On the Impact of WiFi on 2.4 GHz Industrial IoT Networks

Fabian Rincon, Yasuyuki Tanaka, Thomas Watteyne

On the Impact of WiFi on 2.4 GHz Industrial IoT Networks

Fabian Rincon[1,2], Yasuyuki Tanaka[2], Thomas Watteyne[2]
[1] *ENSTA ParisTech, Palaiseau, France.*
fabian-antonio.rincon-vija@ensta-paristech.fr
[2] *Inria, Paris, France.*
{yasuyuki.tanaka,thomas.watteyne}@inria.fr

***Abstract*—Many Industrial IoT TSCH networks, such as SmartMesh IP, operate in the same 2.4 GHz frequency band as WiFi. As the smart factory becomes more and more connected, WiFi starts being rolled out on many factory floors. It is hence completely legitimate to question whether WiFi impacts Industrial IoT TSCH networks, and in particular their end-to-end reliability. In a setup which replicates a worst case industrial setting, we conduct a thorough experimental study which looks at the performance of a 47-mote SmartMesh IP network undergoing several levels of WiFi interference. The result is that, even though the latency and power consumption of the network increase, end-to-end reliability stays at 100% even under very high WiFi interference. The conclusion is that TSCH technology at 2.4 GHz, such as SmartMesh IP, is perfectly appropriate to for an industrial environment where WiFi is heavily used. This paper is complemented by a research report which contains all the information needed to replicate the measurements, and the raw data.**

***Keywords*-IEEE802.15.4, TSCH, IEEE802.11, coexistence.**

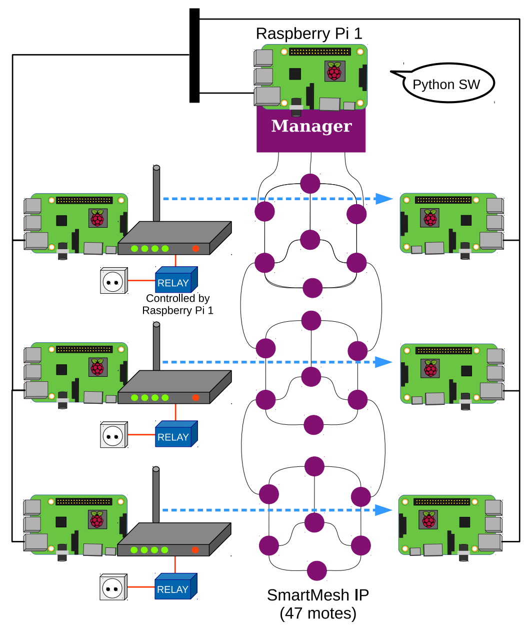


Figure 1: The experimental setup replicates a factory floor on which WiFi networks are deployed alongside an Industrial IoT TSCH network. This setup is a worst case in which any WiFi impactss all parts of the TSCH network. To ensure our measurements reflect the real world, we use the commercial SmartMesh IP, the market leader in TSCH technology.

I. Introduction

Time Synchronized Channel Hopping (TSCH) is the medium access control (MAC) mode that was added to the 2015 revision of the IEEE802.15.4 standard to "better support the industrial markets" [1]. The IETF 6TiSCH working group further defines a fully standardized protocol stack, by associating IEEE802.15.4 TSCH with an "upper stack" which includes the routing and application layers [2]. Other standards such as WirelessHART and ISA100.11a have defined similar TSCH-based solutions on top of IEEE802.15.4 for the industrial markets.

While the IEEE802.15.4 standard supports several frequency band, by far the most used in the TSCH context is the 2.400-2.485 GHz ISM ("Industrial Scientific and Medical") band. The main reasons for that are that *(1)* the band is 85 MHz wide, giving a channel hopping solution enough frequency diversity to efficiently fight external interference and multi-path fading, and *(2)* the band is the same word-wide, so manufacturers can make a single product rather than having to make per-region variants of their hardware/software.

TSCH networks operating at 2.4 GHz have been very successful commercially, with tens of thousands of networks operating today[1]. Their applications are highly industrial, such as monitoring bearing temperature, motor current, well-heads on offshore platforms, air compressors in refineries, etc.

The question about coexistence between WiFi and TSCH is a perfectly legitimate one. First, WiFi and TSCH share the same frequency band, and since a WiFi signal covers 4 TSCH frequencies, a WiFi device transmitting can block up to 4 TSCH devices. Second, a WiFi device emits at a much higher power, overpowering a IEEE802.15.4 signal sent at the same time. This fear is confirmed in single-

[1] One vendor alone, Emerson, announces over 39,000 networks sold, which translates into over 12 Billion operating hours. https://www.emerson.com/en-us/expertise/automation/industrial-internet-things/pervasive-sensing-solutions/wireless-technology

channel IEEE802.15.4-based solutions, such as ZigBee. Those systems typically require a site survey before deployment so the ZigBee operates on a frequency outside of the frequencies used by WiFi in that factory[2]. Third, industrial plants are evolving, with the line between the "office" and the "factory" blurring. This means that it's not unusual to see WiFi rolled out throughout a factory floor.

The goal of this paper is precisely to answer this question: *Will a WiFi network deployed on a factory floor (adversely) impact the Industrial IoT TSCH network operating in the same radio space?* We look at this question purely from an industrial use point of view. That is, the most important is to understand whether WiFi can alter the dependability of the network: can it cause nodes to de-synchronize and loose connectivity to the network, and/or impact the end-to-end reliability of the network? As a second point, we are interested in seeing the impact of WiFi of performance metric that are typically considered secondary for process monitoring applications: end-to-end latency, battery lifetime, and network formation time.

Unlike previous work that addressed similar questions, we adopt an experimental approach. This is due to 3 considerations. First, the topic includes multiple wireless technologies, with behaviors that encompass every layer in the protocol stack, from complex phenomena in propagation, all the way to the algorithms that run on the systems. We don't believe that analysis or simulation is able to accurately capture all of those. Second, we have access to the market leading hardware and have, in practice, the means of conducting an experimental measurements campaign that uses exactly the same technology is being used throughout the Industrial IoT space. We use Analog Devices SmartMesh IP because it's the market leading TSCH commercial product with over 76,000 networks running [3], and because it's performance is understood and published [4]. Finally, we want to ensure the validity and usefulness of our findings to a practical user, and so what to "play user" and report on down-to-earth measurements with real devices.

The contribution on this paper is three-fold:

- We develop a methodology and a set of tools that ensure the repeatability of the results, and that described thoroughly both in this paper and a companion research report, and that is published under an open-source license.
- We show how the TSCH network, even under levels of WiFi interference much higher than would be seen in practice, the functionality of the TSCH network is unaffected, and that it maintains an end-to-end reliability of 100%.
- We quantify the impact of WiFi on end-to-end latency, battery lifetime, and network formation time.

[2] Typical ZigBee networks are configured to operate on IEEE802.15.4 channel 26 (2.480 GHz), which is usually not used by WiFi.

The remainder of this paper is organized as follows. Section II goes through the previous published work most related to this study. Section III describes the overall setup of the experiment. This description, together with the companion research report, ensure repeatability of the experiment. Section IV lists the performance metrics of the SmartMesh IP network we are monitoring during an experiment, details how they are gathered, and discusses why they are representative. Section V presents the results and discusses their importance in the context of Industrial IoT applications. Finally, Section VI summarizes this paper and lists the lessons learned.

II. Related Work

This section gives the necessary background on IEEE802.15.4 and WiFi, then surveys recently published work that we believe it the most related to the study presented in this paper. Our focus in our work that relates to evaluating the robustness of an IEEE802.15.4-based network against interference from other networks, in parciular WiFi.

The IEEE802.15.4 standard cuts the 2.4 GHz ISM band into 16 orthogonal frequencies. An IEEE802.15.4-compliant signal is 2 MHz wide; adjacent frequencies are 5 MHz apart. TSCH uses "Channel Hopping", i.e. when two neighbor nodes exchange data frames, successive frames are sent on different frequencies. Channel hopping is well known to combat external interference and multi-fading, and is therefore used in Bluetooth, some cellular systems, and all TSCH-based low-power wireless soutions [5].

The IEEE802.11 standard (the standard promoted by the WiFi Alliance) uses the exact same 2.4 GHz frequency. It does cut the band into frequencies differently: WiFi signals are 22 MHz wide, and adjacent frequencies are only 5 MHz apart, i.e. they are non-orthogonal. Many commercial deployments use WiFi channels 1 (2.412 GHz), 6 (2.437 GHz) and 11 (2.462 GHz), which are mutually orthogonal.

The output power of a WiFi and IEEE802.15.4 device is also quite different. To have the longest range, WiFi access points can have an output power of +23 dBm (200 mW). In comparison, to conserve battery, IEEE802.15.4 radios are typically used at 0 dBm (1 mW); Analog Devices' SmartMesh product lines are an exception as they output +8 dBm (6 mW). As a rule of thumb, the output poiwer of WiFi is roughly $100\times$ the output power of IEEE802.15.4.

Wagh *et al.* [6] evaluate ZigBee under WiFi traffic by simulation, using NS-2. They collect packet error rate and bit error rate values, for different distances between the WiFi access point and the ZigBee network, and for different frequency offsets between WiFi and ZigBee. They conclude that 8-m distance and 8 MHz frequency offset is a safe configuration to avoid interference by WiFi.

Lim *et al.* [7] conduct a series of experiments using off-the-shelf IEEE802.15.4 and WiFi devices. They vary the physical position of the different devices, as well as the frequencies used by both technologies. They witness the WiFi interference impacting the bit errors of IEEE802.15.4, but also the fact that the CCA (Clear Channel Assessment) returns a “medium busy” reading, causing the IEEE802.15.4 devices to back off.

Wagh *et al.* [6] and Lim *et al.* [7] discuss how some coordination between WiFi networks and IEEE802.15.4 networks is needed to make the latter perform well. We believe that this option is not realistic as background network activities dynamically change, and because one cannot always have full control over all the wireless networks at a deployment site. Measurements by Marcoen *et al.* [8] support this case. They conduct measurements in an office and in a home with two IEEE802.15.4 radio devices. Their results show interference level varying over time, even on the same channel.

The works above all focus on single-channel IEEE802.15.4 technolgies such as ZigBee. The more recent TSCH [1] mode of the same standard is, however, very different.

Vigelm *et al.* [9] evaluate TSCH performance with OpenWSN[3] running on Zolertia Z1 devices, with WiFi background traffic. In one of their three interference scenarios, heavy traffic is generated over two WiFi channels: 1 and 5. The results shows that the mean end-to-end packet delivery ratio is 90% or more, even under heavy WiFi traffic when application traffic is low.

Experiments by Gursu *et al.* [10] also show that the channel hopping nature of TSCH is efficient at combating WiFi interference. In their experiments, three WiFi APs operate on three channels: 1, 5, and 11.

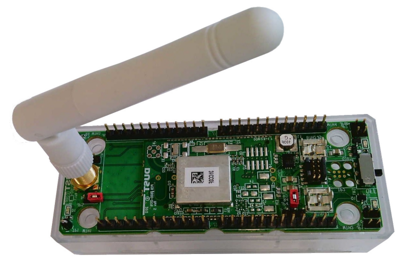
Another insight given by Vilgel *et al.* [9] and Gursu *et al.* [10] is that TSCH could perform better than what they show in terms of end-to-end reliability and latency. The key is the TSCH schedule. Because TSCH operates in a deterministic manner, network nodes need to know when they can transmit a frame and when they should listen for an incoming frame. TSCH performance depends heavily on the schedule being used [11].

Previously published related work on the impact of WiFi on TSCH networks is sparse, and focuses only on academic and canonical settings, in particular in terms of TSCH schedule, number of nodes, and thoroughness of the experiment. We believe that there is a real need, from an end-user point of view, to have a study that shows the impcat of WiFi on a real commercial TSCH implementation. This paper contributes to answering that need.

III. Experimental Setup

The driving requirements of the experimental setup are *(1)* we want to use equipment which matches what is being used in the Industrial IoT, *(2)* we want to have full control over the WiFi interference, i.e. deploying in a building where there is WiFi is not enough, and *(3)* we want to create a worst case scenario in which any WiFi signal impacts the entire SmartMesh IP network.

(a) DC2274 SmartMesh IP manager



(b) DC9025 SmartMesh IP mote

Figure 2: In the experiment, we use one DC2274 SmartMesh IP manager and 47 DC9025 SmartMesh IP motes.

To satisfy requirement *(1)*, we use SmartMesh IP devices (see Section III-A). To satisfy requirement *(2)*, the experiment takes place in an underground parking lot at the Inria-Paris research center. No WiFi infrastructure is deployed in that space, giving us full control over the WiFi interference we generate during the experiment. To satisfy requirement *(3)*, the devices are all co-located on a cart to create a worst-case situation in which WiFi traffic from any WiFi router impacts all nodes in the SmartMesh IP network.

In this section, we explain the hardware, the system configuration, and the scenarios used in our experimental study. The experimental setup is thoroughly explained in the companion research report to this publication [12].

A. Hardware

1) Motes: We use 47 DC9025 prototypes motes, by Analog Devices. At the heart of this mote is an LTP5902 module[4], which itself contains the LTC5800 System-on-Chip (ARM Cortex-M3, IEEE802.15.4-compliant radio). The DC9025 mote further features an MMCX antenna connector, an external whip antenna, a holder for two AA batteries, and a plastic enclosure. Fig. 2b shows the DC9025 mote with the cover of the plastic enclosure removed.

The motes run the unmodified default SmartMesh IP firmware.

[3] A reference open-source 6TiSCH implementation, http://www.openwsn.org/

[4] http://www.analog.com/media/en/technical-documentation/data-sheets/59012ipmfa.pdf

2) Manager: We use the DC2274 SmartMesh IP manager[5]. This manager features the same LTP5902 module as on the mote, but running the embedded manager firmware, and a USB connection to access its serial ports. Fig. 2a shows the DC2274 SmartMesh IP manager.

3) WiFi interference: To generate the WiFi interference, we use three tp-link TL-WR841N WiFi routers, and 7 Raspberry Pi 3 Model B+ single board computers.

B. System Configuration

The experiment consists of a SmartMesh IP network undergoing WiFi interference, as depicted in Fig. 1.

The SmartMesh IP network is composed of 1 manager and 47 motes. The motes send a data packet to the manager every 30 s, and a health report each 5 min. The manager is connected over USB to the master Raspberry Pi. The master Raspberry Pi asks the manager for a snapshot of the network every 15 min. All this data is stored on the master Raspberry Pi in log files.

To create the WiFi interference, we use 3 routers whose power supply is controlled by the master Raspberry Pi using a relay. We are turn any WiFi route on or off programmatically. Each WiFi network operates on a different frequency. We choose WiFi channels 1 (2.401-2.423 GHz), 6 (2.426-2.448 GHz), and 11 (2.451-2.473 GHz) which are orthogonal to one another, and generate interference over the entire 2.4 GHz ISM band. This combination of WiFi channels is commonplace in commercial deployments.

To generate WiFi traffic, we use two slave Raspberry Pi boards per router, and have those continuously exchange a file using Secure Copy Protocol (SCP). SCP allows us to control the maximum data transfer rate, which impacts the amount of interference generated. The transmitting Raspberry Pi is connected by Ethernet to the router, the receiving Raspberry Pi is connected over WiFi. The transmitting Raspberry Pi starts the file transfer process: the data is sent to the router over Ethernet, then from the router to the receiving Raspberry Pi over WiFi.

These operations are orchestrated by the master Raspberry Pi which issues commands over SSH on all slave Raspberry Pi's.

C. Scenarios

We collect the metrics listed in Section IV under different levels of WiFi interference.

The power supply of each WiFi router is controlled through a series of relays by the master Raspberry Pi. We can hence hence have 0, 1, 2 or 3 WiFi routers operating. Having a router at all, even without any active traffic, already generates interference as the router sends WiFi beacons every 100 ms.

[5] http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/dc2274a-a.html

Table I: The 10 configurations used in the experiment, giving us several levels of WiFi interference of 3 WiFi channels.

#	WiFi channel 1 (2.412 GHz)	WiFi channel 6 (2.437 GHz)	WiFi channel 11 (2.462 GHz)
1	*off*	*off*	*off*
2	0 MBps	*off*	*off*
3	1.5 MBps	*off*	*off*
4	3.0 MBps	*off*	*off*
5	0 MBps	0 MBps	*off*
6	1.5 MBps	1.5 MBps	*off*
7	3.0 MBps	3.0 MBps	*off*
8	0 MBps	0 MBps	0 MBps
9	1.5 MBps	1.5 MBps	1.5 MBps
10	3.0 MBps	3.0 MBps	3.0 MBps

Once a router is running, we use the two slave Raspberry Pi's attached to it to exchange data. That exchange can be done at several rates, resulting in more or less interference. We use 3 exchange rates: 0 MBps (no transfer, i.e. only the router is running), 1.5 MBps and 3.0 MBps.

This results in 10 configurations, summarized in Table I.

IV. Performance Metrics

To quantify the impact of WiFi interference, we monitor the performance metrics of the SmartMesh IP network that undergoes that interference. The key metrics are the **average end-to-end latency**, the **link layer stability** and the **end-to-end reliability**. We also monitor the **battery lifetime**, the **average link stability per channel**, the **network formation time**.

All of these metrics are gathered and published by the SmartMesh IP network, through a combination of health reports sent by the motes, and the snapshots taken at the manager. These metrics can hence be extracted from the log files gathered by the master Raspberry Pi. The remainder of this section details how.

The **average end-to-end latency**, **average link layer stability** and **end-to-end reliability** are automatically gathered by the SmartMesh IP manager. The master Raspberry Pi, as part of its snapshot, issues a `getNetworkInfo` command every 15 min, to extract those metrics from the SmartMesh IP manager, into the log file.

We compute the **battery lifetime** from the charge the motes report every 15 min in their device health report. That charge is reported in milli-coulombs (mC). Then use (1) to turn that charge into a battery lifetime, where BC is the battery charge in mAh, ED is the experiment duration in seconds, and C is the charge consumed by the mote in mC. We assume the mote is powered by a pair of 1.5 V AA batteries, holding 2200 mAh of charge.

$$lifetime(years) = \frac{BC \cdot ED}{C} \cdot \frac{1}{24 \cdot 365} \qquad (1)$$

To compute the **average link stability per channel**, we use the extended health report each mote published every 15 min. This contains the number of transmission attempts and the number of transmission failures over the last 15 min, for each neighbor, for each channel. We use (2) to compute link stability[6].

$$stability = 1 - \frac{transmission \quad attempts}{transmission \quad failures} \tag{2}$$

To observe the **network formation time** of each mote, we timestamp when the experiment starts, and when each of the motes joins the network. The start of the experiment is when the manager is reset. A mote has joined the network when the manager publishes an `eventMoteOperational` notification.

V. Experimental Results

We run the network for 9.5 h for each of the configurations in Table I. At each configuration, the SmartMesh IP network completely re-builds. The resulting network is very shallow (mostly 1-hop, with sometimes as 2-hop transmission), as all nodes are co-located.

A. Link Stability

We use the terms link stability and packet delivery ratio interchangeably, to refer to the portion of frames sent from mote to its neighbor that are successfully acknowledged. The link stability only depends on the WiFi interference, and does not reflect in any way the performance of the SmartMesh IP network. It is, however, a good metric to quantify the impact of WiFi interference on the physical layer.

In Fig. 3, we plot the link stability per frequency. We confirm that, when only one router is on (the "1 WiFi" cases) only IEEE802.15.4 channels 11-14 are affected. When adding more WiFi routers (each operating on a different frequency), more IEEE802.15.4 channels are affected. Second, we see that a data transfer at 3 MBps can lower the link stability down to 40% on the frequencies the WiFi activity goes on. As a point of comparison, Brun-Laguna *et al.*measure the link stability in an office building in which WiFi is heavily used, and the average link stability is never below 85%, on any frequency [13]. The maximum level of WiFi interference generated in our experiments is far more than what a TSCH network would experience in a typical WiFi-savvy environment.

Fig. 3 shows that the setup we have chosen results in WiFi interference that does impact the IEEE802.15.4 links, and that does so far more than the typical impact one would see in an office building.

Fig. 4 shows link-layer stability, but averaged over all channels. The result is perhaps more readable than Fig. 3. The green dot in Fig. 4 represents the link-layer stability with no WiFi router on. That average is 96%, which is expected[7]. The average link-layer stability decreases both with the number of WiFi routers on, and with the amount of traffic on those WiFi networks. The worst average link-layer stability is 51%.

[6] The terms "link stability" and "packet delivery ratio (PDR)" can be used interchangeably.

[7] Multi-path fading explains that this isn't 100%

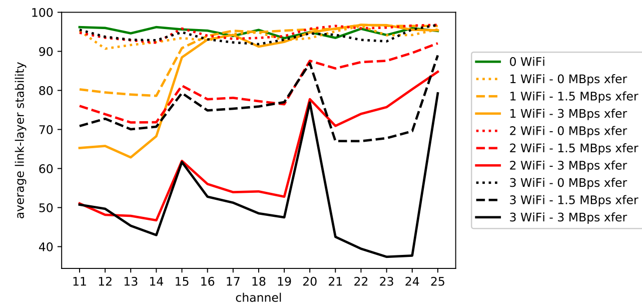


Figure 3: Plotting the link stability as a function of channel clearly shows the frequency band each WiFi router interferes with. We also see that, the worst case WiFi interference is far heavier than what is typically seen in a office environment [13].

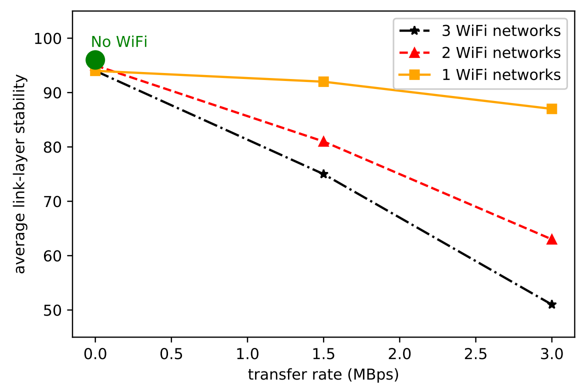


Figure 4: As expected, increasing the number of WiFi routers, or the load on each one decreases the average link-layer stability in the TSCH network.

B. Average End-to-End Latency

The fact that the link-layer stability isn't 100% means that a node sometimes needs to link-layer retry a frame for it to be successfully received. Retrying takes time, so we expect the average end-to-end latency to increase with the level of WiFi interference. Fig. 5 shows this average end-to-end latency, measured on the SmartMesh IP network. As a baseline, without WiFi interference, the end-to-end latency is 700 ms. This value can be decreased or decreased by changing the configuration of the network (see [4] for an in-depth discussion). In this paper we use it as a baseline and don't look at its actual value.

As expected, the average latency increases with the level of WiFi interence, up to 2.2 s.

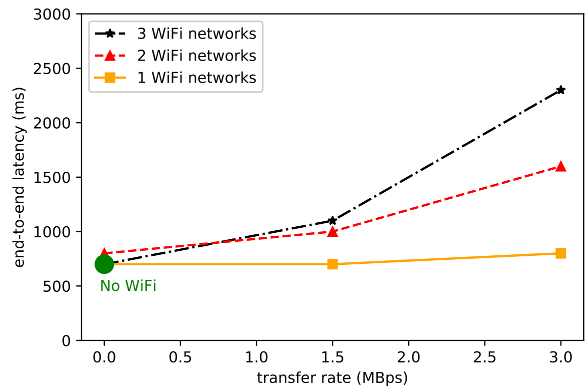


Figure 5: Because WiFi causes nodes to have to retransmit link-layer frames, an increase in WiFi interference causes the end-to-end latency to increase.

C. Battery Lifetime

We compute the battery lifetime indirectly by gathering the average current draw of the nodes after the 9.5 h of the experiment, then compute the expected lifetime assuming *(1)* the motes are powered by a 2,200 mAh 2×AA battery pack, and *(2)* that the batteries are a perfect bucket of charge. Both these assumptions can be challenged, and, given the fact that only one experiment is conducted for each WiFi interference level, the results only indicate general trends.

We plot the battery lifetime in two ways.

Fig. 6 shows the distribution of battery life for each level of WiFi interference. As the link layer stability decreases, the number of retransmissions increases, which necessary increases the energy consumption of the motes. The mean battery lifetime for all levels is between 6 and 7 years, and while the impact of WiFi interference is visible, it is minimal.

Fig. 7 plots the network lifetime, i.e. the time until the first mote depletes its battery. Fig. 7 corresponds to the "min" values of Fig. 6, i.e. the bottom of the whiskers.

D. End-to-End Reliability

We define end-to-end reliability as the portion of the data packets generated by the motes that reach the manager. SmartMesh IP announces end-to-end reliability above 99.999%, the "five nines" rule which is used across the industrial networking domain.

Fig. 8 shows end-to-end reliability measured in our experiment. In all cases, end-to-end reliability is 100%, not a single packets is lost. This holds even with the highest level of interference where the average link-layer stability is as low as 51%, and which the SmartMesh IP stack compensates for through more link-layer retransmissions.

This result is of utmost importance for industrial applications, for which any data packet lost could result in factory

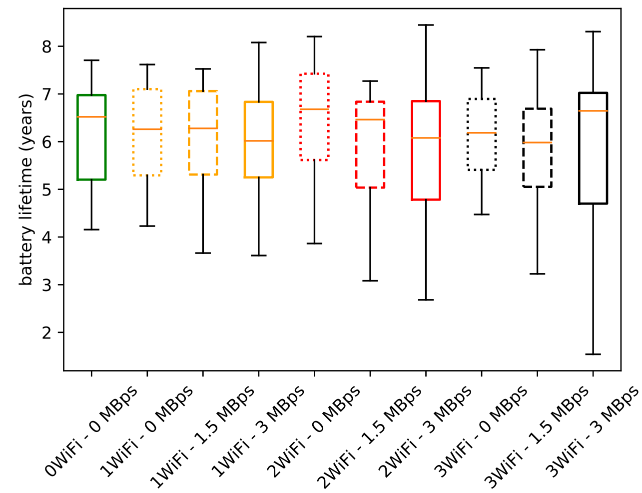


Figure 6: Battery lifetime distribution of all motes. Whiskers show the min/max lifetime, the box the standard deviation around the average lifetime. Battery lifetime for all levels is between 6 and 7 years, and while the impact of WiFi interference is visible, it is minimal.

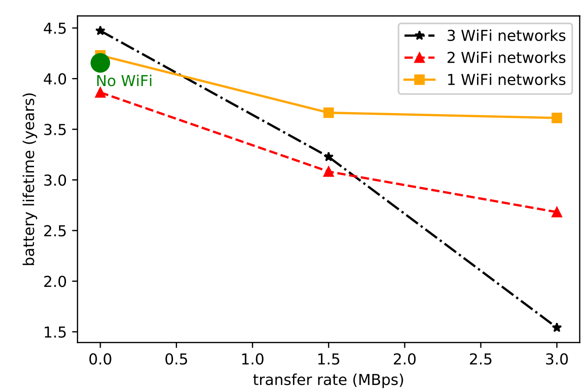


Figure 7: Because WiFi causes nodes to retransmit link-layer frames, they consume more, and the motes' battery lifetime decreases.

down-time.

E. Network Formation Time

All TSCH networks today have a similar join procedure. A new node starts by listening for beacon frames sent by nodes already in the network, synchronizes to the network, and performs a security handshake with a security manager which is usually running next to the gateway of the network. The same general procedure is shared by WirelessHART. ISA100.11a and 6TiSCH.

In our experimental setup, all nodes are running and try to join the network at the same time as soon as the manager is

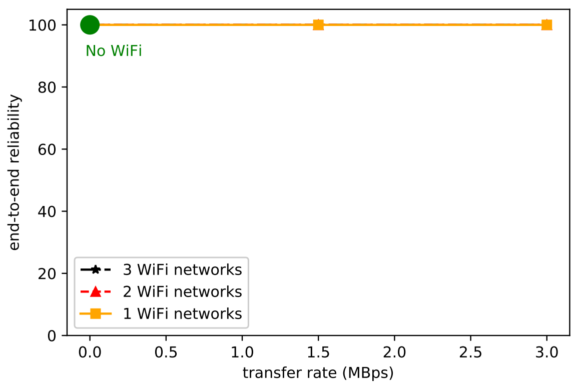


Figure 8: Even under extreme levels of WiFi interference, the end-to-end reliability of the SmartMesh IP network remains at 100%. Not a single packet is lost in any of the 10 experiments.

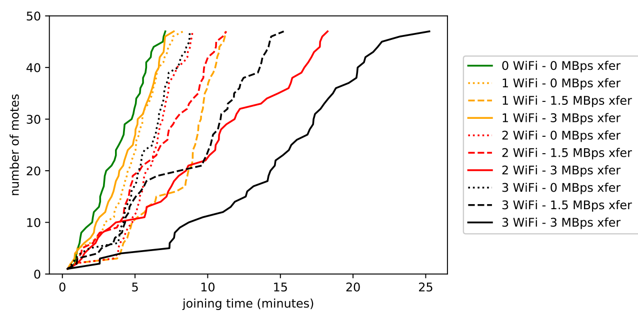


Figure 9: Network formation time.

switch on. This creates congestion, which results in a back-off mechanism kicking in to resolve the congestion between the different security handshake. WiFi interference indirectly causes the network formation time to increase, because it decreases the link layer stability, which in turn increases the retransmission and overall congestion.

We witness this is Fig. 9, which shows how the portion of joined nodes increases as time elapses after the manager has been switched on. Without interference, the 47 mote network forms in roughly 7 min. At very high WiFi interference, the network perfectly forms, only it takes close to 25 min.

VI. Conclusions

SmartMesh IP and WiFi both operate in the 2.4 GHz ISM frequency band. This paper reports on a thorough experimental study which looks at the performance of a 47-mote SmartMesh IP network undergoing several levels of WiFi interference. The result is that the latency, power consumption and formation time of the SmartMesh IP network only very slightly increase even under very high WiFi interference, without any effect on the 100% end-to-end reliability or overall function of the network. The conclusion is that TSCH technology at 2.4 GHz, and SmartMesh IP in particular, are perfectly appropriate to the used in an environment where WiFi is heavily used.

References

[1] *802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks*, IEEE Std., April 2016.

[2] T. Watteyne, M. R. Palattella, and L. Grieco, *Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement*, IETF Std. RFC7554, May 2015.

[3] T. Watteyne, L. Doherty, J. Simon, and P. Pister, "Technical Overview of SmartMesh IP," in *Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2013.

[4] T. Watteyne, J. Weiss, L. Doherty, and J. Simon, "Industrial IEEE802.15.4e Networks: Performance and Trade-offs," in *Conference on Communications (ICC)*. IEEE, June 2015, pp. 604–609.

[5] T. Watteyne, A. Mehta, and K. Pister, "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense," in *Sixth ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (ACM PE-WASUN)*, 2009.

[6] S. Wagh, A. More, and P. Kharote, "Performance Evaluation of IEEE 802.15.4 Protocol Under Coexistence of WiFi 802.11b," in *Conference on Recent Trends in Computing (ICRTC-2015)*, 2015.

[7] C. Lim, M. Bolt, A. Syed, P. Ng, C. Goh, and Y. Li, "Dynamic Performance of IEEE 802.15.4 Devices under Persistent WiFi Traffic," in *Conference on Recent Advances in Internet of Things (RIoT)*, 2015.

[8] L. Marcoen, B. Bergh, and S. Pollin, "On the Performance of 802.15.4e in the Real World," in *Symposium on Wireless Communication Systems (ISWCS)*, 2015.

[9] M. Vilgelm, M. Gursu, S. Zoppi, and W. Kellerer, "Time Slotted Channel Hopping for Smart Metering: Measurements and Analysis of Medium Access," in *Conference on Smart Grid Communications (SmartGridComm)*, 2016.

[10] M. Gursu, M. Vilgelm, S. Zoppi, and W. Kellerer, "Reliable co-existence of 802.15.4e TSCH-based WSN and Wi-Fi in an Aircraft Cabin," in *Workshop on Advanced PHY and MAC Technology for Super Dense Wireless Networks (CROWD-Net)*, 2016.

[11] R. Alves and C. Margi, "IEEE 802.15.4e TSCH Mode Performance Analysis," in *Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2016.

[12] F. Rincón, Y. Tanaka, and T. Watteyne, "Experimental Study on the Coexistence between SmartMesh IP and WiFi," Inria, Tech. Rep., September 2018.

[13] K. Brun-Laguna, A. L. Diedrichs, D. Dujovne, C. Taffernaberry, R. Leone, X. Vilajosana, and T. Watteyne, "Using SmartMesh IP in Smart Agriculture and Smart Building Applications," *Elsevier Computer Communications Journal*, 2018.