm} + D_\mathrm{retr}^U. \tag{10}$$

Since packet generation and slotframe transmission are asynchronous cyclic processes, whose periods are prime (30 s and 2.02 s, respectively), we can safely consider D_wait as uniformly distributed between 0 and T_slfr. This means that its probability density function (pdf) is

$$f_{D_\mathrm{wait}}(d) = \frac{1}{T_\mathrm{slfr}} \left(u(d) - u(d - T_\mathrm{slfr})\right), \tag{11}$$

where $u(\cdot)$ is the Heaviside unit step function. It is easy to see that the expected value is equal to half the slotframe duration

$$\mathbb{E}[D_\mathrm{wait}] = \frac{T_\mathrm{slfr}}{2}. \tag{12}$$

The overall time D_retr^T taken by retransmissions, both directions considered, corresponds to the sum of the related contributions (which are i.i.d.), that is, $D_\mathrm{retr}^T = D_\mathrm{retr}^D + D_\mathrm{retr}^U$. Starting from P_r^T values, the pdf of D_retr^T can be found as

$$f_{D_\mathrm{retr}^T}(d) = \sum_{r=0\ldots 2r_\mathrm{L}} P_r^T \cdot \delta(d - r \cdot T_\mathrm{slfr}), \tag{13}$$

where $\delta(\cdot)$ is the Dirac delta function. For any single direction $x \in \{D, U\}$, the expected retransmission latency can be simply expressed as

$$\mathbb{E}[D_\mathrm{retr}^x] = \mathbb{E}[R^x] \cdot T_\mathrm{slfr}, \tag{14}$$

where

$$\mathbb{E}[R^x] = \sum_{r=0\ldots r_\mathrm{L}} r \cdot P_r = \frac{1-\epsilon}{1-\epsilon^{r_\mathrm{L}+1}} \sum_{r=0\ldots r_\mathrm{L}} r \cdot \epsilon^r = \\ = r_\mathrm{L} + \frac{1}{1-\epsilon} - \frac{r_\mathrm{L}+1}{1-\epsilon^{r_\mathrm{L}+1}}, \tag{15}$$

for the known properties of the truncated geometric series. Since $\mathbb{E}[R^x]$ does not depend on the direction x, we will use for it the simplified notation $\mathbb{E}[R]$.

Under our hypotheses, the four contributions that make up the latency D, namely, d_comm, D_wait, D_retr^D, and D_retr^U, can be considered statistically independent. From the above pdfs and (10), by following an approach similar to that described in [56], it is then possible to evaluate the pdf of latency D as

$$f_D(d) = (f_{D_\mathrm{wait}} * f_{D_\mathrm{retr}^T})(d - d_\mathrm{comm}), \tag{16}$$

where operator $*$ denotes convolution. This is a piecewise constant function, where the first and last intervals have infinite width and zero height. Starting at $d = d_\mathrm{comm}$, there are $2r_\mathrm{L}+1$ inner intervals with width T_slfr and height P_r^T/T_slfr.

Concerning the cumulative distribution function (CDF) of the overall two-way latency D, denoted $F_D(\cdot)$, it consists of a continuous piecewise linear function with $2r_\mathrm{L}+2$ knots (see Figs. 6 and 7), where knot r, $r \in [0\ldots 2r_\mathrm{L}+1]$, is located at coordinates $\langle d_\mathrm{comm} + r \cdot T_\mathrm{slfr}, \sum_{i=0\ldots r-1} P_r^T \rangle$.

Finally, the expected value of the latency can be found as the sum of the expected values of the single contributions

$$\mathbb{E}[D] = d_\mathrm{comm} + T_\mathrm{slfr} \cdot \left(\frac{1}{2} + 2 \cdot \mathbb{E}[R]\right). \tag{17}$$

3.6. Channel hopping

In the above analysis, the failure probability ϵ for attempts is assumed not to vary. While this could be acceptable when operating on a single channel, as in legacy IEEE 802.15.4 WSNs, certainly it is not when the channel hopping mechanism of TSCH is enabled, which keeps the transmission frequency changing. By assuming that the number of physical channels and the number of slots in the slotframe are prime (as happens in practice in our 6TiSCH setup, since $N_\mathrm{ch} = 16$ and $N_\mathrm{slot} = 101$), the actual frequency on which subsequent attempts are performed by TSCH hops among all channels according to a pseudo-random sequence. At best, we can reasonably assume that the failure probability ϵ_C of every channel C remains constant over time.

In the following, we will model channel hopping as a truly random process, where all channels are equally likely to be selected. This is possible only provided that channel black-listing techniques are not in use [57, 58]. Even though such techniques could be possibly implemented by modifying the *macHoppingSequenceList* table, maintaining coherence among nodes is not easy. For this reason, a standard black-listing mechanism was not included in the TSCH specification. Under the above hypothesis, the probability P_C for any given channel C to be used is the same and corresponds to $1/N_\mathrm{ch}$. As we will see, doing so provides a reasonable approximation. This

means that each transmission attempt is modeled as a two-step random trial: first, a channel is randomly selected, which is characterized by a specific failure probability; then, the transmission attempt is performed in such conditions. Since the two steps are statistically independent, the attempt can be modeled, on the whole, as a Bernoulli trial where the *equivalent failure probability* $\tilde{\epsilon}$ is equal to

$$\tilde{\epsilon} = \sum_{C=1\ldots N_{\text{ch}}} P_C \cdot \epsilon_C = \frac{1}{N_{\text{ch}}} \sum_{C=1\ldots N_{\text{ch}}} \epsilon_C = \overline{\epsilon}. \tag{18}$$

The same analysis described above can then be applied.

4. Experimental Evaluation

A thorough experimental campaign was carried out to assess the quality of communication achieved by a WSN based on 6TiSCH in the presence of unpredictable external interference. In fact, collisions due to intra-network traffic are prevented thanks to time slotting. In particular, we evaluated the effectiveness of channel hopping to face Wi-Fi traffic.

4.1. Experimental Testbed

A network setup was deployed based on real devices. Since in this paper the case of star network topologies with single-hop links was explicitly considered, two devices are enough to obtain the experimental data relevant to our analysis: the root mote (connected to a PC), which originates requests, and a target mote, which replies with responses.

We used OpenMote B [59] devices running the OpenWSN [60] operating system (version REL-1.24.0). OpenMote B devices are relatively new, and appeared on the market in 2018. They are based on the TI CC2538 System-On-Chip microcontroller, which integrates an IEEE 802.15.4 radio transceiver for transmission in the 2.4 GHz band and an ARM® Cortex™ M3 CPU with 512 KB of flash memory and 32 KB of dynamic RAM memory. In addition, a specific Atmel AT86RF215 chip, not used in our experimental campaigns, is available for sub-GHz transmission (868/915 MHz). Besides OpenWSN, these devices are also compatible with the Contiki operating system.

OpenWSN operating system implementation is aligned to the most recent definitions of the 6TiSCH protocol stack. Its code is freely available: this means that, on the one hand, it can be inspected to check how some features of 6TiSCH are actually implemented on real devices, while on the other hand some modifications can be easily brought to its operations. For instance, in this paper the channel hopping mechanism was disabled in some experiments. This was obtained by setting the variable `ieee154e_vars.singleChannel`, defined in the file `openstack/02a-MAClow/IEEE802154E.c` of the OpenWSN source code, to select a specific, fixed transmission channel. The operating system, and possibly the applications that in OpenWSN are directly linked with the related executable, can be rebuilt by means of a cross-compilation process performed on the PC. In this work, cross-compiling was carried out using the ARM `gcc` toolchain. The produced code is then downloaded to the mote through its USB interface by means of the OpenVisualizer management software, which is provided along with OpenWSN.

4.2. Interfering Traffic

The amount of Wi-Fi traffic in the 2.4 GHz band in our laboratory was not under our control, because of the presence of many APs, notebooks, and mobiles located in the nearby premises. What is worse, the traffic pattern was ever changing, which makes it difficult to reliably compare results obtained in different experiments. To evaluate TSCH performance in different operating conditions, some PCs equipped with Wi-Fi network adapters were used to inject specific amounts of additional interfering traffic on the same frequency range as the WSN. The traffic generated by every Wi-Fi adapter follows a random generation pattern, as detailed in [61], which is managed by a finite state machine consisting of two states: *idle* and *burst*. In the idle state, the interfering node remains inactive for a random time whose duration is modeled according to a truncated exponential distribution. The average and maximum duration of the gap are 280 ms and 20 s, respectively. After the gap, the state machine enters the burst state, in which a burst of packets of size 1500 B is generated with a period of 400 μs. The number of packets generated within each burst is also selected randomly, and follows an exponential distribution with mean 225 packets and truncated so that the maximum number of packets in any burst is 1125. After generating the burst of packets, the state machine returns back to the idle state.

For simplicity, the same configuration was selected for all the interfering sources. Doing so allows any given condition about interference to be identified by specifying (as a subscript) the list of Wi-Fi channels on which the sources were actively transmitting. For instance, $\mathcal{I}_A$ denotes the case where only one interferer was active, tuned on channel A, while $\mathcal{I}_{A,B,B}$ means that two interferers were additionally switched on, both tuned on channel B. Notation $\mathcal{I}_\emptyset$ refers to the case with no interferers, when only the (unknown and variable) background Wi-Fi traffic affects the WSN. No more than two interferers were tuned on the same channel, to prevent as much as possible collisions between them. Concerning interference on a single channel A, what we can reasonably say is that the overall load is strictly increasing for cases $\mathcal{I}_\emptyset$, $\mathcal{I}_A$, and $\mathcal{I}_{A,A}$.

4.3. Measurement Technique

To analyze TSCH behavior when used to support higher-level protocols based on the request-response paradigm, like CoAP, we relied on the conventional `ping` utility. As said before, we wish to investigate those cases where short reaction times are sought, e.g., less than a dozen seconds, as opposed to real-world applications based on WSNs, where the period with which motes are probed is in the order of several minutes or longer. A further reduction of timings does not reflect the typical operating conditions of WSNs, since low power consumption is always given precedence, also in those cases where good reactivity is demanded. To prevent the communication buffers of motes from filling up in the case of prolonged interference, the `ping` timeout was set to 30 s. This means that 120 samples

per hour can be collected in our setup. Since we wished to have (at least) two thousands samples per dataset, each experiment lasted for one whole day.

In every experiment, statistics about the success/failure of every `ping` request were collected and logged in a file. For successful requests, the round-trip time was captured as well, which coincides with d_i values. From logs, the number N_{L} of failed requests was subsequently computed. This permits to evaluate the empirical two-way loss probability $\hat{P}_{\mathrm{L}}^{T}$, defined as the measured fraction of requests for which no response is ever received during the experiment, $\hat{P}_{\mathrm{L}}^{T} = N_{\mathrm{L}}/N_{\mathrm{sam}}$.

4.4. Matching experimental parameters

To check the theoretical model introduced in Section 3 against the behavior of the real testbed, parameters d_{comm} and ϵ have to be inferred from the samples acquired in the experiments. In realistic operating conditions (i.e., when the amount of interference and disturbance is tolerable) and provided that the number N_{sam} of samples is adequate (e.g., some thousands), a reliable estimate of d_{comm} can be satisfactorily evaluated as the minimum among all the measured latencies

$$\hat{d}_{\mathrm{comm}} = d_{\mathrm{min}} \doteq \min_{i=1\ldots N_{\mathrm{sam}}} (d_i). \tag{19}$$

In fact, in the best case both the request and the response packets do not suffer from any transmissions errors and no initial waiting time (or a negligible amount of it) is experienced by the originator of the request. This corresponds to setting $r_i^T = 0$ and $d_{\mathrm{wait},i} \approx 0$ in (5), which leads to (19).

Several fitting techniques can be employed to find ϵ. In the following, two simple yet quite effective ones are described.

4.4.1. Failure rate from latency distribution

Let N_r be the number of `ping` requests that succeeded after exactly r retries (both directions considered). By remembering that the waiting time $d_{\mathrm{wait},i}$ is necessarily shorter than the slotframe duration, $\forall i \in [1\ldots N_{\mathrm{sam}}]$, N_r can be easily determined by counting the number of samples for which $\hat{d}_{\mathrm{comm}} + r \cdot T_{\mathrm{slfr}} \leq d_i < \hat{d}_{\mathrm{comm}} + (r+1) \cdot T_{\mathrm{slfr}}$. In turn, this permits evaluating the empirical probability $\hat{P}_r^T$, which corresponds to the statistical relative frequency of experiencing exactly r retries on the two-way path between the root and the target mote, as $\hat{P}_r^T = N_r/(N_{\mathrm{sam}} - N_{\mathrm{L}})$.

As will be seen, in all experiments the failure rate ϵ was not excessively high, so that a good part of the `ping` requests did not suffer from any frame losses in either direction. This means that $\hat{P}_0^T$ provides a reliable estimate of P_0^T. By recalling from (7) that

$$P_0^T = \frac{(1-\epsilon)^2}{1 - P_{\mathrm{L}}^T}, \tag{20}$$

then a reliable estimate of the attempt failure probability is

$$\hat{\epsilon}_P = 1 - \sqrt{\hat{P}_0^T (1 - \hat{P}_{\mathrm{L}}^T)}. \tag{21}$$

If the number of failed `ping` requests is so low that P_{L}^T cannot be reliably determined from $\hat{P}_{\mathrm{L}}^T$ (in the experiments we carried out, no requests actually failed), then an adequate estimate of ϵ can be found by numerically inverting the equation

$$\hat{\epsilon}_P = 1 - \sqrt{\hat{P}_0^T \left(1 - \hat{\epsilon}_P^{r_{\mathrm{L}}+1}\right)} \tag{22}$$

obtained from (20) and (4).

4.4.2. Failure rate from average latency

The failure rate ϵ can be also derived from the sample mean of the latency, evaluated as

$$\mu_d = \frac{1}{N_{\mathrm{sam}}} \sum_{i=1\ldots N_{\mathrm{sam}}} d_i. \tag{23}$$

By assuming that μ_d provides a good estimation of the expected value of the latency $\mathbb{E}[D]$, a second reliable estimate of ϵ can be found by inverting (17). In particular, given the linearity of the expected value, we can estimate the expected number $\mathbb{E}[R]$ of retries in any direction as

$$\hat{\mu}_r = \frac{1}{2}\left(\frac{\mu_d - d_{\mathrm{comm}}}{T_{\mathrm{slfr}}} - \frac{1}{2}\right) \tag{24}$$

and, by inverting (15),

$$\hat{\epsilon}_D = 1 - \frac{1}{\hat{\mu}_r - r_{\mathrm{L}} + \frac{r_{\mathrm{L}}+1}{1-\hat{\epsilon}_D^{r_{\mathrm{L}}+1}}} = 1 - \frac{1}{\hat{\mu}_r + \frac{1+r_{\mathrm{L}}\cdot\hat{\epsilon}_D^{r_{\mathrm{L}}+1}}{1-\hat{\epsilon}_D^{r_{\mathrm{L}}+1}}}, \tag{25}$$

which can be easily solved numerically.

5. Results

In this section, the results obtained in two experimental campaigns, with and without channel hopping, are described and checked against our theoretical model. For each campaign, several experiments were performed by varying interference conditions. Every single experiment lasted exactly 24 hours, in order to acquire enough samples. Since `ping` is invoked every 30 s, $N_{\mathrm{sam}} = 2880$ samples are available per experiment.

In Table 2, statistics collected from the experiments are reported. In the letfmost part of the table, the number N_{L} of failed `ping` requests and the number N_0 of requests that did not experience any retries are shown, also in relative terms ($\hat{P}_{\mathrm{L}}^T$ and $\hat{P}_0^T$). They are followed by the minimum (d_{min}), mean (μ_d), and maximum (d_{max}) values of the measured two-way latency. Then, the estimated values $\hat{\epsilon}_P$ and $\hat{\epsilon}_D$ of the failure rate, obtained from $\hat{P}_0^T$ and μ_d, respectively, are reported. In addition, the estimate $\hat{\mu}_r$ of the average number of retransmissions, also derived from μ_d, is included. Finally, on the rightmost side of the table, the estimates for P_L^T (i.e., $P_{\mathrm{L},P}^T$ and $P_{\mathrm{L},D}^T$), obtained from the model as described in Section 3 using the above failure rates, are shown.

The upper part of the table refers to experiments where channel hopping was disabled, whereas the lower part refers to those were it was left enabled. As can be seen, the values of d_{min} differ in the two cases (approximately 460 ms and 1940 ms). This depends on the fact that the network had to be restarted

Table 2: Experimental results and estimated model parameters – Channel hopping disabled and enabled

Channel hopping disabled (also see plots in Fig. 6)

		Measured counters / ratios				Measured latencies (ms)			Estimated failure rate			Computed two-way loss ratio	
Exp.		N_L	N_0	$\hat{P}_L^T$	$\hat{P}_0^T$	d_{min}	μ_d	d_{max}	$\hat{\epsilon}_P$	$\hat{\mu}_r$	$\hat{\epsilon}_D$	$P_{L,P}^T$	$P_{L,D}^T$
$\mathcal{I}_{\emptyset}^{(1)}$		0	2286	0.0	0.794	466	1966.00	10723	0.109	0.121	0.108	8.03×10^{-16}	7.02×10^{-16}
$\mathcal{I}_{\emptyset}^{(2)}$		0	2189	0.0	0.760	464	2059.09	11587	0.128	0.145	0.127	1.06×10^{-14}	8.60×10^{-15}
$\mathcal{I}_{6}^{(1)}$		0	1901	0.0	0.660	460	2373.00	11872	0.188	0.224	0.183	4.69×10^{-12}	3.08×10^{-12}
$\mathcal{I}_{6}^{(2)}$		0	1682	0.0	0.584	464	2723.74	14756	0.236	0.309	0.236	1.82×10^{-10}	1.88×10^{-10}
$\mathcal{I}_{6,6}^{(1)}$		0	1092	0.0	0.379	461	3909.81	19755	0.384	0.604	0.376	4.51×10^{-07}	3.25×10^{-07}
$\mathcal{I}_{6,6}^{(2)}$		0	1318	0.0	0.458	466	3399.57	18553	0.324	0.476	0.323	2.88×10^{-08}	2.75×10^{-08}
$\mathcal{I}_{\emptyset}^{(+)}$	•	0	4475	0.0	0.777	464	2012.55	11587	0.119	0.133	0.118	3.05×10^{-15}	2.69×10^{-15}
$\mathcal{I}_{6}^{(+)}$	•	0	3583	0.0	0.622	460	2548.37	14756	0.211	0.267	0.211	3.16×10^{-11}	3.01×10^{-11}
$\mathcal{I}_{6,6}^{(+)}$	•	0	2410	0.0	0.418	461	3654.69	19755	0.353	0.541	0.351	1.17×10^{-07}	1.06×10^{-07}

Channel hopping enabled (also see plots in Fig. 7)

Exp.		N_L	N_0	$\hat{P}_L^T$	$\hat{P}_0^T$	d_{min}	μ_d	d_{max}	$\hat{\epsilon}_P$	$\hat{\mu}_r$	$\hat{\epsilon}_D$	$P_{L,P}^T$	$P_{L,D}^T$
$\mathcal{I}_{\emptyset}^{(1)}$	•	0	2465	0.0	0.856	1937	3278.97	9197	0.075	0.082	0.076	1.94×10^{-18}	2.44×10^{-18}
$\mathcal{I}_{1}^{(1)}$		0	2133	0.0	0.741	1945	3613.18	14003	0.139	0.163	0.140	4.07×10^{-14}	4.40×10^{-14}
$\mathcal{I}_{1}^{(2)}$	•	0	2320	0.0	0.806	1943	3409.05	11278	0.102	0.113	0.101	2.96×10^{-16}	2.51×10^{-16}
$\mathcal{I}_{5}^{(1)}$		0	2481	0.0	0.861	1941	3263.55	10667	0.072	0.077	0.072	1.01×10^{-18}	1.00×10^{-18}
$\mathcal{I}_{1,1}^{(1)}$		0	2109	0.0	0.732	1940	3621.55	12681	0.144	0.166	0.143	7.04×10^{-14}	5.80×10^{-14}
$\mathcal{I}_{1,5}^{(1)}$		0	1926	0.0	0.669	1940	3859.07	12332	0.182	0.225	0.184	2.96×10^{-12}	3.36×10^{-12}
$\mathcal{I}_{1,5}^{(2)}$	•	0	2149	0.0	0.746	1938	3575.46	11289	0.136	0.155	0.134	2.80×10^{-14}	2.28×10^{-14}
$\mathcal{I}_{1,5,9}^{(1)}$		0	1524	0.0	0.529	1940	4438.65	16942	0.273	0.368	0.269	1.86×10^{-09}	1.53×10^{-09}
$\mathcal{I}_{1,5,9}^{(2)}$	•	0	1848	0.0	0.642	1944	3944.73	12761	0.199	0.245	0.197	1.21×10^{-11}	1.02×10^{-11}
$\mathcal{I}_{1,5,13}^{(1)}$		0	1952	0.0	0.678	1941	3810.58	15999	0.177	0.213	0.175	1.81×10^{-12}	1.61×10^{-12}
$\mathcal{I}_{1,5,9,13}^{(1)}$	•	0	1659	0.0	0.576	1942	4277.65	17288	0.241	0.328	0.247	2.59×10^{-10}	3.85×10^{-10}
$\mathcal{I}_{1,5,9,13}^{(2)}$		0	1768	0.0	0.614	1943	4076.80	13649	0.216	0.278	0.218	4.66×10^{-11}	5.06×10^{-11}
$\mathcal{I}_{1,1,5,5}^{(1)}$		0	1638	0.0	0.569	1945	4316.97	16035	0.246	0.337	0.252	3.56×10^{-10}	5.33×10^{-10}

between the two sets of experiments (enabling/disabling channel hopping requires the code of motes to be modified and rebuilt). The 6TiSCH Experimental Scheduling Function (SFX) foresees that 6P cell negotiation is performed starting from a randomly selected set of cells. This means that a different matrix is obtained on every network startup, which implies a different relative position of the downstream and upstream slots in the slotframe. Additional experiments, not reported here, confirm this behavior.

5.1. Channel Hopping Disabled

The first experimental campaign was devoted to analyze the behavior of the TSCH protocol when channel hopping is not in use. When doing so, intra-network interference among motes belonging to the WSN is still prevented, thanks to the time slotting mechanism. However, the ability to face narrowband disturbance and interference from nearby Wi-Fi network infrastructures is lost, which means that communication quality may be affected negatively.

The transmission frequency of motes was tuned on WSN channel 17, while interferers were tuned on Wi-Fi channel 6 that, according to Fig. 4, overlaps with the WSN. Three experimental conditions were considered, denoted $\mathcal{I}_{\emptyset}$, $\mathcal{I}_6$, and $\mathcal{I}_{6,6}$, where zero, one, or two interferers were selected, respectively. For every interference condition, two experiments were performed, to provide some hints on the intrinsic variability of the spectrum conditions due to fluctuations of the background traffic. A numeric superscript (in parentheses) is used to distinguish among experiments that refer to the same nominal injected interference, e.g., $\mathcal{I}_6^{(1)}$ and $\mathcal{I}_6^{(2)}$. Experiments identified

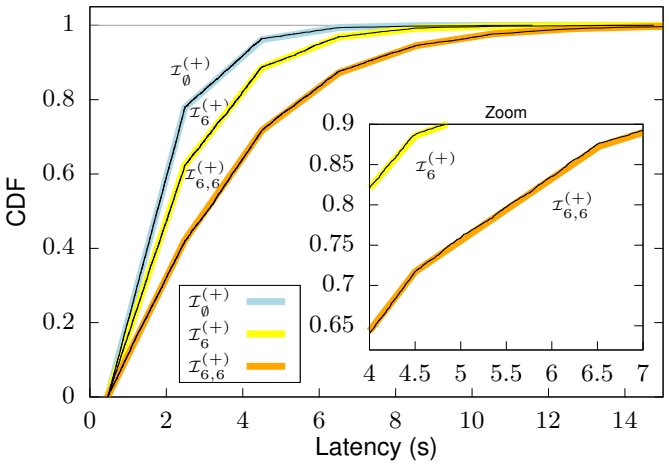


Figure 6: Measured and theoretical CDFs of d (channel hopping disabled).

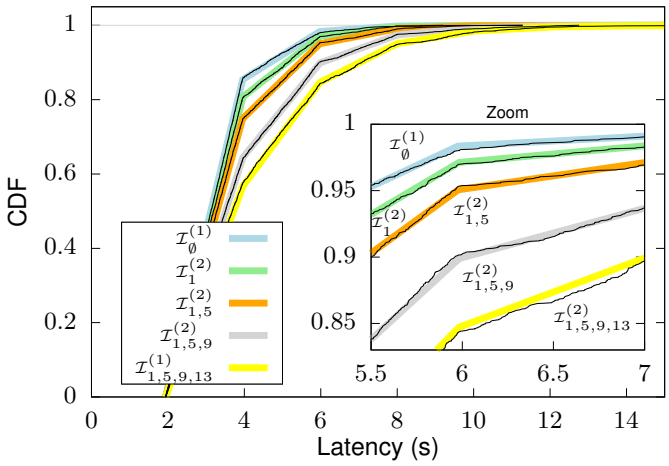


Figure 7: Measured and theoretical CDFs of d (channel hopping enabled).

with the superscript (+), i.e., $\mathcal{I}_{\emptyset}^{(+)}$, $\mathcal{I}_{6}^{(+)}$, and $\mathcal{I}_{6,6}^{(+)}$, correspond to aggregate datasets, obtained by merging all the samples acquired in the experiments performed with the same interference conditions. Only for them the number of samples is higher, $N_{\text{sam}} = 5760$.

Concerning the estimates $\hat{\epsilon}_P$ and $\hat{\epsilon}_D$ for the failure rate provided by the two proposed methods, they are actually very similar. Likely, relying on the measured mean latency provides a more reliable estimate, because the contribution of all samples is considered. Conversely, the other method models in a more faithful manner those samples that experienced less retries (and hence, lower delays). As expected, increasing the amount of injected traffic generally leads to a higher failure probability. However, a noticeable variability can be observed for the failure probability in the same conditions of injected interference. For instance, in the $\mathcal{I}_{6,6}$ case, two quite different values were obtained when estimating ϵ, that is, 32.4% and 38.4% (derived from $\hat{P}_0^T$).

In Fig. 6, the measured cumulative frequency distributions of latencies are shown using thin black lines. The datasets employed to obtain the plots have been highlighted in Table 2 with the symbol • (solid circle). In particular, we relied on the merged datasets $\mathcal{I}_{\emptyset}^{(+)}$, $\mathcal{I}_{6}^{(+)}$, and $\mathcal{I}_{6,6}^{(+)}$ to try reducing variability of the background traffic. Also depicted, using thick colored lines, are the CDFs derived from our theoretical model. As can be seen, there is a good match between measured and calculated values, which implies that the hypotheses we made on the channel error model are not particularly restrictive. It is worth noting that experimental and theoretical data also match in the tail portion of the curves, located on the rightmost side. This means that the approximation provided by the model is acceptable for a wide range of latency values.

5.2. *Channel Hopping Enabled*

The second experimental campaign was carried out by leaving channel hopping on, to analyze TSCH performance in standard operating conditions. Up to four Wi-Fi interferers were activated, no more than two on the same channel. The same notation as before, based on Wi-Fi channels on which interference is injected, has been retained to identify experiments. Results, reported in the lower part of Table 2, do not differ drastically with respect to the previous case. Again, the two estimates $\hat{\epsilon}_P$ and $\hat{\epsilon}_D$ of the failure probability, obtained from $\hat{P}_0^T$ and μ_d, respectively, are very similar. This confirms that modeling attempts on air in TSCH as Bernoulli trials is a sensible choice.

Concerning improvements brought by channel hopping, it can be seen that, in this case, the failure rate experienced by attempts is generally lower than when transmissions are performed on a single, fixed channel that overlaps with the interfering traffic. Again, variability of the background traffic undermines repeatability of experiments, and makes a direct comparison of the results obtained at different times only partially meaningful. This appears very clearly by comparing side-by-side experiments carried out with the same amount and pattern of injected interfering traffic. For example, $\mathcal{I}_{1,5,9}^{(1)}$ and $\mathcal{I}_{1,5,9}^{(2)}$ led to failure rates ϵ equal to 27.3% and 19.9% (derived from $\hat{P}_0^T$), respectively. What can be inferred from results is that increasing the amount of injected interference generally leads to a worsening of ϵ, which grows from about 7.5% with no interferers up to 21.6%–24.6% in the case when four interferers are active (derived from $\hat{P}_0^T$). Interestingly, the equivalent failure rate achieved by channel hopping with four interfering nodes tuned on separate channels (hence, overlapping with the whole set of channels used by motes) is, on average, lower than the failure rate experienced when hopping is disabled and two identical interferers are tuned on the same frequency as motes.

Measured and theoretical CDFs obtained for five different interfering conditions (those marked with a solid circle in the lower portion of the table) when channel hopping is enabled are shown in Fig. 7. Again, curves highlight a good match between the analytical model and experimental results.

To double check results, a number of additional experiments were carried out by setting the retry limit to much lower values, i.e., $R = 1$, 3 (typical of TSCH), and 5. The estimates of $\hat{\epsilon}_P$ we found were in the range 9–14%. In spite of the spectrum variability, such values are comparable to results in the table. Moreover, when only a single retry is allowed (which means

that packet transmission is not unlikely to fail), the estimated and measured `ping` failure rates closely match ($P_L^T = 1.9\%$ and $\hat{P}_L^T = 1.7\%$, respectively).

5.3. Comments on Channel Hopping Effectiveness

By comparing results with and without channel hopping, reported in the lower and upper parts of Table 2, respectively, it can be seen that the quality of communication perceived by motes during the experiments is related to the mean amount of Wi-Fi traffic purposely injected on air that may affect WSN transmissions. For instance, the fraction of failed attempts with 4 Wi-Fi interferers tuned on distinct channels in the case the transmission frequency of motes is changed on every attempt somehow resembles the case when the frequency of motes is left fixed and only one Wi-Fi interferer is activated that transmits on an overlapping frequency range. In fact, under the assumption that the background Wi-Fi traffic on the different channels is almost the same and does not vary consistently among experiments (which is indeed a rough approximation), the overall interference seen by the transmission attempts performed by motes is about the same, irrespective of the channel their radio transceivers are tuned on at the time of transmission, which explains the above behavior.

This corroborates in some way the assumption that channel hopping makes communicating motes see an equivalent spectrum that averages the behavior of all the involved physical channels, as postulated by equation (18), hence reducing the variability of the communication quality. In other words, our experimental campaigns also confirmed, to some extent, the ability of TSCH to flatten narrowband interference.

6. Conclusions

A major cause of packet losses and delays in WSNs, when they are deployed in residential and industrial areas, as well as in campuses, hospitals, malls, etc., is the interference generated by co-located Wi-Fi network infrastructures, which nowadays can be found everywhere and whose diffusion is steadily increasing over time. In fact, the related traffic may impair communication in the WSN severely, either by causing repeated collisions or by delaying transmissions because of the carrier sensing mechanism. TSCH, and in particular the channel hopping mechanism, were purposely introduced with the aim of counteracting the effects of such unpredictable phenomena.

In this paper, the behaviour of a 6TiSCH network based on a star topology has been evaluated when the amount of interfering Wi-Fi traffic is varied. In particular, we considered the quality of communication (latencies and losses) provided to applications when motes are accessed according to a CoAP-like request-response paradigm. A very simple theoretical model has been developed to describe this kind of interactions, where attempts are assumed to be independent and subject to a time-invariant failure probability. Then, the model has been fitted to experimental data captured from a real WSN.

Results show that, thanks to the combined effects of time and frequency diversity, the above assumptions mostly hold in a real-world setup. This permits to determine the complete probability distribution of round-trip delays starting from a swift characterization of the wireless spectrum based on simple statistics obtained from the `ping` command. In addition, they confirmed the ability of TSCH to smoothen narrowband disturbance and interference, by offering motes an equivalent quality of communication that roughly averages what is seen on the different physical channels. This means reducing the chance that the quality of communication between motes drops below an acceptable threshold for prolonged periods of time due to bulk traffic occasionally sent on co-located Wi-Fi networks.

The proposed model is not meant to compare the behavior of TSCH against legacy WSN solutions. In fact, in many of the latter, subsequent attempts cannot be considered statistically independent (which is one of the reasons that led to the definition of solutions relying on time slotting and channel hopping). As part of our future work we plan to extend the above analysis from star to mesh networks. However, we expect transmission latencies over multiple hops to grow and become less predictable, making the TSCH approach less appealing for applications subject to specific timing constraints (e.g., to close control loops with slow dynamics).

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393–422. doi:https://doi.org/10.1016/S1389-1286(01)00302-4.

[2] X. Yu, P. Wu, W. Han, Z. Zhang, A survey on wireless sensor network infrastructure for agriculture, Computer Standards Interfaces 35 (1) (2013) 59–64. doi:https://doi.org/10.1016/j.csi.2012.05.001.

[3] M. Srbinovska, C. Gavrovski, V. Dimcev, A. Krkoleva, V. Borozan, Environmental parameters monitoring in precision agriculture using wireless sensor networks, Journal of Cleaner Production 88 (2015) 297–307. doi:https://doi.org/10.1016/j.jclepro.2014.04.036.

[4] Y. Liao, M. Mollineaux, R. Hsu, R. Bartlett, A. Singla, A. Raja, R. Bajwa, R. Rajagopal, SnowFort: An Open Source Wireless Sensor Network for Data Analytics in Infrastructure and Environmental Monitoring, IEEE Sensors Journal 14 (12) (2014) 4253–4263. doi:10.1109/JSEN.2014.2358253.

[5] B. Rashid, M. H. Rehmani, Applications of wireless sensor networks for urban areas: A survey, Journal of Network and Computer Applications 60 (2016) 192–219. doi:https://doi.org/10.1016/j.jnca.2015.09.008.

[6] M. Erdelj, M. Krl, E. Natalizio, Wireless Sensor Networks and Multi-UAV systems for natural disaster management, Computer Networks 124 (2017) 72–86. doi:https://doi.org/10.1016/j.comnet.2017.05.021.

[7] T. Miyazaki, K. Anazawa, Y. Maruyama, S. Kobayashi, T. Segawa, P. Li, Resilient Information Management for Information Sharing in Disaster-Affected Areas Lacking Internet Access, in: M. R. Palattella, S. Scanzio, S. Coleri Ergen (Eds.), Ad-Hoc, Mobile, and Wireless Networks, Springer International Publishing, Cham, 2019, pp. 3–17.

[8] F. Civerchia, S. Bocchino, C. Salvadori, E. Rossi, L. Maggiani, M. Petracca, Industrial Internet of Things monitoring solution for advanced predictive maintenance applications, Journal of Industrial Information Integration 7 (2017) 4–12. doi:https://doi.org/10.1016/j.jii.2017.02.003.

[9] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, M. Mukherjee, A survey on fault diagnosis in wireless sensor networks, IEEE Access 6 (2018) 11349–11364. doi:10.1109/ACCESS.2018.2794519.

[10] IEEE, IEEE Standard for Low-Rate Wireless Networks, IEEE Std 802.15.4-2015 (Rev. of IEEE Std 802.15.4-2011) (2016) 1–709doi:10.1109/IEEESTD.2016.7460875.

[11] Z. Shelby, K. Hartke, C. Bormann, The Constrained Application Protocol (CoAP), IETF RFC 7252 (2014) 1–111.

[12] R. Heynicke, D. Krush, C. Cammin, G. Scholl, B. Kaercher, J. Ritter, P. Gaggero, M. Rentschler, IO-Link Wireless enhanced factory automa-

tion communication for Industry 4.0 applications, Journal of Sensors and Sensor Systems 7 (1) (2018) 131–142. doi:10.5194/jsss-7-131-2018.

[13] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni, Evaluation of the IoT LoRaWAN Solution for Distributed Measurement Applications, IEEE Transactions on Instrumentation and Measurement 66 (12) (2017) 3340–3349. doi:10.1109/TIM.2017.2746378.

[14] 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) (2012) 1–225doi:10.1109/IEEESTD.2012.6185525.

[15] D. D. Guglielmo, S. Brienza, G. Anastasi, IEEE 802.15.4e: A survey, Computer Communications 88 (2016) 1–24. doi:https://doi.org/10.1016/j.comcom.2016.05.004.

[16] IEC, Industrial networks - Wireless communication network and communication profiles - WirelessHART^{TM}, International Standard IEC 62591:2016, International Electrotechnical Commission (2016).
URL `https://webstore.iec.ch/publication/24433`

[17] P. Thubert, An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4, draft-ietf-6tisch-architecture-28 (2019) 1–68.

[18] Q. Wang, K. Jaffrs-Runser, Y. Xu, J. Scharbarg, Z. An, C. Fraboul, TDMA Versus CSMA/CA for Wireless Multihop Communications: A Stochastic Worst-Case Delay Analysis, IEEE Transactions on Industrial Informatics 13 (2) (2017) 877–887. doi:10.1109/TII.2016.2620121.

[19] M. Mongelli, S. Scanzio, A neural approach to synchronization in wireless networks with heterogeneous sources of noise, Ad Hoc Networks 49 (2016) 1–16. doi:https://doi.org/10.1016/j.adhoc.2016.06.002.

[20] G. Anastasi, M. Conti, M. D. Francesco, A. Passarella, Energy conservation in wireless sensor networks: A survey, Ad Hoc Networks 7 (3) (2009) 537–568. doi:https://doi.org/10.1016/j.adhoc.2008.06.003.

[21] M. R. Palattella, P. Thubert, X. Vilajosana, T. Watteyne, Q. Wang, T. Engel, 6TiSCH Wireless Industrial Networks: Determinism Meets IPv6, Springer International Publishing, Cham, 2014, pp. 111–141. doi:10.1007/978-3-319-04223-7_5.

[22] A. D. Wood, J. A. Stankovic, G. Zhou, DEEJAM: Defeating Energy-Efficient Jamming in IEEE 802.15.4-based Wireless Networks, in: 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2007), 2007, pp. 60–69. doi:10.1109/SAHCN.2007.4292818.

[23] G. Cena, S. Scanzio, A. Valenzano, Improving Effectiveness of Seamless Redundancy in Real Industrial Wi-Fi Networks, IEEE Transactions on Industrial Informatics 14 (5) (2018) 2095–2107. doi:10.1109/TII.2017.2759788.

[24] T. Watteyne, A. Mehta, K. Pister, Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense, in: Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN '09, ACM, New York, NY, USA, 2009, pp. 116–123. doi:10.1145/1641876.1641898.

[25] G. Cena, S. Scanzio, L. Seno, A. Valenzano, Comparison of Mixed Diversity Schemes to Enhance Reliability of Wireless Networks, in: M. R. Palattella, S. Scanzio, S. Coleri Ergen (Eds.), Ad-Hoc, Mobile, and Wireless Networks, Springer International Publishing, Cham, 2019, pp. 118–135.

[26] P. H. Gomes, T. Watteyne, B. Krishnamachari, MABO-TSCH: Multihop and blacklist-based optimized time synchronized channel hopping, Transactions on Emerging Telecommunications Technologies 29 (7) (2018) e3223, e3223 ett.3223. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.3223, doi:10.1002/ett.3223.

[27] IEEE Standard for Information technologyTelecommunications and information exchange between systems Local and metropolitan area networksSpecific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012) (2016) 1–3534doi:10.1109/IEEESTD.2016.7786995.

[28] R. Koutsiamanis, G. Z. Papadopoulos, X. Fafoutis, J. M. D. Fiore, P. Thubert, N. Montavont, From Best Effort to Deterministic Packet Delivery for Wireless Industrial IoT Networks, IEEE Transactions on Industrial Informatics 14 (10) (2018) 4468–4480. doi:10.1109/TII.2018.2856884.

[29] G. Cena, S. Scanzio, A. Valenzano, C. Zunino, Experimental Analysis and Comparison of Industrial IoT Devices based on TSCH, in: 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2019), 2019, pp. 184–191. doi:10.1109/ETFA.2019.8869410.

[30] ISO/IEC, Information technology - Home electronic system (HES) architecture - Part 3-11: Frequency modulated wireless short-packet (FMWSP) protocol optimised for energy harvesting - Architecture and lower layer protocols, International Standard ISO/IEC 14543-3-11:2016, International Organization for Standardization/International Electrotechnical Commission (2016).

[31] G. Lu and B. Krishnamachari and C. S. Raghavendra, An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks, in: International Parallel and Distributed Processing Symposium (IPDPS 2004), 2004, pp. 224–. doi:10.1109/IPDPS.2004.1303264.

[32] R. Piyare and A. L. Murphy and C. Kiraly and P. Tosato and D. Brunelli, Ultra Low Power Wake-Up Radios: A Hardware and Networking Survey, IEEE Communications Surveys Tutorials 19 (4) (2017) 2117–2157. doi:10.1109/COMST.2017.2728092.

[33] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, T. Engel, On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH, IEEE Sensors Journal 13 (10) (2013) 3655–3666. doi:10.1109/JSEN.2013.2266417.

[34] M. R. Palattella, T. Watteyne, Q. Wang, K. Muraoka, N. Accettura, D. Dujovne, L. A. Grieco, T. Engel, On-the-Fly Bandwidth Reservation for 6TiSCH Wireless Industrial Networks, IEEE Sensors Journal 16 (2) (2016) 550–560. doi:10.1109/JSEN.2015.2480886.

[35] L. Lo Bello, A. Lombardo, S. Milardo, G. Patti, M. Reno, Software-Defined Networking for Dynamic Control of Mobile Industrial Wireless Sensor Networks, in: 23rd International Conference on Emerging Technologies and Factory Automation (ETFA 2018), Vol. 1, 2018, pp. 290–296. doi:10.1109/ETFA.2018.8502457.

[36] H. Nishimoto, Y. Kawahara, T. Asami, Prototype implementation of ambient RF energy harvesting wireless sensor networks, in: IEEE SENSORS, 2010, pp. 1282–1287. doi:10.1109/ICSENS.2010.5690588.

[37] F. K. Shaikh, S. Zeadally, Energy harvesting in wireless sensor networks: A comprehensive review, Renewable and Sustainable Energy Reviews 55 (2016) 1041–1054. doi:https://doi.org/10.1016/j.rser.2015.11.010.

[38] S. Y. Shin, H. S. Park, W. H. Kwon, Mutual interference analysis of IEEE 802.15.4 and IEEE 802.11b, Computer Networks 51 (12) (2007) 3338–3353. doi:https://doi.org/10.1016/j.comnet.2007.01.034.

[39] J. Kim, W. Jeon, K. Park, J. P. Choi, Coexistence of Full-Duplex-Based IEEE 802.15.4 and IEEE 802.11, IEEE Transactions on Industrial Informatics 14 (12) (2018) 5389–5399. doi:10.1109/TII.2018.2866307.

[40] S. Pollin, I. Tan, B. Hodge, C. Chun, A. Bahai, Harmful Coexistence Between 802.15.4 and 802.11: A Measurement-based Study, in: 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom), 2008, pp. 1–6. doi:10.1109/CROWNCOM.2008.4562460.

[41] L. Angrisani, M. Bertocco, D. Fortin, A. Sona, Assessing coexistence problems of IEEE 802.11b and IEEE 802.15.4 wireless networks through cross-layer measurements, in: IEEE Instrumentation Measurement Technology Conference (IMTC), 2007, pp. 1–6. doi:10.1109/IMTC.2007.379454.

[42] M. Petrova, L. Wu, P. Mahonen, J. Riihijarvi, Interference Measurements on Performance Degradation between Colocated IEEE 802.11g/n and IEEE 802.15.4 Networks, in: Sixth International Conference on Networking (ICN 2007), 2007, pp. 93–93. doi:10.1109/ICN.2007.53.

[43] B. Polepalli, W. Xie, D. Thangaraja, M. Goyal, H. Hosseini, Y. Bashir, Impact of IEEE 802.11n Operation on IEEE 802.15.4 Operation, in: 2009 Int. Conference on Advanced Information Networking and Applications Workshops, 2009, pp. 328–333. doi:10.1109/WAINA.2009.102.

[44] F. Yao, S. Yang, W. Zheng, Mitigating interference caused by IEEE 802.11b in the IEEE 802.15.4 WSN within the environment of smart house, in: IEEE International Conference on Systems, Man and Cybernetics, 2010, pp. 2800–2807. doi:10.1109/ICSMC.2010.5641899.

[45] S. Ben Yaala, F. Tholeyre, R. Bouallegue, Performance study of co-located IEEE 802.15.4-TSCH networks: Interference and coexistence, in: IEEE Symposium on Computers and Communication (ISCC), 2016, pp. 513–518. doi:10.1109/ISCC.2016.7543790.

[46] J. Umer, H. Di, L. Peilin, Y. Yueming, Frequency hopping in IEEE 802.15.4 to mitigate IEEE 802.11 interference and fading, Journal of Systems Engineering and Electronics 29 (3) (2018) 445–455. doi:10.21629/JSEE.2018.03.01.

[47] S. Zoppi, H. M. Grsu, M. Vilgelm, W. Kellerer, Reliable hopping sequence design for highly interfered wireless sensor networks, in: IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN 2017), 2017, pp. 1–7. doi:10.1109/LANMAN.2017.7972164.

[48] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal on Selected Areas in Communications 18 (3) (2000) 535–547. doi:10.1109/49.840210.

[49] F. Babich and M. Comisso, Theoretical analysis of asynchronous multipacket reception in 802.11 networks, IEEE Transactions on Communications 58 (6) (2010) 1782–1794. doi:10.1109/TCOMM.2010.06.090186.

[50] Pham, Peter P., Comprehensive Analysis of the IEEE 802.11, Mobile Networks and Applications 10 (5) (2005) 691–703. doi:10.1007/s11036-005-3363-x.

[51] D. De Guglielmo and B. Al Nahas and S. Duquennoy and T. Voigt and G. Anastasi, Analysis and Experimental Evaluation of IEEE 802.15.4e TSCH CSMA-CA Algorithm, IEEE Transactions on Vehicular Technology 66 (2) (2017) 1573–1588. doi:10.1109/TVT.2016.2553176.

[52] Ouanteur, Celia and Bouallouche-Medjkoune, Louiza and Aïssani, Djamil, An Enhanced Analytical Model and Performance Evaluation of the IEEE 802.15.4e TSCH CA, Wireless Personal Communications 96 (1) (2017) 1355–1376. doi:10.1007/s11277-017-4241-0.

[53] Luong, Phuong and Nguyen, Tri Minh and Le, Long Bao, Throughput analysis for coexisting IEEE 802.15.4 and 802.11 networks under unsaturated traffic, EURASIP Journal on Wireless Communications and Networking 2016 (1) (2016) 127. doi:10.1186/s13638-016-0586-4.

[54] Soo Young Shin, Throughput analysis of IEEE 802.15.4 network under IEEE 802.11 network interference, AEU - International Journal of Electronics and Communications 67 (8) (2013) 686–689. doi:https://doi.org/10.1016/j.aeue.2013.02.007.

[55] A. Conta, S. Deering, M. Gupta, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, IETF RFC 4443 (2006) 1–23.

[56] G. Cena, I. C. Bertolotti, A. Valenzano, C. Zunino, Evaluation of Response Times in Industrial WLANs, IEEE Transactions on Industrial Informatics 3 (3) (2007) 191–201. doi:10.1109/TII.2007.903219.

[57] T. Watteyne, A. Mehta, K. Pister, Reliability through Frequency Diversity: Why Channel Hopping Makes Sense, in: ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN 2009), PE-WASUN '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 116–123. doi:10.1145/1641876.1641898.

[58] P. Du and G. Roussos, Spectrum-aware wireless sensor networks, in: IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2013), 2013, pp. 2321–2325. doi:10.1109/PIMRC.2013.6666532.

[59] OpenMote, `https://www.industrialshields.com/open-mote-b-industrial-shields-open-source-device-ready-for-internet-of-things`, accessed: 2020-02-21.

[60] OpenWSN, `https://openwsn.atlassian.net/wiki/`, accessed: 2020-02-21.

[61] G. Cena, S. Scanzio, A. Valenzano, Experimental Evaluation of Techniques to Lower Spectrum Consumption in Wi-Red, IEEE Transactions on Wireless Communications 18 (2) (2019) 824–837. doi:10.1109/TWC.2018.2884914.