Denial-of-Sleep Attacks against IoT Networks

Antoine Gallais, Thin-Hinen Hedli, Valeria Loscrì, Nathalie Mitton

Denial-of-Sleep Attacks against IoT Networks

Antoine Gallais[‡§] and Thin-Hinen Hedli[‡]
[‡]ICube Laboratory, CNRS / University of Strasbourg,
Pole API, Boulevard Sebastien Brant,
67412 Illkirch Cedex, France
Email: antoine.gallais@inria.fr

Valeria Loscri[§] and Nathalie Mitton[§]
[§]FUN - Self-organizing Future Ubiquitous Network
Inria Lille - Nord Europe, avenue Halley,
Villeneuve d'Ascq, France
Email: {valeria.loscri, nathalie.mitton}@inria.fr).

***Abstract*—Numerous medium access control (MAC) have been proposed for Low-power Lossy Networks (LLNs) over the recent years. They aim at ensuring both energy efficiency and robustness of the communication transmissions. Nowadays, we observe deployments of LLNs for potentially critical application scenarios (e.g., plant monitoring, building automation), which require both determinism and security guarantees. They involve battery-powered devices which communicate over lossy wireless links. Radio interfaces are turned off by a node as soon as no traffic is to be sent or relayed. Denial-of-sleep attacks consist in exhausting the devices by forcing them to keep their radio on. We here focus on jamming attacks whose impact can be mitigated by approaches such as time-division and channel hopping techniques. We use the IEEE 802.15.4e standard to show that such approaches manage to be resistant to basic jamming but yet remain vulnerable to selective jamming. We discuss the potential impacts of such onslaughts, depending on the knowledge gained by the attacker, and to what extent envisioned protections may allow jamming attacks to be handled at upper layers.**

I. Introduction

Numerous Internet of Things (IoT) deployments have grown in maturity. The typical sensor nodes that compose the so-built networks embed communication technologies along with storing and processing abilities [1]. These objects offer a wide variety of possibilities and numerous applications are now relying on those solutions. The constantly increasing demand for accurate monitoring systems imposes to collect the expected measurements for long periods of time with minimal human intervention. Their low cost and small size yet limit their computing capacity, bandwidth, memory and energy. While energy expense is critical for the overall network lifetime, these devices also face demanding radio environments, inducing radio links of highly varying qualities. These so-called Low-power Lossy Networks (LLNs) benefit from a large variety of solutions to fill the requirements of the end applications. Physical transmissions, medium access control and routing allow the deployed devices to report to a base station which collects, stores and processes the received data.

More specifically, the main source of energy consumption being the radio communications, much work has been done at physical and medium access control (MAC) layers to limit the energy expense while allowing efficient wireless transmissions. Radio interfaces can be turned off or reduced to some basic operations (i.e., idle) in order to save energy without endangering the deployment. The nodes thus operate according to a cycle of operation, which consists in alternating phases of sleep and activity. The MAC layer allows to identify those periods, either centrally or in a distributed fashion, the goal being to minimize the duty cycle of the objects in order to save as much energy as possible. MAC protocols classification depends on whether nodes are time-synchronized or not [2]. In addition, slow channel hopping has recently gained much attention as it has been proved to combat efficiently narrow band noise [3], thus becoming highly common in industrial networks.

As they allow to manage critical facilities (e.g., power grid, water treatment plant), these so-called Industrial Internet of Things (IIoT) networks require security at every layer of the communication stack [4], while inheriting vulnerabilities of classical IoT networks. Denial of service attacks are some of the security problems that must be tackled once deploying such IoT networks [5]. Thus, energy awareness and security have become two interrelated challenges to be addressed [6].

Denial-of-Sleep (DoS) attacks consist in exhausting the batteries of the devices by increasing their duty cycle. By forcing nodes to awake at unnecessary times or by inducing additional duty (e.g., listening, retransmissions), these attacks aim at reducing the expected lifetime of the constituted IoT network [7]. Detecting abnormal uses of highly energy-consuming tasks (e.g., idle listening, overhearing, retransmissions) yet remains challenging [8].

In this paper, we focus on the vulnerabilities of MAC solutions and the mechanisms they can embed to provide security by nature. We especially investigate physical jamming scenarios where attackers prevent communications from taking place, thus leading to further retransmissions and additional duty of the target devices. We detail some existing attacks before focusing on communication technologies which are being investigated by some of the main standardization bodies. We consider time-synchronized and channel hopping (TSCH) networks that are being designed for industrial wireless devices. Those solutions minimize the risks of collisions and reduce idle listening, while providing some cryptographic suites to ensure authentication and encryption if needed. We describe how some Denial-of-Sleep attacks could yet be successful over such networks. We anticipate their potential impact, depending on the knowledge an attacker would be able to gain. Some preliminary simulation results show how various scenarios of selective jamming can perform against TSCH networks.

Section II details some existing Denial-of-Sleep attacks and countermeasures. Section III introduces the TSCH mode of the IEEE 802.15.4e and details how jamming can be performed against such networks which are resistant *by design* (i.e., due to channel hopping). We present some preliminary results in Section IV before discussing some conclusions and perspectives in Section V.

II. Background and related work

Denial-of-Sleep attacks may target various IoT protocols [6]. Various strategies can be used, ranging from useless data packets that receivers must process to jamming of radio channels that causes errors and thus costly retransmissions [9]. Furthermore, malicious control packets may lead to excessive energy consumption due to e.g., illegitimate duty-cycle modifications. From the Medium Access Control (MAC) standpoint, those attacks target the main provided service, namely energy-efficient point-to-point communications that allow IoT networks to collect data for long periods of time.

MAC solutions for LLNs can be divided into two main families, depending on whether time synchronization among nodes is a prerequisite or not. On the one hand, non synchronized devices are required to send preamble before data in order to awake periodically sampling neighbors for the upcoming transmission [10]. Attacks would focus on waking up as many nodes as possible while preventing from long periods of sleep. On the other hand, time-synchronized objects must agree on transmission periods to wake-up synchronously in order to send and receive data [2]. Attacking the scheduling devices or even adding malicious control traffic in the network would result in nodes sending or receiving during wrong slots and thus spending energy for potentially long periods of time. We here review some attacks able to exhaust such networks.

A. Denial-of-Sleep over asynchronous IoT networks

ContikiMAC incorporates most of the innovative features proposed for preamble-sampling mechanisms [11]. It however lacks some defenses to counter denial-of-sleep attacks appropriately. In [12], authors identify three different DoS attacks that can be made against ContikiMAC, namely ding-dong ditching, collision attacks and pulse-delay attacks. They detail some optimizations that allow to secure some critical operations. For instance, a secure phase-lock optimization confines the maximum length of sequences of unicast frames throughout a session by sending a keep-alive frame to a permanent neighbor whose wake-up time was not updated for a critical period of time. Even though this solution comes at a low overhead, authors acknowledge that some attacks could still lead to some routing instabilities thus leading to an energy-consuming reorganization of the routing topology.

In [13], replay attacks over preamble-sampling devices are studied. Some anti-replay mechanisms are investigated e.g., a Bloom filter to find out if a packet has already been received. Each node includes a varying key in its preambles, whose value depends on the transmission time. Receiving nodes can then check that the packet is not being replayed.

The IEEE 802.15.4 standards includes a coordinated sampled listening (CSL) mechanism that mimics preamble-sampling. Although protected against basic eavesdropping, injection and replay attacks, it offers no denial-of-sleep protection. In [14], Krentz et al. apply some countermeasures in the context of IEEE 802.15.4 and show that CSL can be made resistant to most of DoS attacks.

Other kinds of asynchronous MAC layer solutions rely on Wake-up radio where nodes are equipped with two interfaces, one being responsible to wake up the receiver [15]. DoS attacks can thus be led by generating control traffic intended to wake up as many receivers as possible. In [16], the proposed AntiDoS protocol counteract Denial-of-Sleep attacks by using cryptographic primitives (i.e., hash, certificates) to provide flexible and secure peers authentication and keys exchange. This solution can be combined with some IoT standards (e.g., IEEE 802.15.4, 6LoWPAN).

B. Denial-of-Sleep over synchronous / hybrid IoT networks

Attacks can be categorized by taking into account the attacked layer and the attacker's intelligence (e.g., knowledge of the used protocol) [17]. Popular sleep-denial attacks consist in either transmitting unauthenticated packets or replaying a recorded traffic [18]. Even though unauthenticated packets would be discarded due to failed authentication, their decoding causes receivers to waste energy. Replaying a recorded traffic allows to pass this authentication phase and can be used to inject false information that endangers the network or increase energy expense (e.g., false routing information, wrong increase of duty-cycle). An attacker simply has to observe the ongoing traffic in order to identify when to send which packets.

In [19], authors show that an attacker can discover some of the nodes schedules by accessing the IEEE 802.15.4 Guaranteed Time Slots (GTS) descriptor. Such gained knowledge allows to send fake packets to the concerned nodes at appropriate periods in order to create collisions. The attacker can also wait for the completion of a reception task to keep a node awake longer, by e.g., sending a stream of unicast packets to this destination node.

The IEEE 802.15.4 synchronous protocol offers options for securing the network against such network-fatal replayed packets. By checking the frame counter included in the Auxiliary Security Header field, a node verifies that a data exchange is taking place with a given transmitter. Such verification does not come for free however. IEEE 802.15.4 uses AES-CCM mode, whose usage would result in an increased energy expense [20], due to the additional computation and communication tasks (e.g., data encryption/description, authentication, attack detection and defending). In [21], Cao et al. evaluate the vulnerability of such security primitives. They propose to secure the access control by assuming a key known to all nodes, which further use frame counters for every neighbor, thus ensuring message integrity, confidentiality, and replay protection.

C. Denial-of-Sleep with jamming attacks

As discussed above, numerous mechanisms exist to protect IoT networks against most of DoS attacks (e.g., replay, packet injection). Communication protocols handle operations at upper

layers which all eventually rely on the service provided by the physical layer. Therefore, jamming attacks appear as the hardest to protect against. As studied in [22] in the similar context of IEEE 802.11 wireless networks, such attacks cannot be avoided with traditional security methods since attackers do not need to know about the employed protocols to continually transmit on a wireless channel.

However, some smarter attackers would transmit traffic only if the channel is busy, or use the knowledge of the deployed protocols. Using the example of synchronized protocols (e.g., S-MAC [23], IEEE 802.15.4), an attacker would analyze the traffic and recognize the active period thanks to the control packets. It would then be able to alternate between phases of attack and sleep and thus save its energy while endangering the network. In the following sections, we investigate jamming against networks robust by nature against such attacks.

III. Jamming attacks against time-synchronized and channel hopping networks

We here investigate to what extent time-synchronized and channel hopping networks can be resistant to jamming attacks. Such attacks are commonly assumed as impossible to prevent since they target the communication medium, before any communication protocol can protect the endangered devices.

A. TSCH networks

We aim at focusing on the IEEE 802.15.4-2015 standard [24]. Its Time-Slotted Channel Hopping (TSCH) mode is destined to challenging industrial environments where both reliability and determinism are expected. The devices here deal with harsh radio conditions (e.g., external interferences) by using a Time Division Multiple Access (TDMA) and a slow channel hopping. They agree on a transmission schedule that guarantees enough transmission opportunities while avoiding collisions. Nodes not involved in a communication are allowed to turn off their radio to save energy. The channel hopping feature is especially considered efficient against jamming attacks since communicating nodes constantly change the radio channel used for their data exchange.

The TSCH slotframe is represented as a matrix. Each cell corresponds to a timeslot and a channel offset which is translated into the radio frequency to use. The slotframe is repeated as long as the network is running and cells are assigned to communicating devices. Cells can be either shared or dedicated. While dedicated cells are assigned to a given transmitter and its receiver(s), the shared ones rely on a contention mechanism (i.e., slotted ALOHA) to let nodes send and receive frames.

Figure 1 depicts a flow of three slotframes during which a node A would have a dedicated timeslot and channel offset to send frames to nodes B, C and D. Depending on the amount of traffic to be sent, more cells could be requested by A, either for all receivers or for a subset only. The Absolute Sequence Number (ASN) denotes the number of timeslots since the network started and X thus corresponds to the slotframe length on this example. Note that the first timeslot of each slotframe is here used for shared cells that convey control traffic (e.g., network joining).

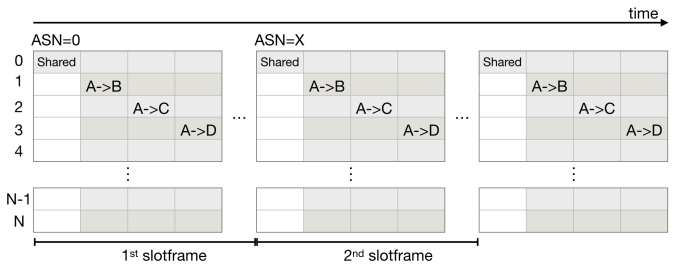


Fig. 1: Repetition of slotframes in TSCH mode of IEEE802.15.4e standard.

Most importantly, the channel offset (here ranging between 0 and N) results in a different radio frequency for each slotframe.

This frequency f is computed in a pseudo-random manner, with a formula known to all devices:

$$f = F\left[(chOffset + ASN) \ mod \ NbChannels\right] \quad (1)$$

where $chOffset$ denotes the channel offset and $NbChannels$ the number of physical channels. The $F[]$ function maps an integer with a radio frequency.

B. IETF 6tisch working group

The IEEE 802.15.4e standard does not define how to build the communication schedule. The IETF 6tisch working group does, in addition to other tasks (e.g., network formation, multi-hop topology, resource management) [25]. Several scheduling solutions have been proposed already [26]. Here, we assume periodic traffic whose bandwidth requirements can be satisfied with a basic static scheduling. Nodes in a TSCH network send Enhanced Beacons (EB) frames periodically to announce the presence of the network.

As mentioned in Section II, some security suites are available at the IEEE 802.15.4 standard, allowing to enforce the frame integrity, encryption and authentication while also providing replay protection. Key management is however not addressed. The 6tisch working group initially aimed at describing security in the join process and data-frame protection.

C. Jamming TSCH networks

A potential attacker must first determine during which timeslots its attack should take place. Indeed, constantly jamming a given channel would result in excessive energy consumption and would also increase the chances to be detected. Similarly, jamming all channels appears unrealistic thus the attacker should gain knowledge about which channel to jam for a given timeslot.

Therefore, TSCH networks are assumed to be resistant to jamming attacks by nature (i.e., time-division and channel hopping). We here propose to investigate the robustness of 6TiSCH networks against some jamming attacks. We especially focus on jamming attacks performed by an attacker whose knowledge ranges from blind to complete view of the neighborhood's timeslots and channels used, thanks to some observation.

More specifically, Table I summarizes the investigated scenarios. The knowledge of an attacker refers to its understanding of the network upon aiming at the target.

TABLE I: Target of selective jamming attacks, required knowledge and potential impact.

Target	Knowledge	Impact
Shared cells	Used channel and slotframe length	Jeopardized receptions of EBs, endangered network formation and maintenance
Dedicated cells	Timeslot and channel used by the victim(s)	Jeopardized receptions of data frames, endangered data communications

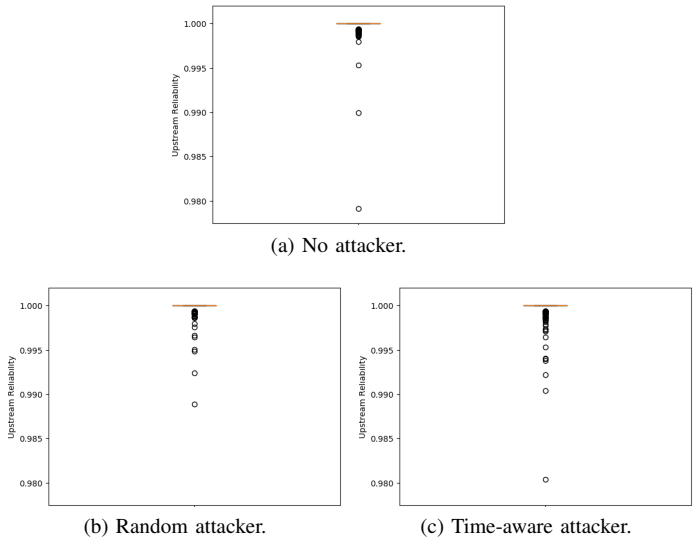
In this paper, we assume three scenarios, depending on the knowledge of the attacker:

- The **random** scenario relates to a blind attacker jamming over random channels at random timeslots. In such case, an attacker should select a jamming duration (the amount of jammed timeslots depends on the transparency targeted by the attacker) and one channel (out of 16 by default);
- The **time-aware** scenario involves an attacker able to anticipate the timeslots used by its victim(s) in each slotframe. The radio channel remains unknown however leaving the attacker with a $\frac{1}{NbChannels}$ probability to select the right one;
- the **fully-aware** scenario refers to a time-aware attacker who would have also computed the channel hopping sequence for the upcoming slotframes. To do so, an attacker should know $NbChannels$ and listen for a given channel during multiple slotframes. It then identifies the timeslots during which this channel is used and can easily anticipate the upcoming communications by using Equation 1.

IV. Performance Evaluation

Our objective is to evaluate the impact of random, time-aware and fully-aware jamming against networks relying on both time division and channel hopping mechanisms. We aim at quantifying the potential of jamming attacks described in Section III, depending on the knowledge of the attacker. More specifically, we wonder to what extent random and time-aware scenarios would endanger the network performances (e.g., latency, reliability) and how fully-aware jamming could be set. Unlike previous studies that have focused on synchronized protocols only (e.g., [19], [14]), we also considered the channel hopping feature which should normally lead to more *secure by design* medium access control protocols.

Our simulation campaign relies on the 6TiSCH Simulator [27] that was created as part of the standardization activity, and which has been used extensively by the `6tisch` working group. Our simulation study involves two nodes trying to communicate while an attacker is trying to selectively jam the radio channels. Our simulation setup consists in runs of 1000 slotframes each, which serve a periodic application sending a 90 byte packet per second. We use the Minimal Scheduling Function (MSF). The slotframe consists in 101 timeslots whose duration is 15 ms each. The packet delivery ratio over the link varies over time, in accordance with the Pister-Hack model [27]. Considered wireless links are configured with a minimum packet delivery ratio of 0.95, in order to clearly identify the impact of attacks. A total of 16 channels are assumed, in accordance with the standard. In this study, we evaluated the jamming over dedicated cells (i.e., data traffic). Our objective is indeed to clearly observe the impact of such attacks only, i.e., without adding cascade effects with interferences over control messages.



(a) No attacker.

(b) Random attacker.

(c) Time-aware attacker.

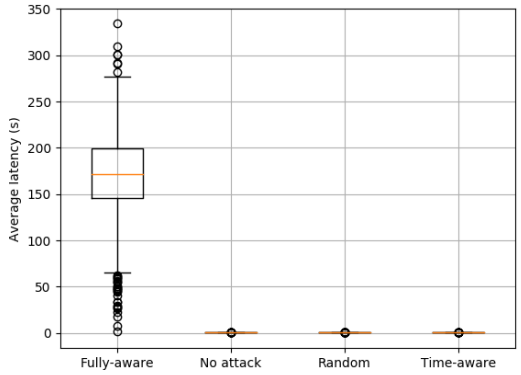
Fig. 2: Upstream reliability as impacted by random and time-aware attackers.

The presented results include the **random** and **fully-aware** scenarios. In the former, we implement an attacker which jams for a limited duration, up to half of a slotframe, thus reaching a success probability of $\frac{1 \times jammingDuration}{NbChannels \times slotframelength}$ In the latter, we assume an attacker having gained knowledge of both the used timeslots and the used channel sequence. Some uncertainty is yet added to mimic both the duration of the learning process and its potential failure.

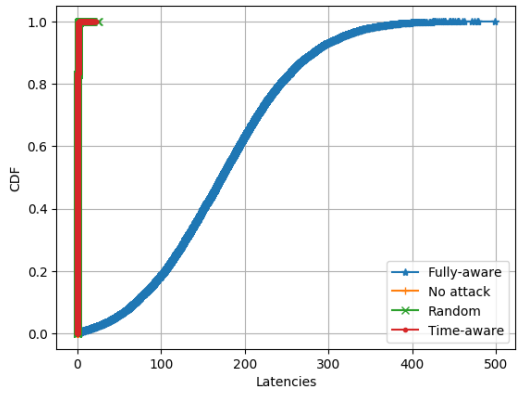
Please note that some of the following figures represent boxplots which depict the distributions of the values, and the minimum and maximum values. Each box represents the median value, the first and third quartiles for all these measurements, while the whiskers here represent the minimum and maximum values. Dots indicate the outliers, i.e., the measures distant from other observations.

Figure 2 depicts the upstream reliability observed for the communication between the two simulated nodes. Upstream reliability is thus a simple computed ratio of the number received packets over the number of packets initiated by the sender. The reliability illustrated in Figure 2a is obtained when no attack is run. Thus, losses are here due to the random variation of link quality only. Figure 2b shows that a random attacker rarely succeeds in affecting the communication reliability. As identifying the timeslots used by a pair of nodes could be done with a simple monitoring, it is also interesting to note that a time-aware attacker is not impacting the communication quality either, as illustrated by Figure 2c. This confirms the efficiency of channel hopping to protect nodes against jamming attacks.

When a jamming attack succeeds, the packet is not received and thus no acknowledgement is sent, leading to a retransmission. Note that a retransmission also happens if the

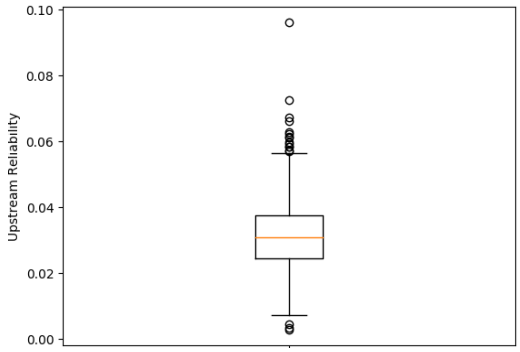


(a) Average latencies.

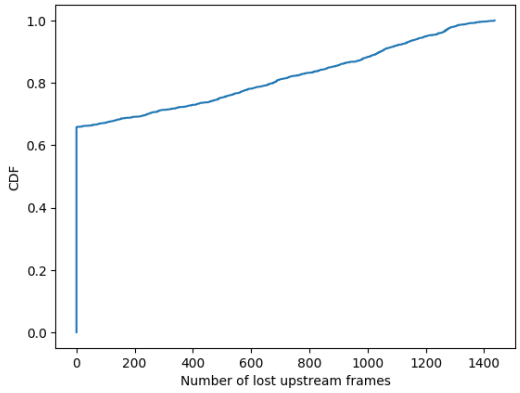


(b) CDF of induced latencies.

Fig. 3: Comparison of latencies (s) induced by considered scenarios.

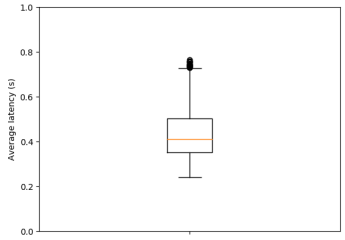


(a) Average upstream reliability.

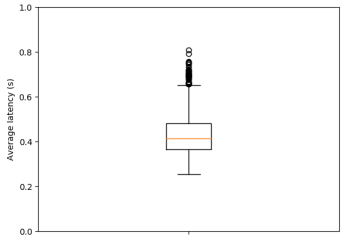


(b) CDF of lost upstream packets.

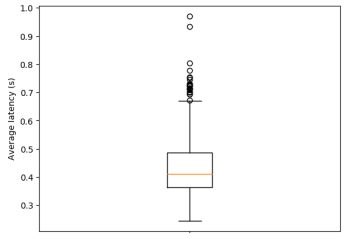
Fig. 5: Upstream reliability achieved with a fully-aware attacker.



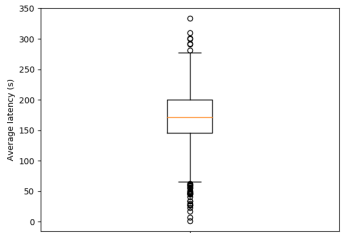
(a) No attack.



(b) Random attacker.



(c) Time-aware attacker.



(d) Fully-aware attacker.

Fig. 4: Average latency (s) induced in considered scenarios.

expected acknowledgement is sent after a successful reception but not received by the sender, due to jamming. Although the latency is logically increased by time-aware and random attackers, the results shown on Figures 3a and 3b confirm the robustness of TSCH networks against random and time-aware jamming as anticipated in Section III-C. Figure 4 further details the average latencies induced on the communication link, for each considered scenario. Here, neither a random nor a time-aware attacker induces a large amount of retransmissions, as confirmed by the comparison between average latencies achieved under no attack (Figure 4a) and induced by random or time-aware attackers (Figures 4b and 4c).

However, once full knowledge is gained, the attack is highly efficient and dramatically impacts the communication reliability (see Figure 5a). In all simulation runs, the very few upstream packets that reach their destination occurred during the slotframes where no jamming was happening, due to the channel hopping sequence yet unknown to the attacker. Figure 5b depicts the cumulative distribution function (CDF) of lost upstream packets. We can observe that numerous losses occur, thus imposing costly retransmissions from the sender. Consequently, the communication latency over the link increases accordingly, as already shown on Figures 3 and 4d.

V. Conclusion and Perspectives

In this paper, we gave an overview of existing denial-of-sleep attacks in the Internet of Things. After detailing some proposed solutions, we focused on the vulnerability of time-synchronized and channel hopping networks where attackers can jam the channels either randomly or selectively by learning from their neighborhood's activity (i.e., timeslots and channels used). Unlike previous studies that have focused on synchronized protocols only, we considered the channel hopping feature

which should normally lead to more *secure by design* medium access control protocols. We illustrated the anticipated impact by using 6TiSCH networks as a target example. After confirming that those networks are more robust to basic jamming (i.e., random, time-aware) by nature (i.e., due to channel hopping), we exhibited the impact of smarter attackers, able to perform selective jamming after anticipating the channel hopping sequence. We confirmed that the latency and reliability achieved under such attacks would totally prevent the network from completing its monitoring task.

We are planning to extend this preliminary study over larger and denser topologies where random jamming is expected to be more efficient. We also aim at investigating the detection and mitigation of such denial-of-sleep attacks. Several approaches already exist to avoid channels of low quality and could thus be used to detect ongoing jamming. Regarding mitigation, some solutions are being proposed in the frame of `6tisch` standardization activities [28]. We then plan to study other Denial-of-Sleep attacks (e.g., replay) over such networks. Especially, the injection of fake control information at MAC or routing layers may dramatically impact those systems.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010.

[2] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, "The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 101–120, First 2013.

[3] T. Watteyne, A. Mehta, and K. Pister, "Reliability through frequency diversity: why channel hopping makes sense," in *PE-WASUN*, (Tenerife, Spain), pp. 116–123, ACM, 2009.

[4] J. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, and Q. Yan, "Industrial Internet: A Survey on the Enabling Technologies, Applications, and Challenges," *IEEE Communications Surveys Tutorials*, vol. 19, pp. 1504–1526, thirdquarter 2017.

[5] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: A survey of existing protocols and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, pp. 1294–1312, thirdquarter 2015.

[6] A. Merlo, M. Migliardi, and L. Caviglione, "A survey on energy-aware security mechanisms," *Pervasive and Mobile Computing*, vol. 24, pp. 77 – 90, 2015. Special Issue on Secure Ubiquitous Computing.

[7] S. Naik and N. Shekokar, "Conservation of Energy in Wireless Sensor Network by Preventing Denial of Sleep Attack," *Procedia Computer Science*, vol. 45, pp. 370 – 379, 2015. International Conference on Advanced Computing Technologies and Applications (ICACTA).

[8] G. Mahalakshmi and P. Subathra, "A Survey on Prevention Approaches for Denial of Sleep Attacks in Wireless Networks," *Journal of Emerging Technologies in Web Intelligence*, vol. 6, pp. 106–110, Feb 2014.

[9] E. Gelenbe and Y. M. Kadioglu, "Energy Life-Time of Wireless Nodes with Network Attacks and Mitigation," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, May 2018.

[10] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver, "Low energy operation in WSNs: A survey of preamble sampling MAC protocols," *Computer Networks*, vol. 55, no. 15, pp. 3351–3363, 2011.

[11] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," *Swedish Institute of Computer Science, Technical Report*, 2011.

[12] K.-F. Krentz, C. Meinel, and H. Graupner, "Countering Three Denial-of-Sleep Attacks on ContikiMAC," in *International Conference on Embedded Wireless Systems and Networks (EWSN)*, pp. 108–119, 2017.

[13] V. Manju and M. Sasikumar, "Mitigation of Replay Attack In Wireless Sensor Network," *International Journal on Information Technology*, vol. 5, 2014.

[14] K.-F. Krentz and C. Meinel, "Denial-of-sleep defenses for IEEE 802.15.4 coordinated sampled listening (CSL)," *Computer Networks*, vol. 148, pp. 60 – 71, January 2019.

[15] L. Gu and J. A. Stankovic, "Radio-triggered wake-up for wireless sensor networks," *International Journal of Time-Critical Computing Systems (Real-Time Systems)*, vol. 29, no. 2, pp. 157–182, 2005.

[16] A. T. Capossele, V. Cervo, C. Petrioli, and D. Spenza, "Counteracting Denial-of-Sleep Attacks in Wake-Up-Radio-Based Sensing Systems," in *13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–9, June 2016.

[17] M. Jo, L. Han, N. D. Tan, and H. P. In, "A survey: energy exhausting attacks in MAC protocols in WBANs," *Telecommunication Systems*, vol. 58, pp. 153–164, Feb 2015.

[18] D. R. Raymond, R. C. Marchany, M. I. Brownfield, and S. F. Midkiff, "Effects of Denial-of-Sleep Attacks on Wireless Sensor Network MAC Protocols," *IEEE Transactions on Vehicular Technology*, vol. 58, pp. 367–380, Jan 2009.

[19] O. Dagdeviren, R. Sokullu, and I. Korkmaz, "GTS Attack : An IEEE 802.15.4 MAC Layer Attack in Wireless Sensor Networks," *International Journal On Advances in Internet Technology*, vol. 2, no. 1, 2009.

[20] Y. Xiao, H.-H. Chen, B. Sun, R. Wang, and S. Sethi, "MAC Security and Security Overhead Analysis in the IEEE 802.15.4 Wireless Sensor Networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, p. 093830, Oct 2006.

[21] X. Cao, D. M. Shila, Y. Cheng, Z. Yang, Y. Zhou, and J. Chen, "Ghost-in-zigbee: Energy depletion attack on zigbee-based wireless networks," *IEEE Internet of Things Journal*, vol. 3, pp. 816–829, Oct 2016.

[22] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, (Urbana-Champaign, IL, USA), pp. 46–57, 2005.

[23] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, pp. 1567–1576, 2002.

[24] Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011), "Low-Rate Wireless Personal Area Networks (LR-WPANs)," tech. rep., IEEE, April 2016.

[25] P. Thubert and T. Watteyne, "6TiSCH: IPv6 over the TSCH mode of IEEE 802.15.4e." IETF working group. https://datatracker.ietf.org/wg/6tisch/.

[26] R. T. Hermeto, A. Gallais, and F. Theoleyre, "Scheduling for IEEE802.15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey," *Elsevier Computer Communications*, vol. 114, pp. 84 – 105, 2017.

[27] E. Municio, G. Daneels, M. Vucinic, S. Latre, J. Famaey, Y. Tanaka, K. Brun, K. Muraoka, X. Vilajosana, and T. Watteyne, "Simulating 6TiSCH Networks," *Wiley Transactions on Emerging Telecommunications (ETT)*, 2018.

[28] M. Tiloca, S. Duquennoy, and G. Dini, "Robust Scheduling against Selective Jamming in 6TiSCH Networks," draft, IETF, December 2018. draft-tiloca-6tisch-robust-scheduling-01.