Experimental in-depth study of the dynamics of an indoor industrial low power lossy network

Rodrigo Teles Hermeto, Antoine Gallais, Fabrice Theoleyre

To cite this version:

Rodrigo Teles Hermeto, Antoine Gallais, Fabrice Theoleyre. Experimental in-depth study of the dynamics of an indoor industrial low power lossy network. Ad Hoc Networks, Elsevier, 2019, 93, pp.101914. 10.1016/j.adhoc.2019.101914 . hal-02323117

Experimental in-depth Study of the Dynamics of an Indoor Industrial Low Power Lossy Network

Rodrigo Teles Hermeto[a], Antoine Gallais[a,b], Fabrice Theoleyre[a]

[a] *ICube Laboratory, CNRS / University of Strasbourg, Pole API, Boulevard Sebastien Brant, 67412 Illkirch Cedex, France*
[b] *FUN - Self-organizing Future Ubiquitous Network, Inria Lille - Nord Europe, avenue Halley, Villeneuve d'Ascq, France*

Abstract

An increasing number of industrial applications rely on low power embedded devices because of their flexibility. To work properly, the network has to respect requirements concerning specifically the delay and the reliability. Fortunately, low power, and slow channel hopping MAC help to cope with these requirements. For instance, IEEE802.15.4-TSCH relies on a strict schedule of the transmissions, spread over orthogonal radio channels, to set-up a resilient wireless infrastructure. A routing protocol (e.g. RPL) has then to construct energy-efficient routes on top of this link-layer topology. Unfortunately, the radio environment keeps on exhibiting time-varying characteristics, due to e.g. obstacles, and external interference. In a reservation-based stack, the network will have to implement over-provisioning, to cope with small-term variations: additional resources allow the network to operate in the worst situation. Inversely, long-term changes are triggered only when a node/link failure is detected. In this paper, we investigate experimentally the performance stability of a 6TiSCH/ IETF/ RPL stack in collocated deployments. We focus on some key metrics to exhibit the intermittent losses of guarantees (e.g. delivery ratio) under yet static conditions. Our results in large scale testbeds highlight that in the presence of radio oscillations, 6TiSCH introduces frequent network reconfigurations to combat interference and provide high reliability. We perform a multi-layer analysis of the 6TiSCH stack identifying the main sources of instability and proposing solutions to address each one of them. Our performance evaluation highlights the accuracy of our solutions to set up an efficient and reliable network.

Keywords: Industrial Internet of Things, IEEE802.15.4-TSCH, dynamics, stability, routing, link-layer

---

1. Introduction

Industrial networks have been widely deployed for a myriad of utilizations. In smart factories, a large collection of sensors and actuators are connected to programmable logic controllers to take real-time decisions [1]. These industrial networks are also widely used for other general public applications to provide novel high added value services. For instance, a collection of sensors and actuators can detect intrusions, or control the Heating, Ventilation and Air-Conditioning system in building automation [2].

To reduce the deployment costs while maximizing the flexibility, industrial networks start to become wireless [3]. However, radio networks are known to be very challenging since they exhibit time-variable

---

Email addresses: `teleshermeto@unistra.fr` (Rodrigo Teles Hermeto), `antoine.gallais@inria.fr` (Antoine Gallais), `theoleyre@unistra.fr` (Fabrice Theoleyre)

Preprint submitted to Ad Hoc Networks *August 29, 2019*

behaviors, because of obstacles and external interference. Nevertheless, the network must still forward efficiently the packets. For instance, high reliability (e.g. $> 99\%$) and an upper bounded delay are now a requirement for most industrial applications.

Fortunately, several wireless standards have been proposed to make the transmissions more reliable and energy efficient [4]. They rely on a strict schedule of the transmissions, combined with slow frequency hopping to improve the network performance. Multichannel environments are known to mitigate multipath fading [5]. These protocols are often deterministic to be able to provide stable performance, and to make the diagnostic easier when a failure arises.

Typically, IEEE 802.15.4-TSCH relies on a slotframe of *cells*, defined by a pair of timeslot and channel offset [6]. A pair of nodes can negotiate a set of dedicated cells for its own usage. Since this slotframe is repeated indefinitely, these cells denote an amount of reserved bandwidth for this specific link. Over-provisioning helps to cope with unreliable environments: additional cells are reserved to possibly retransmit the data packets, with a stop-and-wait ARQ approach.

Slow frequency hopping helps to combat external interference: when a transmission fails, the packet is retransmitted through a different radio channel, to make the packet losses less repetitive [7]. However, collocated wireless networks using the same ISM band (e.g. Wi-Fi networks) may still damage the reliability [8]. Moreover, external interference exhibits time-variable characteristics [9]. This implies that the network must handle efficiently these variations: the network has to provision enough resources to handle bursts of packet losses.

In multihop low power lossy networks, routes have to be constructed to forward the packets to the destination. The RPL routing protocol [10] constructs dynamically routes when the network exhibits a convergecast traffic pattern. Each node maintains a preferred parent toward the border router to have an upward route. Typically, a trickle timer aims to decrease the volume of control traffic when the network topology is stable. Unfortunately, RPL has been proved to overreact to link quality changes [11]. Thus, dynamic routes may be a source of instability.

The 6TiSCH IETF working group currently defines a set of protocols to execute IPv6 in IEEE 802.15.4-TSCH networks. In particular, it defines the mechanisms to change the schedule on the fly, with the 6P protocol [12]. 6TiSCH also relies on a Scheduling Function to decide how many cells to reserve for each radio link. Since cells are allocated on-the-fly, time-variant conditions may imply oscillations, increasing the control traffic.

In this work, we focus on the performance stability. We focus particularly on long-term deployments, where the network should reach a the steady-state with all nodes synchronized and able to communicate. The radio environment is time-variant, because of e.g. multipath fading, external interference, mobile obstacles. However, we argue that the network should keep on providing a deterministic behavior. Much work has been done to improve the performance with opportunistic mechanisms to take benefit from the instantaneous (best) conditions [13]. To our mind, trying to reach the optimal configuration at any instant is irrelevant if the conditions are time-variant. Indeed, the network has to deal with the worst-case, without triggering continual reconfigurations. The cost of these reconfigurations would exceed the price of operating under sub-optimal configurations.

Our study relies on experiments on FIT IoT-Lab large-scale testbed to assess the performance of the different protocols in realistic conditions. The contributions of this paper are as follows:

1. we present a preliminary study to highlight the presence of instability: the schedule and the routes keep on changing continuously, even for long experiments;
2. we identify several causes of instability in the 6TiSCH stack (collisions for control packets, link quality mis-estimation, negotiation of cells through unreliable links, routing oscillations), and quantify them in an indoor environment;
3. we compare the performance achieved through two different testbeds to highlight the persistence of this instability problem;
4. we propose per-layer solutions to solve this instability, and to make the network performance stable, without continuous reconfigurations.

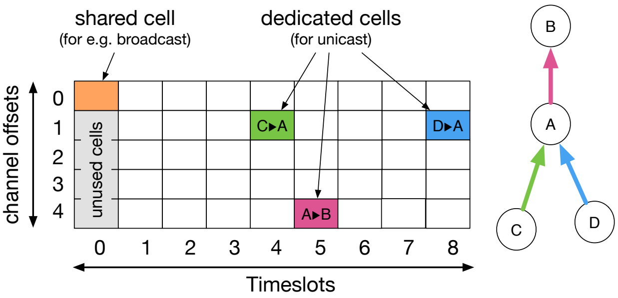


Figure 1: An example of TSCH scheduling for node D. A → D stands for *'A transmits to D'*, while the shared cell is used for broadcast or control frames.

2. Background

We detail here the key mechanisms of the 6TiSCH/IEEE 802.15.4-TSCH stack, to understand more finely how the network bootstraps and maintains its configuration (routes and schedule) to provide high reliability.

2.1. IEEE 802.15.4-TSCH

The IEEE 802.15.4e amendment (2012) introduced the Time-Slotted Channel Hopping (TSCH) mode and was further rolled into the IEEE 802.15.4-2015 standard [6], where nodes follow a collision-free schedule to transmit their packets without collisions. In addition, the standard proposes the use channel-hopping, so that subsequent packets are transmitted over different frequencies to improve the reliability.

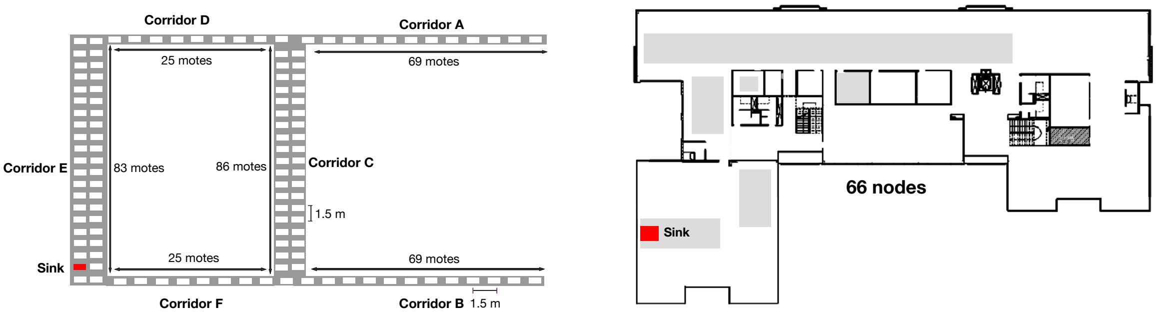
A slotframe in TSCH consists of a matrix of *cells* of equal length, each cell being defined by a pair of *timeslot* and *channel* offsets. A cell may be either *dedicated* or *shared*. Dedicated cells should be assigned to a group of non-interfering radio links. The transmitter does not implement in that case any contention resolution algorithm since it considers it has a *reserved* access. The transmitter may trigger a CCA before transmitting its frame to promote coexistence with other users. However, a CCA cannot handle intra-network interference since the boundaries are closely aligned because of synchronization. Thus, interfering links should never use the same dedicated cell.

On the contrary, shared cells are assigned to a group of possibly interfering transmitters. When a transmitter has a packet in its queue at the beginning of a shared cell, it transmits the packet after a fixed offset, and an optional CCA. If an acknowledgement was required but not received, the transmitter considers a collision occurred. In that case, it selects a random backoff value, and *skips* the corresponding number of shared cells. Since everything is fixed in the timeslot, a shared cell can also be used for synchronization. However, contention resolution is not triggered inside a shared cell, but among different shared cells, decreasing its efficiency.

Each timeslot is labeled with an Absolute Sequence Number (ASN), a counter of the number of timeslots elapsed since the network was established. A node computes the actual physical frequency to use at the beginning of each slot. This frequency is derived from the channel offset and the ASN, such that a cell possibly corresponds to different frequencies in consecutive slotframes.

Many distributed and centralized scheduling algorithms have been proposed so far for TSCH networks [14]. In centralized approaches, a controller has a complete knowledge of the network (topology, traffic) to construct the schedule. Inversely, distributed approaches allow each node to reactively reserve as many cells as required.

Let us consider the TSCH schedule illustrated in Figure 1, with a slotframe of 9 timeslots and 5 channel offsets. The broadcast packets can be transmitted safely during the shared cell: all the nodes have to stay awake during such timeslot. Thus, a single transmission *covers* all the radio neighbors, if we exclude physical errors. The data packets use rather the dedicated cells, and thus are protected against internal collisions. In this schedule, one transmission opportunity is reserved for each radio link.



(a) Grenoble testbed

(b) Lille testbed (2nd floor)

Figure 2: The testbeds used in this study. The sinks were the only fixed nodes in all experiments.

2.2. 6TiSCH overview

The 6TiSCH IETF working group has defined protocols to execute IPv6 (i.e. 6LoWPAN) on top of a reservation based MAC layer (i.e. TSCH). 6TiSCH makes a distinction between the protocol which defines how to negotiate the cells (6P) and the algorithm which decides how many cells have to be allocated in the schedule (i.e. the Scheduling Function).

The 6top Protocol (6P) defines how to allocate/deallocate cells with a neighbor node [12]. Each schedule modification is based by default on a two-way handshake, where the inquirer sends a request to a receiver (e.g. preferred parent) node and waits for its confirmation. Optionally, a three-way handshake can be implemented when the child node lets its parent propose a list of cells. Typically, 6P packets sends its requests through the shared cells since two nodes have no common preallocated dedicated cells to bootstrap a negotiation.

Both sender and receiver detect schedule inconsistencies by incrementing a sequence number that counts the number of modifications made in their respective schedule since the beginning. This sequence number is embedded in every 6P packet, so that the Scheduling Function is able to detect and to handle inconsistencies. For instance, SFx [15] triggers a 6P clear command to reset both schedules when 6P sequence numbers differ between the transmitter and the receiver.

3. Problem statement

In typical industrial deployments, it is not rare to find different wireless technologies cohabiting in the same environment [16]. Because of its low power nature, a 6TiSCH network is particularly in disadvantage when sharing the same unlicensed band with other higher power networks such as Wi-Fi. The interference caused by Wi-Fi devices may increase the packet losses of IEEE 802.15.4 network up to 58% [17]. Additionally, obstacles like concrete walls and heavy metal machines make the wireless channel non-stationary in long-term [18].

Iova *et al.* [19] already demonstrated that RPL changes routes frequently even with a static topology. Indeed, RPL tries to identify efficient routes by exploiting a link quality metric (e.g reliability). Unfortunately, this quality is usually a stochastic variable and can only be estimated with a certain inaccuracy. However, a topology reconfiguration does not come for free and it is usually followed by a burst of control packets transmissions. In particular, 6TiSCH is reservation-based and cells have to be negotiated all along the path toward the sink. Worse, the number of dropped packets tends to increase since reservations take time, generating buffer overflows [20]. Therefore, we need efficient protocols that ensure reconfigurations only when really needed, e.g. sudden fault, definitive link quality changes.

3.1. Definition

Industrial networks are deterministic to provide strict guarantees. It is particularly hard to respect a set of guarantees if the protocols keep on oscillating. Making a diagnostic when everything is dynamic is also particularly challenging. More precisely, we adopt in this paper the following definition:

> A network infrastructure is stable if the state of its protocols remains unchanged for small-term variations (e.g. static topology with sporadic external interference).

In other words, this means that a network has to be reconfigured when a long-term variation is detected. For instance, the routing protocol must change the route when the next hop runs out of energy, but has to keep the same next hop for non significant link quality variations. The optimality has a cost, since the protocols have to exchange many control packets before re-converging eventually to a steady state. Even worse, the network may keep on oscillating, never converging. We argue that this reconfiguration cost exceeds the additional resources to cope with short-term variations.

3.2. Preliminary Study

To illustrate this problem, we investigate the performance stability of 6TiSCH networks when deployed in indoor environments. We use two different testbeds for this study, FIT IoT-LAB in Grenoble and in Lille (Figure 2). In Grenoble the nodes are placed in long corridors without fixed physical obstacles among them. On the other hand, in Lille the nodes are distributed across different corridors separated by walls. These two testbeds correspond to different scenarios with different channel characteristics. Since the testbeds are deployed indoor in a non-dedicated building, the nodes are subject to external interference originated from others wireless devices, such as Wi-Fi access points and users, and concurrent IEEE 802.15.4 networks.

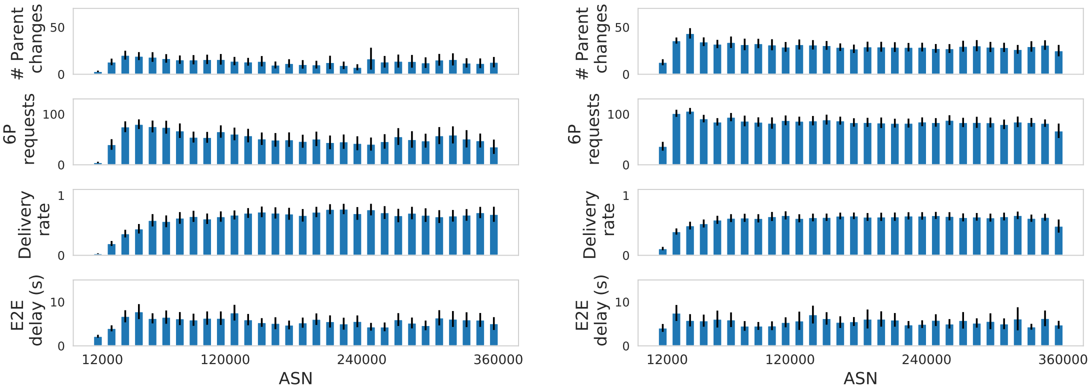
We select 31 nodes, and we perform 30 repetitions of 90 minutes in both testbeds. The sinks are the same in all experiments and we place them at the extremity of the corridors. The other nodes are selected randomly before launching each experiment, providing a different topology for each repetition and increasing the representativity of our experiments. Additionally, we adjust the transmission power in order to have multi-hop networks up to 6 hops. We target here industrial applications where nodes report their data to the sink with a Constant Bit Rate (CBR) traffic (10 seconds). Thus, everything is here static (i.e. topology, traffic), and we expect a convergence to a steady-state.

We employ M3 nodes, based on a STMicroelectronics 32-bit ARM Cortex-M3 micro-controller that embeds an AT86RF231 radio chip, providing an IEEE 802.15.4 compliant PHY layer. Moreover, we execute OpenWSN[1], that provides an open-source implementation of the 6TiSCH stack (i.e. IEEE 802.15.4-TSCH, 6P, SFx, 6LoWPAN and RPL). We set a slotframe composed of 199 slots, where 8 were shared slots restricted to broadcast and 6P control packets. We use the default implementation of OpenWSN. In particular, the timeslot duration is fixed to 15 ms, and no CCA is triggered before a transmission (i.e. OpenWSN does not try to reduce the collisions with other wireless technologies using the same unlicensed band).

Figure 3 illustrates (i) the number of parent changes, (ii) the number of 6P requests, (iii) the packet delivery ratio and (iv) the end-to-end delay. Surprisingly, we keep on identifying continuously routing reconfigurations (i.e. parent changes) during all experiments in both testbeds, highlighting that the instability problem is not restricted to a single testbed. Unfortunately, routing reconfigurations introduce a burst of 6P request packets while nodes reserve dedicated slots. In multihop networks, this problem becomes even more severe since the cells have to be re-negotiated all along the path (i.e. exactly until a common ancestor is reached if it did not yet release the corresponding resources). Worse, these reconfigurations deeply impact the reliability: the packet delivery ratio drops significantly when a burst of RPL reconfigurations occurs. Indeed, the re-convergence takes time as reserving new cells is not instantaneous. Packets are thus dropped because of buffer overflows. Additionally, the frequent routing reconfigurations impacts directly the delay, since the routing protocol may build not optimized paths while the network re-converges.

---

[1] `http://openwsn.org/`



(a) Grenoble

(b) Lille

Figure 3: The network performance in both testbeds for 30 experiments repetitions (95% of confidence level).

3.3. Impact of network instability

Because we expect a wide adoption of the Internet of Things, all the applications will have to share the same unlicensed band, creating a high volume of cross technologies interference, possibly with different transmission power. For instance, the transmission power of IEEE 802.11 devices (30 dBm) is higher than IEEE 802.15.4 devices (0 dBm) and this asymmetry impacts directly the performance and the reliability of IEEE 802.15.4 networks [21]. Thus, as highlighted previously, the networks will rely on dynamic algorithms, with continuous reconfigurations even under unchanged network conditions (same nodes location, same volume of traffic), in order to combat narrow-band noise and to provide strict guarantees. The multichannel feature of IEEE 802.15.4-TSCH should rather allow the network to limit the number of reconfigurations to provide stable performances.

Typically, a node in a 6TiSCH network changes its preferred parent when another neighbor provides a significantly smaller rank. The challenge is to not be too aggressive when estimating the link quality, which may generate routing instability caused by frequent parent changes. Conversely, being too conservative leads to a more stable topology, but possibly performing bad. Overprovisioning may be used to deal with poor links, at the cost of spending more energy to reach the reliability threshold.

On the other hand, network reconfigurations create burst of 6P requests to modify the schedule. For distributed schedule functions, the nodes decide autonomously the cells to use, before detecting and solving collisions. Thus, changes in the schedule also increase the probability of collisions, thus impacting even flows which have not been rescheduled. Additionally, before allocating the bandwidth, a node relies only on shared cells to send its reservation packets to the new parent, where collisions are frequent. As a result, the re-convergence may be deferred while the nodes try to communicate in a best-effort way.

4. Layer 2: collision mitigation in shared cells

MSF [22] advocates to use shared cells for the initial 6P negotiation since no dedicated cell is a priori available. However, broadcast packets such as Enhanced Beacon (EB) and DODAG Information Object (DIO) use already extensively the shared cell. Thus, the collision probability may be quite large.

During moments of high routing instability, several nodes may change their preferred parents, increasing even more the contention in the shared cells. In particular, during the bootstrap phase, nodes probe their neighbors iteratively until they find a suitable preferred parent.

In this section, we discuss approaches to mitigate the collision problem in the shared cells. We demonstrate that a simple re-arrangement of shared cells in the slotframe is enough to reduce the overhead caused by high contention. We do not consider the discovery time for novel nodes, which have to scan all the channels before receiving an EB to get synchronized [23]. The discovery is triggered once, before a node gets synchronized, and is thus less prejudicial to the stability.

4.1. Repeated collisions

In a 6TiSCH network, all nodes broadcast EB and DIO packets periodically. An EB contains crucial information (e.g. slotframe length, hopping sequence) that allows other nodes to synchronize with the sink. However, the IEEE 802.15.4-TSCH standard does not specify the rate at which the EBs have to be transmitted. Similarly, DIO are used to disseminate routing information (rank) over the network. Typically, a trickle timer aims to decrease the volume of DIO when the network topology is stable.

A node starts to send broadcast packets as soon as it joins the network, according to a time interval common to all nodes. Because initially nodes that joined the network at the same time have the same frequency of transmission, their broadcast packets collide repeatedly. Obviously, these collisions impact very negatively the network formation time: a node has to wait longer to receive a correct EB, so that it can be synchronized with the TSCH network.

Using a jitter before broadcasting control packets is an alternative to avoid repeated collisions. Typically, the jitter increases the time window in which a node enqueues broadcast packets. For instance, when the broadcast period is l and the jitter is γ, the time when the next broadcast packet will be enqueued will be randomly selected within the interval $[l-\gamma, l+\gamma]$. Therefore, collisions are less repetitive, since the nodes have now the possibility to transmit broadcast packet at different moments.

4.2. Distributed shared cells

Even when enqueuing broadcast packets randomly, the probability of collision remains high when the shared cells are placed at the beginning of the slotframe. To illustrate this problem, let us consider a slotframe that has only one shared cell placed at the first position of the slotframe. For simplicity, let us consider that a shared cell is only used for transmitting EBs. Additionally, all nodes send EBs periodically every ε seconds. To avoid repeated collisions, we consider a small jitter γ in a way that a node selects its next transmission time by randomly choosing a value between $[\varepsilon-\gamma, \varepsilon+\gamma]$. In this scenario, we denote as Δ_{shared} the time between two consecutive shared cells. The total number of Δ_{shared} repetitions within the interval $[\varepsilon-\gamma, \varepsilon+\gamma]$ is given by the following equation:

$$K = \left\lfloor \frac{2\gamma}{\Delta_{shared}} \right\rfloor \tag{1}$$

The probability of repetitive collisions can be calculated as the probability of at least two nodes reselecting the same Δ_{shared} repetition to enqueue their control packets. In such occurrence, the transmitters will try to send their packets using the next shared cell. The probability of collision is then given by:

$$p[collision] = 1 - \frac{K!}{K^n * (K-n)!} \tag{2}$$

where n is the number of neighboring nodes.

According to Equation 2, the probability of collision for a network sending control packets in a range of 10 seconds, having 101 timeslots, each lasting 10 ms, and having 6 neighbors is approximately 85%. Adding more shared cells to the beginning of the slotframe does not reduce the probability of collision. The reason is that nodes can still enqueue their control packets during the same Δ_{shared} repetition before the next slotframe repetition.

To reduce the probability of collision, we need to consider a larger jitter or to shrink Δ_{shared}. However, in dense networks we would need a large γ to have enough Δ_{shared} repetitions to accommodate a high number of neighbors with low probability of collision. Consequently, the network would be less reactive, as the rate of control packets being transmitted would be lower. On the other hand, by distributing shared cells uniformly over a slotframe, we shrink Δ_{shared}, forcing the nodes to pass through a shared cell more often. In that case, we can modify slightly Equation 1 to:

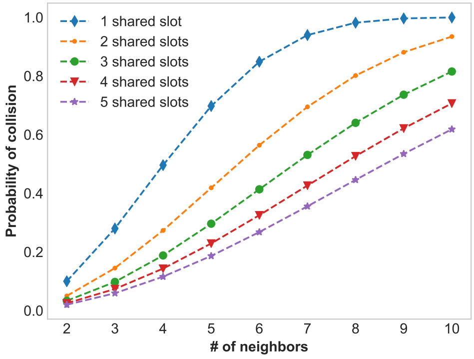


Figure 4: Probability of collision for different number of neighbors and shared cells.

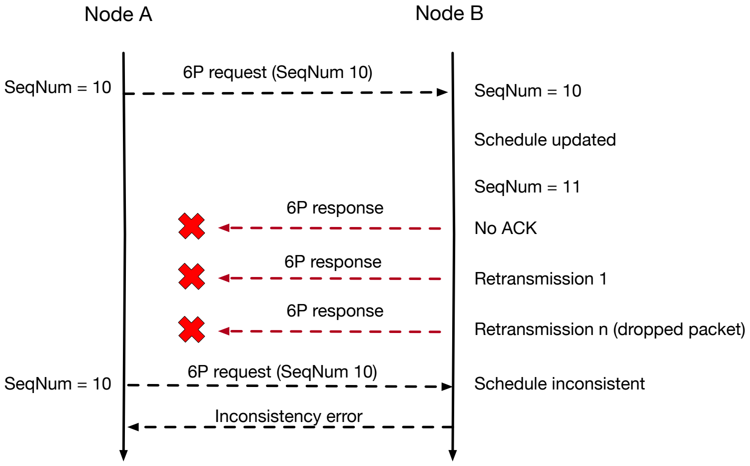


Figure 5: An example where two neighbors have inconsistent schedules.

$$K' = \left(\frac{2\gamma}{\Delta_{shared}}\right) * Total_{shared} \tag{3}$$

where $Total_{shared}$ is the number of shared cells placed distributively over the slotframe.

Figure 4 shows the probability of collision considering different number of shared cells. For a single shared cell case, the probability of collision yields 50% when the number of neighbors is only 4. When we increase the number of distributed shared cells, the probability of collision goes down, as the nodes use different Δ_{shared} repetitions to queue EB. For 5 shared slots, the probability of collision yields approximately 60% considering all 10 nodes. These results highlight that the probability of collision in the shared cells depends on the number and the position of shared cells.

5. Layer 2.5: schedule inconsistency management

Because radio environments are known to be lossy, 6P relies on mechanisms to preserve schedule consistency between a node and its parent. Typically, a 6P sequence number is maintained per link, incremented

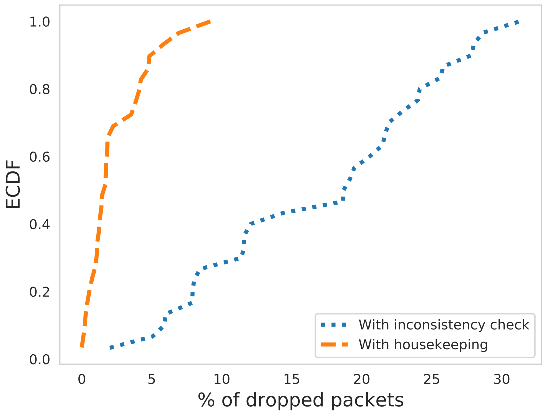


Figure 6: Frequency of dropped packets by experiment caused by buffer overflow after a 6P-clear command in Grenoble.

after every modification triggered by any of the two corresponding nodes. In particular, their schedules are considered inconsistent if the two nodes have a different 6P sequence number. Unfortunately, this may arise when a 6P transaction exceeds the maximum number of retransmissions. The 6P packet is dropped, and both sides of the link have then a different sequence number.

Figure 5 shows a typical scenario to illustrate this problem. When the node B receives the request from node A, B will update its schedule and will return a response back to A. Thus, B increments its 6P sequence number. However, if A never receives the reply from B then it has to use the previous sequence number: its next request has a lower sequence number, which triggers a schedule inconsistency management.

SFx advocates that when two neighbors have inconsistent schedules, they have to flush all their cells with these neighbors after a `6P-clear` command [15]. Then, the two nodes have to restart the reservations from scratch. This corresponds exactly to the same situation as a parent change, when a node redirects the traffic from its children to the new parent, re-allocating all the dedicated cells. Therefore, the shared cells have to be used since the two nodes do not have a common cell anymore. Since collisions are frequent during these cells, the reservation may take a long time.

In our preliminary study (Section 3), we also measured the number of dropped packets caused by buffer overflow. Figure 6 reflects all the 30 experiments performed in the Grenoble testbed. We count the number of packets dropped after a 6P-clear command. Typically, this command is triggered either after a parent change to deallocate the cells in the previous parent or an inconsistency detection by 6P. We can note that a large number of packets are dropped after a 6P-clear command (30% of the packet for one experiment).

Therefore, we consider not using the sequence number: a housekeeping feature [24] is enough for detecting schedule inconsistencies. A housekeeping is an automatic feature that monitors system parameters periodically looking for inconsistencies, i.e. cells that are allocated in the transmitter but not in the receiver or vice-versa. Inconsistent cells have two different effects:

tx-cell: the receiver is not awake, and the transmitter never receives an acknowledgement. Typically, this inconsistency occurs when the transmitter tries to deallocated dedicated cell because of new communication requirements, e.g. some children may leave its subtree. Quickly, the incriminated cell exhibits a poor reliability compared with the other valid cells toward the same neighbor. Thus, the Scheduling Function will trigger a relocation, exactly like when a collision occurs, and will remove the corresponding inconsistency;

rx-cell: the receiver has just to stay awake but does not receive anything. After a timeout, the corresponding cell will be released. This inconsistency just impacts the energy consumption (idle listening) but not the reliability.

Figure 6 illustrates the number of packets dropped after a 6P-clear command when we use a housekeeping function, and we deactivate the 6P inconsistency check. We can clearly see that we reduce largely the number of packets dropped because of 6P. Inconsistencies should be handled individually, and a clear-command appears too aggressive.

6. Layer 3: routing oscillations

The routing protocol aims to construct efficient routes in multihop topologies. By efficiency, low power lossy networks often designate the reliability and energy efficiency. Thus, we have to select the most reliable links while globally limiting the energy consumption.

6.1. Link Metric

Different metrics have been proposed in the literature [25] targeting low power wireless networks. However, due to its lossy nature and dynamicity, to estimate the link quality in a reliable and efficient way is still an open problem.

With MinHop, a node selects as parent the node closest (in hops) to the sink. Thus, the network tends to be conservative with a low rate of reconfigurations [19]. However, using long link for communicating was proved to be inefficient [26] and the network may perform inadequately. RSSI uses the signal strength indicated by the radio chipset to use the most reliable links. Unfortunately, RSSI and reliability are loosely correlated for medium link qualities [27].

Measuring directly the reliability of the link quality seems the most accurate metric. For instance, the transmitter can compute the ratio of packets correctly acknowledged by the receiver. Typically, the Expected Transmission Counter (ETX) evaluates the average number of packets to transmit before receiving an acknowledgement. Let $PDR_{x \to y}$ denote the packet delivery ratio from x to y:

$$ETX = \frac{1}{PDR_{s \to r} * PDR_{r \to s}} \tag{4}$$

ETX is the default link metric for RPL [28]. However, ETX may create instabilities for long routes, because of its cumulative variations [19].

ETX^n was recently proposed by Duquennoy *et al.* [29]. While ETX minimizes the aggregated number of transmissions, ETX^n tries to privilege reliable links by computing the n_{th} power of the ETX value. For instance, the authors advocate that two hops links with a perfect reliability (ETX $= 1 + 1$) constitute a better path than a direct link with a PDR of 50% (ETX $= 2$), although both paths have the same accumulated ETX value. However, ETX^n implicitly penalizes more the intermediate links than ETX. For instance, a link for which the PDR changes only by 0.1 (80% → 70%), exhibits a ETX^2 variation of 0.5. Because of this penalty, ETX^2 may react even for small oscillations, provoking a parent change. One solution to mitigate this instability is to use higher hysteresis values for the routing protocol, but at the cost of decreasing the network reactivity.

6.2. Link quality estimation for inactive neighbors

The link quality indicators provided by the radio chipset are easy to obtain and they provide a quick way of assessing the channel state. However, these indicators have been proved to present good estimates only under specific circumstances. For instance, the accuracy of the Radio Signal Strengthen Indicator (RSSI) decreases when it falls in the grey zone or under the effect of external interference [30].

Estimating directly the reliability of the links in a 6TiSCH network is much more challenging. Indeed, a node may continuously count the number of packets correctly acknowledged by it children and parent nodes. However, it cannot estimate passively the quality for inactive neighbors: no packet is exchanged, by definition. Usually, most solutions rely on probing: some dedicated control packets are transmitted to re-evaluate continuously the link quality [31].

However, probes are very expensive in 6TiSCH networks:

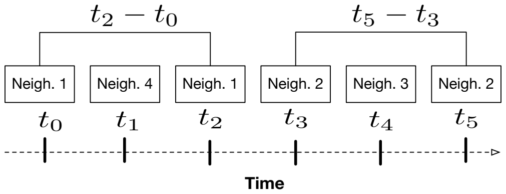


Figure 7: List of circular changes ordered by time.

1. if probes are transmitted through shared cells, they may collide with *beacons*, jeopardizing the network reactivity. For instance, dropping many routing packets (e.g. DIO) may force some children to change their preferred parent arbitrarily. Besides, the transmitter cannot differentiate the probes dropped because of collisions or because of high Packet Error Rates;
2. inversely, if the probes are transmitted through dedicated cells, the transmitter has to reserve some additional dedicated cells for each of its neighbors. This volume of control traffic impacts both the energy consumption and the network capacity, and this method seems unreasonable for very dense networks.

Thus, active measurements are particularly expensive in 6TiSCH networks.

Since active measurements are so expensive, most solutions rely on a default link metric for unknown neighbors. The default OpenWSN[2] implementation considers a default link quality of 4, which corresponds to an average Packet Delivery Ratio of 25%. In other words, if the preferred parent has a reliability larger than 25%, the node will never engage a parent change by default. To our mind, such strategy is too conservative and may lead to suboptimal topologies.

Inversely, if the default link cost is very low (e.g. 1), the node will choose iteratively each of its inactive neighbors as preferred parent. Each parent change also means renegotiating the whole bandwidth. This solution is clearly inapplicable.

To quantify the number of useless parent changes, we count the number of circular changes. By circular change, we mean a list of parent changes so that the node reselects back the initial parent of the list. For instance, in Figure 7 the neighbor 1 is selected at the instant t_0 and reselected at t_2. Similarly, the neighbor 2 is selected at t_3 and t_5. This chain of changes is certainly useless and it wastes energy: over-provisioning additional cells would have been much more efficient than changing the preferred parent, and renegotiating several cells (de/re-allocation).

Figure 8 illustrates the distribution of the time difference between two changes to the same parent. We select 7 experiments randomly, from our preliminary study in Lille (Section 3), to quantify the number of circular changes. We observe that the selected experiments follow typically the same distribution. In these experiments, approximately 50% of all circular changes occurred with less than 15 minutes. In IEEE 802.15.4-TSCH networks, we must account the control packets exchanged during the negotiation and the required time to reallocate the active cells. In long-term deployments (e.g. weeks, months), circular changes should be present only when the gain exceeds the reconfiguration cost. The reconfiguration has also to be correctly handled to configure properly the novel paths *before* the flows are redirected, to respect the reliability and delay constraints.

In our preliminary evaluation (section 3), we identified several parent changes due to what we call the existence of a *misleading zone*. A *misleading zone* is composed of nodes with low ranks, *virtually close* to the sink. Long links have been proved to exhibit often a very poor reliability [26]. Thus, a node may receive `beacons` from neighbors in the misleading zone time to time, infrequently.

Let us consider the topology illustrated in Figure 9. The node G has a preferred parent (F) for which the link quality degrades, and has to search for another parent. If it picks the neighbor with the lowest rank, it

---

[2] `http://www.openwsn.org`, an open-source implementation of a 6TiSCH protocol stack.

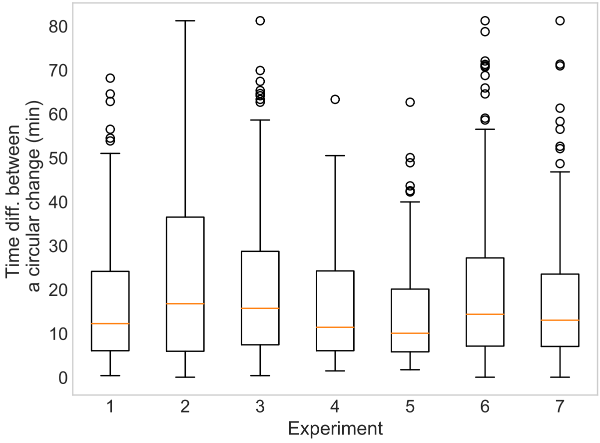


Figure 8: Distribution of the time difference between two changes to the same parent for 7 experiments selected randomly.

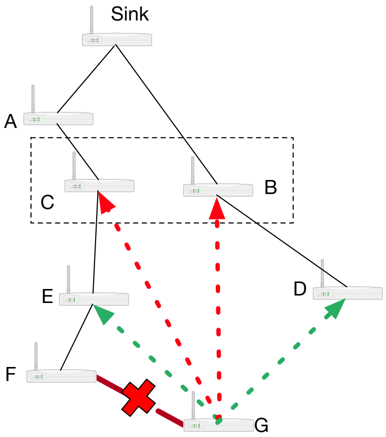


Figure 9: Misleading zone of the node G.

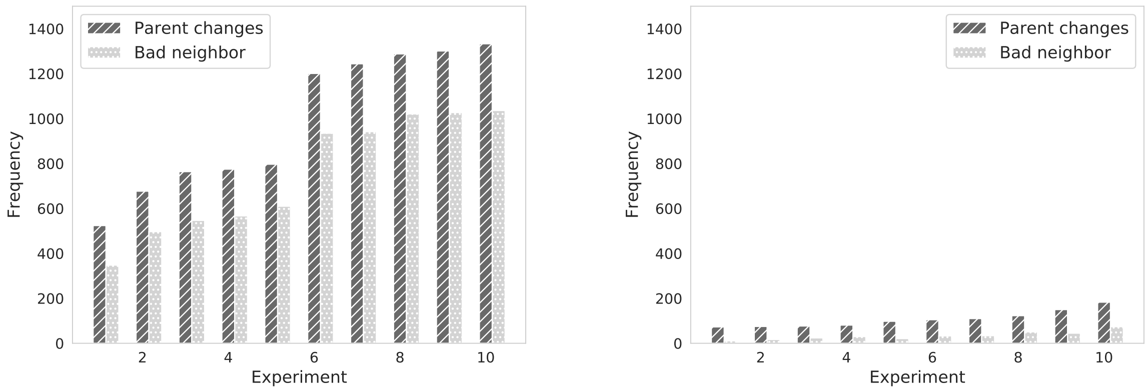
will select probably the nodes B or C, which provide a very low PDR. The node G should rather select E or D, with a much better tradeoff between reliability, and routing progress (i.e. closer to the sink).

To avoid selecting a bad parent, we propose to exploit a passive link quality estimation by monitoring the broadcast packets transmitted periodically by all the nodes. We demonstrated in a previous work the existence of a correlation between this unicast and broadcast Packet Delivery Ratios [32].

We propose to apply a filtering approach:

1. a node counts the number of EB packet received from each of its neighbors. Then, a WMEWMA is applied to smooth this metric;
2. the link quality is then estimated proportionally to this broadcast rate metric. This way, the best neighbors are scanned preferentially.

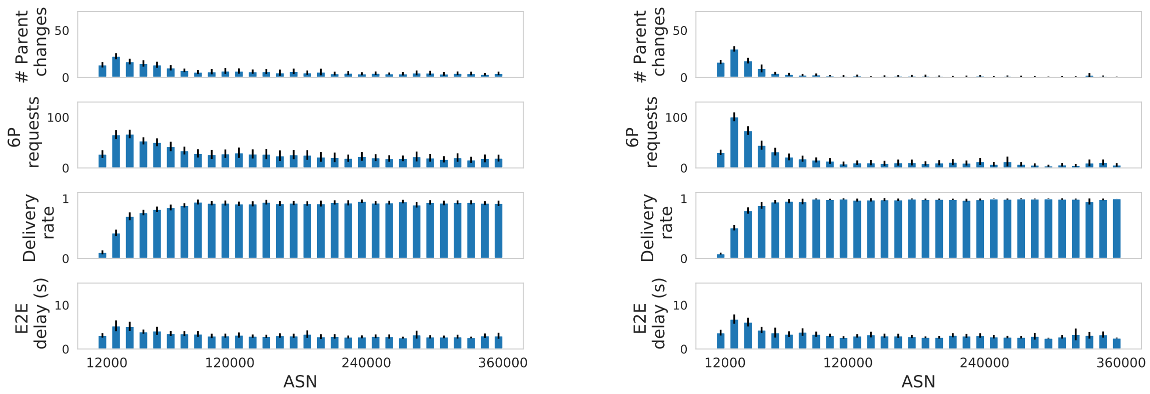
Figure 10a illustrates the number of parent changes and the number of times that a bad preferred parent was selected. A new parent is considered a *bad neighbor* when the negotiation between a node and its new parent fails or when the new parent has a lower PDR than the previous one. We selected 10 experiments randomly before and after introducing the filtering approach from our preliminary study in Lille. Without pre-filtering the neighbors, the number of parent changes is quite high, between 500 and 1300 depending



(a) No filtering

(b) Filtering

Figure 10: Frequency of parent changes and bad parent selection.



(a) Grenoble

(b) Lille

Figure 11: The performance stability in both testbeds after adding the housekeeping and filtering approaches (95 % of confidence level).

on external interference. On the contrary, using the broadcast rate (Figure 10b) helps to identify the best neighbors, without using an erroneous default link metric. Consequently, we select a good parent at the first try, and we have much less parent changes. Alternatively, we may here execute an active (and expensive) measurement method for the second step, to discriminate the best neighbors. However, our experimental evaluation demonstrates this passive measurement is sufficient to construct a stable and efficient topology, without requiring any additional control packet.

7. Experimental Evaluation

We focus now on measuring the network performance stability after integrating our modifications to the 6TiSCH stack (schedule housekeeping, misleading zone identification, filtering). We conduct our experiments in the same deployments (Grenoble and Lille), using the same platform as in Section 3. Our proposals were implemented in OpenWSN and are freely available on GitHub[3]. We use the same settings as in Section 3

---

[3] `https://github.com/rodrigoth/openwsn-fw/tree/convergence`

(e.g. slotframe size, shared cells, traffic load, sink position). In these experiments, all nodes execute our housekeeping approach (Section 5). Every 10 minutes, each node looks for schedule inconsistencies, i.e. cells with no rx or tx success. When a node has an allocated rx-cell that was never used, the housekeeping algorithm removes it. Additionally, a tx-cell that has never received an ACK is also removed.

We use a shared platform, deployed in office buildings, where Wi-Fi networks are also used. We do not have any control on the level of external interference. Thus, we run 30 different experiments for each value, picking randomly the nodes in the testbed (except the sink which remains the same). Each experiment runs for 90 minutes independently on each of the two testbeds. We plot confidence intervals to verify that the results are significant, i.e. exhibit small variations. The dataset generated in our experiments is also freely available online[4]to be analyzed by anyone. We expect to conduct experiments with controlled external interference in a controlled environment in a future work to analyze finely the impact of external interference.

Moreover, all nodes register the broadcast packets (EB+DIO) that were sent every 30 seconds in average by their neighbors. As explained in section 6.2, each node uses this broadcast rate to rank its neighbors. It selects its neighbor with the highest broadcast rate as preferred parent. A WMEWMA estimator helps to smooth the variations. In particular, we decided to *not* implement any probing mechanism, particularly expensive in IEEE 802.15.4-TSCH environments. Thus, the best candidate neighbors are those who are ranked in the first positions and a node prioritizes them when it needs to change its preferred parent.

Figure 11 highlights the network performance when using our approaches. Since our filtering approach relies on counting broadcast packets, the convergence time depends on the intensity of collisions in the shared cells. Indeed, the nodes send initially a burst of 6P requests and the probability of collisions in a shared cell is very high. However, as soon as the nodes reserve dedicated slots, the shared slots are less used and our solution becomes more precise.

Because the nodes have a precise view of their environment, they select a good candidate neighbor in the first attempts, thus largely reducing parent changes. This is reflected in a shorter convergence time and a smoother routing instability period during the operating phase. Routing reconfigurations are very infrequent, compared with the 6TiSCH stack without any optimization (Fig. 3). We reduce by 88% the number of parent changes during the 90 minutes of the experiment.

Similarly, we also reduce the number of 6P requests. Indeed, a parent change means also that the cells have to be re-negotiated with the novel parent. Moreover, a 6P request has a limited length, and contains only a small number of cells to insert. Thus, several 6P handshakes may be required before having a sufficient number of cells to forward all the packets in the queue. 6P requests are still required to adjust the number of cells when the link quality changes significantly. However, these changes just allow the node to empty its queue more easily, and do not impact negatively the reliability.

Additionally, our modified version of the 6TiSCH stack provides a end-to-end reliability of 99% for both testbeds. These results differ from the unoptimized 6TiSCH stack (Fig. 3), where the network performance was very unstable in all experiments.

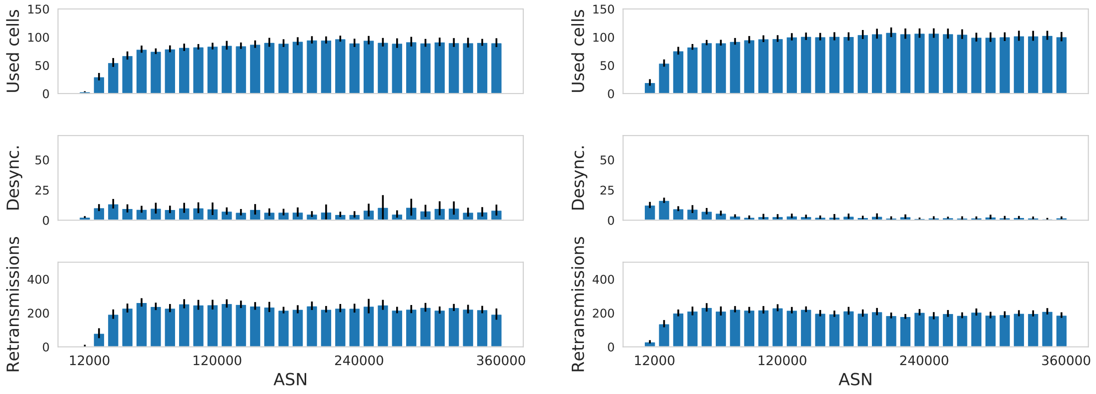
Differently from the results in our preliminary study (Section 3.2), we observe now a near constant end-to-end delay across all experiments in both testbeds. The routing stability achieved after adding our modifications allows the construction of reliable paths, with an upper bounded delay.

Finally, we compare the impact of our modifications on the schedule occupancy (Figure 12). We can note that we reserve approximatively the same number of cells in Grenoble, but the number of cells is higher for Lille after our modifications. Indeed, we remind that the delivery rate before our modifications was very low, and many packets were dropped (Fig. 3b). Mechanically, these non-transmitted packets do not need cells to be scheduled.

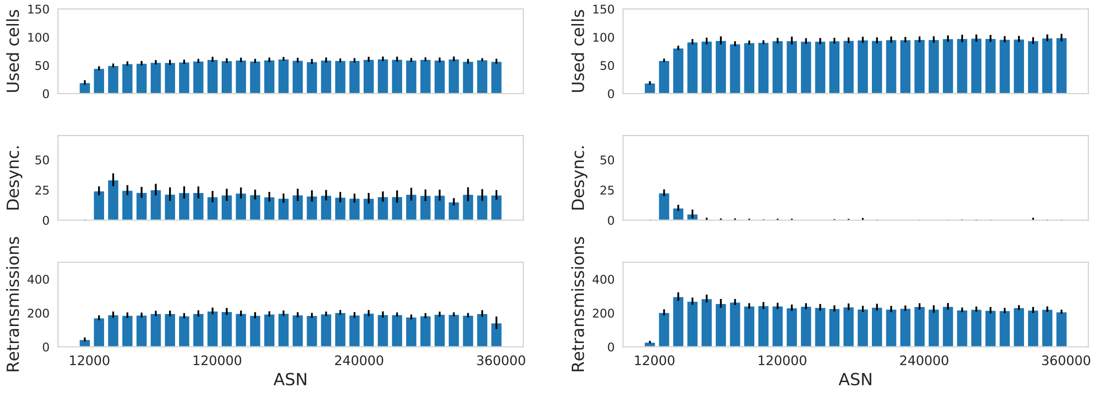
In timeslotted networks, the number of retransmissions is tightly related to the energy consumption. Indeed, the nodes can quickly turn off their radio during idle slots, and their energy consumption is much smaller compared with busy slots [33]. In figure 12, we can verify that enforcing stability is not detrimental to the network reliability. In Grenoble, we keep on using the most reliable links, with the same number of

---

[4] `http://www.rodrigoteleshermeto.com/#dataset`



(a) Grenoble before (b) Grenoble after



(c) Lille before (d) Lille after

Figure 12: Network efficiency before and after our modifications (95 % of confidence level).

retransmissions as Grenoble while providing an higher packet delivery rate. In Lille, we succeed to deliver the packets through long and unreliable routes, which increases both the delivery rate and the number of retransmissions.

Finally, we also measure the number of desynchronizations. A node becomes desynchronized typically after a schedule's inconsistency, when it has to flush all its dedicated cells. We can see that we reduced largely the number of desynchronizations in Lille: all the nodes keep synchronized, able to forward the packets.

Therefore, our approaches allow the deployment of stable, reliable and predictable multihop networks even in presence of external interference (e.g. Wi-Fi, other concurrent experiments on the testbed).

8. Related Work

Stability of network performance represents a key challenge in radio environments. In particular, a route has to be constructed, selecting the most stable nodes/links, so that their link quality fluctuations are minimized [34].

Estimating accurately the link quality for low power lossy networks has been extensively studied in the literature [16]. Because of the stochastic nature of radio transmissions, statistical estimators are required. A Window-Mean Exponentially Weighted Moving Average (WMEWMA) combined with ETX is a popular estimator [35]. However, it behaves poorly for medium link qualities [36].

RPL [10] is the most popular routing standard for low power lossy networks. Clausen *et al.* have highlighted some network instabilities under real deployment [37]. Because of the oscillations in the link quality, a node tends to change its parent aggressively, impacting negatively the convergence.

One may argue that routing changes are not prejudicial: the existence of a valid next hop could be sufficient. Unfortunately, RPL relies on a trickle timer to reduce the volume of control packets when the network has converged. Thus, oscillations mean also a larger overhead. Kermajani and Gomez [38] have conducted a sensitivity analysis of RPL. The parameters have been proved to have a strong impact on the convergence delay. They propose to use DODAG Information Solicitation (DIS) to accelerate the convergence: each node already attached to the DODAG has to reply with a DIO. However, these control packets would also use shared cells in TSCH, increasing the number of collisions.

Kim *et al.* [39] have proposed multipath routing to balance more efficiently the load among the different nodes, and to reduce the occurence of buffer overflows. Congestion often impacts negatively the stability, forcing the nodes to reconsider their routing decisions. Each node can tune its transmission power to force a node to detach from the routing tree, to balance more efficiently the load, and to increase the stability [40].

Iova *et al.* [19] evaluate the impact of different link quality metrics on the network routing stability. They highlighted the existence of a tradeoff between stability and performance. When using the MinHop metric, RPL operates steadily with low frequency of reconfigurations but performs badly. On the other hand, with ETX and LQI, the network presents higher frequency of reconfigurations and higher end-to-end reliability. Alvi *et al.* propose rather to change the objective function, combining the ETX and min hop metrics [41]. However, this method is quite conservative: a node tends to be stuck with the same parent, even if a better alternative choice exists.

Vucinic *et al.* focus on the problem of broadcast in the IEEE 802.15.4-TSCH layer [42]. Shared cells are used for broadcast packets, when the network bootstraps: too many collisions also induce a longer convergence delay. To reduce the collisions, broadcast packets are transmitted probabilistically, instead of using a fixed time period.

While much attention has been given to period traffic, some industrial applications may generate sporadic and high priority traffic in the presence of critical events. For instance, a bursty traffic upon unanticipated event (e.g., fire monitoring application [43]) may require that nodes renegotiate extra cells to accommodate the new communication flow, possibly dropping packets while the network re-converges. Ben Yaala *et al.* [44] handle sporadic traffic by multiplexing different high priority flows using shared cells in IEEE 802.15.4-TSCH networks. While the solution works well for low density networks, the collision rate degrades heavily the network performance for high densities. Zhang *et al.* [45] combine periodic and sporadic traffic, with the last having higher priority. In their solution, the network can drop periodic traffic so higher priority packets can be transmitted, attending the delay requirements. The sink is responsible of deciding which periodic packets should be discarded based on the Lawler's algorithm [46] and to push its decision to all nodes. However, they rely on good and stable link conditions, hard to obtain in real industrial deployments.

9. Conclusion and Future Work

We investigated in this paper the performance stability of 6TiSCH networks in two indoor testbeds with different channel conditions. We showed that the network does not succeed to converge to a steady-state through two different testbeds, even in static situations. For both testbeds, we can identify moments of performance instability due to oscillations in the radio conditions caused by external interference and obstacles. We identified the causes of instabilities, and proposed solution for each of the layers in the 6TiSCH stack. First, we demonstrated that a rearrangement of shared cells in the slotframe reduces the probability of collisions for control packets, paving the way to a faster negotiation during topology reconfigurations. Next, we simplified the schedule consistency management between two nodes to reduce the network instability caused by renegotiating from scratch all the cells when they detect a schedule inconsistency. We also exploited the existing correlation between the broadcast packet reception rate and the unicast link quality to create a two-step parent selection, avoiding bad choices leading to instabilities. We finally obtain a network that converges faster and that reacts accurately during moments of instabilities.

In a future work, we expect to go further in the convergence, and to study in-depth the network re-convergence in the presence of long-term modifications of the environment. The network must be robust enough to keep on guaranteeing a minimum reliability when e.g. a node has crashed. When reconfiguring one part of the network, we must be sure to be able to respect the reliability and delay constraints after the reconfiguration. Thus, the network has to be reconfigured before the flow is redirected. Multipath, over-provisioning, and worst-case analysis should help to solve such challenge. We also expect to investigate the stability of a 6TiSCH stack when considering bursty traffic, e.g. alarms with strict delay constraints. The scheduling functions must be designed to not over-react to changes. Tuning the network resources for the worst case then apears particularly challenging.

References

[1] B. Galloway, G. P. Hancke, Introduction to industrial control networks, IEEE Communications Surveys Tutorials 15 (2) (2013) 860–880. `doi:10.1109/SURV.2012.071812.00124`.

[2] W. Kastner, G. Neugschwandtner, S. Soucek, H. M. Newman, Communication systems for building automation and control, Proceedings of the IEEE 93 (6) (2005) 1178–1203. `doi:10.1109/JPROC.2005.849726`.

[3] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, B. Yin, Smart factory of industry 4.0: Key technologies, application case, and challenges, IEEE Access 6 (2018) 6505–6519. `doi:10.1109/ACCESS.2017.2783682`.

[4] M. Raza, N. Aslam, H. Le-Minh, S. Hussain, Y. Cao, N. M. Khan, A critical analysis of research potential, challenges, and future directives in industrial wireless sensor networks, IEEE Communications Surveys Tutorials 20 (1) (2018) 39–95. `doi:10.1109/COMST.2017.2759725`.

[5] T. Watteyne, S. Lanzisera, A. Mehta, K. S. J. Pister, Mitigating multipath fading through channel hopping in wireless sensor networks, in: International Conference on Communications (ICC), IEEE, 2010, pp. 1–5. `doi:10.1109/ICC.2010.5502548`.

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

[7] C. Shih, A. E. Xhafa, J. Zhou, Practical frequency hopping sequence design for interference avoidance in 802.15.4e tsch networks, in: IEEE International Conference on Communications (ICC), 2015, pp. 6494–6499. `doi:10.1109/ICC.2015.7249359`.

[8] M. Gürsu, M. Vilgelm, S. Zoppi, W. Kellerer, Reliable co-existence of 802.15.4e TSCH-based WSN and Wi-Fi in an aircraft cabin, in: IEEE International Conference on Communications Workshops (ICC), 2016, pp. 663–668. `doi:10.1109/ICCW.2016.7503863`.

[9] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, F. Tholeyre, Is local blacklisting relevant in slow channel hopping low-power wireless networks?, in: ICC, IEEE, Miami, FL, USA, 2017, pp. 1–6. `doi:10.1109/ICC.2017.7996980`.

[10] T. Winter, el at., RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, RFC 6550, IETF (March 2012).

[11] O. Iova, F. Theoleyre, T. Watteyne, T. Noel, The Love-Hate Relationship between IEEE 802.15.4 and RPL, IEEE Communications Magazine 55 (1) (2017) 188–194. `doi:10.1109/MCOM.2016.1300687RP`.

[12] Q. Wang, X. Vilajosana, T. Watteyne, 6top Protocol (6P), draft, IETF, draft-ietf-6tisch-6top-protocol-11 (October 2018).

[13] E. Ghadimi, O. Landsiedel, P. Soldati, S. Duquennoy, M. Johansson, Opportunistic routing in low duty-cycle wireless sensor networks, ACM Trans. Sen. Netw. 10 (4) (2014) 67:1–67:39. `doi:10.1145/2533686`.

[14] R. T. Hermeto, A. Gallais, F. Theoleyre, Scheduling for IEEE802.15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey, Computer Communications 114 (2017) 84 – 105. `doi:10.1016/j.comcom.2017.10.004`.

[15] D. Dujovne, L. Grieco, M. Palattella, N. Accettura, 6TiSCH 6top Scheduling Function Zero / Experimental (SFX), Internet-draft, IETF, draft-ietf-6tisch-6top-sfx-00 (Sept 2017).

[16] R. D. Gomes, D. V. Queiroz, A. C. L. Filho, I. E. Fonseca, M. S. Alencar, Real-time link quality estimation for industrial wireless sensor networks using dedicated nodes, Ad Hoc Networks 59 (2017) 116 – 133. `doi:10.1016/j.adhoc.2017.02.007`.

[17] R. Musaloiu-E., A. Terzis, Minimising the Effect of WiFi Interference in 802.15.4 Wireless Sensor Networks, Inderscience International Journal of Sensor Networks 3 (1) (2008) 43–54. `doi:10.1504/IJSNET.2008.016461`.

[18] P. Agrawal, A. Ahlén, T. Olofsson, M. Gidlund, Long Term Channel Characterization for Energy Efficient Transmission in Industrial Environments, IEEE Transactions on Communications 62 (8) (2014) 3004–3014. `doi:10.1109/TCOMM.2014.2332876`.

[19] O. Iova, F. Theoleyre, T. Noel, Stability and efficiency of RPL under realistic conditions in Wireless Sensor Networks, in: PIMRC, IEEE, London, UK, 2013, pp. 2098–2102. `doi:10.1109/PIMRC.2013.6666490`.

[20] F. Theoleyre, G. Papadopoulos, Experimental Validation of a Distributed Self-Configured 6TiSCH with Traffic Isolation in Low Power Lossy Networks, in: MSWiM, ACM, Malta, Malta, 2016, pp. 102–110. `doi:10.1145/2988287.2989133`.

[21] S. Y. 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:10.1016/j.aeue.2013.02.007`.

[22] T. Chang, etl al., 6TiSCH Minimal Scheduling Function (MSF), draft 3, IETF (April 2019).

[23] C. Vallati, S. Brienza, G. Anastasi, S. K. Das, Improving network formation in 6tisch networks, IEEE Transactions on Mobile Computing 18 (1) (2019) 98–110. `doi:10.1109/TMC.2018.2828835`.

[24] K. Muraoka, T. Watteyne, N. Accettura, X. Vilajosana, K. S. J. Pister, Simple Distributed Scheduling With Collision Detection in TSCH Networks, IEEE Sensors Journal 16 (15) (2016) 5848–5849. `doi:10.1109/JSEN.2016.2572961`.

[25] N. Baccour, A. Koubâa, L. Mottola, M. Zúñiga, H. Youssef, C. Boano, M. Alves, Radio link quality estimation in wireless sensor networks: A survey, ACM Transactions on Sensor Networks (TOSN) 8 (4) (2012) 34.

[26] K. Heurtefeux, H. Menouar, Experimental evaluation of a routing protocol for wireless sensor networks: RPL under study, in: WMNC, IFIP, Dubai, United Arab Emirates, 2013, pp. 1–4. `doi:10.1109/WMNC.2013.6548990`.

[27] B. Pavkovic, F. Theoleyre, D. Barthel, A. Duda, Experimental analysis and characterization of a wireless sensor network environment, in: PE-WASUN, ACM, Bodrum, Turkey, 2010, pp. 25–32. `doi:10.1145/1868589.1868595`.

[28] P. Thubert, Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL), RFC 6552, IETF (March 2012).

[29] S. Duquennoy, J. Eriksson, T. Voigt, Five-Nines Reliable Downward Routing in RPL, `http://arxiv.org/abs/1710.02324` 1710.02324v1, arXiv (2017).

[30] F. Barać, M. Gidlund, T. Zhang, Ubiquitous, yet deceptive: Hardware-based channel metrics on interfered WSN links, IEEE Transactions on Vehicular Technology 64 (5) (2015) 1766–1778.

[31] E. Ancillotti, C. Vallati, R. Bruno, E. Mingozzi, A reinforcement learning-based link quality estimation strategy for RPL and its impact on topology management, Computer Communications 112 (2017) 1 – 13. `doi:10.1016/j.comcom.2017.08.005`.

[32] R. T. Hermeto, A. Gallais, K. V. Laerhoven, F. Theoleyre, Passive Link Quality Estimation for Accurate and Stable Parent Selection in Dense 6TiSCH Networks, in: EWSN, ACM, Madrid, Spain, 2018, pp. 114–125.

[33] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, K. S. J. Pister, A realistic energy consumption model for tsch networks, IEEE Sensors Journal 14 (2) (2014) 482–489. `doi:10.1109/JSEN.2013.2285411`.

[34] M. Boushaba, A. Hafid, M. Gendreau, Node stability-based routing in wireless mesh networks, Journal of Network and Computer Applications 93 (2017) 1 – 12. `doi:10.1016/j.jnca.2017.02.010`.

[35] A. Woo, T. Tong, D. Culler, Taming the underlying challenges of reliable multihop routing in sensor networks, in: International Conference on Embedded Networked Sensor Systems (SenSys), ACM, 2003, pp. 14–27. `doi:10.1145/958491.958494`.

[36] A. Becher, O. Landsiedel, G. Kunz, K. Wehrle, Towards short-term wireless link quality estimation, in: Workshop on Embedded Networked Sensors (Hot EmNetS), ACM, 2008, pp. 1–5.

[37] T. Clausen, U. Herberg, M. Philipp, A critical evaluation of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL), in: WiMob, IEEE, Shanghai, China, 2011, pp. 365–372. `doi:10.1109/WiMOB.2011.6085374`.

[38] H. Kermajani, C. Gomez, On the Network Convergence Process in RPL over IEEE 802.15.4 Multihop Networks: Improvement and Trade-Offs, Sensors 14 (2014) 11993–12022. `doi:10.3390/s140711993`.

[39] H. S. Kim, H. Kim, J. Paek, S. Bahk, Load Balancing Under Heavy Traffic in RPL Routing Protocol for Low Power and Lossy Networks, IEEE Transactions on Mobile Computing 16 (4) (2017) 964–979. `doi:10.1109/TMC.2016.2585107`.

[40] H. Kim, J. Paek, D. Culler, S. Bahk, Do not lose bandwidth: adaptive transmission power and multihop topology control, in: International Conference on Distributed Computing in Sensor Systems (DCOSS), IEEE, Ottawa, Canada, 2017, pp. 99–108. `doi:10.1109/DCOSS.2017.23`.

[41] S. A. Alvi, F. u. Hassan, A. N. Mian, On the energy efficiency and stability of RPL routing protocol, in: IWCMC, IEEE, 2017, pp. 1927–1932. `doi:10.1109/IWCMC.2017.7986578`.

[42] M. Vučinić, T. Watteyne, X. Vilajosana, Broadcasting strategies in 6tisch networks, Internet Technology Letters 1 (1) (2 2018). `doi:10.1002/itl2.15`.

[43] L. Zhang, G. Wang, Design and implementation of automatic fire alarm system based on wireless sensor networks, in: International Symposium on Information Processing (ISIP), Academy Publisher, 2009, p. 410.

[44] S. Ben Yaala, F. Théoleyre, R. Bouallegue, Experimental Analysis of the Efficiency of Shared Access in IEEE 802.15.4-TSCH Networks with Sporadic Traffic, in: International Conference on Advanced Information Networking and Applications (AINA), IEEE, 2018, pp. 9–16. `doi:10.1109/AINA.2018.00015`.

[45] T. Zhang, T. Gong, C. Gu, H. Ji, S. Han, Q. Deng, X. S. Hu, Distributed dynamic packet scheduling for handling disturbances in real-time wireless networks, in: Real-Time and Embedded Technology and Applications Symposium (RTAS), IEEE, 2017, pp. 261–272. `doi:10.1109/RTAS.2017.11`.

[46] J. Y. Leung, Handbook of scheduling: algorithms, models, and performance analysis, CRC press, 2004.