On the (over)-Reactions and the Stability of a 6TiSCH Network in an Indoor Environment

Rodrigo Hermeto, Antoine Gallais, Fabrice Theoleyre

To cite this version:

Rodrigo Hermeto, Antoine Gallais, Fabrice Theoleyre. On the (over)-Reactions and the Stability of a 6TiSCH Network in an Indoor Environment. MSWIM '18 - 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Oct 2018, Montreal, Canada. 10.1145/3242102.3242104 . hal-01886716

HAL Id: hal-01886716
https://hal.archives-ouvertes.fr/hal-01886716

Submitted on 29 Jan 2020

On the (over)-Reactions and the Stability of a 6TiSCH Network in an Indoor Environment

Rodrigo Teles Hermeto
ICube Lab, CNRS / University of Strasbourg
Illkirch, France
teleshermeto@unistra.fr

Antoine Gallais
ICube Lab, University of Strasbourg / CNRS / Inria Lille - Nord Europe
antoine.gallais@inria.fr

Fabrice Theoleyre
ICube Lab, CNRS / University of Strasbourg
Illkirch, France
theoleyre@unistra.fr

ABSTRACT

Industrial networks differ from others kinds of networks because they require real-time performance in order to meet strict requirements. With the rise of low-power wireless standards, the industrial applications have started to use wireless communications in order to reduce deployment and management costs. IEEE802.15.4-TSCH represents currently a promising standard relying on a strict schedule of the transmissions to provide strong guarantees. However, the radio environment still exhibits time-variable characteristics. Thus, the network has to provision sufficient resource (bandwidth) to cope with the worst case while still achieving high energy efficiency. The 6TiSCH IETF working group defines a stack to tune dynamically the TSCH schedule. In this paper, we analyze in depth the stability and the convergence of a 6TiSCH network in an indoor testbed. We identify the main causes of instabilities, and we propose solutions to address each of them. We show that our solutions improve significantly the stability.

CCS CONCEPTS

• **Networks** → **Network experimentation**; **Network performance analysis**; **Sensor networks**;

KEYWORDS

IIoT; 6TiSCH; IEEE802.15.4-2015-TSCH; Stability; Convergence

ACM Reference Format:
Rodrigo Teles Hermeto, Antoine Gallais, and Fabrice Theoleyre. 2018. On the (over)-Reactions and the Stability of a 6TiSCH Network in an Indoor Environment. In *21st ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM '18), October 28-November 2, 2018, Montreal, QC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3242102.3242104

1 INTRODUCTION

Industrial networks aim to provide strong guarantees to enable real-time communications. They are typically used to monitor safety-related industrial processes, such as in the Industry 4.0 and smart factories [1]. Consequently, best-effort solutions are not acceptable: high reliability (e.g. > 99%) and an upper bounded delay must be ensured.

With recent wireless standards, industrial networks rely more and more on wireless communications. However, low-power wireless networks are known to be lossy with no delivery guarantees. Thus, we have now to attend deterministic performance on top of an unreliable link layer.

IEEE 802.15.4-TSCH (Time-Slotted Channel Hoping) targets low-power, deterministic and reliable wireless industrial networks. It carefully schedules the transmissions such that each device has enough transmission opportunities while avoiding collisions. In addition, slow-channel hopping allows the nodes to combat the effect of external interference. However, higher power networks such as Wi-Fi can still disrupt IEEE 802.15.4-TSCH networks when deployed in the same environment. Moreover, external interference exhibits time-variable characteristics [2]. This implies that the network must handle efficiently these time variations, and must reserve enough resources to support the *worst* case.

Low-power routing protocols (e.g. RPL [3]) try to deal with instabilities by adjusting the topology dynamically and by ensuring that each device has one appropriate neighbor to communicate with. Typically, a trickle timer aims to decrease the volume of control traffic when the network is stable. Unfortunately, RPL has been proved to overreact to link quality changes [4].

In this work, we focus on multihop networks with real-time guarantees, and we highlight the main causes of instability and their impact on the network convergence. We also propose improvements to existing solutions to reduce the convergence time. We are convinced that investigating these factors is of primal importance to low-power wireless networks with deterministic performance requirements. The full understanding of these factors will help in the development of more efficient protocols targeting real scenarios with different sources of instability (e.g. nodes failures, link quality oscillations, etc.).

The contributions of this paper are as follows:

1. our experimental results demonstrate the instability of a 6TiSCH network in an indoor testbed;
2. we identify the main causes of instability and their impact on the network convergence;
3. we provide solutions to handle the pathological situations conducting to instability. In particular, we propose a two-step parent selection to avoid oscillations, and we improve the schedule inconsistency management which may arise in a lossy network.

MSWIM '18, October 28-November 2, 2018, Montreal, QC, Canada

ACM ISBN 978-1-4503-5960-3/18/10…$15.00
https://doi.org/10.1145/3242102.3242104

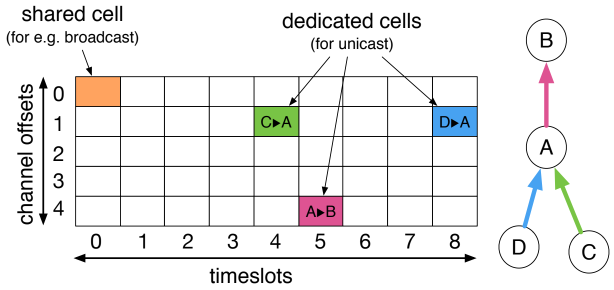


Figure 1: An example of TSCH scheduling for node D. A → D stands for *'A transmits to D'*, while the shared cell is used for broadcast or control frames.

2 BACKGROUND

We present here the main mechanisms of the 6TiSCH stack, and how it constructs and maintains the network structure (e.g. routes, schedules). We finally discuss the link quality metrics to set up efficient routes.

2.1 IEEE 802.15.4-TSCH

IEEE 802.15.4-2015 has defined the TSCH mode [5], where nodes schedule the transmissions such that each application has enough transmission opportunities while avoiding collisions. In addition, the standard proposes the use of a channel-hopping technique, so that subsequent packets are transmitted over different frequencies.

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. 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 immediately. If an acknowledgement is required but wasn't received, the transmitter considers a collision occurred. In that case, it selects a random backoff value, and *skips* the corresponding number of shared cells.

Each timeslot is labeled with an Absolute Sequence Number (ASN), a variable which counts the number of timeslots elapsed since the network was established. A node computes the frequency to use at the beginning of each slot, derived from the channel offset assigned to the cell, and the ASN.

Many distributed and centralized scheduling algorithms have been proposed so far for TSCH networks [6]. Centralized approaches assume the Path Computation Element (PCE) has a complete knowledge of the network (topology, traffic). 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 cells: all the nodes have to stay awake during this 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.

2.2 6TiSCH overview

The 6TiSCH IETF working group aims to define 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 [7]. 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. Typically, 6P packets use 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 schedules since the beginning. This sequence number is embedded in every 6P packet, so that the Scheduling Function is able to detect and to handle an inconsistency. For instance, SFx [8] triggers a 6P clear command to reset both schedules when both 6P sequence numbers differ.

2.3 Link quality metrics

Different metrics have been proposed in the literature [9] 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. When using MinHop, the network tends to be more stable with a low rate of reconfigurations [10]. However, using long link for communicating was proved to be inefficient [11] 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 [12].

The Expected Transmission Counter (ETX) is widely used in the literature [13]. Let $PDR_{x \to y}$ denote the packet delivery ratio from x to y. ETX estimates the average number of transmissions before the emitter receives a link-layer acknowledgement:

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

However, ETX may create instabilities for long routes, because of its cumulative variations [10].

ETX^n was recently proposed by Duquennoy *et al.* [14]. 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 radio links with a perfect reliability constitute a better path than a direct link with a PDR of 50%, although both paths have the same ETX value.

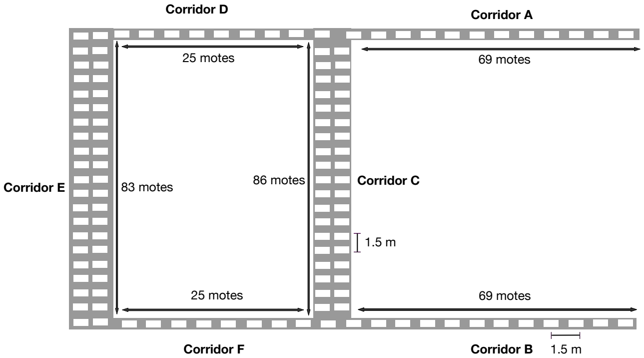


Figure 2: Nodes deployment on the IoT-LAB in Grenoble.

3 MOTIVATION & PRELIMINARY STUDY

To the best of our knowledge, while much attention has been given to construct schedules to provide high reliability in 6TiSCH networks [6] [15], the network convergence has not been investigated in details, yet. More specifically, we are interested in studying how a 6TiSCH network reacts to local changes. For instance, link quality drops and sudden faults caused by hardware malfunctions are situations likely to occur in real deployments. Theoleyre *et al.* [16] highlighted that many packets are dropped during the bootstrapping phase since reservations take time, and create buffer overflows.

Iova *et al.* [10] already demonstrated that RPL changes routes frequently in the presence of link quality oscillations. This instability disturbs the convergence of the Scheduling Function, specially in multihop networks where the bandwidth needs to be reserved hop by hop toward the sink.

To illustrate our motivation, we measure the stability of a network where the nodes report periodically their data to the sink. This scenario represents typically a monitoring application used in industrial deployments [17]. We deploy this network on the FIT IoT-LAB testbed, an open platform where several concurrent experiments are executed in parallel, providing a realistic environment for IoT-related systems and applications experiments. We employ M3 nodes, based on a STMicroelectronics 32-bit ARM Cortex-M3 micro-controller (ST2M32F103REY) that embeds an AT86RF231 radio chip, providing an IEEE 802.15.4 compliant PHY layer. Because the nodes are deployed in an indoor environment, they are also subject to external interference originated from others wireless devices, such as Wi-Fi access points and users, and concurrent IEEE 802.15.4 networks.

In this preliminary study, we investigate the network stability when possibly other networks operating in the same unlicensed band are colocated in the same area. We measure the performance of a 31 nodes topology. Moreover, we execute OpenWSN[*], that provides an open-source implementation of the 6TiSCH stack (i.e. IEEE 802.15.4-TSCH, 6P, SFx, 6LoWPAN and RPL). We configure a slotframe composed of 199 slots, where 8 were shared slots restricted to broadcast and 6P control packets.

We select the nodes in the corridor C (Figure 2), where the sink was placed on the extreme side of the network in order to have a multihop topology with up to 5 hops. All nodes generate a convergecast CBR traffic (one data packet every 20 seconds). We investigate the behavior of the network for 8 hours during a regular workday from the morning until the night. We used the default OpenWSN values for all the parameters.

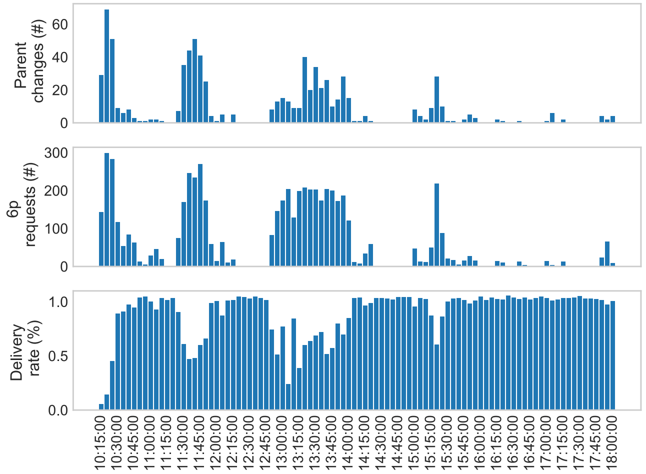


Figure 3: Parent changes, 6p requests and packet delivery ratio over the day (5 minutes of interval).

Figure 3 shows (i) the number of parent changes, (ii) the number of 6p requests and (iii) the packet delivery ratio. Surprisingly, we keep on identifying routing reconfigurations during the 8 hours: RPL detects a better parent, and triggers reconfigurations. Unfortunately, 6TiSCH is reservation based, and new cells have to be reserved with the new parent: 6P requests are still generated several hours after the network has bootstrapped. Worse, these reconfigurations deeply impact the reliability: the packet delivery ratio drops significantly when a burst of RPL reconfigurations occurs. Indeed, the re-convergence time is quite long (reserving new cells is not instantaneous), and packets are dropped because of buffer overflows.

4 CAUSES OF NETWORK INSTABILITY AND THEIR SOLUTIONS

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 interference, which is already the case for the 800MHz band [18]. 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 provide strict guarantees.

Network reconfigurations create burst of 6P requests to modify the schedule. For distributed Scheduling Functions (SF), the nodes decide autonomously the cells to use, before detecting and solving collisions. Thus, changes in the schedule also increase the probability of collisions, which impact even flows which have not been rescheduled.

We focus now on the identification of factors that led to these periods of turbulence, when the network transits from a stable to an highly unstable state. We consider that a network leaves the stable state at the moment a node switches its preferred parent (routing) or sends a 6P packet (scheduling).

---

[*] http://openwsn.org/

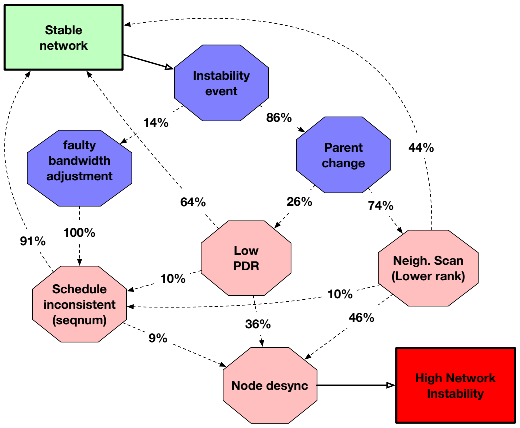


Figure 4: Cause and consequence diagram of the factors that led to network instability.

4.1 Instability causes

Figure 4 illustrates the causes of instability and the frequency of their occurrence. More specifically, we reference all the events (i.e. parent change, schedule change, seqnum inconsistency, etc.) in the experiment described in section 3. The events are chronologically ordered with their ASN to construct the *flow*.

We remark that in complex situations, the different problems are not isolated. For instance, a schedule inconsistency creates a parent change, with a low PDR, leading to a cascade effect (i.e. the paths leading to a desynchronization are not unique). We will now describe in details each cause and the solutions we propose to resolve them.

Two events can be the origin in the flow of events leading to instability. A *faulty bandwidth re-adjustment* occurs 14% of the time. We consider a rescheduling is faulty when the negotiation fails and leads to a reset. It typically happens when a node adjusts its schedule because of topology changes in its subtree. In some cases, the schedules of the node and its parent become inconsistent [8].

On the other hand, a *parent change* is responsible to create instability 86% of the time. Indeed, a parent change in all cases implies that the cells have to be released from the previous parent, and renegotiated with the new one. Practically, this negotiation is not instantaneous and negatively impacts the reliability.

4.2 Parent Change

In a 6TiSCH network, a node changes its preferred parent when another neighbor provides a significantly smaller rank. Typically, when the link quality between them degrades, another neighbor will be selected. Many metrics have been proposed in the literature to measure the link quality [9]. However, due to its lossy nature and dynamicity, to estimate the link quality in a reliable way is still an open problem. In particular, it represents a very challenging problem to estimate the link quality for unused neighbors [19].

In our preliminary study, 26% of the parent changes were caused by a *low PDR*. In such case, the node simply looks for another neighbor with a lower rank to forward its packets. On the other hand, 74% of the cases were caused by nodes changing their preferred parents when the next-hop (parent and its associate parent) link deteriorates or when the node finds a new neighbor with a rank substantially lower than its current parent. This occurs even if the link quality between the node and its immediate parent is high.

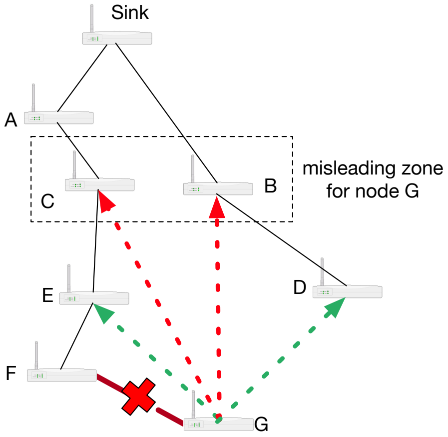


Figure 5: Misleading zone in a multihop network where G has to change its preferred parent. Here, A and B are sufficiently close to G to be recognized as neighbors.

We consider that a parent change may come either from a bad initial choice, or from an overreaction to a link quality change. To our mind, the two problems require specific solutions.

4.2.1 Inaccurate blind estimation. When a node changes its preferred parent, it selects by definition a next hop which he did not exploit previously. Unfortunately, estimating the link quality requires often to send unicast packets (either data or probe packets) to the given neighbor. Since 6TiSCH requires to reserve some cells for such probes, active measurements are very expensive, and most solutions rely on a default link metric for unknown neighbors.

In particular, when a packet is received from a bad neighbor, with a rank smaller than its own preferred parent, the node will probably engage a parent change. If the new parent provides actually a bad link quality, the cell's negotiation is very long, and the node may re-change its parent. Even worse, a bad parent selection may even jeopardize the synchronization process, and the node may become disconnected. As shown by Heurtefeux and Menouar [11], longer links tend to present a lower packet reception rate. Thus, this pathological situation is quite common.

We highlight the existence of a *misleading zone*, which is composed of neighbors that are much closer to the sink, but very far from the node. We illustrate the concept of a misleading zone in Figure 5. The node G has to select a new parent (e.g. because of external interference), and two of its neighbors have a very low rank (B and C). Unfortunately, these two neighbors are very far, and it is highly probable they will provide a poor reliability. On the contrary, the nodes D and E would constitute much better parents, but provide a larger rank and are less probable to be selected.

Solution: two-steps parent selection with filtering. We demonstrated in a previous work the hazardousness of using a default link quality blindly for unused links *et al.* [19]. In particular, we demonstrated the existence of a correlation between the broadcast reception rate and the unicast link quality in 6TiSCH networks.

Thus, we propose to use this broadcast reception rate as a first step to filter really bad neighbors, even if a node does not communicate with them. Secondly, we restrict the default link cost only to the best neighbors, i.e. the neighbors that sent more broadcast packets. Conversely, we define a larger value (i.e. default value + c) in a way that selecting a bad neighbor becomes more costly for a node.

Using the broadcast link rate solves the misleading zone problem, as we will highlight in our experimental performance evaluation. In particular, a bad neighbor from the misleading zone may succeed to deliver an enhanced beacon (TSCH frame containing information on synchronization, channel hopping and timeslot used in the network, further noted as EB) or a RPL DIO. However, its broadcast rate would be sufficiently low to never engage a parent change.

4.2.2 Overreaction to Changes. The environment characteristics are time-variable [2]. Some external sources of interference may stop or start randomly, an obstacle (e.g. moving object/person) may temporarily impact the link quality.

In our preliminary experiments, a node may change its parent to come back after a while to its previous parent. These oscillations are very prejudicial: many packets may be dropped until the new cells are reserved along the new path, and control packets have been transmitted while such reconfiguration is useless. We argue that overprovisionning would have more efficiently tackled the problem: the link quality hadn't change drastically.

Solution: stable link quality metric. We argue that the link quality metric should provide a good tradeoff between reactivity and stability to avoid excessive parent changes. Short-term metrics like ETX may add instability to the network due to its continuously search for the best immediate link quality [10]. Nevertheless, ETX is one of the most used metrics in low-power wireless networks and is used by default in the Minimum Rank with Hysteresis Objective Function (MRHOF).

In section 5, we compare experimentally different metrics (cf. Section 2.3) and their impact on the stability.

4.3 Schedule consistency check

Because of the links lossy nature, the 6P protocol relies on mechanisms to ensure the schedule consistency between a node and its parent. Typically, a 6P sequence number is maintained per link, incremented 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.

Because of lossy links, the reply may be dropped when the link layer triggers too many retransmissions (at most 3 by default). Figure 6 shows a typical scenario to illustrate this inconsistency creation. 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, A never receives the reply from B and 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 [8]. Then, the two nodes have to restart the reservations from scratch. Even worse, the shared cells have to be used since the two nodes haven't anymore a common cell. Since collisions are frequent during these cells, the reservation may take a long time.

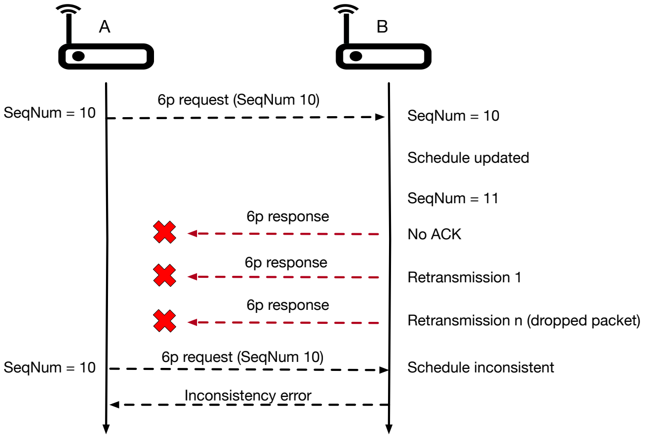


Figure 6: An example where two neighbors have inconsistent schedules.

The 6P requests may also collide with the EB, and the nodes may become desynchronized after ≈ 35 seconds by default. Although not very frequent (occurring in 9% of the time), this desynchronization is critical since all the packets start to be buffered, and possibly dropped.

Solution: simplified inconsistency management. We argue that the benefit-cost-ratio is here negative. This approach is particularly inadequate when a node has many dedicated cells already allocated to its preferred parent. Reallocating all the cells requires a long time, and packets may be dropped before all the cells are correctly allocated.

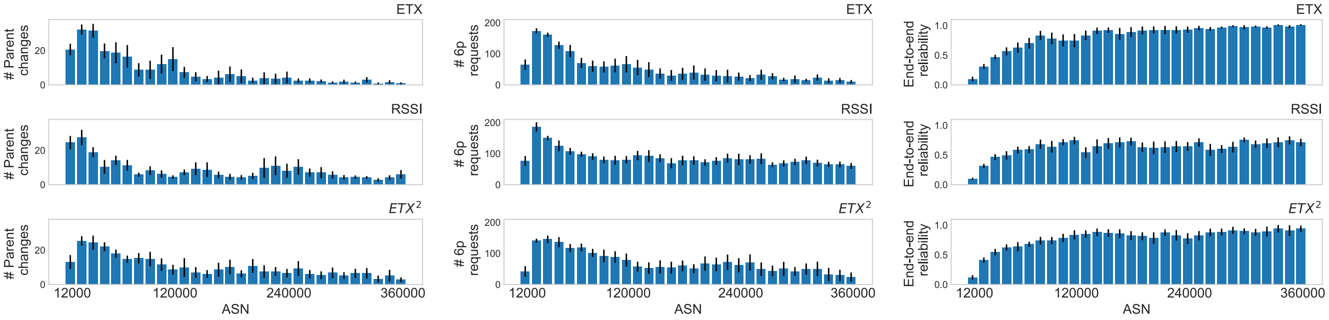
Therefore, we consider not using the schedule check: a housekeeping feature [20] is enough for detecting schedule inconsistencies. Inconsistent cells have two different effects:

- **TX-cell:** the receiver is not awake, and the transmitter never receives an acknowledgement. Quickly, the incriminated cell exhibits a poor reliability compared with the other cells toward the same neighbor. Thus, the SF 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.

Thus, inconsistencies should be handled in an individual way, and a clear-command is to our mind too aggressive.

5 PERFORMANCE EVALUATION

In this Section, we present our experimental campaign over the FIT IoT-LAB platform. We conduct our experiments in the same deployment (Grenoble), using the same platform as in Section 3. Differently from the previous experiment, we reduced the transmission power to -9dBm to run a larger multihop network (up to



(a) Parent changes

(b) 6P requests

(c) End-to-end reliability

Figure 7: Network stability when using the 3 evaluated metrics (95% of confidence).

7 hops). This power reduction allows us to deploy a multihop network without placing the nodes too far from each other, reducing the chance of concurrent experiments and increasing the reproducibility of our experiments. We implemented our proposals in OpenWSN and they are freely available on GitHub [†]. The dataset generated in our experiments is also freely available on GitHub [‡].

To evaluate the accuracy of each of our propositions from Section 4, we perform 10 experiments of 90 minutes using 31 nodes in the corridor B (Figure 2). We plot also the 95% confidence intervals for each point in the graphs, over the 10 experiments. We place the sink on the extreme side of the network and we reserve a pair of nodes every 4.5 meters over the corridor. Each mote sends its packets toward the sink according to a CBR traffic (1 packet every 10 seconds). Additionally, the nodes send Enhanced Beacon and RPL DIO every 23 and 17 seconds respectively. To limit the collision probability, we use 8 shared cells placed uniformly in the slotframe.

We present our findings in the upcoming sections.

5.1 Impact of link quality metrics on the stability

First, we investigate how ETX, ETX^2 and RSSI impact on the network stability. We design a simple RSSI-based RPL rank metric based on the Min-Max normalizer, which normalizes the RSSI to a value between 0 and 1. To define the Min and Max values, we based on the experiments performed by Lee *et al.* [21]. When the RSSI is greater than -80dBm, the packet loss rate is approximately 0 and our normalized value yields 1. In this case, our metric works like MinHop and a node will prioritize neighbors with higher RSSI level that are closer to the sink. Conversely, the packet loss rate increases when the RSSI level is in the gray zone and the normalized value will be lower depending on the RSSI level.

When using the RSSI metric, a node n computes its rank to a neighbor p using the following Equation:

$$rank(n) = rank(p) + \left(\frac{1}{MinMax_{RSSI(p)}} \right) * 256 \qquad (2)$$

Figure 7 compares the 3 metrics in terms of number of parent changes, number of 6P packets sent and end-to-end reliability. During the bootstrap phase, the 3 metrics exhibit similar performance, a pike of parent changes and 6P requests that decrease over the time. In this phase, the number of parent changes of the RSSI metric is slightly lower, as the nodes can use the RSSI level of their neighbors to select a new preferred parent. However, the RSSI is sensitive to external interferences [9] and its value oscillates frequently triggering new parent changes.

These frequent parent changes were also observed for the ETX^2 metric. ETX^2 implicitly penalizes more intermediate links than the other two. 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 penalization, ETX^2 reacts even for small oscillations, i.e. provokes a parent change. One solution to reduce this instability is to use higher hysteresis values for the routing protocol, but at the cost of decreasing the network reactivity. Conversely, after the convergence, the network using ETX works more steadily with few moments of instability and more regularly regarding the end-to-end reliability.

5.2 Accurate parent changes

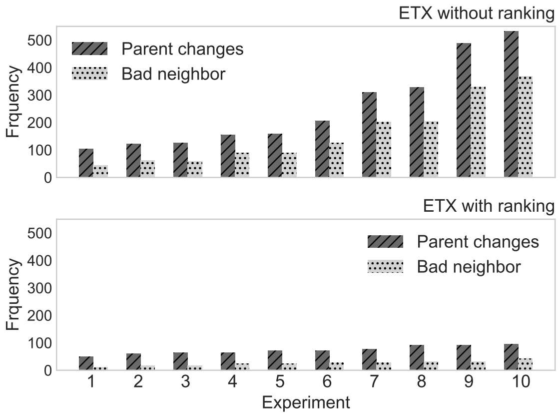
We focus now on evaluating our solution filtering the neighbors with their broadcast rate, as described in Section 4.2.1. We compare it with the blind (default link metric) approach. In our approach, all nodes register the broadcast packets that were sent by their neighbors and they use this information to rank them according to the number of packets received. Thus, the best candidates neighbors are those who are ranked in the firsts positions and a node prioritizes them when it needs to change its preferred parent.

Figure 8a illustrates the number of parent changes and the number of times that a bad preferred parent was selected in each experiment. 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. Without prefiltering the neighbors, the number of parent changes is quite high (between 100 and 500 depending on external interference). Not surprisingly, any 6P negotiations fail when using default link metric: the neighbor is so bad that the handshake fails repetitively. On the

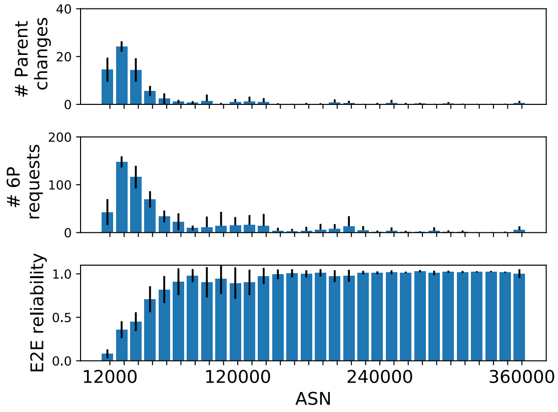
---

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

[‡] https://github.com/rodrigoth/mswim2018



(a) Frequency of bad neighbor selection



(b) Network stability over the time

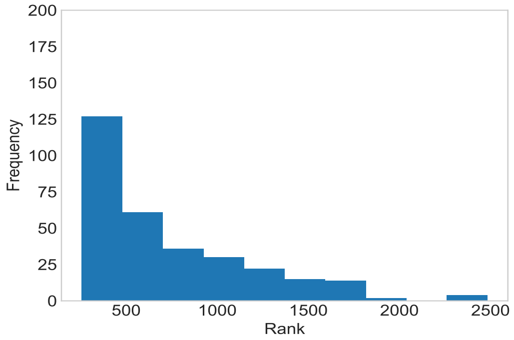
Figure 8: Impact when using the broadcast reception rate on the network stability (95% of confidence).

contrary, using the broadcast rate 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.

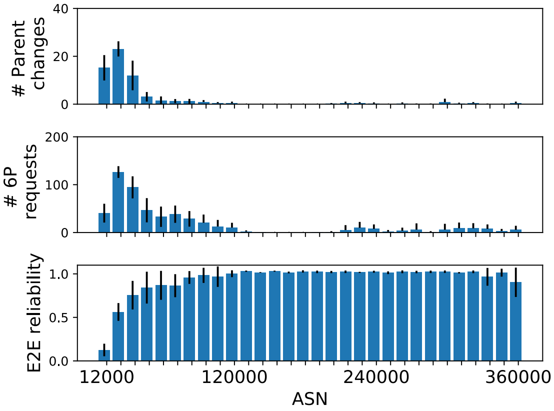
Figure 8b illustrates the time-variations of the different performance metrics (averaged over the 10 experiments). We can still observe a pike of parent changes during the bootstrap phase. Indeed, the nodes send initially a burst of 6P requests and the probability of collisions in a shared cell is very 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, the number of parent changes is much lower and precise in our approach, as they can select a good candidate neighbor in the first attempts. This is reflected in a shorter convergence time and a smoother instability periods during the operating phase.

5.3 Schedule consistency simplification

We measure now the impact of simplifying the schedule consistency check on the network performance. As in the previous experiment, we use the broadcast reception rate . We implemented a simple and conservative housekeeping algorithm that runs periodically (5 min), looking for schedule inconsistencies. 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.



(a) Frequency of schedule fixes by rank



(b) Network stability over the time

Figure 9: Impact of simplified schedule consistency checking on network stability (95% of confidence).

Figure 9a shows the impact of the nodes rank on the frequency of schedule changes due to the housekeeping algorithm. The nodes closer to the sink are more likely to have inconsistencies in their schedule as they send more 6P packets. Indeed, they handle a large number of cells, proportional to the traffic to forward. Thus, they are potentially more likely to make the network to transit from a stable to a highly unstable state, as seen in the experiment in Section 3.

Additionally, we observe that the network instability in this scenario (Figure 9b) was similar to the result shown in the previous experiment (Figure 8b). This similarity suggests that an accurate parent selection reduces the chances of scheduling inconsistencies between a node and its preferred parent.

6 RELATED WORK

The stability has been studied over the last years focusing mainly on the routing layer. Some works have studied the Trickle algorithm and its impact on the convergence. Clausen *et al.* [22] highlight the network instability under real deployment when using the Trickle algorithm. Because of the oscillations in the link quality, the number of parent changes increases as the emission of DIO results in longer convergence. Kermajani and Gomez [23] investigate the influence of the Trickle algorithm parameters on the convergence time. Increasing the value of the Redundancy Constant (k) and reducing the Minimum Interval (I_{min}) achieves a lower convergence time. In addition, they propose a new mechanism that schedules DIS messages for a faster convergence.

Other works have focused on load balancing approaches to achieve higher end-to-end delivery rate and stabler topologies. Kim *et al.* [24] avoid overloading a forwarding node by introducing the queue size to RPL objective function. The approach minimizes the occurrence of packet losses and reconfigurations due to buffer overflow. Additionally, the network stability can also be increased by adding an adaptive mechanism that adjusts the transmission power to minimize the occurrence of collisions [25] .

Iova *et al.* [10] evaluate the impact of different link quality metrics on the network 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 performing badly. On the other hand, ETX and LQI the network present higher frequency of reconfigurations and higher end-to-end reliability. Alvi *et al.* [26] propose rather to change the objective function, combining the ETX and min hop metrics. However, this method is quite conservative: a node tends to be stuck with the same parent, even if a better alternative choice exists.

7 CONCLUSION AND PERSPECTIVES

We investigated here the stability of 6TiSCH networks in an indoor environment. We identified the main causes of instability and we proposed solutions to each one of them. We evaluated different routing metrics commonly used in the literature. In our experiments, ETX performed more steadily than ETX^2 and an RSSI-based metrics after the network convergence. Next, we exploited the existing correlation between the broadcast packet reception rate and the unicast link quality to create a 2-step parent selection, avoiding bad choices leading to instabilities. Finally, we simplified the schedule consistency management between two nodes to reduce the instability caused by renegotiating from scratch all the cells when they detect a schedule inconsistency. We obtain finally a network that converges faster and that reacts accurately during moments of instabilities.

In a future work, we expect to study in-depth the network re-convergence time in the presence of hardware failures, or mobile nodes. Fault-tolerance is a key characteristic in industrial networks. Surprisingly, few attention has been given considering the re-convergence process in industrial wireless networks with real-time requirements. We aim also to study the network stability when considering a bursty traffic, e.g. alarms with strict delay constraints. Additionally, we aim to increase the variability of our experiments by considering networks with different traffic load and network topologies.

REFERENCES

[1] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin. Smart factory of industry 4.0: Key technologies, application case, and challenges. *IEEE Access*, 6:6505–6519, 2018.

[2] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Tholeyre. Is local blacklisting relevant in slow channel hopping low-power wireless networks? In *ICC*, pages 1–6, Miami, FL, USA, May 2017. IEEE.

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

[4] O. Iova, F. Theoleyre, T. Watteyne, and T. Noel. The Love-Hate Relationship between IEEE 802.15.4 and RPL. *IEEE Communications Magazine*, 55(1):188–194, January 2017.

[5] IEEE Standard for Low-Rate Wireless Networks. IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011), April 2016.

[6] 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. *Computer Communications*, 114:84 – 105, 2017.

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

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

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

[10] O. Iova, F. Theoleyre, and T. Noel. Stability and efficiency of RPL under realistic conditions in Wireless Sensor Networks. In *PIMRC*, pages 2098–2102, London, UK, Sept 2013. IEEE.

[11] K. Heurtefeux and H. Menouar. Experimental evaluation of a routing protocol for wireless sensor networks: RPL under study. In *WMNC*, pages 1–4, Dubai, United Arab Emirates, April 2013. IFIP.

[12] B. Pavkovic, F. Theoleyre, D. Barthel, and A. Duda. Experimental Analysis and Characterization of a Wireless Sensor Network Environment. In *PE-WASUN*, pages 25–32, Bodrum, Turkey, 2010. ACM.

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

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

[15] D. De Guglielmo, S. Brienza, and G. Anastasi. IEEE 802.15.4e: A survey. *Computer Communications*, 88:1 – 24, 2016.

[16] F. Theoleyre and G.Z. Papadopoulos. Experimental Validation of a Distributed Self-Configured 6TiSCH with Traffic Isolation in Low Power Lossy Networks. In *MSWiM*, pages 102–110, Malta, Malta, 2016. ACM.

[17] A. C. Lima-Filho, R. D. Gomes, M. O. Adissi, T. A. B. da Silva, F. A. Belo, and M. A. Spohn. Embedded system integrated into a wireless sensor network for online dynamic torque and efficiency monitoring in induction motors. *IEEE/ASME Transactions on Mechatronics*, 17(3):404–414, June 2012.

[18] M. Lauridsen, B. Vejlgaard, I. Z. Kovacs, H. Nguyen, and P. Mogensen. Interference Measurements in the European 868 MHz ISM Band with Focus on LoRa and SigFox. In *WCNC*, pages 1–6. IEEE, March 2017.

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

[20] K. Muraoka, T. Watteyne, N. Accettura, X. Vilajosana, and K. S. J. Pister. Simple Distributed Scheduling With Collision Detection in TSCH Networks. *IEEE Sensors Journal*, 16(15):5848–5849, Aug 2016.

[21] T. H. Lee, X. S. Xie, and L. H. Chang. Rssi-based ipv6 routing metrics for rpl in low-power and lossy networks. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1714–1719, Oct 2014.

[22] T. Clausen, U. Herberg, and M. Philipp. A critical evaluation of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL). In *WiMob*, pages 365–372, Shanghai, China, Oct 2011. IEEE.

[23] H. Kermajani and C. Gomez. On the Network Convergence Process in RPL over IEEE 802.15.4 Multihop Networks: Improvement and Trade-Offs. *Sensors*, 14:11993–12022, 2014.

[24] H. S. Kim, H. Kim, J. Paek, and S. Bahk. Load balancing under heavy traffic in rpl routing protocol for low power and lossy networks. *IEEE Transactions on Mobile Computing*, 16(4):964–979, April 2017.

[25] H.S. Kim, J. Paek, D. Culler, and S. Bahk. Do not lose bandwidth: adaptive transmission power and multihop topology control. In *DCOSS*, pages 99–108, Ottawa, Canada, 2017. IEEE.

[26] S. A. Alvi, F. u. Hassan, and A. N. Mian. On the energy efficiency and stability of RPL routing protocol. In *IWCMC*, pages 1927–1932. IEEE, June 2017.